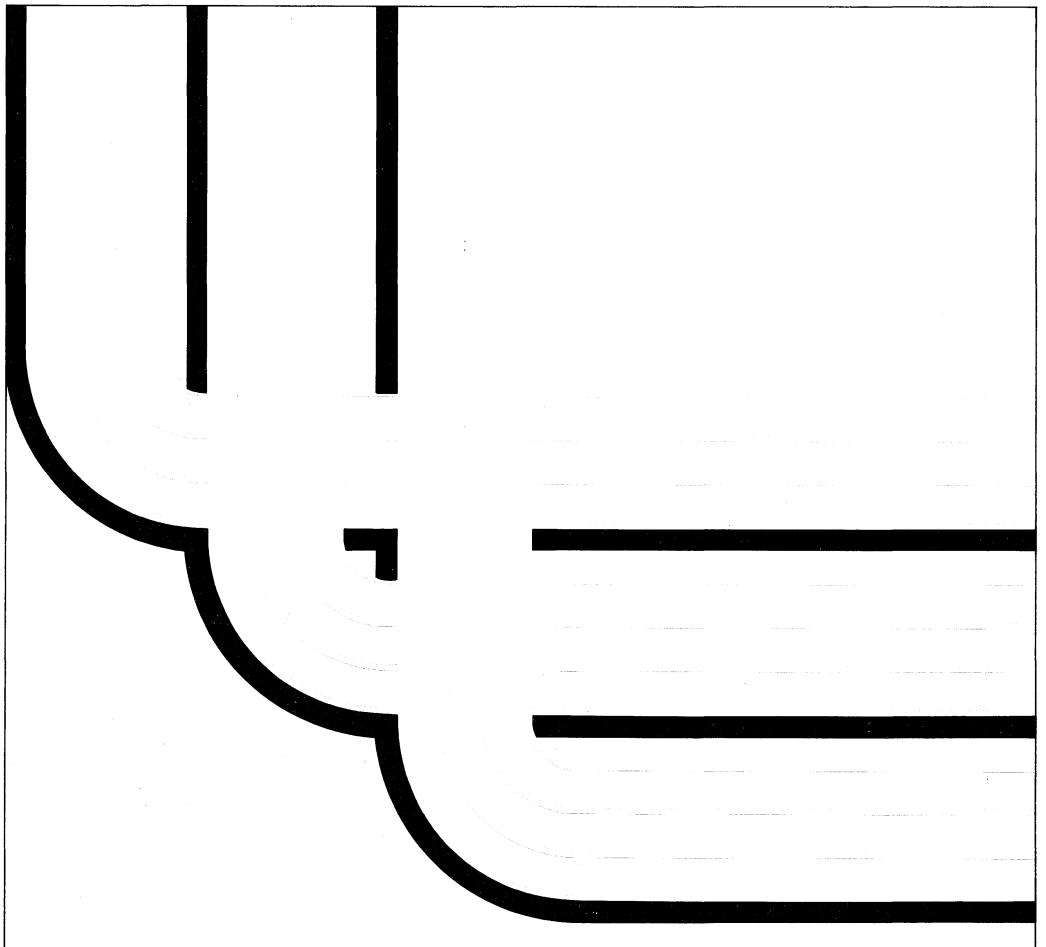
Application System/400

SC41-0056-01

Operating System/400 Workstation Customization Function Programmer's Guide

Version 2



System and Application Support

Application System/400

SC41-0056-01

Operating System/400 Workstation Customization Function Programmer's Guide

Version 2

Take Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page vii.

Second Edition (November 1993)

This edition applies to the licensed program IBM Operating System/400, (Program 5738-SS1), Version 2 Release 3 Modification 0, and to all subsequent releases and modifications until otherwise indicated in new editions. This major revision makes obsolete SC41-0056-0. Make sure you are using the proper edition for the level of the product.

Order publications through your IBM representative or the IBM branch serving your locality. Publications are not stocked at the address given below.

A Customer Satisfaction Feedback form for readers' comments is provided at the back of this publication. If the form has been removed, you can mail your comments to:

Attn Department 245
IBM Corporation
3605 Highway 52 N
Rochester, MN 55901-7899 USA

or you can fax your comments to:

United States and Canada: 800+937-3430
Other countries: (+1)+507+253-5192

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you or restricting your use of it.

Contents

Part 3. Printer Customization Technical Reference

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577, U.S.A.

This publication could contain technical inaccuracies or typographical errors.

This publication may refer to products that are announced but not currently available in your country. This publication may also refer to products that have not been announced in your country. IBM makes no commitment to make available any unannounced products referred to herein. The final decision to announce any product is based on IBM's business and technical judgment.

Changes or additions to the text are indicated by a vertical line (|) to the left of the change or addition.

| Refer to the "Summary of Changes" on page xi for a summary of changes made to the Operating System/400 licensed
| program and how they are described in this publication.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

This publication contains small programs that are furnished by IBM as simple examples to provide an illustration. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. All programs contained herein are provided to you "AS IS". THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

Application System/400
AS/400
IBM
InfoWindow
OfficeVision
OfficeVision/400
Operating System/400
OS/400
PS/2
Personal System/2
400

The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other companies as follows:

Dasher	Data General Corporation
DEC	Digital Equipment Corporation
Epson	Epson America Incorporated
Hewlett-Packard	Hewlett-Packard Company
LaserJet	Hewlett-Packard Company
NEC	NEC Corporation
Okidata	Okidata Corporation
TeleVideo	TeleVideo Systems, Incorporated
Wyse	Wyse Corporation
WY30	Wyse Corporation
WY50	Wyse Corporation
WY60	Wyse Corporation

About This Guide

This guide provides introductory, procedural, and reference information about the OS/400* workstation customizing functions. The workstation customizing functions allow you to map both supported and unsupported twinaxial keyboards, ASCII displays and keyboards, and ASCII printers for use with the AS/400* system.

This guide is intended for experienced technical professionals who are knowledgeable about configurations of keyboards, displays, and printers.

Before using the workstation customizing functions, your AS/400 system should be up and running. You should have your licensed programs installed, including IBM* Application Development Tools/400 (Program 5738-PW1). The new devices you want to use with the system should be physically attached to the system to complete the customizing procedures. See the *AS/400* 9402 Attaching Workstation and Communications Cables*, SA41-0005, the *9404 Attaching Workstation and Communications Cables*, SA41-0004, or the *9406 Attaching Workstation and Communications Cables*, SA41-9957, to attach your workstation cables to the AS/400 system. You should also be familiar with the procedures in the *Device Configuration Guide*, SC41-8106, and the *ASCII Work Station Reference and Example*, SA41-9922.

You should also have the *National Language Support Planning Guide*, GC41-9877, readily available to look up information about the EBCDIC code pages associated with the language you are using.

You may need to refer to other IBM manuals for more specific information about a particular topic. The "Bibliography" lists publications especially relevant to the topics discussed in this book. The *Publications Guide*, GC41-9678, provides information on all the manuals in the AS/400 library.

Summary of Changes

This manual has been divided into three parts.

Part 1 describes the tasks you need to follow to use the workstation customization functions.

- "Using the Tag Language to Customize Your Workstations" (formerly Chapter 6), is now part of Chapter 3 in Part 1 of this manual.

Part 2 contains technical reference material required to customize displays and keyboards.

- Chapter 6 is a new chapter that provides an overview of the two ways you can customize the functions of displays and keyboards.
- Chapter 7 describes how to customize twinaxial displays. The content of this chapter is basically unchanged from Version 2 Release 2.
- Chapter 8 describes how to customize ASCII displays. The content of this chapter is basically unchanged from Version 2 Release 2.

Part 3 contains technical reference material required to customize ASCII printers.

- Chapter 9 is a new chapter that provides an overview of the three different ways you can customize ASCII printer functions.
- Chapter 10 is a new chapter that describes how to customize ASCII printers that use the host print transform function. The host print transform function on the AS/400 system transforms the data sent to ASCII printers. To customize a printer that uses the host print transform function, you can retrieve printer attributes from the AS/400 system-provided customizing objects. The retrieved attributes can then be changed using source entry utility (SEU).

  Customizing ASCII printers that use host print transform function has advantages over customizing printers that use the emulator on the display or the workstation controller. The host print transform function:
  - Provides support for more ASCII printers than are supported by a particular device emulator.
  - Provides support for more printer functions than are supported by a particular device emulator.
  - Provides consistent function across all attachment methods.
- Chapter 11, formerly Chapter 9, describes how to customize ASCII printers that use the emulator on the display station. This chapter has been reorganized to highlight the functions supported by different twinaxial display types.
- Chapter 12, formerly Chapter 10, describes how to customize ASCII printers that use the emulator on the workstation controller. The content of this chapter is basically unchanged from Version 2 Release 2. The customization example, however, has been replaced with a new example.

In addition, appendixes have been updated to include work sheets and a source code example for ASCII printers that use the host print transform function.

Part 1. Customizing Workstations

Chapter 1. Planning For and Setting Up Workstation Customizing

This chapter provides a brief overview of the workstation customizing procedure and then describes the planning step in the procedure. A list of the possible errors and the necessary steps for recovery is provided, followed by a planning checklist that tells you what you need to have when the planning step is complete. You can find examples of the planning step for the workstation customizing procedure in the reference chapters in Part 2 and Part 3 of this guide.

Workstation Customizing Overview

The workstation customizing functions provided by the IBM* Operating System/400* licensed program allow you to do the following:

- Change the way a local or remote workstation controller supports displays, keyboards, and ASCII printers that you use with your AS/400* system.
- Change the way a twinaxial display supports ASCII printers that you use with your AS/400 system.
- Change the way the host print transform function supports the ASCII printers that you use with your AS/400 system.

Some of the workstation characteristics you can add or change are listed below:

Command key sequences
Special character support (display and print)
Function key support
Shift state key processing
Bold, underlined, and double width printing
Font and pitch support
Paper orientation
Paper selection

Customizing Limitations

Workstation customizing supports only those functions supported by the workstation controller, the display, or the host print transform function and the device itself.

If the workstation controller, the display, or the host print transform function cannot support a particular function, the function cannot be added using the workstation customizing functions.

If a device is not capable of performing a function, or displaying or printing a character, workstation customizing does not and cannot provide the additional function.

The workstation customizing functions require that you provide hexadecimal data corresponding to the characters and functions supported by your device. You must have this information to customize device functions.

To change the characteristics of workstations and printers, you change copies of the mapping tables used for character translation and function translation. You then create a customizing object. A road map showing the basic steps of the workstation customizing procedure is shown in the following figure. The steps are denoted throughout the first part of this guide by the reference numbers (for example, **1**) that appear in the figure.

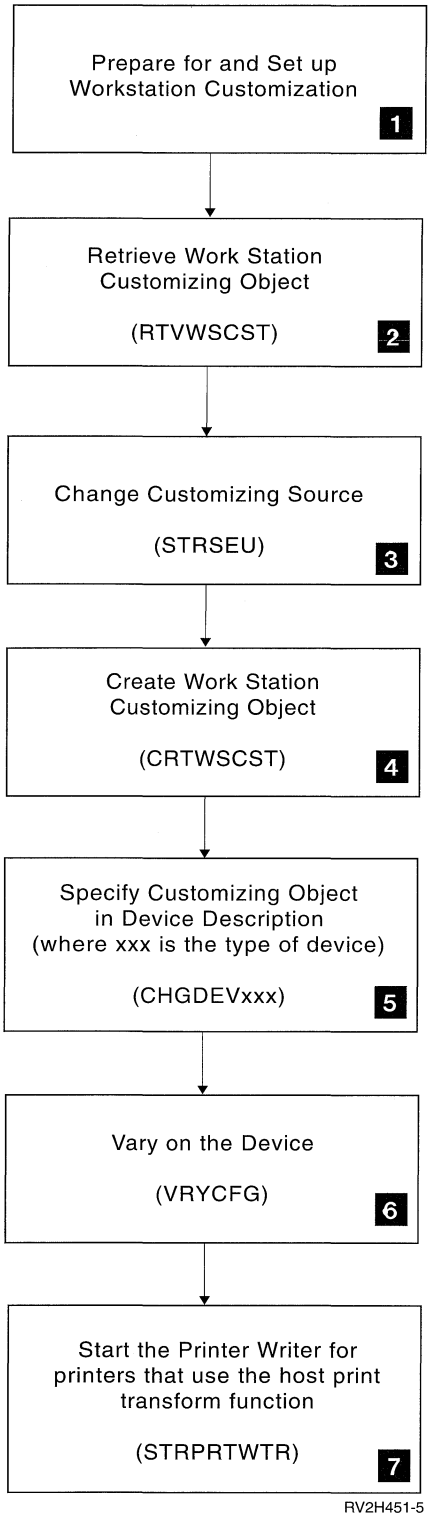


Figure 1-1. Workstation Customizing Procedure Road Map. Step 7 applies only to printers that use the host print transform function with spooled print jobs. For printers that use the host print transform function with nonspooled print jobs, the customized table is activated when the job is printed.

To begin the workstation customizing procedure, there are some things you should do before you actually customize the device. First, you need to familiarize yourself with the display or printer device you want to customize. This means you should read the manuals and available reference material for the device and understand the characteristics you want to change. If the device has never been used with the AS/400 system before, or it is a device that is not supported by IBM, you need to connect the device to the system, see how it works, and then create a list of the characteristics you want to change. You also need the hexadecimal codes for the device. The hexadecimal codes provide you with values for customizing the appropriate mapping tables.

If you are using a language other than English, you may also have language concerns you need to address. The language a workstation uses determines how you should define several characteristics when you first set up the AS/400 system and the workstation itself.

You may be working with more than one language on your AS/400 system. For example, suppose the primary language used on the system is US English, but some of your workstations are set up to display French. The device description for the workstations using the secondary language must indicate the correct keyboard language type and character identifier. Indicating the correct keyboard language type and character identifier ensures the correct mapping tables are used to display data.

For more information about language-specific concerns with workstation customizing, see the language considerations discussed in the following chapters in the reference section of this guide:

- Chapter 7, "Customizing Twinaxial Displays"
- Chapter 8, "Customizing ASCII Displays"
- Chapter 10, "Customizing ASCII Printers That Use the Host Print Transform Function"
- Chapter 11, "Customizing ASCII Printers That Use the Emulator on the Display"
- Chapter 12, "Customizing ASCII Printers That Use the Emulator on the Controller"

For more general information about setting up and using national language support, see the sections about using and changing a secondary language and system coded character set identifier (CCSID) implementation in the *National Language Support Planning Guide* and the other language support manuals listed in the "Bibliography."

The next step in the workstation customizing procedure is to retrieve or create a customizing source file. To retrieve a source file member based on existing AS/400 supported devices, use the Retrieve Work Station Customizing Object Source (RTVWSCST) command. The retrieved source contains copies of the system mapping tables for the device or manufacturer type and model you specified in the command. The source also contains the correct tags for the type of device or manufacturer type and model. For examples of retrieved source file members that have not been changed, see Appendix B, "Source Code Examples."

A tag language is provided for you to change the mapping tables that you retrieve. The tag language provides a shell for the system-based mapping tables that are used to map data and functions between the AS/400 system and the device you are customizing. For a description of the tag language for workstation customizing, see Chapter 3, "Changing the Source to Customize Your Workstations."

After you have completed your changes in the source file member, you use the Create Work Station Customizing Object (CRTWSCST) command to create a customizing object. Specifying the name of that object in the device description for the display or printer links the object to the device. If a customized object is used by the host print transform function, it is activated when:

- The printer writer is started (spooled print jobs)
- A job is printed (nonspooled print jobs)

Otherwise, the customized mapping tables are downloaded to the workstation controller when you vary on the device. The customized tables are used to map the data from the device to the AS/400 system and from the system back to the device. If you need to delete a workstation customizing object from your system, use the Delete Work Station Customizing Object (DLTWSCST) command.

Preparing for Workstation Customizing

If you are already using an ASCII workstation or printer, you or your users may have identified certain functions that are not supported the way that you would want or expect. For example, you press a key and the character displayed is not the character engraved on the key. Or, although your ASCII printer prints most characters correctly, there are some characters specific to the language you use that you would also like to have print correctly. The workstation customizing functions can help you correct these problems.

The following list shows the device types that can be customized using the OS/400* workstation customizing functions:

- Twinaxial keyboards (twinaxial workstations)
- ASCII displays and keyboards (ASCII workstations)
- ASCII printers that use the host print transform function
- ASCII printers (attached to 3477 Model H, 3486, 3487, or 3488 twinaxial workstations) that use the emulator on the display
- ASCII printers (directly attached to the AS/400 system) that use the emulator on the workstation controller

Before you begin using the workstation customizing functions, your AS/400 system must be up and running, and you need to have at least one workstation configured and usable. The workstations you want to customize should be physically attached to the AS/400 system, and you also should have created all the appropriate line (if remote), controller, and device descriptions.

You may need to refer to some or all of the following manuals when customizing a device:

- Reference manual for the device you want to customize
- *ASCII Work Station Reference and Example*
- *National Language Support Planning Guide*

It is absolutely critical that you have the hexadecimal codes for the device you want to customize. This information is often included in the reference manual for the device. The device manual also gives you an understanding of what functions the device is capable of performing and which language or character sets it supports.

The following sections briefly describe some of the equipment and the mapping tables for which you use the workstation customizing functions.

Workstation Controllers

The following local and remote workstation controllers can use the customized mapping tables you create with the workstation customizing functions.

Note: Customized printers that use the host print transform function do not have workstation controller dependencies.

- Local Workstation Controllers

  2661: Twinaxial local workstation controller
  2638: Twinaxial local workstation controller
  6040: Twinaxial local workstation controller
  6050: Twinaxial local workstation controller
  6140: Twinaxial local workstation controller
  2637: ASCII local workstation controller
  6041: ASCII local workstation controller
  6141: ASCII local workstation controller

- Remote Workstation Controllers

  5494 Remote workstation controller (Release 1 Modification 1)

  Note: The 5494 is a twinaxial workstation controller. The customizing process for a remote workstation attached to the 5494 is the same as it would be for a local twinaxial workstation.

The workstation controller uses the mapping tables loaded from the AS/400 system to translate data back and forth between the workstation or printer and the system. When you change the hexadecimal values in the mapping tables, you customize the associated device. The customized tables and attributes are then compiled and stored in the customizing object you create. This object is used to create customized mapping tables that are downloaded to the workstation controller the next time you vary on the workstation or printer. You can use a single customizing object for more than one workstation or printer, or you may want a different customizing object for each device.

Displays

The IBM Corporation sells a wide variety of displays that can be connected to the AS/400 system. The OS/400 workstation customizing functions support both the twinaxial display and the ASCII display. Displays are attached to the AS/400 system by way of a workstation controller. Both the ASCII workstation controller and the twinaxial workstation controller use mapping tables downloaded from the AS/400 system to support the input/output (I/O) for these devices.

The workstation controller processes I/O for a display in two ways. **Inbound processing** refers to keystroke and function processing from the display to the AS/400 system. **Outbound processing** refers to character and function processing from the AS/400 system to the display. The workstation customizing functions allow you to customize both types of processing for an ASCII display, but only the inbound, or keystroke processing can be customized for a twinaxial display.

ASCII Printers

A wide variety of ASCII printers can be attached to the AS/400 system. ASCII printers receive and process data by way of an ASCII data stream. One example of an ASCII data stream is the IBM page printer data stream (PPDS). ASCII printers understand only the ASCII data stream and commands. The AS/400 system and applications produce an EBCDIC data stream. Therefore, the AS/400 system treats all ASCII printers as emulated twinaxial printers.

This means that the EBCDIC data stream is converted to an ASCII data stream. Some restrictions exist on the printing functions that are supported because of the selected emulation type, the ASCII printer type, and the connection type. Some of the more general functions that are not supported include graphics, images, and bar codes. For more information on customizing printers, see Chapter 9, "Customizing Printers."

OS/400 Workstation Customizing Function Limitations

Most manufacturers design displays and printers that support the specific character sets for the countries in which the devices are sold. The character set support is either built into the hardware for the device or embedded into the internal code for the device. This places limitations on the OS/400 customizing functions.

Setting Up Workstation Customizing

In most cases, you are already aware of some characteristics that you would like to change about the way your device works with the AS/400 system. In other cases, the device has never been attached to the AS/400 system. You may need to experiment with it to see which characteristics you want to change. Either way, you need to gather the following information before you put together your workstation customizing plan:

- Reference manuals for your device
- A copy of the customizing source for the device
- Information about the tag language used for customizing
- A list of characters, command, and control sequences you want to change

Notes:

1. If you are customizing an ASCII display station, you also need to have the *ASCII Work Station Reference and Example*.
2. If you are customizing an ASCII printer that uses the emulator on the workstation controller, you also need to have the *ASCII Work Station Reference and Example*.
3. If you are using one or more national languages, you also need to have the *National Language Support Planning Guide*.

The second and third items in this list are covered in the reference part of this manual. The last item comes from your or your user's experience with the device to be customized. If you are customizing the tables for a device that is not supported by IBM, you may also want to have the reference information for a similar IBM-supported device so that you can compare the character values and control sequences. To complete the planning for customizing your workstation, do the following:

1. Read the reference manuals for the device and know what the control codes and function codes are and where they are located in the manual.
2. Experiment with the device to find out which characteristics you would like to change. For example, you may need to change the way a character is displayed or printed. You may want to change the way a certain character is displayed when used with the Shift key or change the function of a command or control key sequence.
3. Use the work sheets provided in Appendix E, "Workstation Customizing Planning Work Sheets" to make a note of these characteristics and check the device reference manuals to find out the following:
   - Does the device itself support the character or function you want to add or change? (Is the device capable of producing the character or performing the function?)

Note: If a device cannot support a character or function, the workstation customizing functions cannot provide the missing support.

- If the device can support the character or function, what is the hexadecimal or integer representation for that character or function?

4. Familiarize yourself with the way the AS/400 workstation controller, the display, or the host print transform function handles the mapping tables for the type of device you want to customize.
5. Read the tag language descriptions in Part 2 and Part 3 of this guide to understand the tags you use to customize your device.
6. Use the information in the following sections of this guide to determine how to customize your device:
   - Twinaxial keyboard: see "Determining Which Twinaxial Keyboard Translation Table to Customize" on page 7-9.
   - ASCII display: see "Determining Which ASCII Display Tables to Customize" on page 8-14.
   - ASCII printer that uses the host print transform function: see "Choosing the Customizing Source" on page 10-2.
   - ASCII printer that uses the emulator on the display: see"Determining Which ASCII Printer Tables to Use" on page 11-2 .
   - ASCII printer that uses the emulator on the workstation controller: see "Determining Which ASCII Printer Tables to Use" on page 12-4.

   Examples showing this step of the workstation customizing procedure are provided for each device class in the corresponding reference chapters in Part 2 and Part 3 of this guide. When the planning step is complete, you should have an understanding of the device you want to customize, the customizing procedures, and a list of the characteristics (and their hexadecimal values if applicable) for the device you want to customize.

Errors and Recovery

The following list describes errors that can occur during the steps of planning and setting up workstation customizing. The list also recommends some things to do to recover from the error.

- **Missing reference information for the device**

  If you do not have the reference information for the device you want to customize, you will not be able to use the workstation customizing functions. OS/400 workstation customizing is dependent on the hexadecimal codes and sequences that a device sends or understands. The reference manual for the device should provide this information. If you do not have the reference manual for the device, contact your sales representative or the manufacturer of the device and request a copy.

- **ASCII display does not show Sign On display**

  If you cannot sign on to the AS/400 system, you cannot experiment with the display to see what needs to be customized. Use the procedure provided in Appendix D, "Setting Up to Customize a Display." If you still cannot sign on using the display, you can try to continue to customize the display by changing a few characteristics in the source at a time, compiling the object, and testing it with the display. The best characteristics to start with are the Clear Screen and Set Cursor Address commands.

- **ASCII printer does not produce any printouts**

  If your printer does not produce any printouts, one of the following errors may be causing the problem:

  - Your hardware may be set up incorrectly. Check the reference manuals for your display, workstation controller, and printer, as appropriate, to resolve this problem.
  - Your printer may be configured incorrectly. If the ASCII printer is supported by IBM and the AS/400 system, you need to check the device type and model number and verify that this information was entered correctly in the device description.

    If the ASCII printer is not supported by IBM, you may need to try specifying another device type for the device description. The *ASCII Work Station Reference and Example* provides descriptions of some supported ASCII printers that you can use for comparison with the characteristics of your unsupported ASCII printer.
  - Your 3477 display is not a Model H. Use the procedure in "Setting Up a 3477 Twinaxial Display" on page D-2 to verify your 3477 display is a Model H. If your printer uses the emulator on the twinaxial display, the 3477 display must be a Model H. If it is not a Model H, you cannot use the workstation customizing functions to customize an attached printer. If you would like to upgrade your 3477 to a Model H, contact your marketing representative or service representative.

Verifying That Planning Is Complete

When the planning step is complete, you should have collected the following:

- A list of the device characteristics you want to customize
- The hexadecimal data or integer values associated with the device characteristics

You should be familiar with the way the workstation controller, the display, or the host print transform function handles the mapping tables for your device. You should also have an understanding of the tags you use in the customizing step of the procedure. (See the chapters for each device type in Part 2 and Part 3 of this guide.)

Chapter 2. Retrieving the Workstation Customizing Source

After you have completed the planning for workstation customizing, you are ready to retrieve copies of the workstation customizing source for the device you want to customize. Use the Retrieve Work Station Customizing Object Source (RTVWSCST) command to retrieve copies of the workstation customizing source for the device.

When you retrieve copies, the internal forms of the tables and control sequences are converted into the source format of the customizing tag language. The source is then put into a source file member.

Although it is possible to create an entire source file member without retrieving a copy of existing tables, it is not recommended. The retrieved source provides you with the correct number of mapping tables for the device or the manufacturer, type, and model you specify. The retrieved source also provides you with the correct layout and syntax for the tag language code. To look at examples showing the source retrieved for the different device classes, see Appendix B, "Source Code Examples."

Devices That Use the Emulator on the Workstation Controller or Display

If you are customizing a device that uses the emulator on the workstation controller or the display, have the following information available. You need this information to retrieve the correct source for the device:

- Device type

  If you are customizing an unsupported device, select a device type that is most like your device. Doing this allows you to retrieve a source file member that contains the mapping tables for you to customize. If you cannot determine which supported device is most like your device, select any ASCII device. Be sure to differentiate between an ASCII display and an ASCII printer.

- Keyboard language type to be used on this device
- Library and source physical file name for storing your customizing source code

Note: If you are customizing the keyboard for an unsupported twinaxial display, you must select the supported twinaxial display that your display emulates.

After you determine which device type to select, use the following procedure to retrieve your workstation customizing source.

Retrieving Source for Devices That Use the Emulator on the Workstation Controller or the Display

Follow these steps to retrieve workstation customizing source for devices that use the emulator on the workstation controller or the display station.

1. Type RTVWSCST (Retrieve Work Station Customizing Object Source command) and press F4 (Prompt).
2. Specify the device type for the device you want to customize. This is a required parameter. For the lists of IBM-supported devices, see one of the following:
   - "Types of Displays and Keyboards" on page 7-1
   - "Beginning to Customize ASCII Displays" on page 8-1
   - "Supported ASCII Printers Attached to Twinaxial Displays" on page 11-1
   - "Supported ASCII Printers" on page 12-1

   Note: If you are customizing an ASCII printer attached to a twinaxial display, specify the display type for this parameter. Do not specify the printer type. (If you are customizing an ASCII printer attached to a 3488 display, specify that the printer is attached to a 3487 display. The OS/400 customizing functions treat a 3488 display station as a 3487 display station.)
3. Specify the keyboard language type you want to use for this device. This is a required parameter. A list of the languages you can specify here is in Table 7-21 on page 7-16, or you can use the help information on the system to determine which language type to specify.

   Note: It is important that the keyboard language type you specify here is the same as the keyboard language type specified in the device description. If the two keyboard language types specified do not match, then the device will not vary on.
4. Specify a name for the source file member to be created for the customizing tag language source that you want to retrieve. This is a required parameter. This should be a name you can easily remember, possibly related to the device or type of device you are customizing.
5. Specify the keyboard type. This parameter is required only for customizing twinaxial displays and twinaxial displays with attached ASCII printers. For illustrations of the keyboard types you can use with twinaxial displays, see Appendix A, "Twinaxial Keyboard Layouts." Each illustration is associated with a keyword value to assist you in making your selection.

   Note: For twinaxial displays, this is a required parameter. The value specified here must match the actual keyboard type of the display being customized; otherwise the display will not vary on successfully.

6. Specify a library and source file name to create a source physical file in which to store the source file member. The library you specify must exist. If the source physical file already exists, the system adds the new file member to it. If the source file does not exist, the system creates it for you. The coded character set identifier for the newly created file is *HEX.

   The default value for this field is the system-supplied source file QTXTSRC in library QGPL.

7. Specify a text description for the source file if it does not already have one. The description should be unique so that you can use it to help you identify the devices associated with the source file member. This is helpful when you have more than one source file for more than one device type.

Each device that can be customized has one or more specific mapping or translation tables associated with it. The source physical file member you create in this step should contain all the source code for each of the tables required by the device you specified.

Examples showing this step of the workstation customizing procedure are provided for each device class in the corresponding reference chapters in Parts 2 and 3 of this guide. After you have retrieved the workstation customizing source, you can begin changing the source that allows you to customize your device.

Printers that Use the Host Print Transform Function

If you are customizing a printer that uses the host print transform function, have the following information available. You need this information to retrieve the correct source for the device:

- Manufacturer, type, and model

  If you are customizing an unsupported printer, select a manufacturer, type, and model that is most like your device. Doing this allows you to retrieve a source file member that contains the mapping tables for you to customize.
- Library and source physical file name for storing your customizing source code

After you determine which manufacturer, type, and model to select, use the following procedure to retrieve your workstation customizing source.

Retrieving Source for ASCII Printers That Use the Host Print Transform Function

Follow these steps to retrieve workstation customizing source for ASCII printers that use the host print transform function.

1. Type RTVWSCST (Retrieve Work Station Customizing Object Source command) and press F4 (Prompt).
2. Specify *TRANSFORM for the device type. This is a required parameter.
3. Specify the manufacturer, type, and model for the ASCII printer you want to customize. Press F4 (Prompt) in the *Manufacturer type and model* field to see a list of values you can specify. This is a required parameter.
4. Specify a name for the source file member to be created for the source that you want to retrieve. This is a required parameter. This should be a name you can easily remember, possibly related to the manufacturer, type, and model name.
5. Specify a library and source file name to create a source physical file in which to store the source file member. The library you specify must exist. If the source physical file already exists, the system adds the new file member to it. If the source file does not exist, the system creates it for you. The coded character set identifier for the newly created file is *HEX.

   The default value for this field is the system-supplied source file QTXTSRC in *LIBL.
6. Specify a text description for the source file if it does not already have one. The description should be unique so that you can use it to help you identify the devices associated with the source file member. This is helpful when you have more than one source file for more than one manufacturer, type, and model.

Examples showing this step of the workstation customizing procedure are provided for the TRANSFORM device class in the corresponding reference chapter in Part 3 of this guide. After you have retrieved the workstation customizing source, you can begin changing it to customize your device.

Retrieving Source for Devices Not Supported by IBM

If you are customizing the keyboard for an unsupported twinaxial display, you must select the supported twinaxial display that your display emulates.

If you want to create a workstation customizing object for an ASCII device that is not supported by IBM and you want to retrieve a source file for the object, you need to compare the specifications for the unsupported device with the specifications of IBM-supported devices. You can also use the *ASCII Work Station Reference and Example* to help you match your display with a supported ASCII display. If you find a moderately close match between sets of specifications, you can then specify the device type and model of that IBM-supported device in the RTVWSCST command. This allows you to retrieve source that may be compatible with the unsupported device you want to customize.

Note: The source you retrieve for an unsupported device based on an IBM-supported device will not be totally compatible with the device. You may have to use trial-and-error

methods when working with the source to create a customizing object that works effectively.

If you do not want to attempt to match device specifications or you cannot find an IBM-supported device that has similar specifications to your device, you can create a source file and blank source file member to enter the tag language and table information yourself. This method of creating workstation customizing source is not recommended; however, if you cannot find any IBM-supported devices with matching specifications, it may be simpler to create a new source file member yourself than to retrieve and try to change a source file member for a totally different device type.

Errors and Recovery

The following list describes some errors that can occur during this step and what you can do to recover from the error.

- **One or more parameters on the RTVWSCST command not valid**

  To find out which parameters were not valid, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) to look at the detailed messages that were sent when you ran the RTVWSCST command. Note the parameter values that the system found to be not valid. Use the planning information in "Preparing for Workstation Customizing" on page 1-3 to make any necessary corrections and try the RTVWSCST command again.

- **You are not authorized to use the RTVWSCST command**

  Have the security administrator (QSECADM) for your system use the Grant Object Authority (GRTOBJAUT) command to give you the proper authority to use the command. You can then try the command again.

- **Parameter KBD required for this device type**

  The keyboard (KBD) parameter is required when you specify a twinaxial display for the device type (DEVTYPE) parameter. To look at the valid keyboard types for a twinaxial display, see "Types of Displays and Keyboards" on page 7-1.

- **Parameter KBD not allowed for this device type**

  The keyboard (KBD) parameter is not allowed when you specify an ASCII display or printer for the device type (DEVTYPE) parameter. Remove the value you specified for this parameter and try the RTVWSCST command again.

- **Device type specified not compatible with the actual device**

  This may occur when retrieving workstation customizing source for devices not supported by IBM. Be sure that the device type you specify for an unsupported device is the same as the device type of the actual device you are using. For example, If your unsupported device is an ASCII display, be sure that the supported device you select for the device type parameter of the RTVWSCST command is an ASCII display.

  The mapping tables retrieved will not be completely compatible with your device; however, the source retrieved should contain the appropriate mapping tables and corresponding customizing language tags. You should be able to make the necessary changes to the mapping tables in the source using the hexadecimal code information in the reference manual for the device.

- **Keyboard language type not allowed**

  The keyboard language type (KBDTYPE) parameter is not allowed when you specify *TRANSFORM for the device type (DEVTYPE) parameter. Remove this value and try the RTVWSCST command again.

- **Manufacturer type and model not allowed**

  The manufacturer type and model (MFRTYPMDL) parameter is not allowed when you specify anything other than *TRANSFORM for the device type (DEVTYPE) parameter. Remove this value and try the RTVWSCST command again.

- **Manufacturer type and model required**

  The manufacturer type and model (MFRTYPMDL) parameter is required when you specify *TRANSFORM for the device type (DEVTYPE) parameter. To see the allowed values, type in the RTVWSCST command, and press F4 (Prompt). In the MFRTYPDML field, press F4 (Prompt). Specify one of the values shown for the MFRTYPMDL parameter, and try the RTVWSCST command again.

- **Keyboard language type required**

  The keyboard language type (KBDTYPE) parameter is required when you specify anything other than *TRANSFORM for the device type (DEVTYPE) parameter. Specify a keyboard language type and try the RTVWSCST command again.

- **Source member required**

  The source member (SRCMBR) parameter is a required parameter. Specify a source member and try the RTVWSCST command again.

Verifying the Source Is Retrieved Successfully

When you have completed the procedure to retrieve the source for the customizing object, you can use the Source Entry Utility (SEU) to verify the source file member has been created.

1. Type STRSEU (Start Source Entry Utility) and press F4 (Prompt).
2. Type the name of the source file you created for the workstation customizing source. (Specify QTXTSRC for

this value if you used the system default source file for the RTVWSCST command.)

3. Type the name of the library where the source file you specified is stored. If you used the system default source file for the RTVWSCST command, specify:
   - QGPL as the library for devices that use the emulator on the display or the emulator on the workstation controller.
   - *LIBL as the library for printers that use the host print transform function.
4. Type the name of the source file member you specified for the RTVWSCST command and press the Enter key.

The workstation customizing source should appear on your display. If the system cannot find the source, press F9 (Retrieve) as needed to look at the names you specified when you ran the RTVWSCST command. Try the verification procedure again.

If you still cannot find the source, verify that the source was created successfully by entering the Display Job Log (DSPJOBLOG) command and pressing F10 (Display detailed messages) to look at the detailed messages that were sent when you ran the RTVWSCST command.

Chapter 3. Changing the Source to Customize Your Workstations

Now that you have the source file for the customizing object, you can make the changes necessary for the device to display or print correctly. For this step you need to have the following information available:

- The list of characteristics you want to change and the corresponding hexadecimal values needed to make the changes (see Chapter 1, "Planning For and Setting Up Workstation Customizing")
- The tag language reference information
- The guide and reference information for the device

The workstation customizing source you retrieved in step **2** (see Figure 1-1 on page 1-2) contains the mapping tables for the specified device converted into the tag language structure. To change these tables, you need to understand the hexadecimal codes used by the device and the workstation controller or the host print transform function. Understanding the hexadecimal codes helps you make the changes and additions correctly. These hexadecimal codes are classified as control codes, escape sequences, and control sequences.

Using Control Codes, Escape Sequences, and Control Sequences

Some applications send commands to the devices they use in the form of control codes, escape sequences, or control sequences. These commands are sent within the data stream to provide instructions for the display and printer devices. The codes and sequences may be called other names by device manufacturers other than IBM. The important thing is that the ASCII command sequence is a string of hexadecimal data that directs the device to perform a specific function.

For IBM devices, control codes are the simplest hexadecimal codes and provide the basic device functions. A control code is usually a single byte of hexadecimal data that tells the device to perform a basic function. Some examples of functions that are called by control codes are the vertical tab, carrier return, and backspace functions.

Escape sequences and control sequences are strings of hexadecimal data, 2 or more bytes long, that tell the device to perform more complicated functions. For example, character highlighting, selecting character sets, and selecting fonts are among the functions called by escape sequences. Selecting the code page, setting the cursor position, and selecting global fonts are functions called by control sequences.

Within your workstation customizing source, these codes and sequences are indicated by specific tags. For example, there are two tags associated with underlining for an ASCII printer attached to a 3477 display. The STRUS (start underscore) and ENDUS (end underscore) tags each have a data parameter. The data parameter specifies the hexadecimal ASCII sequence that tells the printer to start and end underlining.

For example, you have a Hewlett-Packard** LaserJet** 4 printer and you want to underline text. The printer does not currently underline text, although it is capable of doing so, as stated in the printer reference manual. To add support for this function, change the hexadecimal data for the STRUS and ENDUS tags. The following example section of source shows this change:

```
   .
   .          /* Add next tags, hex values         */
   .          /* from the device reference manual  */
:STRUS
   DATA ='1B26643044'X.
:ENDUS
   DATA ='1B266440'X.
    .
    .
    .
```

Structure of Customizing Objects

Each source file member contains the source code for a single workstation customizing object. The source file member always begins with a WSCST (workstation customizing object) tag and ends with an EWSCST (end workstation customizing object) tag. Secondary tags must follow their associated primary tags in each source file member. This is strictly enforced by the workstation customizing object compiler. The content and structure of the source file member is determined by the device class, indicated by the DEVCLASS parameter of the WSCST tag. The general source structure for a workstation customizing source file member is shown in Figure 3-1.

```
:WSCST

         DEVCLASS=TWINAXDSP|ASCIIDSP|
         TWINAXPRT|ASCIIPRT|TRANSFORM.
       .
       .
       .
  /*tag or comment*/
       .
       .
       .
       .
       .
:EWSCST.
```

Figure 3-1. Syntax for the Workstation Customizing Object Tag

The possible values for the DEVCLASS parameter on the WSCST tag are:

TWINAXDSP An EBCDIC display connected by a twinaxial cable.

ASCIIDSP An ASCII display connected by way of a local ASCII workstation controller.

TWINAXPRT An ASCII printer connected to one of the following twinaxial displays: 3477, 3486, 3487, or 3488 that use the emulator on the display.

Note: The 3477 twinaxial display must be a Model H display. To verify this, see the procedure described in "Setting Up a 3477 Twinaxial Display" on page D-2.

ASCIIPRT An ASCII printer (connected by way of a local ASCII workstation controller) that uses the emulator on the workstation controller.

TRANSFORM An ASCII printer that uses the host print transform function.

When you retrieve the source file member for a device, the tags are set up in the appropriate structure and the values on the tags are from the system mapping tables for that particular device. The allowable structures for source file members (hereafter called source) for each device class are shown in the corresponding reference chapters in this part of the manual.

Using Source Entry Utility to Change the Tags and Keywords

To change the mapping tables used, change the tags and their associated keywords and values using the source entry utility (SEU). SEU is a function of the AS/400 Application Development Tools licensed program that is used to create and change source members.

Use the following procedure to change the source file members you created previously in step **2** (Retrieving the source).

1. Type

   `STRSEU SRCFILE(`*`library/source file name`*`)`

   on any command line. The library and source file you specify should be the one you created in step **2**. Press the Enter key.

2. Type a 2 (Edit) in the *Option* column next to the file member containing the customizing source you want to change and press the Enter key.

3. Use the tag language as described in Part 2 and Part 3 to change the values for mapping tables used by your device. The following list will help you find the tags specific to the device you are customizing.

   Twinaxial display (keyboard)
   See "Working with the Tag Language for Twinaxial Displays" on page 7-15.

   ASCII display
   See "Working with the Tag Language for ASCII Displays" on page 8-14.

   ASCII printer that uses the host print transform function
   See "Changing the Source" on page 10-2.

   ASCII printer attached to a 3477, 3486, 3487, or 3488 twinaxial display that uses the emulator on the display
   See "Working with the Tag Language for ASCII Printers Attached to Twinaxial Displays" on page 11-2.

   ASCII printer that uses the emulator on the workstation controller
   See "Working with the Tag Language for Directly Attached ASCII Printers" on page 12-5.

Note: A device that uses the emulator on the workstation controller may have entire tables you do not want to customize. You can delete the mapping tables you do not want to customize. For example, if you did not need to change the ASCII-to-EBCDIC mapping table, you could delete this table from your workstation customizing source. Doing this reduces the storage requirements on the workstation controller. The workstation controller then uses the default mapping tables associated with your device type to map the data properly between the device and the system. You should not however, delete the tags for individual functions, such as the tags for the printer definition table. When you delete these tags and values, the workstation controller does not map the associated function at all.

Examples showing this step of the workstation customizing procedure are provided for each device class in the corresponding reference chapters in Part 2 and Part 3 of this guide. When you have finished making the changes to the workstation customizing source, you can create the customizing object that allows you to customize your device.

When you have completed the changes to the source member, use the Create Work Station Customizing Object (CRTWSCST) command. This command creates the new customizing object for the device.

General Programming Considerations for Workstation Customizing

The general syntax for the tags used by the workstation customizing tag language is as follows:

```
tagid
    keyword = value
          .
          .
          .
    delimiter
```

The following rules apply to all the tags and keywords used for the workstation customizing tag language.

- Tag identifiers (IDs) begin with a colon followed by a character string. The total maximum length of a tag ID is 15 characters.
- Keywords that are associated with a tag ID have a maximum length of 10 characters.
- Parameters (for tags that require parameters) are represented by pairs of keywords and values.
- Values that are not hexadecimal are limited to a length of 32KB. The allowable length for hexadecimal values depends on the size of the table you are working with. The allowable length is shown in specific places within the tag descriptions.
- The ending delimiter for all tags is a period.
- Blanks are not allowed within the tag names, identifiers, keywords, or values. Blanks are allowed anywhere else in the source.
- The tag language for workstation customizing can be coded in free form. There are no column-dependent restrictions. Tags may begin in any column.
- Comments can appear anywhere in the source as long as they are not in the middle of a tag, keyword, or value. A comment is coded in the following form:

```
/* . . . character string . . . */
```

- Values associated with the keywords for the tags are in one of the following formats:

  Character constants
  These values are to be coded exactly as shown in the tag descriptions.

  Integer values
  These values may be signed or unsigned, 0 through 9. The default for integer values is unsigned, unless noted for a specific value. The value 0 is valid unless it is explicitly excluded in the value portion of the tag description.

  Hexadecimal values
  These values are coded using the following notation:

  ```
  'xxxx'X
  ```

  The xxxx within the apostrophes in the example represents an *even* number of the characters 0–9 and A–F. You can code a hexadecimal value as a single string, or the string can be broken into many contiguous strings. Contiguous strings are logically concatenated so that the resulting value is equal, internally, to a single large string.

Finding the Hexadecimal Values for the Tags

The hexadecimal and integer values required for the workstation customizing tags are in the reference manual or reference section of the manual that came with the device you want to customize. You create some of the data parameter values based on the combination of keys (and codes) you use to represent a control or command sequence. In these cases, you concatenate the hexadecimal values for several individual keystrokes to create a "single keystroke."

Twinaxial scan codes for the 122-key, 5250, and enhanced keyboards are listed in "Keyboard Scan Codes" on page 7-2.

For each device class supported by the workstation customizing functions, there is a corresponding reference chapter. The reference chapters provide the detailed technical information you need. The reference chapters also provide examples to help you work with the tag language and the workstation customizing source. A number of appendixes also provide keyboard layouts, source examples, additional procedures, and hexadecimal conversion tables. Appendix C, "Character to Hexadecimal Value Tables" provides the hexadecimal values for the United States ASCII character codes. "EBCDIC Character Code to Hexadecimal Value Chart" on page C-2 provides an EBCDIC character to hexadecimal value conversion table.

Errors and Recovery

The following list describes some errors that can occur during this step and what you can do to recover from the error.

- **Library containing the workstation customizing source not found**

  The library in which the workstation customizing source is stored does not exist or could not be found by the system. Use the Display Library (DSPLIB) command to verify that the library you specified is the correct library and to verify that the library exists. Use the Display Job Log (DSPJOBLOG) command to verify that you spelled the name of the library correctly when you used the Retrieve Work Station Customizing Object Source (RTVWSCST) command. If you specified the default value when you retrieved the source, the library name is QGPL.

- **Source file containing the workstation customizing source not found**

  The source file where the workstation customizing source is stored does not exist or could not be found by the system. Use the Display Library (DSPLIB) command to display the library you used to store the workstation

customizing source. Type the option to display the objects next to this library name and look for your source file. If it is not present, use the Display Job Log (DSPJOBLOG) command to verify that you spelled the name of the source file correctly when you used the RTVWSCST command. If you specified the default value when you retrieved the source, the source file name is QTXTSRC. If you cannot find the source file you created in Chapter 2, you may need to use the RTVWSCST command again.

- **Source file member containing the workstation customizing source not found**

  The source file member where the workstation customizing source is stored does not exist or could not be found by the system. Use the Display File Description (DSPFD) command specifying your source file and library names. Page down through the file information and look at the member names. If your source file member is not listed, use the Display Job Log (DSPJOBLOG) command to verify that the member was created successfully and that you spelled the name of the library correctly when you used the RTVWSCST command. If you cannot find the source file member, you may need to use the RTVWSCST command again.

- **Part of workstation customizing source typed over or deleted**

  As previously stated, the workstation controller is dependent on having all necessary mapping tables for a specific device. If one of the mapping tables embedded in your source is accidentally typed over or partially or completely deleted, you need to recover the lost tables and code. One way to do this is to use the RTVWSCST command to retrieve a new copy of the source you started to change. With this new source, you can either copy the missing parts into the source you were originally changing or, if you have not made too many changes to the original source, you can make these changes to the new copy of the source and delete your original source when the changes are complete.

Verifying That the Source Changes Are Complete

Your workstation customizing source changes are complete when you have changed all the hexadecimal and integer values for the DATA parameters on the tags representing the appropriate table values for the characters, command, and control sequences for your device. These are the values you listed in the planning steps of this procedure described in Chapter 1, "Planning For and Setting Up Workstation Customizing."

Chapter 4. Creating the Workstation Customizing Object

After you change the values in the source for the workstation or ASCII printer you are customizing, you are ready to create the customizing object. To do this, you use the Create Work Station Customizing Object (CRTWSCST) command to compile the source file members containing the tags you added or changed in step **3** (see Figure 1-1 on page 1-2). You need to use this command for each source file member you created and changed in step **3** of the customizing procedure.

Compiling and Creating the Workstation Customizing Object

When you are ready to use the CRTWSCST command, see the maps of matching parameters in Table E-1 on page E-1 and Table E-2 on page E-1. To compile and create the workstation customizing object, do the following:

1. Type CRTWSCST on any command line and press F4 (Prompt).
2. Specify a name for the customizing object. This name should be unique to the device for which the object is being created. For example, if you are planning to use this object for a group of ASCII workstations, you might want to name the object ASCWSC1. This is a required parameter.
3. Press F10 (Additional parameters) to display the remaining parameters.
4. Specify the name of the library and the name of the source physical file that contains the customized tag language. (You created this file in step **2** of the customizing procedure).
5. Specify the name of the source file member you created and changed in step **3** of this procedure.
6. Specify the authority you want to grant to users who do not have specific authority to the object.
7. Indicate whether or not this customizing object should replace an existing customizing object of the same name. (Specify *YES for the *Replace* field when you have made new changes to existing customized tables.)
8. Specify a text description for the customizing object if it does not already have one.

   Your description should be unique so that you can use it to help you identify a customizing object you created to customize one or more specific devices. If you do not specify any text here, the text is taken from the source member you specified previously.
9. Press the Enter key.
10. Press F10 (Include detailed messages) on the Command Entry display to look at the messages associated with the CRTWSCST command and to verify the command completed successfully.

After you create the customizing object, you can use it to customize one or more devices of the same type. For example, you may create a customizing object for an ASCII workstation and you may want to use seven of these workstations with your AS/400 system. You can use the single customizing object you created for all seven of the workstation configurations, or you can use a different customizing object for each of the seven identical workstations.

You can also save the customizing object in a save file or on tape. You can then distribute the object to another AS/400 system in a network. At the receiving system, the object can be restored. The object can then be used by:

- Workstation controllers attached to the receiving system
- The host print transform function on the receiving system

For more information about saving, restoring, and distributing objects, see the *Advanced Backup and Recovery Guide* and the *Central Site Distribution Guide*.

Examples showing this step of the workstation customizing procedure are provided for each device class in the corresponding reference chapters in Part 2 and Part 3 of this guide. When you have finished creating the workstation customizing object, you can change the device description for your device. For devices that use the emulator on the workstation controller or the display, you can vary the device off and on. This loads the object into the workstation controller for the device. For printers that use the host print transform function, you can start the printer writer to use the customized object for the printer.

Errors and Recovery

The following list shows some of the error messages that can be sent to the job log during this step of the workstation customizing procedure. The recovery for each error is in the help information on the AS/400 system for the individual message and is not duplicated here. This list is provided to give you an idea of the kinds of errors that can occur.

- Combination of keyword values not valid
- Combination of keyword values already specified
- Comment starting on line X not closed
- Ending delimiter on hexadecimal value not valid
- Ending period missing on tag
- Ending period not found for tag
- Format of keyword starting on line X not valid
- Hexadecimal value on line X not correct
- Internal error occurred
- Keyword not allowed on tag
- Keyword is not allowed
- Language type value not valid
- Length of associated value for keyword incorrect
- Mapping for code page occurred multiple times
- Required tag not found

- Required keyword not found for tag
- Required combination of keyword values missing
- Required mapping for code page not found
- Start comment delimiter found inside comment
- Starting delimiter on hexadecimal value not valid
- System monocase table not found
- Tag not found
- Tag on line X not allowed
- Too few tags found
- Too many tags found
- Value for keyword not allowed
- Workstation customizing object not created

The following is a list of errors that can also occur during this step in the procedure but do not correspond to a specific AS/400 error message.

- **You are not authorized to use the CRTWSCST command**

  Have the security administrator (QSECADM) for your system use the Grant Object Authority (GRTOBJAUT) command to give you the proper authority to use the command. You can then try the command again.

- **You are not authorized to use the library in which the source file is stored.**

  Have the security administrator (QSECADM) for your system use the Grant Object Authority (GRTOBJAUT) command to give you the proper authority to use the library (create and store files and members there). You can then try using the CRTWSCST command again.

- **Programming errors**

  The workstation customizing functions are based on the use of a tag language that has syntax rules much like any other programming language. When you have a coding error, you can use the messages in the job log associated with your AS/400 session to help you resolve the error. If the workstation customizing object is created and the character or function you are trying to add or change does not work, you need to go back to your device manual and verify the hexadecimal values you used for the data parameters. As with most types of programming, customizing a device involves some trial-and-error.

Verifying the Workstation Customizing Object Is Created

To verify that the workstation customizing object has been successfully created, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) to show the messages that were sent when you used the Create Work Station Customizing Object (CRTWSCST) command. If the object was created, a message is shown indicating that the creation was successful.

Note: If you need to delete a workstation customizing object from your system, you use the Delete Work Station Customizing Object (DLTWSCST) command.

Chapter 5. Testing the Customizing Object

After the customizing object has been successfully created, test the results of your changes. To test the results of your changes, specify the new object in the device description and then vary on the device. For this step, you need the following information:

- The name of the device description
- The library and name for the customizing object
- Security authorization for both the object and the library in which it is stored

For examples of this step in the workstation customizing procedure, see the example sections in the reference chapters for each device type in Part 2 and Part 3 of this guide.

Changing the Device Description

To specify the workstation customizing object in the device description for the device you are customizing, use the following procedure:

1. Type the appropriate change device description command (either CHGDEVDSP for a display or CHGDEVPRT for a printer) and press F4 (Prompt).
2. If the device you are customizing is an ASCII printer that uses the host print transform function, you must enable the host print transform function.
   a. Specify the value for your printer in the *Manufacturer type and model* field. You can press Prompt (F4) on the *Manufacturer type and model* field to display a list of values for this field.

      Note: If your printer is not shown in the list of manufacturer type and model values, specify *WSCST.
   b. Specify the following parameters in the device description:
      - Paper source 1
      - Paper source 2
      - Envelope source
      - ASCII code page 899 support

      For a description of these parameters, see the *CL Reference*.
3. Press F10 (Additional parameters) and page through the information until the *Workstation customizing object* field is shown.
4. Type the name of the workstation customizing object followed by the name of the library where it is stored and press the Enter key.

Varying On the Device

Use the Vary Configuration (VRYCFG) command to vary on the devices you have customized. For devices using the emulator on the workstation controller or the display, the mapping tables you have changed are downloaded to the workstation controller. The mapping tables are actively used to convert data between the device and the AS/400 system. For printers using the host print transform function, the mapping table you changed is not used when the device is varied on. The printer still needs to be varied on, however. Do the following:

1. Type VRYCFG on any command line and press F4 (Prompt).
2. Type the names of the devices you are ready to vary on (from the device descriptions you just changed).
3. Specify the device type (*DEV in most cases).
4. Specify *ON for the status, and press the Enter key. The devices move to a status of `Vary on pending`.

To view the status of the devices, use the Work with Configuration Status (WRKCFGSTS) command. Your devices should show a status of `Vary on pending`, `Active`, or `Signon display`.

Note: If for any reason, your displays are not functioning properly or the customization is not successful (that is, the displays do not respond at all), you can use the QCONSOLE display to reset the system so that the original system mapping tables are used until you can correct the customizing source and recompile the customizing object.

Starting the Printer Writer for a Printer That Uses Host Print Transform Function

Use the Start Printer Writer (STRPRTWTR) command to start the printer writer for the customized printer that uses the host print transform function. When you start the printer writer, the host print transform function uses the customized AS/400 mapping table to transform the data from the AS/400 system to the printer for spooled print jobs. For nonspooled print jobs, the customized mapping table is used when the job is printed.

Note: For ASCII printers attached to the AS/400 system through PC Support, the printer writer is automatically started when the workstation function printer session is started.

Errors and Recovery

The following list describes some errors that can occur during this step and the things to do to recover from the error.

- **Unchanged characters and control sequences not displayed or printed**

  This can happen when the tags or the mapping tables for characters or control sequences that you did not customize have been accidentally deleted from the source. The customizing object created from such source no longer contains the necessary mappings for the characters and control sequences you did not want to change. The workstation controller or the host print transform function has no information to use to map these characters and control sequences.

  Save the source from which these tags or tables have been deleted. Create a new source file using the RTVWSCST command just as you did when you created the original source for the object. Using the original source as a guide, make the changes for customization as you did for the original source, taking care not to delete any of the tags or mapping tables. When this is complete, use the Create Work Station Customizing Object (CRTWSCST) command to create the object again. Be sure the customizing object is specified in the device description. Then vary on the device. If the device is using the host print transform function, start the printer writer for the printer. The results should be what you expected.

- **Mapping table changes (customizing) not producing expected results**

  This can occur when one or more of the hexadecimal or integer values you changed in the workstation customizing source is not correct. For example, because some of the hexadecimal values are very large, you may have typed a value incorrectly or recorded a different hexadecimal value from the device manual than the value you needed. First, check the list of changes you created when you did the planning for customizing (step 1 in Figure 1-1 on page 1-2). Verify that the values you recorded when planning the customizing are the correct values by rechecking the device reference manual. Be certain that the values you recorded are the correct values for the keys, characters, and control functions you wanted to change. Next, verify that the values you entered into the customizing source match the values you recorded when planning the customization. Make any necessary changes. Use the Create Work Station Customizing Object (CRTWSCST) command to create the customizing object again. Vary on the device. If the device is using the host print transform function, start the printer writer for the printer. Then test the keys or functions you changed.

- **Customizing object or library not found**

  Either the workstation customizing object or the library where it resides was not found by the system. Use the Display Library (DSPLIB) command to look at the contents of the library where the customizing object should be stored and verify that the library you specified is the correct library and that the object has been created.

  If the object is in the list of objects for the library, check the spelling of the object name. Verify that you typed this name correctly in the device description.

  If the customizing object is not in the library, check to see if the source member was added to the user file QTXTSRC in the following libraries:

  - QGPL for devices that use the emulator on the display or the emulator on the workstation controller.
  - *LIBL for printers that use the host print transform function.

  You may want to move it to one of your own libraries. If you cannot locate the customizing object at all, use the CRTWSCST command to create the object again.

- **Library not found**

  The library where the workstation customizing object should reside does not exist or could not be found by the system. Use the Display Library (DSPLIB) command to verify that the library you specified is the correct library and to verify that the library exists. Verify that you spelled the name of the library correctly in the device description.

- **Value not valid for parameter**

  The value you entered for one of the command parameters is not valid or is incorrect. Use the AS/400 online help information or the *CL Reference* to verify the values of parameters. Change the parameter value and try the command again.

- **You are not authorized to the workstation customizing object**

  Have the security officer or the owner of the workstation customizing object use the Grant Object Authority (GRTOBJAUT) command to change the authorities for the object so that you can use it. The authority can be set to *USE or *ALL for one or more user IDs. *ALL also allows you to change and delete the object as well.

- **Workstation customizing object damaged**

  Delete the damaged object. Either use the source you used to create the object originally and create a new workstation customizing object, or If you periodically save your system and data, use the Restore Object (RSTOBJ) command to restore the object. If the problem persists, it may be a system problem. Use the Analyze Problem (ANZPRB) command to describe and report the problem to your service representative.

- **You are not authorized to the library where the workstation customizing object is stored**

  Have the security officer or the owner of the library in which the workstation customizing object is stored use

the Grant Object Authority (GRTOBJAUT) command to change the authorities for the library so that you can use it. The authority can be set to *USE or *ALL for one or more user IDs. *ALL also allows you to change and delete the object as well.

- **Device not varying on**

  There are three cases for this problem. The first involves the storage limitations of the workstation controller. The second case involves the keyboard type you selected when you retrieved the workstation customizing source. The third case involves a mismatch between the language you selected for the workstation customizing source and the language you specified in the description for the device. Table 5-1 describes what you can do to look at and correct this problem.

  Note: This problem does not occur for printers customized through the host print transform function.

Table 5-1. Device Does Not Vary On

Problem	Recovery
Workstation controller storage capacity exceeded	If you have varied a device off and on many times to test a workstation customizing object, new or changed mapping tables have been downloaded to the workstation controller each time you have varied on the device. It is possible that the storage for the workstation controller is full to capacity. To verify this, check the error messages in the QSYSOPR message queue. If the workstation controller storage is full, you need to vary off the controller, and then vary it on again specifying RESET(*YES). Use the following values on the Vary Configuration display to vary on the controller, specifying your controller name for *ctlname*. • CFGOBJ(*ctlname*) • CFGTYPE(*CTL) • STATUS(*ON) • RESET(*YES) To display the *Reset* parameter on the Vary Configuration display, you need to press F10 (Additional parameters).
Keyboard attached value specified on RTVWSCST command is not correct	If you are customizing a twinaxial display or an ASCII printer attached to a twinaxial display, verify that the value you selected for the keyboard attached parameter when you retrieved the workstation customizing source is the correct keyboard type for your display. "Keyboard Translation Table (TKBDTBL) Tag" on page 7-15 describes the language types and keyboards that match the IBM-supported twinaxial displays and the shift states supported for those keyboards. If the keyboard type does not match one of those supported by the display, the device will not vary on and a message indicating the vary on status is `Failed` will appear on the Configuration Status display. If you have specified an incompatible value for the keyboard attached parameter when you retrieved the source for customizing a twinaxial display or ASCII printer attached to a twinaxial display, you need to use the Retrieve Work Station Customizing Object Source (RTVWSCST) command again, specifying a compatible keyboard for this value. You can then make the same changes to this new source that you made in the original source, create the new workstation customizing object, and try to vary on the device again.
Language type in workstation customizing object does not match language type specified in device description	When you are customizing any type of device, you must be sure that the keyboard language type you specify when you retrieve the source exactly matches the keyboard language type (for displays) or the language type (for printers) in the device description. If these values do not match, the device fails to vary on and messages are sent to the job log. To correct this, you must change either the language type you selected to retrieve the workstation customizing source, or the language type in the device description. If you decide to change the value you specified for the source, you need to use the Retrieve Work Station Customizing Object Source (RTVWSCST) command again, specifying the correct language type for this value. You can then make the same changes to this new source that you made in the original source, create the new workstation customizing object, and try to vary on the device again.

- **Workstation controller failed**

  The workstation customizing object has caused the workstation controller to fail. Check the QSYSOPR message queue and the error log to look for the cause of the failure. You may need to run problem analysis from the QSYSOPR message queue, create an APAR using the Create Authorized Program Analysis Report (CRTAPAR) command, or contact your software support specialist. You may have a problem with the workstation controller.

Verifying a Successful Vary On

To see whether or not the device has been varied on, use the Work with Configuration Status (WRKCFGSTS) command to look at the status of the device. The status should be `Active`, `Sign On display`, `Varied on` or `Vary on Pending`. When the display or printer is varied on, you should check to see whether or not the customization procedure was a success. You can do this by looking at the plan

you created and verifying that the characteristics you wanted to change are working (printed or displayed correctly). If all is working according to your plan, then the workstation customizing procedures are complete. Otherwise, you may need to make more changes to your workstation customizing source. To do this, return to Chapter 3, "Changing the Source to Customize Your Workstations" and complete the workstation customizing procedures again from step **3**, changing your source.

Part 2. Display and Keyboard Customization Technical Reference

Chapter 6. Customizing Displays and Keyboards

This chapter provides a brief overview of the workstation customizing functions to prepare you for customizing displays and keyboards. It describes the two different methods you can use to customize the functions of displays and keyboards. It helps you understand which functions can be customized and whether your displays or keyboards can be customized.

Determining Whether a Display or Keyboard Can Be Customized

You can use the test function provided by the display to determine whether or not certain characters are already supported.

Before you start to use the test function, be sure the AS/400 Sign On display is showing. If the Sign On display does not appear on the device, you may need to change the device type (DEVTYPE) parameter in your device description. Change the device type to a type that is more closely matched to your display device. (For a procedure to help you get to a Sign On display, see Appendix D, "Setting Up to Customize a Display.")

For twinaxial displays, use the following keystrokes to call the test function. The set of keystrokes you use is determined by the type of keyboard that is attached to the twinaxial display.

5250	Press the Cmd key followed by the Backspace key
122-key	Press the Alt key and the Play key simultaneously
Enhanced	Press the Alt key and the Help key simultaneously

For an ASCII display, press the key sequence Esc+t to call the test function.

The test function uses the tables that were loaded into the workstation controller when you varied on the display. These tables are the system-supplied tables if you did not specify a customizing object in the device description before you varied on the display. If you did specify a customizing object in the device description before you varied on the display, these tables are user-defined tables.

If the characters you need to display are not shown, check the *National Language Support Planning Guide* to see if they are defined for the code page you specified in the device description. If they are defined for that code page, check to see if the manual for the device indicates you can change the code page. (Some displays do not provide this function.) Remember that workstation customizing functions support only the characters and functions that the device and the workstation controller support.

Understanding Function Support for Displays and Keyboards

The workstation customizing functions support twinaxial displays and ASCII displays attached to the AS/400 system by way of a workstation controller. The workstation controller uses mapping tables that are downloaded from the AS/400 system to support the input/output (I/O) for these devices.

The workstation controller processes I/O for a display as inbound processing and outbound processing. Inbound processing is keystroke and function processing from the display to the AS/400 system. Outbound processing is character and function processing from the AS/400 system to the display.

The workstation customizing functions allow you to customize only the inbound, or keystroke processing, for a twinaxial display. The workstation customizing functions allow you to customize both inbound and outbound processing for an ASCII display.

Customizing Twinaxial Displays

The workstation customizing functions allow you to customize keystroke processing for the following supported twinaxial displays:

3179	3486
3180	3487
3196	3488
3197	5251
3476	5291
3477	5292

You can also customize keystroke processing for unsupported twinaxial displays. For more information on customizing unsupported twinaxial displays, see "Customizing Unsupported Twinaxial Displays" on page 7-1.

Customize the characters and functions called from the keyboard of a twinaxial display by changing the hexadecimal values in the twinaxial keyboard translation table. You may also need to change the keyboard language type specified in the device description if:

- You are using a language other than your primary language
- You want to use a special character set

The workstation controller uses the twinaxial keyboard translation table to translate the keyboard scan codes from a twinaxial keyboard. The keyboard scan codes are translated into either character codes or internal function codes for processing by the AS/400 system. For complete information on

customizing the twinaxial keyboard translation table, see Chapter 7, "Customizing Twinaxial Displays."

Customizing ASCII Displays

The workstation customizing functions allow you to change both inbound and outbound processing for the following supported ASCII displays:

3101
3151
3161
3162
3163
3164
D220 (Data General Dasher** D220)
T910 (TeleVideo** 910)
T925 (TeleVideo 925)
T955 (TeleVideo 955)
V100 (DEC** VT-100)
V220 (DEC VT-220)
W30 (Wyse** WY30**)
W50 (Wyse WY50**)
W60 (Wyse WY60**)

You can also customize inbound and outbound processing for unsupported ASCII displays. For more information on customizing unsupported ASCII displays, see "Customizing Unsupported ASCII Displays" on page 8-1.

Customize the characters and functions called from the ASCII display and keyboard by changing the hexadecimal values in a set of system-provided mapping tables. For a better understanding of the default character and control sequence mappings, retrieve the source for the display and keyboard as described in Chapter 2, "Retrieving the Workstation Customizing Source." You can then use the source entry utility (SEU) to view the default character and control sequence mappings.

Customizing the System-Supplied Mapping Tables:

Customizing the ASCII workstation for use with the AS/400 system is accomplished by changing the hexadecimal values in one or more of the following tables:

- ASCII-to-EBCDIC mapping table

  This table converts an ASCII character value in the range '20'X to 'FF'X into a single EBCDIC character code value.

- ASCII-to-keyboard function mapping table

  This table converts an ASCII control character value or a character sequence beginning with a control character into one of the following:

  - A twinaxial display function key request
  - A local display function (screen refresh, toggle display indicators, or terminal disconnect)
  - A request to set the state (shift-in and shift-out support) for processing subsequent data from the device

- EBCDIC-to-ASCII mapping table

  This table converts an EBCDIC character value in the range '40'X to 'FE'X into a single ASCII character code value. This table is also used to map twinaxial display control characters in the range '00'X to '1F'X into ASCII character codes or ASCII control codes.

- Update screen table

  This table is used by the workstation controller to perform the following types of operations with the ASCII display:

  - Set the type of highlighting that is used at a given screen location
  - Set the screen position at which data should be written
  - Clear the screen
  - Sound the audible alarm at the display
  - Initialize the display parameters

The choices you make for character and control sequence mapping are based on the:

- Reference information for the ASCII display
- Supported device you select when you retrieve the tables
- Characteristics of the display
- National language code page and character set you are using

For more complete information on customizing ASCII displays and keyboards, see Chapter 8, "Customizing ASCII Displays."

Chapter 7. Customizing Twinaxial Displays

This chapter provides detailed technical information to help you customize twinaxial display keyboards. Background information about the workstation controller, keyboard mapping tables, scan codes, and languages is provided, followed by the descriptions of the tags you use to customize twinaxial displays.

For a twinaxial display, the OS/400 workstation customizing functions provide you with a way to create a customized version of the keyboard translation table. The translation table is used by the workstation controller to convert scan codes from the twinaxial display keyboard to EBCDIC codes or function codes that the AS/400 system can interpret. To customize this table, you retrieve a workstation customizing object source (a source file member) that contains the entries for a translation table corresponding to the device type, keyboard type, and language specified in the device description for the display. The source and its structure allow you to create an object that is compatible with the system translation tables normally used by the workstation controller.

Note: The OS/400 licensed program does not support the customizing of double-byte character set (DBCS) keyboards.

Types of Displays and Keyboards

The AS/400 system and the twinaxial workstation controller use the following pieces of information to determine the translation table to be used for a particular twinaxial display:

The type of display attached
The type of keyboard attached to the display
The keyboard language type

There are five basic types of keyboards you can attach to displays on the twinaxial workstation controller. The different keyboard layouts are distinguished from each other by the physical layout of the keys. For example, the number of keys, the relative position of different groups of keys, and the way in which certain types of numeric and graphic characters are laid out on the keyboard are different for different national languages. Keyboard layout is usually either a typewriter or a data entry style. For illustrations of the different keyboard layouts, see Appendix A, "Twinaxial Keyboard Layouts."

Table 7-1 shows the different types of twinaxial displays that may be attached to a twinaxial workstation controller. For each display type, the table shows the types of keyboards that can be attached and the parameter value you would use for the Keyboard attached parameter on the Retrieve Work Station Customizing Object Source (RTVWSCST) command to retrieve the object source for a twinaxial display with that type of keyboard.

Note: There are no twinaxial displays to which all the different types of keyboards can be attached.

Table 7-1. IBM Twinaxial Displays and Supported Keyboards

Display	Supported Keyboards	Parameter Values
5251	5250 Typewriter 5250 Data Entry	*TYPE5250 *DATA5250
5291	5250 Typewriter 5250 Data Entry	*TYPE5250 *DATA5250
5292	5250 Typewriter 5250 Data Entry	*TYPE5250 *DATA5250
3180	122-key Typewriter 122-key Data Entry	*TYPE122 *DATA122
3179	122-key Typewriter	*TYPE122
3196	122-key Typewriter Enhanced Typewriter	*TYPE122 *ENHANCED
3197	122-key Typewriter Enhanced Typewriter	*TYPE122 *ENHANCED
3476	122-key Typewriter Enhanced Typewriter	*TYPE122 *ENHANCED
3477	122-key Typewriter Enhanced Typewriter	*TYPE122 *ENHANCED
3486	122-key Typewriter Enhanced Typewriter	*TYPE122 *ENHANCED
3487	122-key Typewriter Enhanced Typewriter	*TYPE122 *ENHANCED
3488	122-key Typewriter Enhanced Typewriter	*TYPE122 *ENHANCED

Note: The 3488 display station must be specified as a 3487 display station when using the RTVWSCST command. The workstation customizing functions support a 3488 display station as a 3487 display station.

The table of twinaxial displays does not include double-byte character set devices. It also does not include any devices that can be attached, but for which the workstation controller does no keystroke processing, such as personal computers running PC Support/400. Keystroke processing for these devices is done in the device itself; therefore, no translation table is downloaded to the workstation controller for these devices.

Customizing Unsupported Twinaxial Displays

When you want to customize an unsupported twinaxial display for use with the AS/400 system, you still need to retrieve a source file member based on a supported IBM device type, keyboard type, and keyboard language type. To do this effectively, you need to find out which supported IBM twinaxial display your unsupported display emulates. You can try using the automatic configuration functions provided by the AS/400 system to help you determine the most similar IBM-supported device type and model. You can also try

reading the reference manual, talking to the manufacturer of the device, or talking to your technical support specialist.

To determine the type of keyboard that most closely matches your keyboard, compare your keyboard to the keyboard illustrations in Appendix A, "Twinaxial Keyboard Layouts."

To determine the keyboard language type for your display, check the device description for the display (if it exists), or check with your system administrator to verify that the language type used by the system is the language you want to configure for the device. The language type used by the system is defined in the QCHRID system value.

Note: The keyboard language type you specify when you retrieve the workstation customizing source for a display must exactly match the keyboard language type you specify in the device description for the display. If these values do not match, the device will not vary on when you specify the workstation customizing object in the device description.

Working with Keyboard Translation Tables and the Workstation Controller

The twinaxial workstation controller processes all keystrokes entered at an attached display. The workstation controller then determines the function to perform for a given key using the keyboard translation table. This table allows the workstation controller to convert the scan codes generated by display keyboards into either character codes or internal function codes, which are then used to determine the function to be performed.

The workstation controller supports a wide variety of display types, keyboard styles, and languages. Using different translation tables, the controller can support all the different combinations of displays, keyboards, and languages. There are separate translation tables to support each unique combination of display type, keyboard style, and language.

Because of the wide variety of displays, keyboard styles, and languages supported by the workstation controller, a large number of translation tables exist on the AS/400 system. However, only a small number of these tables are stored in the workstation controller when it is first varied on.

The OS/400 licensed program determines the particular translation table required to support the unique requirements of a given display. When the display device and the workstation controller are first varied on, the correct translation tables are downloaded to the controller.

Keyboard Scan Codes

When you press a key on a twinaxial keyboard, the display passes a code back to the workstation controller that uniquely identifies the key that you pressed. This code is called a keyboard scan code.

Every key on the keyboard has a particular scan code value assigned to it. The assignment of a scan code is not directly related to the label on the key top. A key at a certain position on a particular keyboard layout always has the same scan code value assigned to it. For example, a key at a given location on a 5250-style keyboard always returns the same scan code, although the key may be labeled differently for different national languages.

Keyboard scan codes are passed to the controller in a 1-byte code. The 7 least significant bits within this byte contain the actual scan code. Some keys, called make/break keys, use the most significant bit of this byte to indicate whether the scan code corresponds to a make of the key or a break of the key. When a make/break key is pressed, a scan code is generated when the key is first pressed (**make**) and another scan code is generated when the key is released (**break**).

The shift keys (left and right), Alt key, Shift Lock key, and Caps Lock key on a keyboard are usually all make/break keys. In general, it is not recommended that you change these keys using the workstation customizing functions.

Figure 7-1 shows the format for a scan code byte passed to the workstation controller from a display.

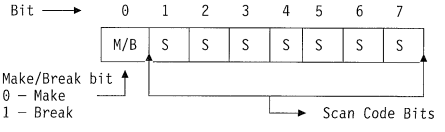


Figure 7-1. Scan Code Byte Format

Scan code values are assigned in the range '01'X to '7F'X. Not all values in this range are assigned to keys on the keyboard. The set of valid scan codes depends on the style of keyboard. For example, the set of scan codes generated by a 5250-style keyboard is different from the set of scan codes generated by an Enhanced keyboard.

Figure 7-2 and Figure 7-3 show the scan code values assigned to each of the different key positions for the 5250 and Enhanced keyboard layouts, respectively. The scan codes for corresponding keys on the 5250 typewriter and 5250 data entry keyboards are the same.

Note: Some of the existing 5250-style translation tables are set up to handle scan codes that cannot be generated by an actual 5250-style keyboard. Attached devices using 5250 emulation use these scan codes to support additional capabilities that are not available with the 5250-style keyboard.

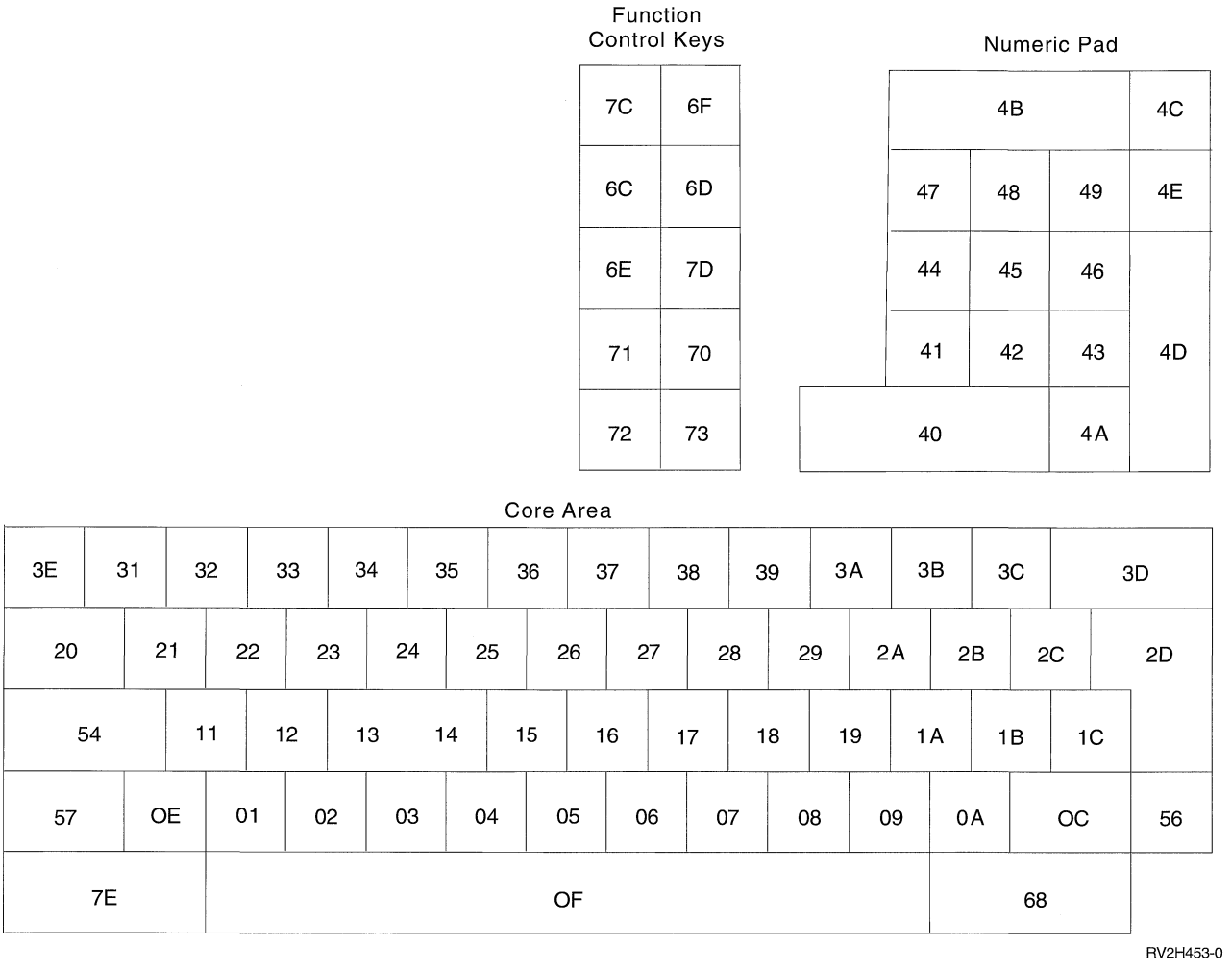


Figure 7-2. 5250-Style Keyboard Scan Codes

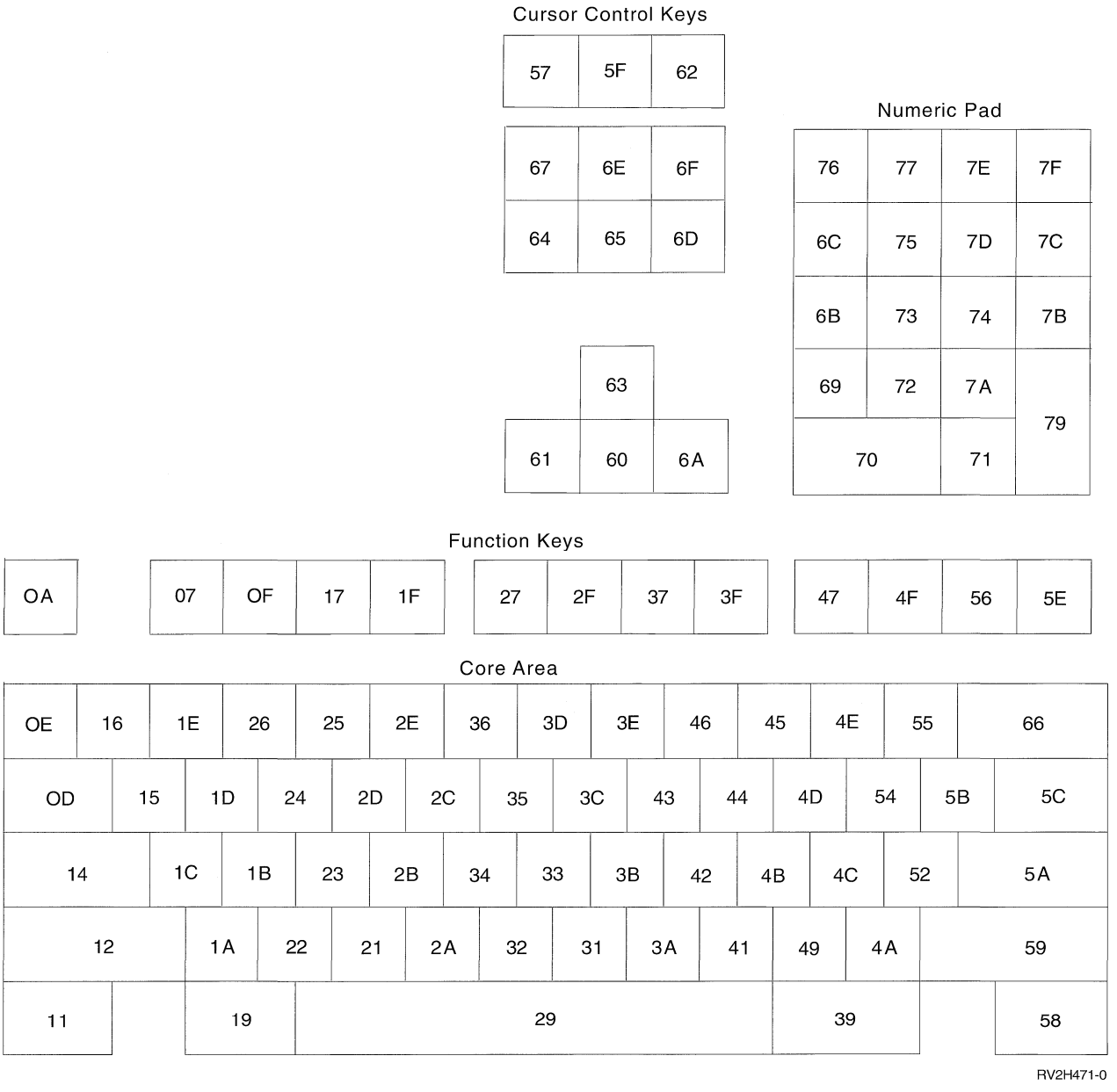


Figure 7-3. Enhanced Keyboard Scan Codes

The 122-key keyboard is shown in Figure 7-4 on page 7-5. Displays with this layout that attach to the twinaxial workstation controller always operate in what can be referred to as a 5250-style emulation mode. When a key on a 122-key keyboard is pressed, the scan codes assigned to each individual key are not passed directly back to the workstation controller. Instead, the display passes back a scan code or a sequence of scan codes corresponding to the scan code required for that same function on a 5250-style keyboard. For example, pressing the key for PF1 on a 122-key keyboard actually results in the generation of 5250 keyboard scan codes for the Cmd key and the key in the core area of the keyboard normally labeled with the number 1.

Note: For certain national languages, the 122-key displays are capable of also passing back additional scan codes that cannot be generated on an 5250 keyboard.

In handling a particular function on the 122-key keyboard, the 5250 keyboard emulation process also takes care of generating any 5250 scan codes for shift key sequences that are required to emulate the entry of that same function on a 5250 keyboard. For example, pressing the F13 key on a 122-key keyboard generates the 5250 keyboard scan codes for keys containing the Cmd key, the shift (make) key, the top-row key labeled with the number 1, and finally, another shift (break) key.

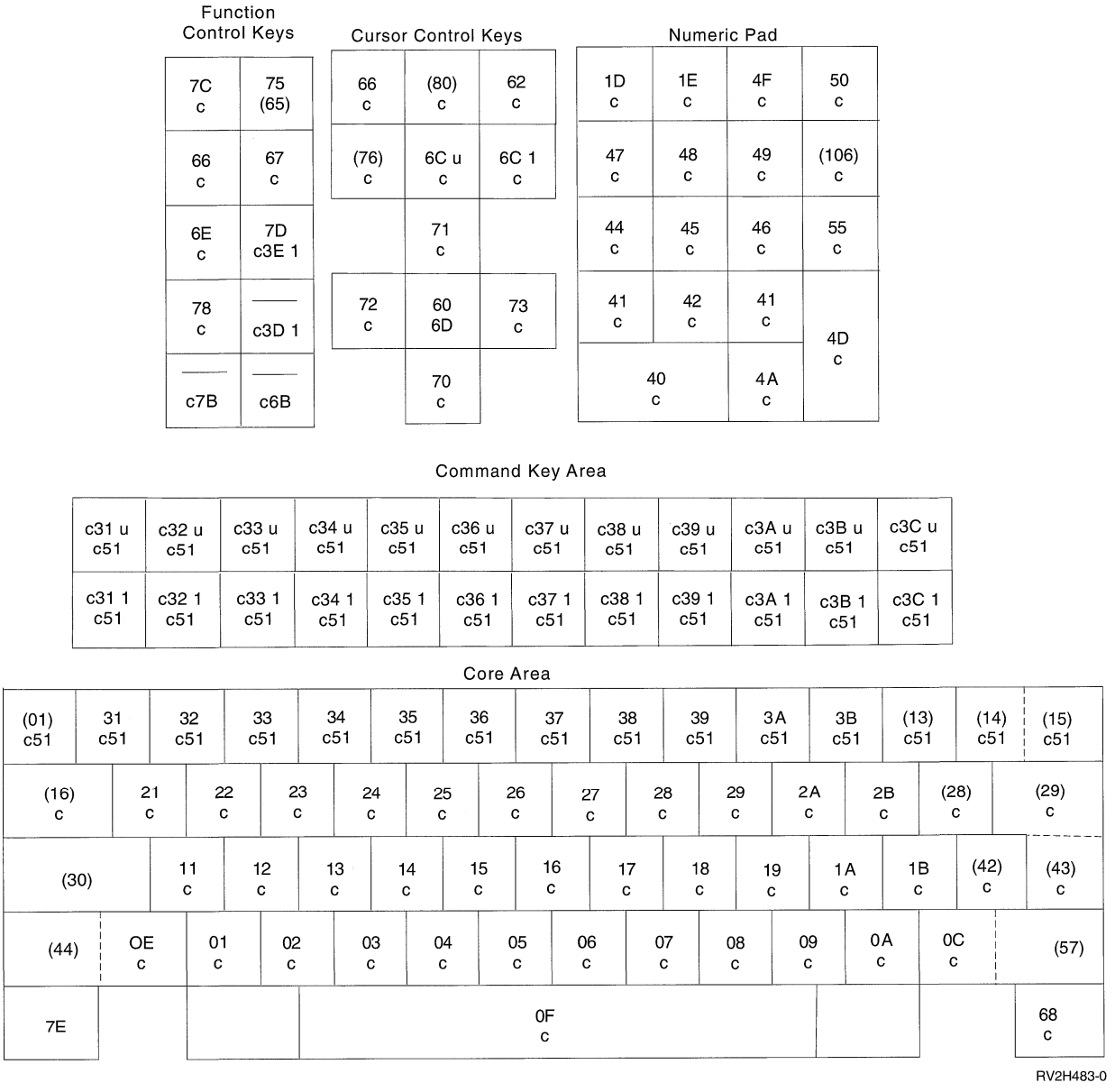


Figure 7-4. 122-Key Keyboard Scan Codes

In Figure 7-4, the top line shown for each key shows the emulation scan code generated for that key. The scan codes are for the key when:

- It is pressed simultaneously with the shift key (the upper-shift state)
- It is pressed by itself (the lower-shift state)

The bottom line shown for each key shows the emulation scan code generated for that key when it is pressed simultaneously with the ALT key.

The following additional symbols are used in Figure 7-4:

c Emulation generates scan code for 5251 CMD key (X'6F'), followed by the scan code shown on the top line.

c*XX* Emulation generates scan code for 5251 CMD key (X'6F'), followed by the scan code (*XX*) specified.

u Emulation forces upper shift state when generating this scan code sequence

l Emulation forces lower shift state when generating this scan code sequence

( ) Parentheses around a scan code, for example (28), indicate that the emulation scan code generated for that key is dependent on the keyboard type. The following keyboards have emulation scan codes that are dependent on the keyboard type:

- 1A data processing keyboard
- 1A data entry keyboard
- 1A Katakana data processing keyboard
- 1A Katakana data entry keyboard

Table 7-2 on page 7-6 shows the emulation scan codes generated for keys that are dependent on the keyboard layout.

Table 7-2. 1A Keyboard Type Dependent Emulation Scan Codes

Key Number	1A Data Processing	1A Data Entry	1A Katakana Data Processing	1A Katakana Data Entry
(01)	3E	3E	30	3E
(13)	3C	3D	3C	30
(15)	3D	3F	3D	3F
(16)	20	5E	20	5E
(28)	2C	2E	2C	2D
(30)	54	10	57	57
(42)	1C	1C	1C	1F
(43)	2D	2D	56	56
(44)	57	57	53	53
(57)	56	56	52	52
(65)	6F3D	6F3D	6F61	6F3D
(76)	0C	63	63	63
(80)	4C	65	4C	65
(106)	4E	2C	4E	2C

There are slight differences in other national language versions of keyboards from the U.S. English versions; for example, some keys in the core area can be arranged differently. These other versions are not shown; however, a more complete listing of the supported keyboards is available in the *National Language Support Planning Guide*.

National Language Requirements

Each country and national language type supported by the OS/400 licensed program establishes standards to which data processing equipment used in that country must conform. For each national language, the two main standards affecting the local twinaxial workstation controller are the following:

- EBCDIC code page standards
- Keyboard layout for displays

EBCDIC Code Page Standards

The EBCDIC code page standard for a given national language provides the set of graphic characters that an attached device should be able to display. It also identifies the EBCDIC code point value that should be used to represent each graphic character. For example, the EBCDIC code value of 'F1'X is used to identify the graphic character for the number 1. Different countries or national language groups have their own code page standards to which equipment manufacturers are expected to conform.

For most of the Latin-based languages supported by the OS/400 licensed program, such as those used in US/Canada, Latin America, and most of Western Europe, a common set of characters, known as character set 697, makes up the different code pages. The set of characters in the code pages is the same from one code page to the next; however, the code point values assigned to given characters may be different from one code page to the next.

One of the code pages that contains all the characters in character set 697 is the Multinational EBCDIC code page, or code page 500. This code page does not conform to the code page standard of any particular national language. It does, however, contain all the same characters as those code pages used by the Latin-based languages. Displays that support the Multinational EBCDIC code page also usually support one or more national language code pages containing the characters in code page 697. For more information about code pages and EBCDIC code point values, see the *National Language Support Planning Guide*.

AS/400 systems required to handle data originating from a number of different countries are often set up to use this Multinational EBCDIC code page. You can set up most displays and printers that are manufactured for a particular national language to run using either the specific EBCDIC code page defined for that national language or the Multinational EBCDIC code page.

The workstation controller uses the translation tables to convert the keyboard scan codes into EBCDIC character codes. This conversion is done taking into account the layout of characters on the keyboard for that country and the EBCDIC code page to which a scan code is being mapped. The conversion is done differently depending on whether the conversion is being done for a specific national language EBCDIC code page or the Multinational EBCDIC code page. This means that for the keyboards of many countries, there are separate translation tables so that many countries using national language support can handle the conversion into either a specific national language EBCDIC code page or the Multinational EBCDIC code page.

Keyboard Layout Standards

Every country has standards for the layouts of keyboards that are manufactured for use in that country. These standards define the country's national language requirements for the layout of keys on a keyboard containing alphanumeric data. The standards apply to what is generally called the core area on a keyboard.

Note: Some countries use more than one national language.

Over time, a country may change its standards. Not only the keyboard standard may change, but the EBCDIC code page standard may change as well. New displays and printers are expected to conform to the new standards, but existing hardware devices are usually not changed to meet the new standards. The workstation controller supports the existing displays that conform to older standards, as well as the new displays that conform to the latest standards. For example, many displays with the 5250 keyboard layout that are supported by the workstation controller conform to older standards.

Customizing Restrictions for Twinaxial Display Keyboards

When considering the customizing of a twinaxial display, keep in mind the following restrictions of the OS/400 workstation customizing functions. Read these restrictions carefully before you continue to customize a twinaxial display.

- The workstation customizing functions require that you provide hexadecimal data corresponding to the characters and functions supported by your twinaxial display. You must have the reference manuals for the display to obtain this information.
- You cannot customize the mapping of characters sent to the display from AS/400 application programs. Only the characters and functions that are entered from the keyboard can be customized.
- In most cases, the workstation hardware supports only a subset of the code pages that the AS/400 system can support. Be sure that you select and set the correct code page for the hardware and then select a compatible keyboard language type in the device description for the workstation.
- Twinaxial keyboards usually support different states by combining keystrokes. For example, pressing the Alt key or Shift key in combination with other keys may display a special character or call a display function. However, there is a restriction on the 122-key keyboard. The Alt key on 122-key keyboards is not supported as a separate shift state in the 122-key keyboard translation tables. To find out which states your keyboard supports look at the following sections:

  See Appendix A, "Twinaxial Keyboard Layouts" to determine the keyboard type for the workstation you want to customize.

  See Table 7-22 on page 7-18 and the corresponding mode and shift table to determine which states are supported for your keyboard.
- Keys used for text processing, such as those used with the OfficeVision/400* licensed program, cannot be directly customized. These keys may be affected when you customize other keyboard functions. See "Keyboard Functions Not Specified in Translation Tables" on page 7-14 for information on how customizing other keyboard functions can affect the text processing keyboard functions.
- It is not recommended that you customize keys such as Alt, Caps Lock, Shift, and Shift Lock as this can produce unpredictable results.
- The twinaxial translation tables contain entries that indicate which characters on the keyboard are subject to monocasing. However, you cannot customize the keyboard so that characters not currently subject to monocasing are now converted to uppercase. Entries in the keyboard translation table for a twinaxial display that are subject to monocasing have a first byte equal to '0A'X. The workstation controller uses separate internal tables that allow it to determine the EBCDIC character to be displayed when monocasing is used to convert a given character. The internal monocase tables cannot be customized.

  If the internal monocase table does not indicate that an EBCDIC character has an uppercase character associated with it, monocasing is not done even if you change the translation table entry to designate the character as subject to monocasing support.

  On the other hand, if the translation table entry for a character currently identified as subject to monocasing is changed so that monocasing is no longer supported (first byte value is changed from '0A'X), monocasing is no longer done for that character.
- The display must support the code page associated with the character identifier (CHRID) assigned to the display in the device description, and the CHRID must match the coded character set identifier (CCSID) of the job you are running on that display.
- When you choose the keyboard language type for the display, you must select one that uses the same code page mapping supported by your display. Many twinaxial displays provide a test request function, which you can use to show the characters that the display is capable of supporting. Use this function when available to verify that the graphics character mapping is correct. You cannot customize this mapping because this character set is built into the device. All you can do is specify a compatible choice for the language type in the device description, and specify the same language type for the customizing object (these should match).

There are also certain scan codes that you should not customize. The translation table entries for these scan codes should not be changed in the source you retrieve.

Changing these particular translation table entries could cause your keyboards to be unusable. For example, if certain shift key functions were converted to scan codes for keys that are not make/break keys, a user might not be able to get out of a certain shift state. For another example, when the scan code assigned to the Cmd key function in a 122-key keyboard translation table is changed, many of the Cmd key functions on a display with that keyboard become inaccessible because there is no single key on the 122-key keyboard that generates this particular scan code. The scan code for the Cmd key is part of a scan code sequence generated because the 122-key keyboard operates in a 5250 keyboard emulation mode.

The scan codes for translation table entries that should not be changed are shown in the following tables, Table 7-3 through Table 7-7. The translation table entries for these scan codes in any customized translation table should match the corresponding entries within the existing system table, regardless of the shift state to which a given translation table entry applies. Some of the scan codes shown may apply only to the keyboards for certain national languages.

However, the entries associated with these scan codes should not be changed from the way they appear in the original system table that you retrieve in your workstation customizing source.

Note: The Caps Lock/Shift Lock key on an Enhanced keyboard is an exception to the restrictions shown in the following tables. See the note following "Enhanced Keyboard Translation Table Restrictions" for more information about this exception.

122-Key Typewriter Keyboard Translation Table Restrictions

Table 7-3. 122-Key Typewriter Keyboard Scan Codes Restricted from Remapping

Scan Code Value	Assigned Function
'52'X	Language Layer shift (4-shift keyboards) or Katakana shift on Katakana keyboards
'53'X	Latin Layer shift (4-shift keyboards) or Alphanumeric shift on Katakana keyboards
'54'X	Shift Lock
'56'X	Left Shift, or Katakana Symbols shift on Katakana keyboards
'57'X	Right Shift
'6F'X	Scan code for Cmd key in emulation sequence

122-Key Data Entry Keyboard Translation Table Restrictions

Table 7-4. 122-Key Data Entry Keyboard Scan Codes Restricted from Remapping

Scan Code Value	Assigned Function
'0F'X	Space/Proof Space Character Mapping
'52'X	Katakana shift (Katakana 122-key data entry keyboards only)
'53'X	Katakana Symbol shift (Katakana 122-key data entry keyboards only)
'56'X	Alphanumeric shift
'57'X	Left Shift
'6F'X	Scan code for Cmd key in emulation sequence

5250 Typewriter Keyboard Translation Table Restrictions

Table 7-5. 5250 Typewriter Keyboard Scan Codes Restricted from Remapping

Scan Code Value	Assigned Function
'52'X	Language Layer shift (4-shift keyboards) or Katakana shift on Katakana keyboards
'53'X	Latin Layer shift (4-shift keyboards) or Alphanumeric shift on Katakana keyboards
'54'X	Shift Lock
'56'X	Left Shift, or Katakana Symbols shift on Katakana keyboards
'57'X	Right Shift

5250 Data Entry Keyboard Translation Table Restrictions

Table 7-6. 5250 Data Entry Keyboard Scan Codes Restricted from Remapping

Scan Code Value	Assigned Function
'0F'X	Space/Proof Space Character Mapping
'52'X	Katakana shift (Katakana 5250 data entry keyboards only)
'53'X	Katakana Symbol shift (Katakana 5250 data entry keyboards only)
'56'X	Alphanumeric shift
'57'X	Left Shift

Enhanced Keyboard Translation Table Restrictions

Table 7-7. Enhanced Keyboard Scan Codes Restricted from Remapping

Scan Code Value	Assigned Function
'12'X	Left Shift or Latin shift (when Alt shift is also active for 4-shift keyboards)
'14'X	Caps Lock or Shift Lock (depending on country and state of Alt shift). See note.
'19'X	Left Alt shift
'39'X	Right Alt shift or Layer Select (on Katakana keyboards)
'59'X	Right Shift or Language shift (when Alt shift also active for 4-shift keyboards)
Note: For the Enhanced keyboard, the entries in a customized translation table for the scan code value of '14'X may be different from the entries in the base table, but the entries should still specify either the Caps Lock or the Shift Lock function.	

Determining Which Twinaxial Keyboard Translation Table to Customize

This section provides a description of the content and format of the workstation customizing source that contains the twinaxial keyboard translation tables used by the workstation controller.

The twinaxial workstation controller uses the keyboard translation tables to map scan codes received from an attached display in one of the following ways:

- Map the scan code for a key into the EBCDIC code for the character that is labeled on the top of the key.

  After the workstation controller has converted this EBCDIC code from the scan code, it transmits the EBCDIC code back to the display so that the proper graphic character appears.
- Map the scan code to an internal workstation controller code that the controller recognizes as a request to perform some local edit or cursor movement function.

  For example, a particular scan code can be converted into a code that is a request to move the cursor in a particular direction. When the controller receives such a request, it directs the display to move the cursor in the direction indicated.
- Map the scan code into an internal workstation controller code that the controller recognizes as a request to either send data to a system application or signal the application that a particular system function is being requested.

  For example, you may have pressed the Enter key, the Help key, or any of the other function keys while working with an application.

The translation tables contain information indicating which keys are blank (no graphic character or function assigned) and which scan codes are not valid (cannot be generated by certain keyboard types). The translation table that the workstation controller uses for a given display must be specific to the particular requirements of that display. The following list shows the factors that determine which translation table is to be used:

- Device type
- Keyboard type (5250, 122-key, or Enhanced)
- Keyboard language type

When you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command to retrieve the source for customizing a twinaxial keyboard, you specify parameters relating to these factors that determine which translation table is copied in your source. It is very important that you specify the correct keyboard type and language parameters so that the correct translation table is retrieved.

Note: The device type and keyboard language type you specify for the RTVWSCST command must match the corresponding parameter values in the device description for the display; otherwise, the device will not vary on when the workstation customizing object is specified in the device description. For an example showing the use of the RTVWSCST command, see "Customizing a 3477 Twinaxial Display for Diacritic Character Support" on page 7-19.

Twinaxial Display Source Format

When you retrieve the source for a twinaxial display, the translation table for the display type, keyboard type, and language type you specify is pulled into a source file member and formatted using the workstation customizing tag language. The content and format of the source depends on the shift capabilities of the display and keyboard for which the workstation controller is using that translation table.

The workstation customizing source is divided into a number of different shift regions. Each region is denoted by a TKSTATE (keyboard translation state table) tag and its parameters. A shift region defines the characters on the keyboard when a shift key (or Alt key) is pressed in conjunction with a character or function key. The number of shift regions within a workstation customizing source file member depends on the type of display and keyboard, and the language type specified for the display. Each shift region contains information for converting scan code values from '00'X to '7F'X into EBCDIC or functional code values.

When the workstation controller receives a scan code from a display, the controller first determines the correct shift region of the translation table based on the current shift state of the display. The scan code value is then used to determine the offset within the proper shift region in order to access a particular translation table entry.

There are five basic translation table layouts (according to shift regions):

1. 5250 and 122-key layout for 2-shift national languages (2 shift regions)
2. 5250 and 122-key layout for 4-shift national languages (4 shift regions)
3. Enhanced keyboard layout for 2-shift national languages without special character support (3-shift regions)
4. Enhanced keyboard layout for 4-shift national languages (5 shift regions)
5. Enhanced keyboard layout for 2-shift national languages with special character support (6 shift regions)

Each of the shift regions has a corresponding region in the workstation customizing source that begins at '00'X following a TKSTATE tag.

For most national languages, a 5250 keyboard type has only an upper-shift capability. National languages whose keyboards contain this type of shift capability are called 2-shift national languages (unshifted and upper shift).

Some national languages also support a language and Latin shift key or keys on their keyboard, in addition to keys for the upper shift state. National languages that use the language

and Latin shift key or keys are called 4-shift national languages. The data keys on this type of keyboard can be viewed functionally as being split into a language layer and a Latin layer. The language layer of the data keys usually contains characters unique to a particular language, while the Latin layer contains those characters common to most Latin-based languages. Each layer of a key is a character defined for the unshifted position and another character defined for the upper shift position. Therefore, a key may have 4 characters on the key top, one corresponding to each possible combination of the upper/lower shift and language/Latin shift.

Although 122-key keyboards contain an Alt shift key, there is no shift region reserved within the translation table for this keyboard for the Alt shift state. This is because all twinaxial displays using 122-key keyboards function in a 5250 keyboard emulation mode, and the 5250 keyboard does not have a defined Alt key.

Enhanced keyboards also have Alt keys in their layout. When you press the Alt key on this type of keyboard, it is recognized by the workstation controller as a unique shift state. Therefore, the translation tables used for an Enhanced keyboard also include shift regions to handle the Alt shift state. For some national languages, the Enhanced keyboard also contains language and Latin shift keys. The translation tables for these types of keyboards have regions to handle the different combinations of language/Latin and upper/lower shift.

For national languages whose EBCDIC code page consists of characters in character set 697, such as United States English, Canadian, Latin American, and many European national languages, the workstation controller also provides support for an Enhanced keyboard function known as special character support. The special character support provides Enhanced keyboard users with yet another shift state.

When you press an Alt key in combination with an upper shift key on one of these Enhanced keyboards, the workstation controller puts the display into special character mode. In this mode, scan codes from the display are translated according to a keyboard layout defined as keyboard ID 100. This mode allows you to enter many special characters that you cannot enter directly from the keyboard because these characters are not included on the standard keyboard layout for a given national language. The Enhanced keyboard translation tables for the national languages that support special characters include separate shift regions to handle scan code translations in this mode.

Note: Special character support is not provided for Enhanced keyboards containing a language/Latin shift capability.

"Source Code for 3477 Twinaxial Display" on page B-1 shows an example of the source for a 3477 Model H twinaxial display with a 122-key keyboard and a keyboard language type of USI. Each shift region in the source contains the individual entries corresponding to each possible scan code value from '00'X to '7F'X in each of the possible shift states.

Changing the Entries in Your Workstation Customizing Source

Each entry in a shift region of a workstation customizing source file member consists of two bytes. The first byte in an entry contains information about the functional category to which this conversion belongs. If the conversion involves a scan code conversion to a graphic character, the second byte of the table entry contains the EBCDIC code for the character into which the scan code is mapped. If the scan code conversion does not involve a conversion to a graphic character, the second byte contains a code that identifies a specific keyboard function to be performed.

Entry Format for EBCDIC Character Conversions:

When a scan code is to be converted into a graphic EBCDIC character, the first byte of the entry in the source for that scan code must have a value in the range '09'X to '10'X. This value indicates to the workstation controller the type of character it is. It also provides an indication of the types of input fields in which each character can be used. Table 7-8 shows the valid first byte values for translation table entries that handle mapping to an EBCDIC character, the types of character each value signifies, and the valid input field types associated with each. For more information about the input field types, see the reference manuals for your workstation controllers.

Table 7-8 (Page 1 of 2). Format for Translation Table Entries That Map to EBCDIC Characters

First Byte Value	EBCDIC Character Type	Valid Field Types
'09'X	Uppercase alphabetic characters	Alphanumeric Shift Alphabetic Numeric Shift
'0A'X	Lowercase alphabetic characters (monocasing supported)	Alphanumeric Shift Alphabetic Numeric Shift
'0B'X	Nonalphanumeric characters	Alphanumeric Shift Numeric Shift
'0D'X	Numeric digit (0—9 in core area of keyboard)	Alphanumeric Shift Numeric Shift Numeric Only Digits Only Signed Numeric
'0E'X	Numeric digit (0—9 in key pad area)	Alphanumeric Shift Numeric Shift Numeric Only Digits Only Signed Numeric
'0F'X	Plus character	Alphanumeric Shift Numeric Shift Numeric Only

Table 7-8 (Page 2 of 2). Format for Translation Table Entries That Map to EBCDIC Characters		
First Byte Value	**EBCDIC Character Type**	**Valid Field Types**
'10'X	Comma, period, minus, or space characters	Alphanumeric Shift Alphabetic Only Numeric Shift Numeric Only

Note: Monocasing is not supported for any entry that has a first byte value other than '0A'X.

The second byte of the translation table entries for EBCDIC character translations must contain a valid EBCDIC character code. The valid range of values for this second byte is from '40'X to 'FE'X.

The uppercase and lowercase alphabetic character types are used for scan codes that are mapped to the alphabetic characters A-Z and a-z, as well as to any other characters that are on the keyboard or included in the alphabet for a given national language. When you enter a character from the keyboard and that character is identified as supporting monocasing, the character is converted to its uppercase character value if the Caps Lock key is active or if the character is entered into a monocase input field.

The process of converting the lowercase version of a character into its uppercase version is called monocasing. The workstation controller performs this conversion when an input field is designated as a monocase input field or when the Caps Lock key is active. The OS/400 operating system has monocasing tables that it uses for the conversion of system data. The workstation controller has its own separate tables for doing this monocase conversion; it does not access or use the system monocasing tables. When you change the source for a twinaxial keyboard translation table, identifying a character as an alphabetic character type that supports monocasing does not result in that character supporting monocasing unless that character is in the workstation controller's internal monocase table. Also, changing a character that is currently identified as an alphabetic character type that supports monocasing to a different character type means that monocasing is no longer done for that character.

The numeric digit character type for the numeric key pad area of the keyboard is used for scan codes corresponding to keys in the numeric key pad that are mapped into a numeric digit character (0-9). The numeric digit character type for the core area of the keyboard is used for scan codes corresponding to keys containing numeric characters anywhere within the core area (either the top row keys of typewriter keyboards or the keys used for numeric characters on a data entry keyboard).

The plus (+) character type is used solely to designate the plus character on a keyboard. There is also a class of characters consisting of the comma, period, minus, and space characters. Translation table entries for characters that do not fall into any of the previously mentioned categories have a first byte value of '0B'X indicating the nonalphanumeric character type.

The workstation controller uses the different character types to determine the validity of data types in different input fields. When customizing a keyboard for a twinaxial display, you should not change the character type value associated with a given EBCDIC character code. Doing so can result in data being allowed in certain types of input fields that an application may not expect or accept. For some applications, this causes unpredictable results.

Entry Format for Diacritic Characters: The keyboards for most national languages have at least one diacritic character. A **diacritic** is an accent near or through a letter or combination of letters indicating a phonetic value different from that given the unmarked or otherwise marked letters.

The workstation customizing source entries for scan codes corresponding to keys that contain a diacritic have the following general format:

Table 7-9. General Format for Translation Table Entries for Diacritics		
Byte Number	**Bit Numbers**	**Values**
1	0-7	'25'X Alphabetic diacritic '26'X Nonalphabetic diacritic
2	0-7	Assigned diacritic value

The first byte of a source entry for a diacritic indicates that the entry is for a diacritic. If the value for an alphabetic diacritic is specified, then that diacritic is treated the same way as an alphabetic character. If a non-alphabetic diacritic is specified, that diacritic is treated the same way as a non-alphabetic character.

The value for the second byte of a source entry for a diacritic indicates the specific diacritic character associated with the entry. The value assigned to a particular diacritic character depends on the language of the translation table (see Table 7-10 on page 7-12). The following example shows a source entry for a nonalphabetic diacritic character from the source for the upper shift region of a 3477 Model H twinaxial display with a 122-key keyboard.

```
          Corresponding                      Shift
Data      Scan Code        Mode              Region

2603      /*3E*/       /*MODE=NONE    SHIFT=UPPER*/
```

The set of diacritic characters that the workstation controller supports depends on the language associated with the translation table. The following list shows the sets of diacritics that are supported by the workstation controller for different groups of languages. For each set, the values for the second bytes in the translation table associated with each diacritic are also shown.

Table 7-10. Sets of Supported Diacritics by Language Group

Language Group	Keyboard Language Type	Hexadecimal Value–Diacritic Character
Character Set 697	AGB AGI BLI BRB CAB CAI DMB DMI FAB FAI FNB FNI FQB FQI ICB ICI INB INI ITB ITI JEB JEI NEB NEI NWB NWI PRB PRI SFI SGI SPB SPI SSB SSI SWB SWI UKB UKI USB USI	01–circumflex 02–grave 03–tilde 04–cedilla 05–diaeresis 06–acute 07–overcircle
Languages of the former Yugoslavia	YGI	01–circumflex 02–grave 03–cedilla 04–diaeresis 05–acute 06–caron 07–breve 08–overcircle 09–ogonek 0A–double acute 0B–overdot
Roece	ROB	01–circumflex 02–cedilla 03–diaeresis 04–acute 05–caron 06–breve 07–overcircle 08–ogonek 09–double acute 0A–overdot
Turkey	TKB	01–circumflex 02–grave 03–tilde
Greece (Old, CP 423)	GKB	01–circumflex 02–grave 03–cedilla 04–diaeresis 05–acute
Greece (New, CP 875)	GNB	01–double strike 02–diaeresis 03–acute

The keyboard language type associated with a customized table is the same as the keyboard language type of the workstation customizing source from which the customized table is created. Because diacritic processing done by the workstation controller depends on the keyboard language type of a translation table, it is important that the keyboard language type you select when retrieving a set of translation tables be the same as the actual language for that keyboard. If it is not the same, the EBCDIC characters resulting from diacritic character processing may be incorrect.

Entry Format for Blank Keys and Unassigned Scan Codes: Not all keys or key positions for a given keyboard necessarily have an assigned character or keyboard function. Source entries for scan codes corresponding to key positions having nothing assigned have a format indicating that the key is blank. Table 7-11 shows the format for blank and unassigned scan codes.

Table 7-11. Translation Table Entry Format for Blank Keys

Byte	Value
1	'22'X Unassigned Key
2	'00'X

When an unassigned key is pressed, the workstation controller discards the scan code. No character is displayed, nor is any kind of error message displayed.

If a particular scan code value cannot be generated by a particular type of keyboard, then all source entries corresponding to that scan code have a format indicating the scan code is not valid. When the workstation controller detects such scan codes, an error message is sent indicating the scan code is not valid. Table 7-12 shows the format for scan codes that are not valid.

Table 7-12. Translation Table Entry Format for Scan Codes That Are Not Valid

Byte	Value
1	'23'X Key Not Valid
2	'00'X

Entry Format for the Proof Space Character: The source entry for the scan code corresponding to a shifted spacebar on a data entry keyboard requires a special format so that certain data entry keyboard functions are properly handled. This entry is called the entry for a proof space character. Table 7-13 shows the format for the proof space character.

Table 7-13. Translation Table Entry Format for Proof Space Character

Byte	Value
1	'27'X
2	'40'X (EBCDIC code for the space character)

Entry Format for Function Keys: In this section, the term **function key** refers to a broad class of keyboard functions that can be performed by the workstation controller. The term does not refer exclusively to the PF or Cmd keys that may be on the keyboard of a twinaxial display. It can also refer to other keys that may result in such controller functions as moving the cursor, putting the display into insert mode, or generating an aid function, such as Help or Page Up.

The controller processes function key requests from a twinaxial display based on a number of different categories of keyboard functions. The source entries for function keys have a format that allows the controller to identify a function key request as belonging to a particular function category.

The first byte of each source entry has the same value for all keyboard functions that are in the same category. The second byte of the entry contains the value that uniquely identifies the keyboard function within a particular category.

Table 7-14 through Table 7-20 on page 7-14 show the formats for function keys.

Table 7-14. Translation Table Entry Formats: Cursor Movement Functions

Byte	Value and Function
1	'01'X Cursor Movement Function
2	'01'X New Line '02'X Cursor Left '03'X Cursor Right '04'X Fast Cursor Left '05'X Fast Cursor Right '06'X Cursor Up '07'X Cursor Down '08'X Character Backspace '09'X Tab Back (Field Backspace) '0A'X Tab Advance '0B'X Tab Advance and Tab Function '0C'X Tab Function '0D'X Home '0E'X Character Advance

Note: In data processing mode, unique text functions are performed for the different tabs with hexadecimal values '0A'X, '0B'X, and '0C'X. In word processing mode, the unique text functions are performed for all the different types of tabs.

Table 7-15. Translation Table Entry Formats: Field Exit Functions

Byte	Value and Function
1	'02'X Field Exit Function
2	'01'X Field Exit '02'X Field – (minus) '03'X Field + (plus) '04'X Dup

Table 7-16. Translation Table Entry Formats: Nonaid Functions

Byte	Value and Function
1	'03'X Nonaid Functions
2	'01'X Insert '02'X Delete '03'X Erase Input '04'X Cancel '05'X Close (Bidirectional languages only) '06'X Reverse (Bidirectional languages only) '07'X Base (Bidirectional languages only) '08'X Screen Reverse (Bidirectional languages only) '09'X Reverse Video Screen '0A'X Erase End of Field (Erase EOF) '0B'X Field Mark '0C'X Cursor Select '10'X Command Key (Cmd) '11'X Hex

Table 7-17. Translation Table Entry Formats: Immediate Functions

Byte	Value and Function
1	'04'X Immediate Functions
2	'01'X Reset '02'X Attention '03'X System Request

Table 7-18. Translation Table Entry Formats: Function (Cmd) Key Functions

Byte	Value and Function
1	'05'X Cmd Key Functions
2	'01'X PF1 (Cmd1) '02'X PF2 (Cmd2) '03'X PF3 (Cmd3) '04'X PF4 (Cmd4) '05'X PF5 (Cmd5) '06'X PF6 (Cmd6) '07'X PF7 (Cmd7) '08'X PF8 (Cmd8) '09'X PF9 (Cmd9) '0Assq.X PF10 (Cmd10) '0B'X PF11 (Cmd11) '0C'X PF12 (Cmd12) '0D'X PF13 (Cmd13) '0E'X PF14 (Cmd14) '0F'X PF15 (Cmd15) '10'X PF16 (Cmd16) '11'X PF17 (Cmd17) '12'X PF18 (Cmd18) '13'X PF19 (Cmd19) '14'X PF20 (Cmd20) '15'X PF21 (Cmd21) '16'X PF22 (Cmd22) '17'X PF23 (Cmd23) '18'X PF24 (Cmd24)

Table 7-19. Translation Table Entry Formats: Aid Generating Functions

Byte	Value and Function
1	'06'X Aid Generating Functions
2	'01'X Print '02'X Roll Up (Page Down) '03'X Roll Down (Page Up) '04'X Enter '05'X Clear '06'X Test Request '07'X Help '08'X PA1 '09'X PA2 '0A'X PA3

Table 7-20. Translation Table Entry Formats: Shift Key Functions

Byte	Value and Function
1	'16'X Shift Key
2	'01'X Shift Lock '02'X Left Shift '03'X Right Shift '04'X Alphanumeric Shift '05'X Katakana Symbol Shift '06'X Katakana Shift '07'X Left Alt Shift (Enhanced keyboards only) '08'X Right Alt Shift (Enhanced keyboards only) '09'X Caps Lock '0A'X Latin Shift '0B'X Language Shift '0C'X Layer Select Shift '0D'X Left Special Character support '0E'X Right Special Character support

Changing Source Entries for Scan Codes That Are Not Valid

The set of scan codes that a particular display keyboard can generate is a subset of the entire range of scan code values in a translation table. The complete range is from '00'X to '7F'X. Translation table entries for scan code values that are not in a specific subset are usually coded as scan code entries that are not valid (see "Entry Format for Blank Keys and Unassigned Scan Codes" on page 7-12). Although most twinaxial displays should never generate such scan code values, you can change the corresponding source entries for these scan codes so that something other than a scan code that is not valid is specified.

A device that is emulating a twinaxial display may be capable of generating scan code values that an actual twinaxial display cannot. Also, there are a number of twinaxial displays manufactured to meet certain language requirements that can generate scan code values besides those shown in "Keyboard Scan Codes" on page 7-2. To accommodate such displays, the translation table entries for all scan codes (except the ones noted for each keyboard type) can be changed in your workstation customizing source.

Keyboard Functions Not Specified in Translation Tables

There are some keyboard functions that you cannot directly specify in a translation table. Some of these functions cannot be changed using workstation customizing, while others can be indirectly affected by it.

Examples of keyboard functions that you cannot directly specify are functions that require a Command (Cmd) key sequence on a 5250-style keyboard. When you press the Cmd key on a twinaxial keyboard followed by one of the

top-row keys on the keyboard, the workstation controller performs one of the Command or PF key functions (Cmd1-Cmd12). However, there are no individual entries in a 5250-style keyboard translation table for each of these functions. There is only a translation table entry for the Cmd key itself. When the Cmd key is pressed and the scan code for a top-row key is received, the controller calls one of the Cmd1-Cmd12 functions.

Any keyboard functions normally performed by the workstation controller that are not coded in a given translation table cannot be customized using the workstation customizing functions. Therefore, the key sequences for Cmd1-Cmd12 on a 5250-style keyboard must always involve pressing a given key (like Cmd) followed by one of the top-row (numeric) keys. You can change the key required to put the display into the Cmd state; however, the subsequent key required to call one of the functions Cmd1-Cmd12 cannot be changed.

On an Enhanced keyboard, the equivalent keyboard functions (Cmd1-Cmd12) are accessed by way of single keys. There are separate entries in an Enhanced keyboard translation table that specifically identify the Cmd function to be performed.

Another group of keyboard functions that can be indirectly affected by workstation customizing are the OfficeVision/400 editor functions. Many of these functions are called using Alt key sequences on 122-key and Enhanced keyboards, and using Cmd key sequences on 5250 keyboards. The translation tables contain no information specifying these particular functions. The workstation controller is notified that a display is in the editor (text edit) mode, and the base translation table function that is called determines whether or not one of these editing functions should be performed.

The text editing functions are only affected by workstation customizing when you decide to change the base keyboard functions with which a given text edit function is associated. For example, the editing function for underlining text is initiated by pressing Alt U on an Enhanced keyboard when the editing functions are active. If a customized translation table changes the key required for a user to enter the character U, then the underscore function follows the U function to that new key.

Working with the Tag Language for Twinaxial Displays

When the device class parameter specified for the RTVWSCST command is a twinaxial display other than 3477, 3486, or 3487 (DEVCLASS=TWINAXDSP), your source structure looks like the following:

```
:WSCST DEVCLASS=TWINAXDSP.

  :TKBDTBL with parameters.

    :TKSTATE with parameters.
      /*one keyboard state table*/
                .
                .

     /*from 1 - 5 additional state*/
     /*tables depending on keyboard type*/
                .
                .

:EWSCST.
```

Figure 7-5. Source Structure for Twinaxial Displays

The order and placement of the primary tags in each source file member is enforced by the workstation customizing object compiler.

Using the Tags to Customize Twinaxial Display Keyboards

The following sections describe the tags you can use to customize a twinaxial keyboard. For a twinaxial device, the keyboard is the only part of the device that you can customize.

To retrieve a source file for customizing a twinaxial display, be sure to specify a twinaxial display for the device type when you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command. To see a complete source code example for a twinaxial display, see "Source Code for 3477 Twinaxial Display" on page B-1.

Keyboard Translation Table (TKBDTBL) Tag

The TKBDTBL (keyboard translation table) tag in your source defines the twinaxial keyboard translation table you want to customize. When the TKBDTBL tag is missing from the source, the default translation tables are used when the device is varied on. The syntax for this tag is:

```
:TKBDTBL
         LANGTYPE = keyboard language type
         KBDTYPE = DATA5250|TYPE5250|
                   DATA122|TYPE122|
                   ENHANCED.
```

Figure 7-6. Syntax for the Keyboard Translation Table Tag

The TKBDTBL tag indicates the beginning of the keyboard translation table and is used to specify a keyboard and language type for a twinaxial display or an ASCII printer attached to a twinaxial display.

LANGTYPE

A required parameter that specifies the type of keyboard language. Valid values for this parameter are listed in Table 7-21 on page 7-16.

Note: Not all combinations of the language type parameter and the keyboard type parameter are valid. For a list of the valid combinations, see Table 7-22 on page 7-18.

Table 7-21 (Page 1 of 2). Values for the Language Type (LANGTYPE) Parameter

Language Type Value	Language Description	Associated Code Page
AGB	Austria/Germany	273
AGI	Austria/Germany Multinational	500
BLI	Belgium Multinational	500
BRB	Brazil	037
CAB	Canadian French	260
CAI	Canadian French Multinational	500
CLB	Arabic	420
CYB	Cyrillic	880
DMB	Denmark	277
DMI	Denmark Multinational	500
FAB	France (Azerty)	297
FAI	France (Azerty) Multinational	500
FNB	Finland	278
FNI	Finland Multinational	500
FQB	France (Qwerty)	297
FQI	France (Qwerty) Multinational	500
GKB	Greece	423
GNB	Greece	875
ICB	Iceland	871
ICI	Iceland Multinational	500
INB	International	500
INI	International Multinational	500
ITB	Italy	280
ITI	Italy Multinational	500
JEB	Japan English	281
JEI	Japan English Multinational	500
KAB	Japan Katakana	290
NCB	Hebrew	424
NEB	Netherlands	037
NEI	Netherlands Multinational	500
NWB	Norway	277
NWI	Norway Multinational	500
PRB	Portugal	037
PRI	Portugal Multinational	500
ROB	Latin 2	870
SFI	Switzerland/French Multinational	500
SGI	Switzerland/German Multinational	500
SPB	Spain	284
SPI	Spain Multinational	500
SSB	Spanish Speaking	284

Table 7-21 (Page 2 of 2). Values for the Language Type (LANGTYPE) Parameter

Language Type Value	Language Description	Associated Code Page
SSI	Spanish Speaking Multinational	500
SWB	Sweden	278
SWI	Sweden Multinational	500
THB	Thailand	838
TKB	Turkey	1026
UKB	United Kingdom	285
UKI	United Kingdom Multinational	500
USB	United States/Canada	037
USI	USA/Canada Multinational	500
YGI	Languages of the former Yugoslavia	870

KBDTYPE
A required parameter that specifies the keyboard type. The following list shows the valid values and their meanings.

DATA122
122-key data entry keyboard

TYPE122
122-key typewriter keyboard

DATA5250
5250 data entry keyboard

TYPE5250
5250 typewriter keyboard

ENHANCED
Enhanced keyboard

Table 7-22 on page 7-18 maps the available language types with the types of keyboards for which workstation customizing is supported.

The TKBDTBL tag can be followed by from two to six TKSTATE (keyboard translation state table) tags. Each tag indicates one state table for the keyboard. For a given language type, there are valid combinations of the KBDTYPE parameter with the MODE and SHIFT parameters on the TKSTATE tags. See Table 7-22 on page 7-18 for the format value associated with each valid KBDTYPE value for the language type you selected. This format value dictates the number of TKSTATE tags that are required. The formats are described with the TKSTATE tag in the following section.

Keyboard Translation State Table (TKSTATE) Tag

The TKSTATE (keyboard translation state table) tag defines a twinaxial keyboard translation table for one state of a keyboard. Keyboards can have anywhere from two to six states, depending on the type. Therefore, there are two to six TKSTATE tags following each TKBDTBL tag. This tag is required whenever the TKBDTBL tag is specified. The syntax for the TKSTATE tag is:

```
:TKSTATE
        MODE = NONE|LATIN|LANGUAGE|SPCCHR
        SHIFT = UNSHIFT|UPPER|ALT
        DATA = table data.
```

Figure 7-7. Syntax for the Keyboard Translation State Table Tag

MODE
A required parameter. Specifies the keyboard mode for which the translation table is used.

NONE
No keyboard mode is required.

LATIN
Latin character entry mode

LANGUAGE
Language character entry mode

SPCCHR
Special character keyboard mode.

SHIFT
A required parameter. Specifies the keyboard shift for which the translation table is used.

UNSHIFT
The translation table is used if the keyboard is not in any shift state.

UPPER
The translation table is used if the keyboard is in an upper shift state.

ALT
The translation table is used if the keyboard is in an Alt key shift state.

DATA
A required parameter. Specifies the translation table data for one keyboard state. The data is the hexadecimal values for the keys on the keyboard in this state. The table data must be hexadecimal and must be exactly 256 bytes in length.

table data
Hexadecimal values for the keyboard keys in this state.

Table 7-22 maps the available language types with the keyboards and shift states they support. A list following the table shows the MODE / SHIFT combinations for the various language and keyboard types. These combinations can occur in any order within the group, but all are required to be present for a particular keyboard and its requirements.

Table 7-22. Language Type and Keyboard Type Considerations

Language Type Value	Number of TKSTATE Tags				
	DATA122	TYPE122	DATA5250	TYPE5250	ENHANCED
AGB, AGI	—	2	2	2	6
BLI	—	2	2	2	6
BRB	—	—	—	2	6
CAB	2	2	2	2	3
CAI	2	2	2	2	6
CLB, CYB	—	4	—	4	5
DMB, DMI	—	2	2	2	6
FAB, FAI	—	2	2	2	6
FNB, FNI	—	2	2	2	6
FQB, FQI	—	—	2	2	—
GKB, GNB	—	4	—	4	5
ICB, ICI	—	2	—	2	6
INB, INI	—	—	2	2	—
ITB, ITI	—	2	2	2	6
JEB, JEI	—	—	2	2	—
KAB	4	4	4	4	5
NCB	—	4	—	4	5
NEB, NEI	2	2	2	2	6
NWB, NWI	—	2	2	2	6
PRB, PRI	—	2	2	2	6
ROB	—	2	—	2	3
SFI, SGI	—	2	—	2	6
SPB, SPI	—	2	2	2	6
SSB, SSI	2	2	2	2	6
SWB, SWI	—	2	2	2	6
THB	—	—	—	—	5
TKB	—	2	—	2	3
UKB, UKI	—	2	2	2	6
USB, USI	2	2	2	2	6
YGI	—	2	—	2	3

The MODE / SHIFT combinations that are required for the different numbers of TKSTATE tags specified in a workstation customizing source are defined as follows:

- When 2 TKSTATE tags are specified, the required MODE / SHIFT combinations are:

  NONE / UNSHIFT
  NONE / UPPER

- When 3 TKSTATE tags are specified, the required MODE / SHIFT combinations are:

  NONE / UNSHIFT
  NONE / UPPER
  NONE / ALT

- When 4 TKSTATE tags are specified, the required MODE / SHIFT combinations are:

  LATIN / UNSHIFT

LATIN / UPPER
LANGUAGE / UNSHIFT
LANGUAGE / UPPER

- When 5 TKSTATE tags are specified, the required MODE / SHIFT combinations are:

  LATIN / UNSHIFT
  LATIN / UPPER
  LANGUAGE / UNSHIFT
  LANGUAGE / UPPER
  NONE / ALT

- When 6 TKSTATE tags are specified, the required MODE / SHIFT combinations are:

  NONE / UNSHIFT
  NONE / UPPER
  NONE / ALT
  SPCCHR / UNSHIFT
  SPCCHR / UPPER
  SPCCHR / ALT

Customizing a 3477 Twinaxial Display for Diacritic Character Support

In this example, you are customizing a 3477 Model H twinaxial display with a 122-key typewriter keyboard and you want to use and display some diacritic characters.

It is assumed that the 3477 Model H workstation is physically attached to the AS/400 system, configured to work with the system (appropriate controller and device descriptions created), and is varied on.

Step 1: Planning the Customizing

After looking over the reference manual for the display to familiarize yourself with the display device, see the reference information for customizing twinaxial displays in Chapter 7, "Customizing Twinaxial Displays" of this guide.

In this example, the diacritic characters for which you want to add support are the following:

^	circumflex
~	tilde
ç	cedilla (the accent beneath the c)
..	diaeresis
´	acute

From experimenting with the display functions, you find that five keys on the numeric key pad do not correspond to or provide any function. Using the scan code illustrations in "Keyboard Scan Codes" on page 7-2, you find the scan codes for these unassigned keys. These are shown in the following figure:

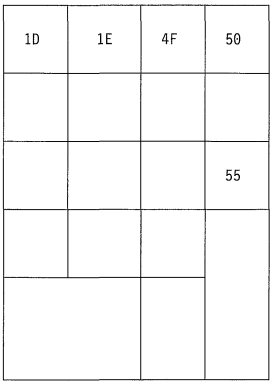


Figure 7-8. Scan Codes for Unassigned Keys on 3477 Model H 122-Key Typewriter Keyboard

Step 2: Retrieving the Workstation Customizing Source

To create a workstation customizing source, you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command and specify the following:

Device type
DEVTYPE(3477)

Keyboard language type
KBDTYPE(USI)

Note: This value must match the keyboard language type you specified in the device description for the display; otherwise the display will not vary on.

Source member
SRCMBR(CST3477)

Keyboard attached
KBD(*TYPE122)

Source file
SRCFILE(CSTSRC)

Library LIB(CSTLIB)

Text TEXT('Workstation customizing source for 3477 diacritics')

Step 3: Changing the Source

After you retrieve the source, use the source entry utility (SEU) to change the hexadecimal values that correspond to the codes for the keys on the numeric key pad. The command is:

```
STRSEU SRCFILE(CSTLIB/CSTSRC) SRCMBR(CST3477)
```

To be certain that the keys you want to customize are unassigned, look at the first state table in your source denoted by

```
:TKSTATE MODE = NONE  SHIFT = UNSHIFT.
```

The values for unassigned keys are set to '2200'X. The first byte of the new entries was selected from Table 7-9 on

page 7-11. The second byte was selected from Table 7-10 on page 7-12. Write these values on your planning work sheet as shown in the following table:

Table 7-23. Twinaxial Keyboard Translation Table Entries–Example Work Sheet

Twinaxial Keyboard Translation Table Entries (Mode= NONE Shift State= UNSHIFT)				
	Key or Function		Hexadecimal Data	
Scan Code	Old	New	Old	New
1D	Num Lock	^	2200	2501
1E	/	~	2200	2503
4F	*	cedilla	2200	2504
50	—	..	2200	2505
55	+	´	2200	2506

Change the '2200'X entries for the appropriate keys in your source to the entry values shown in the previous table and save the changed source.

Step 4: Creating the Workstation Customizing Object

After you change and save the source, create the workstation customizing object using the Create Work Station Customizing Object (CRTWSCST) command. Specify the following parameter values:

WSCST name WSCST(CST3477)
Library LIB(CSTLIB)
Source member SRCMBR(CST3477)
Text TEXT('Workstation Customizing Object for 3477 diacritics')
Source file SRCFILE(CSTSRC)
Library LIB(CSTLIB)

If any errors occur, messages are sent to the job log to help you correct them.

Step 5: Varying On the Device

Using the Change Device Description Display (CHGDEVDSP) command, change the device description for the 3477 Model H display and specify the customizing object CST3477 for the workstation customizing object (WSCST) parameter.

To activate the workstation customizing function, use the Work with Configuration Status (WRKCFGSTS) to vary off and then vary on the display.

Chapter 8. Customizing ASCII Displays

This chapter provides more detailed technical information to help you customize ASCII displays. This chapter describes the function and operation of the mapping tables used by the AS/400 system's local ASCII workstation controller to support an ASCII display. Some of the background information about the local ASCII workstation controller is provided to help you understand the way the different mapping tables are used with one another and how the ASCII workstation controller processes keyboard and display data. This information may help you customize your ASCII display.

The tag language source structure for customizing the mapping tables for ASCII displays is also described in this chapter, followed by the syntax and descriptions for the individual tags related to customizing ASCII displays.

Using the OS/400 workstation customizing for ASCII displays you can:

- Customize the functional characteristics of a supported ASCII display
- Customize the functional characteristics and specify all the necessary parameters required to provide support for an unsupported ASCII device type

Beginning to Customize ASCII Displays

When you customize a supported ASCII display, you are adding or changing the functional or character support the ASCII workstation controller provides for the display. If you want to change the way the ASCII keyboard sends information to the ASCII workstation controller, you need to change the mapping tables used to perform inbound processing. If you want to change the way that characters appear on the display, you need to change the mapping tables used to perform outbound processing. In most cases you want to change both. (See "Inbound and Outbound Processing" on page 8-8 for a more detailed definition.)

To begin customizing your ASCII display, start by changing the characteristics for outbound processing first. This allows you to see the information being sent to the display correctly. It is then easier to determine the changes you want to make for keyboard processing. After you have made the necessary additions and changes to the display mapping tables, you should create the customizing object and test the characteristics for outbound processing that you changed. When you have these characteristics working correctly, you can then change the workstation customizing source a second time to change the characters and functions called by the keyboard.

If you are customizing a supported ASCII display, you can skip the next section and go on to the technical overview that describes the way the ASCII workstation controller performs both inbound and outbound processing. Following the overview is more detailed technical information about the ASCII display mapping tables and the tags you can use to customize your display.

Customizing Unsupported ASCII Displays

When you want to customize an ASCII display that is not currently supported by the ASCII workstation controller, you need to retrieve a source file member based on a supported device type and keyboard language type. To do this, you need to find out which device type is most like your unsupported display. This may involve reading the reference manual to find out if the display emulates any supported display types, talking to the manufacturer of the device, or talking to your technical support specialist. The following are some other questions you need to answer before you begin the workstation customizing process:

- What display functions or characteristics and national characters do I want this display to support?

  Write these down so that you can answer the next question.

- Does the display itself support the functions you need?

  Check the reference manual to determine this. If neither the display nor the ASCII workstation controller can support the functions you need, you cannot customize the display to support those functions.

- Does the display emulate or support the emulation of an IBM-supported display?

  If so, set it up to use the emulation because it could make your customizing easier.

- Which supported ASCII display has similar characteristics to my unsupported ASCII display?

  To find out more about the characteristics of the ASCII displays supported by IBM and the AS/400 system, you can do one or more of the following:

  - Check the reference manuals for the supported displays. These may be available through your technical support specialist or marketing representative. You may already be using one or more of the supported displays and have a manual on hand, or you could contact the manufacturer of a supported display and ask for a list of the display characteristics or the manual for the supported display.
  - The *ASCII Work Station Reference and Example* contains a list of the general display characteristics (such as the line speed, parity, and size of the screen) for supported ASCII displays.

When you know or have an idea which supported display your ASCII display is most like, you can use the device type for the supported display to retrieve a workstation customizing source file member to be the basis of your workstation

customizing object. Use the Retrieve Work Station Customizing Object Source (RTVWSCST) command to retrieve mapping tables for a supported workstation that is most like your unsupported workstation.

After you have answered the previous questions, you also need to do the following:

- Set up all the necessary hardware to connect the display to the ASCII workstation controller
- Set up any programmable features provided by the display

  This is the time to set up any emulation (if supported), a well as any language sets and other parameters that affect the display.
- Create the necessary controller and device descriptions if needed

  The controller description is for the ASCII workstation controller and may already exist. The device description is for the ASCII display.

After you have set up and turned on the display, vary it on and see if the Sign On display is shown. If so, you can test the characteristics of the display immediately. Otherwise, you may need to create a workstation customizing source and change the hexadecimal values for the CLRSCN (Clear Screen), and CSRADR (Set Cursor Address) attributes in the source to those for the unsupported display. For more information about changing these tags initially, see "Display Commands for Unsupported Device Types" on page 8-12.

The workstation customizing procedure for ASCII displays that are not currently supported can involve some trial-and-error. For an unsupported display, you usually have a lot of additions or changes to make. To avoid compiling errors, you can do this by making one or two changes to the source at a time and then creating the workstation customizing object. Test the object (vary off the display, specify the customizing object in the device description, and then vary the display back on) to see if the results are what you expected. If so, then make the next changes; otherwise, go back to the source and the reference manual for the display to determine the cause of the problem before going further with the workstation customizing procedure.

The remaining sections in this chapter provide more detailed technical information about the display mapping tables, the tags you use to customize an ASCII display, and some examples.

Working with ASCII Displays and the Local ASCII Workstation Controller

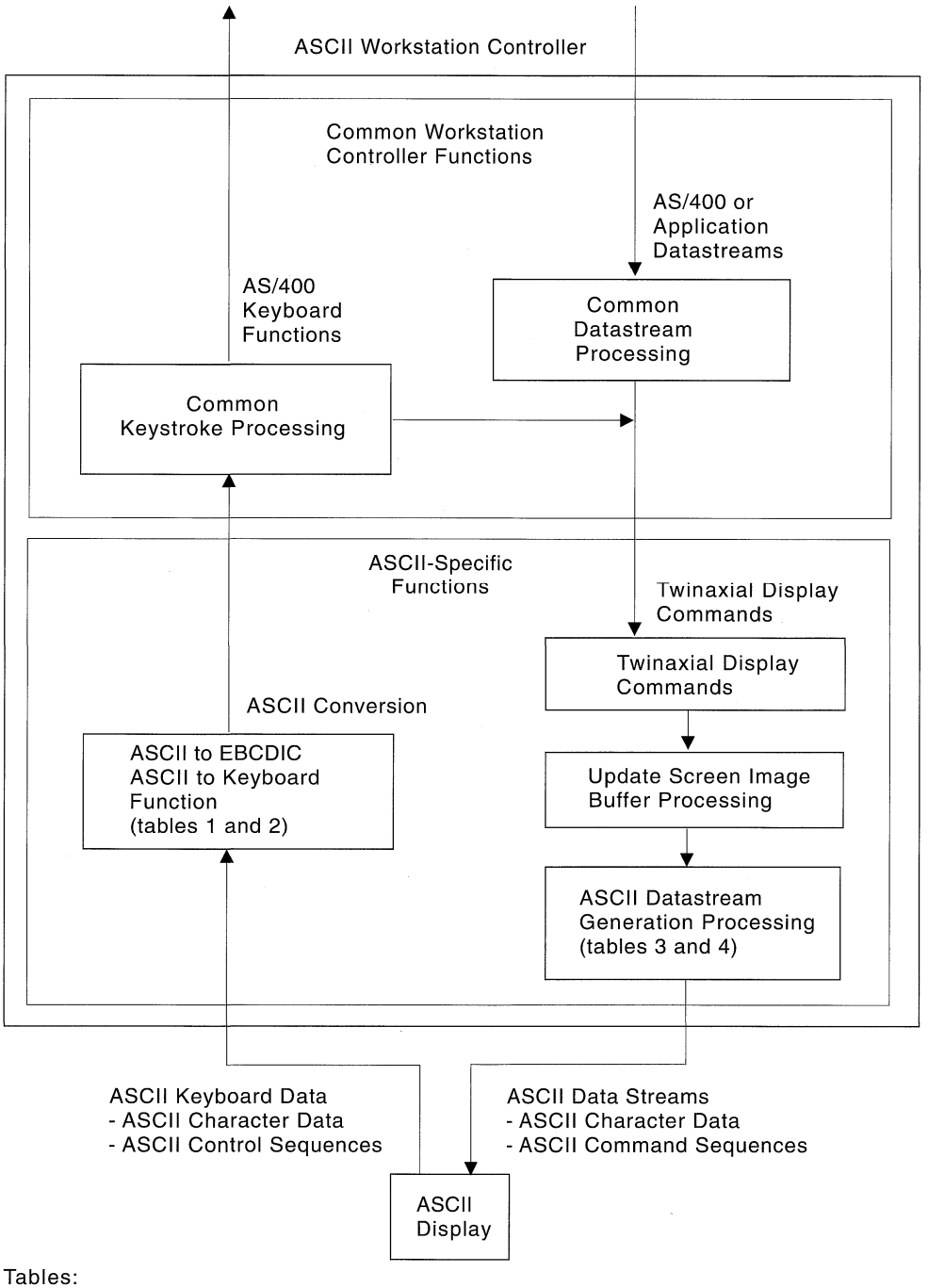
The AS/400 local ASCII workstation controller works similarly to the local twinaxial workstation controller in terms of the display and printer interface it provides to the AS/400 operating system. Whether the attached device is an ASCII or a twinaxial device does not matter to applications sending data to a display or printer from the AS/400 system.

The function of the local ASCII workstation controller is based on that of the twinaxial workstation controller. Much of the functional code within the two controllers is common code. This common code provides most of the major functions provided by the controllers, for example, data flow control, data stream processing, and keystroke processing.

An ASCII-specific layer of code provides the function that is unique to supporting ASCII displays with the ASCII workstation controller. Among the functions included in this layer of code are a twinaxial device emulation function (ASCII displays are supported as emulated twinaxial displays), ASCII device configuration support, and the necessary interface between the ASCII displays and the ASCII-unique hardware of the ASCII controller.

The ASCII-specific layer of code has two main interfaces. One is an interface to the common code, which sends and receives commands, status, and data as though a twinaxial display is attached. The other is the interface to the actual ASCII displays. The character data sent to and received from the devices must be ASCII character data and the command format must conform to the command format for the specific type of ASCII display that is attached.

Figure 8-1 on page 8-3 illustrates the interfaces between the code (the ASCII-specific code and the common workstation controller code) and the devices themselves:



Tables:

1. ASCII to EBCDIC Character Mapping Table
2. ASCII to Keyboard Function Mapping Table
3. Update Screen Table
4. EBCDIC to ASCII Mapping Table

RV2H455-2

Figure 8-1. Twinaxial Device Emulation Processes

The ASCII-specific code in the ASCII workstation controller processes keyboard data by converting it to EBCDIC character codes or twinaxial keyboard functions. The common code then performs the keystroke processing necessary to echo a character back to the display, move the cursor, or signal an application that a particular function key has been pressed. The ASCII-specific code then generates ASCII commands and data from the twinaxial display commands passed to it from the common code.

To support an ASCII display, the ASCII-specific code must map data and commands back and forth between the format

required by the common code and the format required by the devices themselves. This is done using a number of different mapping tables. The following list shows the mapping tables required to support an ASCII display.

ASCII display tables:

- ASCII-to-EBCDIC character mapping table
- ASCII to keyboard function mapping table
- Update screen table
- EBCDIC-to-ASCII character mapping table

In most cases, the ASCII to keyboard function table and the update screen table used for an ASCII display depends exclusively on the device type. Which ASCII-to-EBCDIC character tables and EBCDIC-to-ASCII character tables are used for an ASCII display depends on both the device type and the national language selected for the device description.

Twinaxial Device Emulation

Before looking at the display and mapping tables used by the ASCII workstation controller, it is helpful to have some background information on ASCII character sets, code pages, control codes, and command sequences.

The ASCII workstation controller uses the mapping tables combined with twinaxial device emulation to map data between an ASCII workstation and the AS/400 system. The twinaxial device emulation assures that the following objectives are met:

- Appearance of application displays sent to an ASCII display are the same (or nearly the same) as if those application displays were sent to a twinaxial display. This means that the same characters should be displayed in the same locations on the screen, and the highlighting of data should be consistent with what it would be on a twinaxial display.
- Data passed back to the application from an attached ASCII display is in the same format as data passed back from an attached twinaxial display.
- ASCII keyboard functions are the same as the keyboard functions available for a twinaxial display.

Part of the twinaxial device emulation involves the maintenance of an internal screen image buffer within the ASCII workstation controller itself. This internal screen image buffer is equivalent to the screen buffer for a twinaxial display. The type of data residing in the internal screen image buffer for the ASCII workstation controller is exactly the same as the type of data that would be in the buffer for a twinaxial display. For example, null character data, control character data, attribute control characters, and EBCDIC character data are stored in this buffer.

As Figure 8-1 on page 8-3 shows, the interface between the common workstation controller code and the ASCII-specific code involves twinaxial commands from the common code to the ASCII-specific code. For displays, these twinaxial commands contain information on the data that should be displayed and where on the screen it should be displayed. The ASCII-specific code processes these commands, updating its internal screen image buffer so that it contains the same data that would be in a twinaxial display's internal screen buffer.

Maintaining the internal screen image buffer and mapping the data from this buffer into ASCII data streams allows the appearance of the ASCII display to be as close as possible to the appearance of a twinaxial display. This also allows the ASCII workstation controller to be sure that data being passed back to the application is equivalent to data that would be passed back from a twinaxial display.

To provide you with the same type of keyboard functions that are available on a twinaxial display, the ASCII-specific code in the ASCII workstation controller maps the ASCII control characters and control sequences generated from the keyboard of your ASCII display into a format which the common workstation controller code recognizes as twinaxial keyboard function requests. The common workstation controller code then completes the keystroke processing.

ASCII Character Sets and Code Pages

An ASCII character set represents another method of encoding character data (the same way that an EBCDIC character set represents a way of encoding character data). "ASCII Character Code to Hexadecimal Value Chart" on page C-1 shows how graphical character data is encoded for the United States (US) ASCII character set. The encoding scheme shown is also often referred to as the US ASCII code page.

Code values in the range '20'X through '7E'X are reserved for graphical character data. Code values in the range '00'X through '1F'X, plus '7F'X, are used for control codes.

Just as there are different varieties of EBCDIC character sets and code pages, there is also a variety of ASCII character sets and code pages. For most ASCII devices manufactured in the US, the US ASCII character set is the same from one ASCII device to the next. The particular ASCII character set used for a given national language varies from one display manufacturer to the next.

The US ASCII character set is a 7-bit character set, meaning all the data and control codes within this set can be represented by different combinations of 7 bits. Some ASCII devices support 8-bit ASCII character sets. For 8-bit ASCII character sets, the assignment of character code values and control code values is split into two ranges, a lower range of values from '00'X through '7F'X, and an upper range of values from '80'X through 'FF'X. The assignment of character values in this upper range is similar to the assignment used in the lower range as shown in the following list.

Lower range '00'X-'7F'X

Lower range control characters	'00'X-'1F'X
Graphic character data	'20'X-'7E'X

Control code:	'7F'X
Upper range '80'X-'FF'X	
Upper range control characters	'80'X-'9F'X
Graphic character data	'A0'X-'FE'X
Character data (usually)	'FF'X

ASCII devices that support an 8-bit character set can be set up to operate in 7-bit mode. In this case, the display still needs a way to access data in the upper range of the character set using a 7-bit character code. To access the upper range of the character set, the control characters, **shift-in** (SI) and **shift-out** (SO) are used.

When a shift-out control character (usually '0E'X) is sent to the display, the display is signaled that it should now treat the subsequent data it receives in the range '20'X through '7F'X as though it is character data in the upper range ('A0'X through 'FF'X) of the character set.

When a shift-in control character (usually '0F'X) is sent to the display, the display is signaled that it should return to treating data in the range '20'X through '7F'X as the character data defined for those values.

The two ranges that make up the 8-bit character set are often referred to as character spaces. IBM ASCII displays refer to those spaces as the G0 and G1 character spaces. These displays support commands that allow the set of accessible characters in one of these character spaces to be swapped with a different set of graphic characters. Therefore, a single display may actually support three or more sets of characters simultaneously with specific character sets being loaded into the two character spaces as needed to display specific characters.

For example, many ASCII displays that are not manufactured in the United States are designed so that, by default, the US ASCII character set is assigned to the character space '20'X through '7F'X. Another character set containing many of the special language characters is assigned to the space from 'A0'X through 'FF'X. These displays usually support a command that allows the special language character set to be replaced with another graphic character set.

A command to switch back to the national language character set is also provided allowing the ASCII workstation controller to force the display back and forth between character sets as needed. The ASCII workstation controller makes use of these character space exchanging functions in its support for text symbol characters.

ASCII Control Codes

ASCII control codes are sent to an ASCII display to indicate a specific command that the device is to perform. For example, the control code of '07'X has the mnemonic name BEL, and is usually a command to sound the audible alarm at the display. The control code of '08'X has the mnemonic name BS (backspace), and is usually a command to the display indicating that it should perform a backspace from the current cursor position.

ASCII Command Sequences

Besides recognizing commands in the form of control codes, ASCII devices also recognize commands sent in the form of ASCII command sequences. Command sequences consist of a string of ASCII characters, where the first character is a control code. For most ASCII devices, command sequences begin with the Escape (Esc) control code, which has the hexadecimal value '1B'X.

The data stream sent to an ASCII display or printer consists of ASCII graphic character codes, control codes, and ASCII command sequences. Although many ASCII displays and printers perform similar types of functions and support similar types of commands, the set of commands that is supported varies considerably from one device to another. Also, commands that perform similar functions on different types of displays are often called with completely different command sequences or control codes. This means that each ASCII device has its own unique set of commands. Therefore, the ASCII data stream sent to a particular type of device to perform a given function must be specific to that type of device.

ASCII Display Keyboard Operations

Most of the keys on an ASCII display keyboard are assigned ASCII character codes, control codes, or ASCII command sequences. When a key is pressed on the keyboard, the assigned character code, control code, or command sequence is sent to the ASCII workstation controller.

More than one character code, control code, or command sequence can be assigned to a given key on an ASCII keyboard. In such cases, the particular code or sequence sent to the ASCII workstation controller depends on whether the key has been pressed in combination with a Shift or Control (Ctrl) key.

In general, ASCII displays do not send anything to the workstation controller when only the Shift or Ctrl key is pressed. When these keys are pressed in combination with other keys, is data sent to the ASCII workstation controller.

There are usually other keys on an ASCII keyboard that do not send anything to the ASCII workstation controller. Most ASCII displays have several keys reserved for local display functions. Pressing these keys causes the display to perform some internal function that does not involve the workstation controller.

As with command sequences that are sent to an ASCII display to perform certain functions, the control codes or command sequences that are assigned to the keys on an ASCII display depend on the ASCII device type. Each different device generally has its own unique way of assigning control codes or command sequences to the keys.

As an example, Figure 8-2 on page 8-6 shows the keyboard layout for an IBM 3151 ASCII display. The functions and characters shown are labeled on the key tops and do not indicate any kind of specific functions supported by the ASCII workstation controller. Figure 8-3 on page 8-7 shows the ASCII character codes and ASCII code sequences sent to the ASCII workstation controller when the keys on the keyboard are pressed. Table 8-1 on page 8-8 shows the ASCII control sequences for providing the function key support for an IBM 3151 ASCII display.

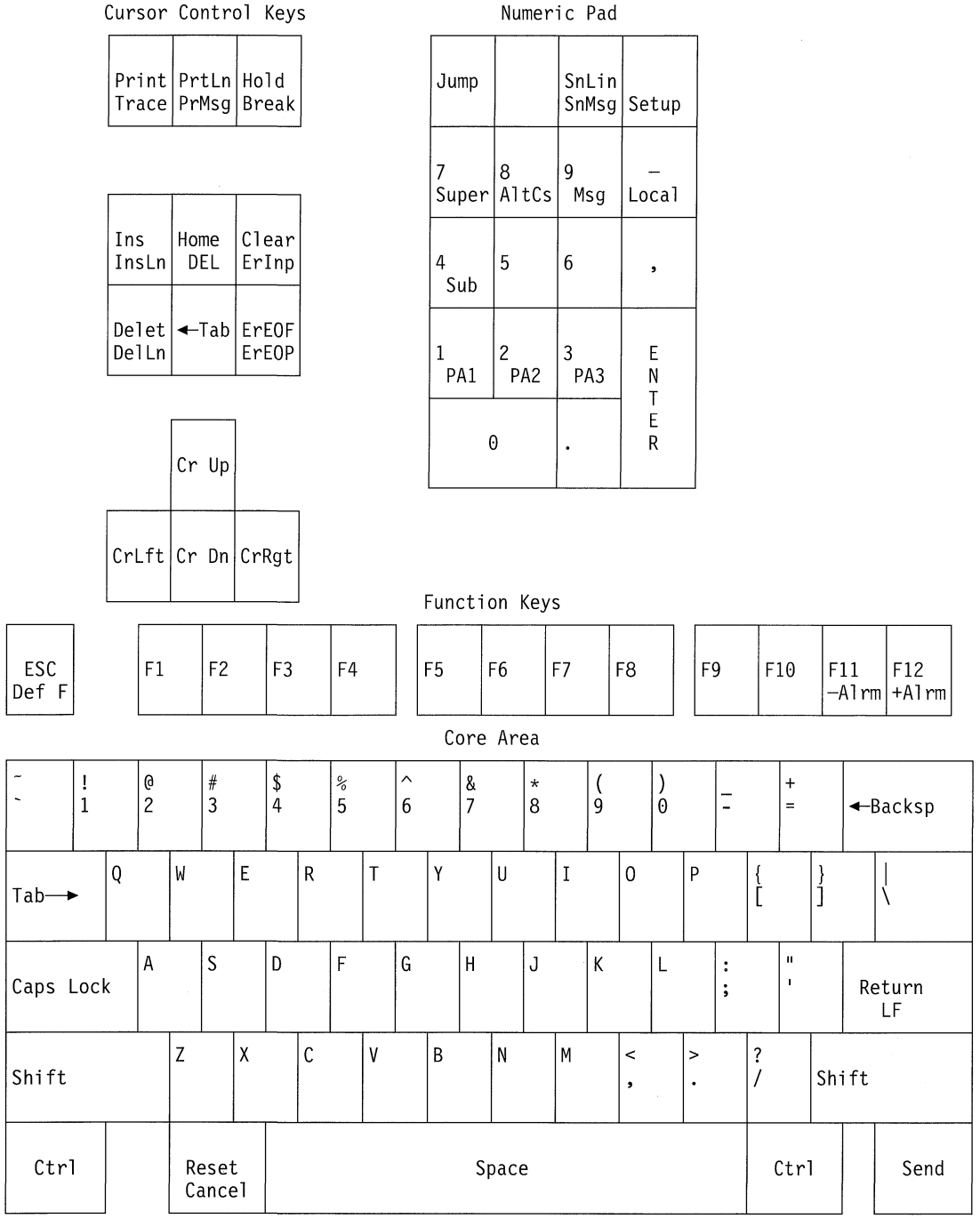
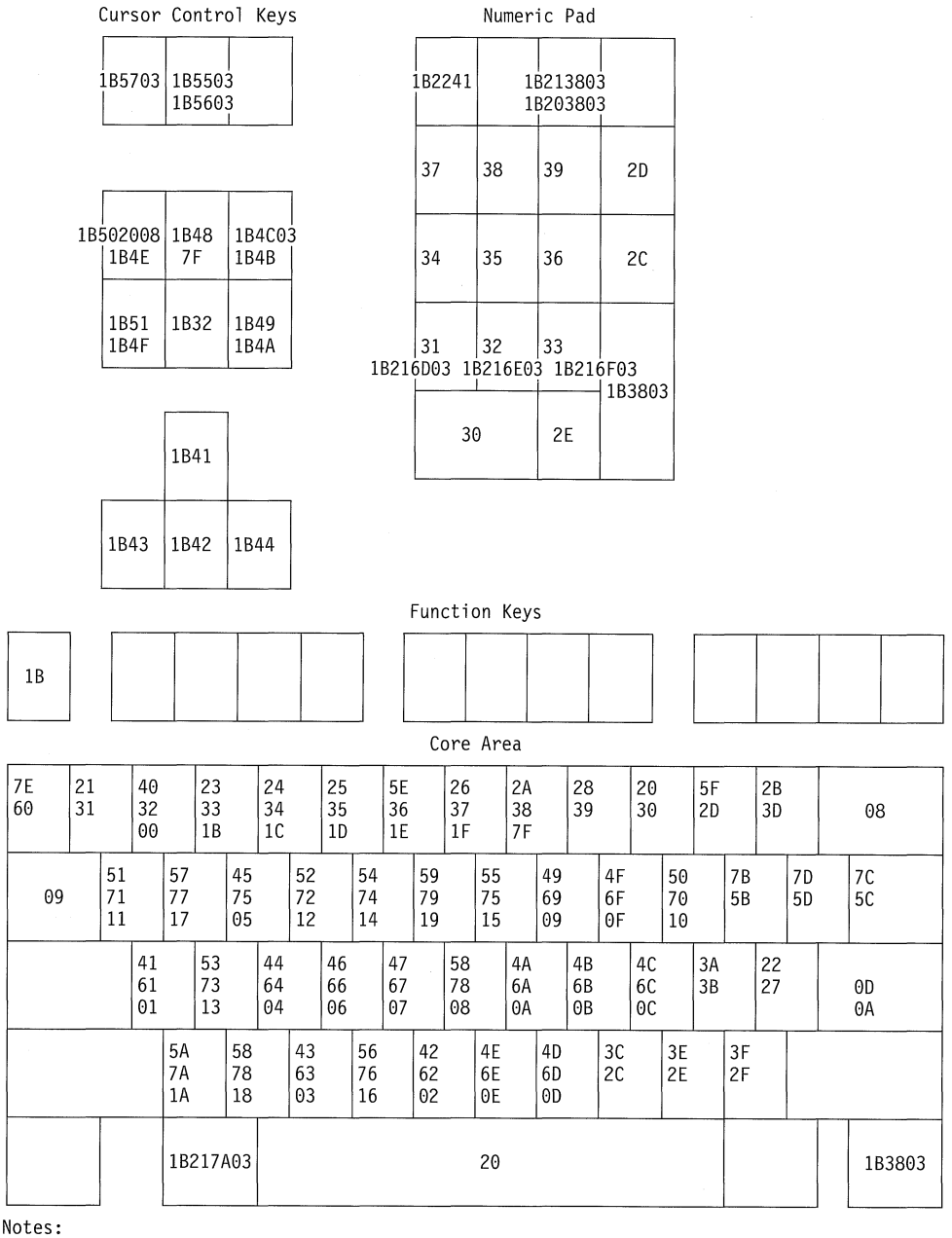


Figure 8-2. IBM 3151 ASCII Display Keyboard Layout



Notes:
1st line Shift state, 2nd line Unshifted, 3rd Line Ctrl

Figure 8-3. Hexadecimal Code Values for the IBM 3151 ASCII Keyboard

Table 8-1. Example: 3151 ASCII Code Sequences

F1–F12 (Unshifted)	Hexadecimal Data	F13–F24 (Shifted)	Hexadecimal Data	F25–F36 (Ctrl+Shift)	Hexadecimal Data
F1	1B6103	F13	1B216103	F25	1B226103
F2	1B6203	F14	1B216203	F26	1B226203
F3	1B6303	F15	1B216303	F27	1B226303
F4	1B6403	F16	1B216403	F28	1B226403
F5	1B6503	F17	1B216503	F29	1B226503
F6	1B6603	F18	1B216603	F30	1B226603
F7	1B6703	F19	1B216703	F31	1B226703
F8	1B6803	F20	1B216803	F32	1B226803
F9	1B6903	F21	1B216903	F33	1B226903
F10	1B6A03	F22	1B216A03	F34	1B226A03
F11	1B6B03	F23	1B216B03	F35	1B226B03
F12	1B6C03	F24	1B216C03	F36	1B226C03

Notes:

1. The keyboards on other ASCII displays may have some of the same functions as those shown for the 3151. In general, however, the keyboards on ASCII displays vary from one manufacturer to another, and the assignment of control codes or character sequences varies a great deal from one display device to the next.
2. If the data sent to the workstation controller for noncharacter keyboard functions contains more than one byte, then the first byte is almost always the Esc control character ('1B'X). This is also the case for most ASCII displays other than the 3151.
3. Different data can be returned when a key is pressed depending on whether the Shift or Control (Ctrl) key is being held down at the same time the key is pressed.
4. Some of the labeled functions shown in Figure 8-2 on page 8-6 do not send data back to the workstation controller. In most cases, a local display function is performed. Examples of this are the Hold and Trace keyboard functions on the 3151 display, which are local display functions.
5. The data passed back to the workstation controller for keys containing graphic character data is always in the range '20'X through '7E'X. The data sent back for all other functions, such as noncharacter keyboard functions, always begins with data in the range '00'X through '1F'X or '7F'X.
6. For many of the character and control codes generated by the keyboard and sent to the ASCII workstation controller, it is not possible to identify exactly which key was pressed. For example, the control code of '09'X is sent to the workstation controller as a result of either pressing the Tab key or pressing the key sequence Ctrl I. Similarly, the code for the number 1 ('31'X) can originate from either a key on the top row of the keyboard or from one of the keys on the numeric key pad.

   The data sent to the ASCII workstation controller from the keyboard does not allow the workstation controller to determine which key has been pressed. Only the particular function or character that you are trying to call is known.

Processing Data for an ASCII Display

The following list shows the names of the mapping tables the workstation controller uses to map data between two formats: the format required by the ASCII display and the format used to process data for AS/400 applications.

- ASCII-to-EBCDIC character mapping table
- ASCII to keyboard function mapping table
- Update screen table
- EBCDIC-to-ASCII character mapping table

Inbound and Outbound Processing: One of the things you need to think about when you customize an ASCII display is inbound and outbound processing. The ASCII-to-EBCDIC character and the ASCII to keyboard function mapping tables are the tables used to process data from the ASCII display keyboard. This type of processing is called inbound processing. The update screen table and the EBCDIC-to-ASCII character mapping table are the tables used to process data coming to the ASCII display from the AS/400 system and the ASCII workstation controller. This type of processing is called outbound processing. Figure 8-4 on page 8-9 and Figure 8-5 on page 8-9 show the flow for both types of processing.

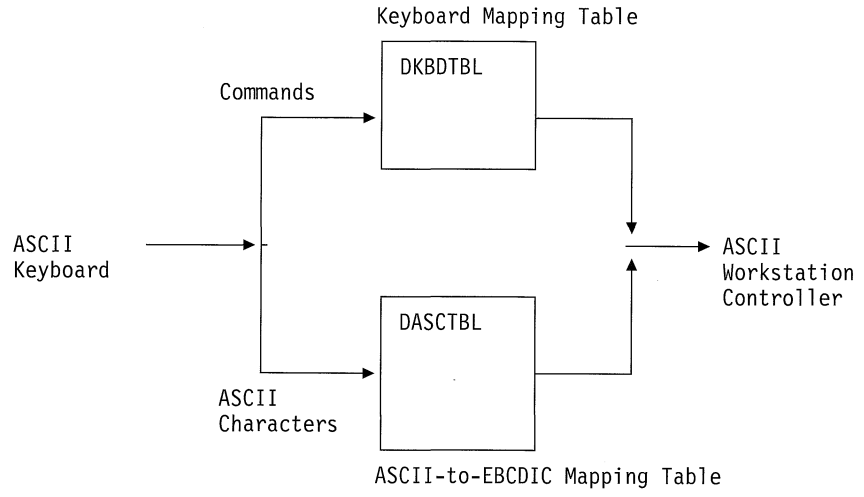


Figure 8-4. Inbound Processing for an ASCII Display

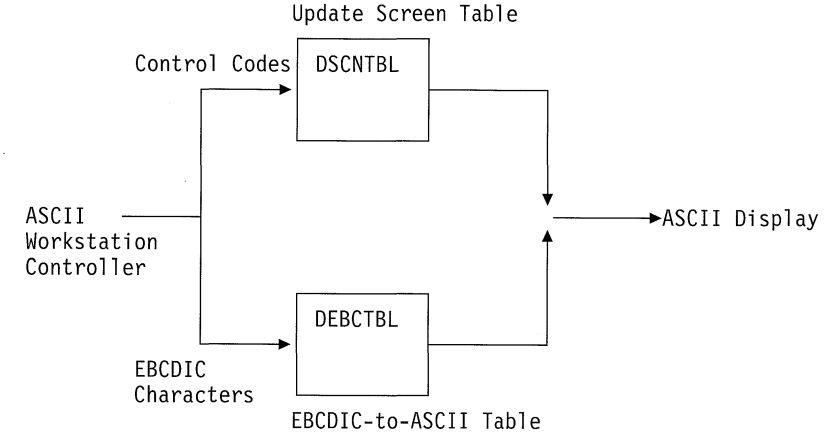


Figure 8-5. Outbound Processing for an ASCII Display

It is recommended that you start customizing the mapping tables for outbound processing first, and then customize the tables for inbound processing. This is so that what you see on the display is correct before you try to customize the keyboard for special characters and functions.

The following sections describe the general characteristics of the mapping tables to help you better understand how to change the workstation customizing source for these tables.

Mapping Tables for ASCII Display Keyboards

The ASCII-to-EBCDIC mapping table and the ASCII to keyboard function table are used to convert data received from the ASCII display keyboard into the internal codes used for processing keystrokes. Figure 8-6 on page 8-10 shows the two tables and how they each fit into the process of generating the internal codes for keystroke processing (inbound processing).

The flow of operations shown in the figure is from the ASCII display to the common code for the workstation controller. The ASCII display sends data to the workstation controller when you press keys on the ASCII keyboard. An example of the type of data passed to the workstation controller for particular keys is shown in Figure 8-3 on page 8-7.

The ASCII-specific code in the workstation controller provides two paths for the processing of inbound data. One path handles the mapping of ASCII code values associated with graphic characters, while the other path handles mapping ASCII control characters and control character sequences.

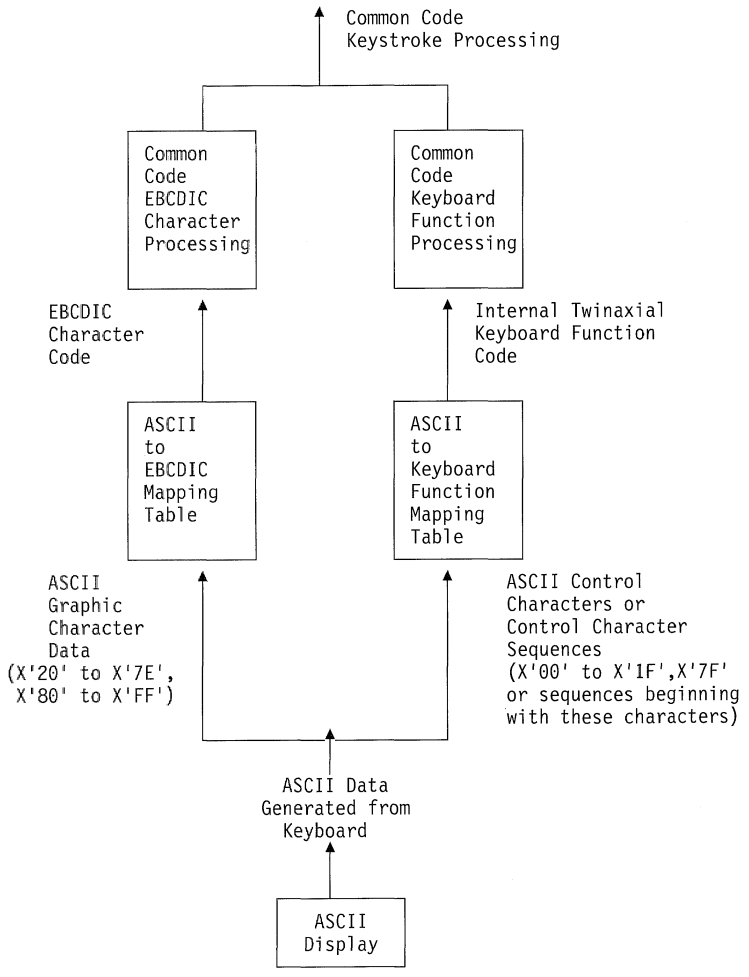


Figure 8-6. ASCII Display Keyboard Mapping Tables

Mapping ASCII Graphic Character Data: The ASCII workstation controller uses the ASCII-to-EBCDIC mapping table to handle the mapping of ASCII code values for graphic characters.

Using the ASCII-to-EBCDIC mapping table, the workstation controller converts an ASCII character code value in the range '20'X through 'FF'X into a single EBCDIC character code value. The particular ASCII-to-EBCDIC mapping table used depends on the ASCII character set used by the display and the EBCDIC code page associated with the national language configured for the display. Using the workstation customizing functions, you can customize the ASCII-to-EBCDIC mapping table that the ASCII workstation controller uses to support an ASCII display configured for a particular national language.

Mapping ASCII Control Characters and Control Sequences: The ASCII workstation controller uses the ASCII to keyboard function table to map ASCII control characters and control character sequences from the ASCII display keyboard to the internal processing codes used by the workstation controller common code.

When the data received from an ASCII display is a control character or a character sequence beginning with a control character, the ASCII workstation controller compares that data with information in the ASCII to keyboard function mapping table to determine whether or not it matches a mapping sequence in the table. The workstation customizing functions allow you to customize this table. Control characters and control sequences are mapped to one of the following:

- A twinaxial display function key request
- A local function supported by a display, such as a screen refresh, toggle indicator, or a terminal disconnect
- A request to set the state for processing subsequent data from the device

Mapping Twinaxial Display Function Key Requests: The twinaxial display function key request is the primary mapping function provided by this table. When the incoming data matches the information specified in the table for a function key request, a code representing the requested twinaxial keyboard function is generated and sent to the common keystroke processing code in the ASCII workstation controller.

When customizing an ASCII to keyboard function table for twinaxial display function key requests, you can be sure that the workstation customizing functions generate the correct

internal code for the function key specified. You only need to be concerned with specifying the ASCII command or control sequence for a given function key tag.

Table 8-6 on page 8-29 shows the workstation customizing tags you can use to customize the twinaxial keyboard functions with the ASCII to keyboard function mapping table.

Mapping Local ASCII Display Functions: The ASCII workstation controller supports three local display functions unique to using a display in an ASCII or asynchronous environment. The ASCII display calls these local display functions using a predefined key sequence that you enter from the keyboard. These functions are local to the ASCII workstation controller and the ASCII display. The following paragraphs describe the local display functions:

Screen Refresh Functions
When this function is called, the ASCII workstation controller retransmits all the data required to display the current image on the screen. This information is maintained by the internal screen image buffer of the ASCII workstation controller. To specify this function or change the way it is called, use the SCNREFRESH (screen refresh) tag following the DKBDTBL (keyboard function table) tag.

Terminal Disconnect Function
When this function is called, the ASCII workstation controller internally simulates a device power-off and power-on sequence for a display. When a display is attached by way of remote modems to the ASCII workstation controller, this ends or drops the remote connection. To specify or change the way this function is called, use the DISC (terminal disconnect) tag following the DKBDTBL (keyboard function table) tag.

Toggle Display Indicators
Twinaxial conditioning indicators are emulated on ASCII displays by showing an asterisk character in the last column of the display. The last column of rows 8–18 are reserved to indicate which display status conditions are currently active; for example, you may be in insert mode, input may be inhibited, or your keyboard may be in a certain shift state. When you turn on the display of indicators, data that might normally appear in these locations on the screen is overlaid by either an asterisk, indicating a status condition is active, or a blank, indicating a status condition is not active. When the indicators are turned off, the data that would normally appear in those locations is displayed. To specify or change the way this function is called, use the TOGIND (toggle display indicator) tag following the DKBDTBL (keyboard function table) tag.

Note: Character data that is overlaid by the conditioning indicators is still present in the internal screen buffer of the ASCII workstation controller. The data is not lost to an application.

Setting the State for Inbound Data Processing: Some other general types of functions specified in the ASCII to keyboard function mapping table are functions to set the state for processing inbound data. The following list describes the three functions for setting the state that you can specify using the workstation customizing functions.

- **Shift-Out State**: This function sets the ASCII workstation controller into a shift-out state for this display. When this state is set, ASCII character data received from the display in the range from '20'X through '7E'X is mapped to EBCDIC character data as though the ASCII character received is in the upper range of the ASCII character set ('A0'X through 'FE'X.) This function may be set by the SHIFTOUT (shift-out) tag following the DKBDTBL (keyboard function table) tag.

  Note: This state is normally used with ASCII displays that have an ASCII code page covering a range of character values from '20'X through '7E'X and 'A0'X through 'FE'X, but are operating in 7 bit mode which only allows a maximum value of '7F'X to be specified. The ASCII display uses the Shift Out control character to switch to the higher range of the code page.
- **Shift-In State**: This function sets the ASCII workstation controller into a shift-in state for this display. When this state is set, ASCII character data received from the display in the range from '20'X through '7E'X is processed as character data. If the display has previously been set to the shift-out state, this resets it. This function may be set by the SHIFTIN (shift-in) tag following the DKBDTBL (keyboard function table) tag.
- **Set Read Status State**: This function sets the ASCII workstation controller into a read status state for an ASCII display. When the attached display is an IBM ASCII display with an auxiliary printer configured for the display, the workstation controller periodically sends the display a command to check whether a printer is attached and powered on at the auxiliary port on the display. IBM ASCII displays return a five-byte response to this command. When the ASCII workstation controller detects the first two bytes of this response, it goes into read status state, verifying that the remaining bytes of the response have been received correctly. If the response data indicates that the printer is powered on, the power-on status for the auxiliary printer is sent back to the AS/400 system and the controller stops sending the read status command to the display.

  The first two bytes of the read status response from the display are specified in the ASCII to keyboard function table by the READSTS (read status) tag that follows the DKBDTBL (keyboard function table) tag. For more information about using these tags, and the restrictions on specifying the READSTS tag, see "Working with the ASCII to Keyboard Function Mapping Table" on page 8-28.

Mapping Tables for ASCII Display Screens

To format the ASCII display screen, the ASCII workstation controller generates ASCII data streams by mapping data, such as EBCDIC character codes, twinaxial display attribute control characters, and other control characters, from its internal screen image buffer into the appropriate string of ASCII characters and control characters.

The ASCII workstation controller uses two tables to perform this function, the update screen table and an EBCDIC-to-ASCII mapping table.

ASCII Update Screen Table: The update screen table contains ASCII display command information that allows the workstation controller to perform the following operations for the ASCII display:

- Set the type of highlighting that will be used at given locations on the screen.
- Position the cursor.
- Set the position on the screen at which data should be written.
- Clear the screen.
- Sound the audible alarm at the display.
- Initialize display parameters (setup commands).

The update screen table also contains information specifying other commands that are sent to support certain special workstation controller functions, such as 132-column support and support for text symbols. This table also contains the information used to specify device support characteristics for the display, such as the way the positions on the screen are addressed and the way display highlighting is handled. You can use the workstation customizing functions to customize this table. See "Working with the Update Screen Table" on page 8-15 for more information about what you can specify in this table.

EBCDIC-to-ASCII Code Mapping Table: The EBCDIC-to-ASCII mapping table converts an EBCDIC character code value from the internal screen image buffer into a single ASCII character code value. This table handles the processing of character code values from '00'X through 'FF'X, even though the range of valid EBCDIC characters that can be displayed is '40'X through 'FE'X. Table entries corresponding to attribute control characters ('20'X through '3F'X) are not used; however, entries for control characters in the range '00'X through '1F'X are used. These control characters usually are mapped into blanks. You can use the workstation customizing functions to customize this table.

For more information about this table, see "Working with the EBCDIC-to-ASCII Code Mapping Table" on page 8-27.

Display Commands for Unsupported Device Types: If you want to create the appropriate customized mapping tables for an ASCII display that is not currently supported by the ASCII workstation controller, you need to specify data in the ASCII display mapping tables that is probably quite different from the data that is retrieved from the source tables on the system. This is especially true for the ASCII to keyboard function table and the update screen table. For the update screen table, it is especially important that certain commands be present in the table. Without those commands, the ASCII workstation controller cannot perform the twinaxial device emulation functions.

At a minimum, you must specify commands for the following tags in an update screen table to obtain a minimal level of ASCII device support:

- CLRSCN (Clear Screen command)
- CSRADR (Set Cursor Address command)

If only these two commands are specified, basic 24x80 display support is provided on an ASCII display. However, there is no highlighting support for the screen. If the ASCII display does support highlighting on the screen, then the attribute commands should also be specified (ATRCMD tag). Most ASCII displays also support a command to sound the display alarm, so the ALARM tag can usually be specified also.

In most cases, you can obtain an acceptable level of display support by specifying the data for the CLRSCN, CSRADR, and ATRCMD tags (and possibly the ALARM tag). For more information about these commands and their associated tags, see the following sections:

"Clear Screen Command" on page 8-21
"Set Cursor Address Command" on page 8-21
"Attribute Command (ATRCMD) Tag" on page 8-24
"Sound Alarm Command" on page 8-21

Customizing Restrictions for ASCII Displays

When considering the customizing of an ASCII display station, keep in mind the following restrictions of the workstation customizing functions. Read these carefully before you continue to customize an ASCII display.

- An unsupported ASCII workstation must have characteristics that are similar to at least one of the supported ASCII workstations for the workstation customizing functions to work for you. If you do not have the reference manuals for any of the supported ASCII displays to use to compare characteristics with your unsupported display, you can check the general characteristics for supported ASCII displays in the *ASCII Work Station Reference and Example*. If you cannot find a supported ASCII display that has similar characteristics to your display, you should select the IBM 3101 display as the device type when you retrieve your workstation custom-

izing source. The 3101 ASCII display is the simplest of all the IBM ASCII displays. (See Appendix D, "Setting Up to Customize a Display" for a procedure to help you get the unsupported workstation to provide minimal function.)

- The ASCII workstation controller has a limited amount of space available for the ASCII to keyboard function table. The ASCII to keyboard function table that is downloaded to the ASCII controller for a given display has a variable length depending on the length of the ASCII control sequences that are specified and the number of keyboard function tags that are specified. When you are customizing an ASCII to keyboard function table, the workstation customizing functions inform you whether the customized table that is created is too large. If this occurs, you need to either delete some of the keyboard function tags from the source, or select ASCII control sequences for mapping that are shorter in length.
- The ASCII to keyboard function mapping table and the update screen table can vary in size depending on the data you specify in these tables. The amount of storage available in the ASCII workstation controller for these tables is limited, and error messages are sent to the job log if the size of the tables you customize is larger than the size allowed by the workstation controller.
- Many 7-bit displays, like the VT-52, do not support all the different national language characters, and you may not be able to display some special characters. Use the reference manual for the ASCII display to verify the characters the display is capable of supporting before you begin to customize it.
- When customizing the highlighting attributes for an ASCII display screen, the display must support highlighting functions on the screen in a way that agrees with either character-based or field-based highlighting. These are the methods that the ASCII workstation controller uses to handle highlighting attributes for an ASCII display. These types of highlighting are described in "The Highlighting Support Parameter (CHARATR)" on page 8-16.

  The appearance of data on the screen may not be correct if the display highlighting does not function in a manner consistent with one of these two methods. For example, some ASCII displays that support character-based highlighting do not change the appearance of a position on the screen when the current highlighting characteristic has been changed, but the character being written is a blank (ASCII '20'X). The ASCII workstation controller's character-based highlighting support assumes that a position on the screen takes on the appearance of the current highlighting characteristic regardless of which character is being written on the screen.

  In some situations, the ASCII workstation controller sends blank characters to clear certain sections of the screen. When using an ASCII display that works in this way, you would probably see the highlighting attributes from previous displays lingering when moving from one display to the next. For such ASCII displays, you may want to remove the ASCII attribute mapping specifications that result in this type of appearance on the screen from your customizing source.
- The READSTS (read status keyboard function) tag appears in the ASCII to keyboard function table when you retrieve your workstation customizing source for an IBM ASCII display (except for the IBM 3101). In general, you should not change this tag and its data when it appears in your workstation customizing source.

  When you do change the ASCII control sequence for the READSTS tag, the following restrictions apply:

  - Only 2 bytes should be specified for this tag.
  - The format of the read status data returned to the workstation controller by the ASCII display must be similar to the format which the ASCII workstation controller expects from an IBM ASCII display:
    - Read status response must be 5 bytes in length
    - Bytes 3 and 4 of the response should contain the status response information.
    - The fifth byte of the response must contain the line turnaround character (LTA) with a value of '03'X.
- The SHIFTOUT (set shift-out) and SHIFTIN (set shift-in) keyboard function tags specify the control character (or control character sequence) that the ASCII workstation controller uses to switch back and forth between mapping inbound ASCII characters from the lower and upper ranges of ASCII character spaces. You should be sure to specify both of these tags, if you decide to use them. When only one of these tags is specified, an ASCII display user can get locked in a state where the ASCII characters from the keyboard are not mapped correctly.
- If you are using port sharing for your ASCII display connections, there are some restrictions when using a workstation customizing object. Device type detection is not supported when you specify a customizing object in the device description; however, line speed, word length and parity checking are supported. The *ASCII Work Station Reference and Example* provides recommendations for the setup of your display and more information about ASCII port sharing for supported combinations of line speed, word length, and parity.
- When adding or changing tags in a workstation customizing source, you cannot specify a substring of the hexadecimal data for one tag as the hexadecimal data for another tag. This can produce unpredictable results. For example, the tag for the F1 key shown in Figure 8-7 on page 8-14 has a data value of '1B31'X. The tag for the F10 key has a data value of '1B3130'X. This could cause the controller to map the F10 key to the function for the F1 key or map the F1 key to the function for the F10 key.

```
:FKEY
        KEY = F1
        DATA = '1B31'X.  /*Substring of F10 key data*/
:FKEY
        KEY = F10
        DATA = '1B3130'X.
        .
        .
        .
```

Figure 8-7. Source Code Showing Data Substring Restriction

- Most ASCII workstations also provide functions that are local to the display; because these functions are local, they cannot be customized. These functions do not send codes to the ASCII workstation controller.

Determining Which ASCII Display Tables to Customize

The OS/400 workstation customizing functions allow you to customize an ASCII display by changing the hexadecimal values in the following mapping tables:

- ASCII to keyboard function mapping table (DKBDTBL tag)
- ASCII-to-EBCDIC mapping table (DASCTBL tag)
- Update screen table (DSCNTBL tag)
- EBCDIC-to-ASCII mapping table (DEBCTBL tag)

The tables used by an ASCII display are downloaded to the ASCII workstation controller at the time that the display is varied on. To accommodate the variety of device types, EBCDIC character sets, and ASCII character sets that the ASCII workstation controller must work with, there are a large number of tables stored on the system. When you retrieve and edit the source for an ASCII display, all four of the tables required to support a given ASCII display configured for a particular language are retrieved and ready to be changed.

If you want to change both the keyboard mappings and the way that characters are displayed, you may need to change all four of the tables the ASCII workstation controller uses to map data.

When you want to change only the way a character or characters appear on the display (outbound processing), you need to change the following tables in the workstation customizing source you retrieve:

- Update screen table
- EBCDIC-to-ASCII mapping table

When you want to change only the way the keyboard is interpreted (inbound processing), you need to change the following tables in the workstation customizing source you retrieve:

- ASCII to keyboard function mapping table
- ASCII-to-EBCDIC mapping table

Note: The ASCII workstation controller currently has restrictions on the language types that may be configured for an unsupported ASCII device type. Also, there are restrictions, regardless of whether the device is a supported or unsupported ASCII display, on selecting certain language types supported by the twinaxial workstation controller. Almost all such restrictions are removed when you specify a customizing object during device configuration. Using the workstation customizing functions, you can create a set of mapping tables to properly support an unsupported ASCII display for a national language other than US English (USB) or US International (USI).

The workstation customizing functions do not support the customization of double-byte character set languages. Restrictions for the Katakana language type still exist and are described where applicable.

Working with the Tag Language for ASCII Displays

When the device class parameter specified for the RTVWSCST command is an ASCII display (`DEVCLASS = ASCIIDSP`), your source structure looks like the following:

```
:WSCST DEVCLASS=ASCIIDSP.

 :DASCTBL with parameters. /*ASCII-to-EBCDIC mapping table*/

 :DKBDTBL.                /*ASCII to Keyboard mapping table*/
         .
         .
   keyboard function tags         /*tags with parameters*/
         .
         .
         .
         .

  :DEBCTBL with parameters. /*EBCDIC-to-ASCII mapping table*/

  :DSCNTBL with parameters. /*Update screen table*/
         .
         .
    update screen tags            /*tag with parameters*/
         .
         .
         .
         .

:EWSCST.
```

Figure 8-8. Source Structure for ASCII Displays

The primary tags in the workstation customizing source for an ASCII display are the following:

DASCTBL ASCII-to-EBCDIC mapping table tag
DKBDTBL ASCII to keyboard function mapping table tag
DEBCTBL EBCDIC-to-ASCII mapping table tag
DSCNTBL Update screen table tag

The default values for the parameters associated with a given tag depend on the device type you specify when you retrieve the workstation customizing source. These tags can occur in any order within the source file member; however, the secondary tags must follow the appropriate primary tag.

When a primary tag is missing from the source, the system default table associated with the missing primary tag is used. This is based on the device type and national language type you specified when you used the RTVWSCST command.

When a DKBDTBL tag is present in the source, but has no keyboard function tags following it, no keyboard functions are mapped when the workstation controller uses this customizing object. This may cause unpredictable results.

When a DSCNTBL tag is present in the source, but has no update screen tags following it, no functions to update the screen are mapped when the workstation controller uses this customizing object. It is probable that no data will appear on the screen when this occurs.

Using the Tags to Customize an ASCII Display Screen

The following sections describe the tags you use to customize the outbound processing for an ASCII display. These tags provide the commands and attributes that the ASCII display uses to show data received from the AS/400 system.

Working with the Update Screen Table

The update screen table contains information about the command sequences the ASCII workstation controller sends to an ASCII display in the process of updating the screen. Among the types of commands that are sent to the display are commands to clear the screen, position the cursor, and control highlighting. This table also contains information indicating the different support characteristics of the ASCII display.

The update screen table used to support an ASCII display is unique to the ASCII device type. It contains commands specific to that ASCII display. You specify the data for the update screen table by changing the hexadecimal values for associated tags in the workstation customizing source you retrieved for the ASCII display.

The following sections describe the tags and commands that allow you to set up the most basic screen characteristics. As previously mentioned, these may be the most important commands you use when using the workstation customizing functions with an unsupported ASCII display.

Update Screen Table (DSCNTBL) Tag: The update screen table tag, DSCNTBL, defines the screen table used to update an ASCII display screen. The syntax for this tag is:

```
:DSCNTBL
     CHARATR = FIELD|CHAR
     ADDRMOD = CHAR|BINARY
     TEXTSYM = NOSUPPORT|SUPPORT
     AUTOSCL = NO|YES.
```

Figure 8-9. Syntax for the Update Screen Table Tag

Note: The default values for the following parameters depend on the device type you specify when you retrieve the workstation customizing source.

CHARATR (Character)
An optional parameter. Specifies whether or not the display supports highlighting on the screen on a character basis or on a field basis. The default for this value depends on the ASCII display you specify for the device type when you retrieve the workstation customizing source. (For more information about the field or character basis, see "The Highlighting Support Parameter (CHARATR)" on page 8-16.)

FIELD
The display that the source is to customize supports highlighting on the screen on a field basis.

CHAR
The display that the source is to customize supports highlighting on the screen on a character basis.

ADDRMOD (Address mode)
An optional parameter. Specifies whether the addressing is by ASCII numeric characters or binary values.

CHAR
Addressing is by numeric characters.

BINARY
Addressing is by binary values.

TEXTSYM (Text symbol)
An optional parameter. Specifies whether text symbols are supported.

NOSUPPORT
Text symbols are not supported.

SUPPORT
Text symbols are supported.

AUTOSCL (Automatic scrolling)
An optional field. Specifies whether automatic paging is supported.

NO
Automatic paging is not supported.

YES
Automatic paging is supported.

The DSCNTBL tag is immediately followed by a number of individual update screen tags, which make up the table entries. The absence of a specific update screen tag after the DSCNTBL tag implies that particular update screen function will not be mapped. When the same update screen tag occurs in an update screen table multiple times following a DSCNTBL tag, warning messages are sent to the job log for each occurrence. The customizing object then uses the last occurrence of the update screen tag in the source. The only exception to this is the SCNSIZE (screen size) tag, which may appear with different values for the size of the screen up to three times in the update screen table.

The following sections describe the parameters for the :DSCNTBL tag in more detail.

The Highlighting Support Parameter (CHARATR): This parameter specifies whether the device supports highlighting on the screen on a field basis or a character basis.

Field-Based Highlighting: For field-based highlighting on an ASCII display, entire regions or fields of data on the screen are automatically shown with a particular highlighting characteristic (for example, blinking, reverse image, or underscore) when commands to set these characteristics are sent to the screen. The commands are directed to specific addresses on the screen that are determined by the current cursor position on the screen. When a command to set the highlighting on the screen for a specific position is sent, all data on the screen is shown with the selected highlighting from that position to the next position on the screen at which a highlighting attribute is defined.

As an example, assume that commands to set highlighting for three positions on the screen have been sent to an ASCII display that supports field-based highlighting as shown in the following figure:

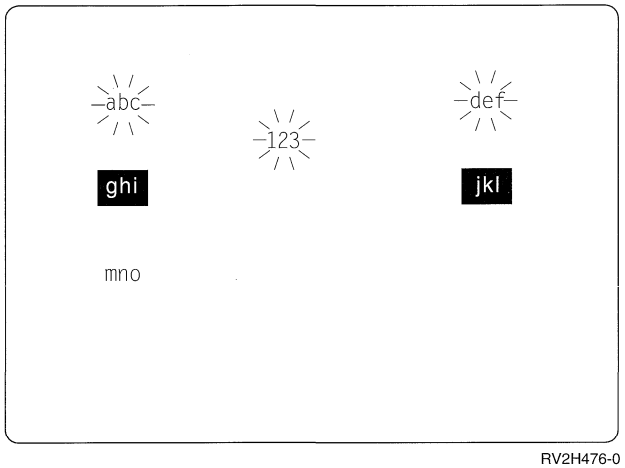


Figure 8-10. Setting Highlighting Attributes for the Screen

For field-based highlighting on an ASCII display, all data between the limits of where highlighting on the screen is defined are displayed with the highlighting attribute defined for the beginning of that region. Therefore, in this example, the characters `abc`, `def`, and `123` are displayed as blinking data. The actual field defined to use the blinking attribute begins just before the character `a` and ends just after the character `3`. The characters `ghi` and `jkl` are shown as reverse image data. This field begins just before the character `g` and ends just after the character `l` The remainder of the display is treated as 1 large field and is set to normal highlighting. Therefore the characters `mno` are displayed with normal highlighting.

If a command is now sent to set the highlighting on the screen at row 5, column 15 (between `abc` and `def`) to underscore, all characters between this position and the next highlighting attribute position (row 7, column 10) are now shown as underlined data. Although the underlined highlighting is only shown for the characters `def` and `123`, the actual field that is highlighted begins with the character `d` and ends with the last space before the character `g`.

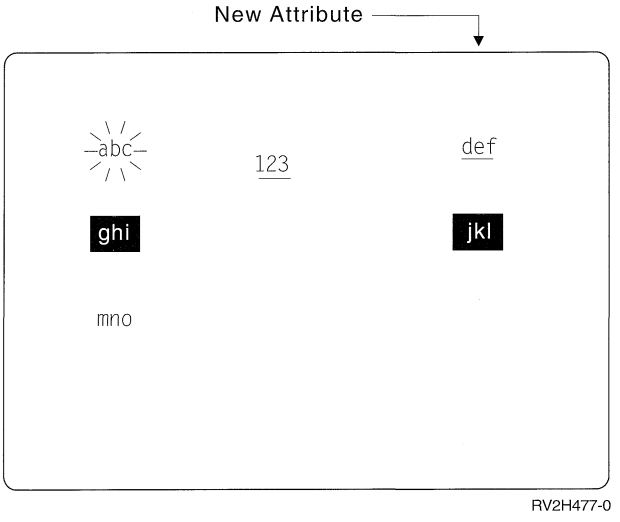


Figure 8-11. Adding New Highlighting Attributes

Character-Based Highlighting: On ASCII displays that support character-based highlighting on the screen, a given character on the screen is displayed with a particular type of highlighting only when that character is shown on the screen following the receipt of a command to set the desired type of highlighting. Sending the command to set the type of highlighting on the screen, by itself, does not change the way that the data currently on the screen is highlighted. However, all character data that is shown on the screen following the command to set the highlighting is shown with the selected highlighting. This highlighting is used to display all characters subsequently written on the screen, until another command to set the highlighting to something different is received by the display screen.

Therefore, sending a command to set the highlighting on the screen on a display that supports character-based highlighting does not change the way that data currently shown on the screen is shown. For example, assume that the following display has been sent to such a display screen with the data highlighted as shown in Figure 8-12 on page 8-17.

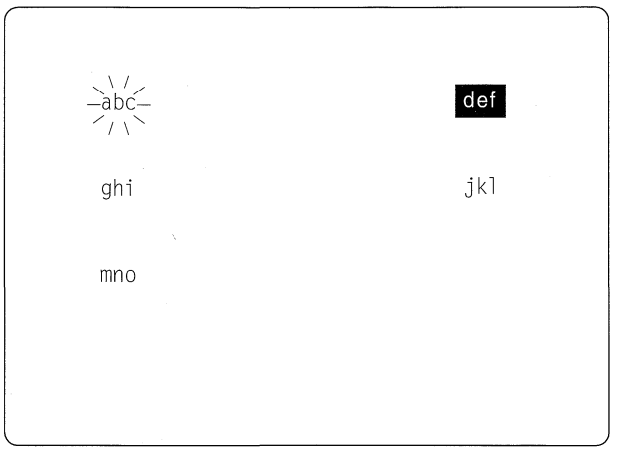


Figure 8-12. Displaying the Highlighted Characters (1)

To write the characters 123 on the screen between the abc and def with the underscore highlighting, the following sequence of commands must be sent to the screen:

1. Position the cursor to the appropriate address on the screen between the abc and def characters.
2. Set the highlighting to underscore.
3. Write the characters 123.

The resulting data on the screen would appear as follows:

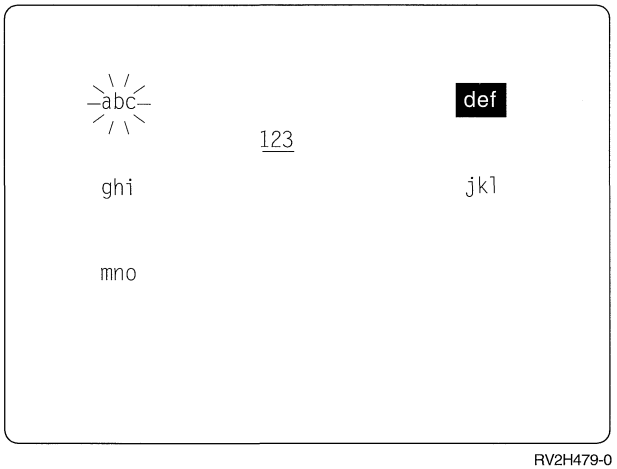


Figure 8-13. Displaying the Highlighted Characters (2)

Note that the data previously shown on the screen does not change. To write the characters def on the screen to be shown as underlined characters in this example, the following additional commands must be sent following the three listed previously listed.

1. Position the cursor from the current screen address to the location on the screen where the character d is shown.
2. Rewrite the characters def on the screen.

When selecting the CHARATR parameter for an ASCII display, you should refer to the manual for the display to determine whether highlighting on the screen is performed on a field or character basis. You can then set this parameter appropriately and use the ATRCMD (attribute command) tag to allow screen highlighting.

The Address Mode Parameter (ADDRMOD): This parameter specifies whether data for the ASCII display is addressed by ASCII decimal numeric characters or binary values.

Decimal Numeric Character Addressing: For ASCII displays that support addressing by numeric characters, the commands for positioning data on the screen require the address data to be specified as numeric character ASCII codes for each digit in the address.

For the decimal numeric character addressing format, the address on the screen is represented by separate character or character sequences for the row and column of the address. The character or character sequence consists of ASCII codes for the numeric characters (graphic characters 0 through 9, represented by the ASCII codes '30'X through '39'X).

For example, a row value of 18 is specified using a sequence of ASCII characters '31'X (for 1) followed by '38'X (for 8). A column address of 63 is represented by the sequence '36'X (for 6) followed by '33'X (for 3). Each row or column value is then represented by a separate ASCII code for each order of ten in the row or column value. For either a row or column value, there is one numeric character code for the tens unit and one numeric character code for the ones unit. There is also a separate character code for a column hundreds unit, if needed.

When specifying data for the command to set the cursor address on a display screen that uses numeric character addressing, the position of each address unit in the command sequence must be specified. This is shown in the following example.

Assume that the general format for setting the cursor address for an ASCII display that uses numeric character addressing is as follows:

ESC [R_1R_2 ; C_1C_2 f

1B 5B - - 3B - - 66

R_1R_2 represent two numeric character values that specify the row address and C_1C_2 represent two numeric characters that specify the column address. For example, R_1R_2 would be '3132'X for a row address of 12, C_1C_2 would be '3637'X for a column address of 67. When the ASCII workstation controller sends this command to the screen, the values in this command for the row and column address at any time depend on the position on the screen that is currently being addressed.

When specifying this command for the CSRADR (set cursor address) tag for the update screen table, some default values need to be specified for the row and column values in the Data parameter of the command. The default values specified should represent a screen address of row 1 and column 1. The following shows how the CSRADR tag should be specified for this particular example:

```
:CSRADR
      ROWTENS = 3
      ROWONES = 4
      COLHUNDS = 0
      COLTENS = 6
      COLONES = 7
      DATA = '1B5B30313B303166'X.
```

Notes:

1. It is very important that the address on the screen for the data specified in the command corresponds to a screen address of row 1 and column 1. The ASCII workstation controller saves the row and column information specified here and later uses these saved values as a base in calculating the row and column values to be used when sending the Set Cursor Address command for a particular screen address. Characters may not be positioned properly on an ASCII display screen if the row and column values specified do not correspond to row 1 and column 1.
2. The ROWTENS, ROWONES, COLTENS, and COLONES parameters are each set to the offset into the DATA sequence of the byte representing that address unit. The third byte in the data is the tens unit of the row address, so ROWTENS is set to 3. The sixth byte in the DATA sequence is the tens unit column address, so COLTENS is set to 6.

   Some ASCII displays use a zero-based addressing scheme for addressing data on the screen. For example, the first screen position may be referenced by the display as row 0 and column 0. In this case, the initial values specified in the set cursor address command should reflect this zero-based addressing scheme. For this example, the Data value specified on the CSRADR tag for a display that uses zero-based addressing would look like the following:

   ```
   :CSRADR
       DATA = '1B5B30303B303066'X.
   ```
3. The COLHUNDS byte has been set to zero in this example. This indicates that the Set Cursor Address command format for this device does not provide for a hundreds unit in the column address. This is the case for most ASCII displays supported by numeric character addressing. When specifying the CSRADR tag, you should be sure not to set the COLHUNDS parameter unless the Set Cursor Address command format for the display absolutely requires it.
4. None of the ASCII displays that use numeric character addressing and that are currently supported by the ASCII workstation controller have a hundreds parameter byte in their column address. The COLHUNDS parameter will always be zero when retrieving the source tables for these displays.

Binary value addressing: For ASCII displays that support binary value addressing, the address locations on the screen are specified in commands for positioning data by ASCII character code values that are arbitrarily assigned to row and column positions. Most ASCII displays that support binary addressing assign the ASCII codes for graphic characters ('20'X through '7F'X) to consecutively represent a given row or column position. Therefore, the first ASCII character code, '20'X, is assigned for a row or column value in the command for positioning data if the data were to be positioned at row 1 or column 1. Following from this, a value of '21'X would represent a row or column address of 2, '22'X would represent a row or column address of 3, and so on.

Other displays that use binary value addressing for their Set Cursor Address command may use a range of ASCII character code values other than '20'X through '7E'X. For example, some ASCII displays use a range of ASCII character codes beginning at '00'X rather than '20'X. You should specify the value corresponding to an address of 1 for the data command sequence of the CSRADR (set cursor address) tag.

When specifying data for the Set Cursor Address command of a display that uses binary value addressing, the position of the row and column address bytes in the command sequence must be specified. These positions are indicated by the ROWTENS and COLHUNDS parameter bytes of the CSRADR tag. This is shown in the following example.

Assume that the Set Cursor Address command for a display has the following format:

```
ESC X  Pr Pc
1B  58 -  -
```

Pr and Pc are single-byte values representing the specific row and column to be addressed. For this example, assume that the range of address values starts with '20'X (so that a row or column value of 2 is specified by '21'X, and so on.).

When specifying this command on the CSRADR tag, the default row and column values specified should be set to a screen address of row 1, column 1. The following lists the specification of the CSRADR tag for this example:

```
CSRADR
  ROWTENS = 3
  ROWONES = 0
  COLHUNDS = 4
  COLTENS = 0
  COLONES = 0
DATA = '1B582020'X.
```

Notes:

1. Only the ROWTENS and COLHUNDS parameters should be set to nonzero values. The values set for each indicate which bytes in the command sequence are the row and column parameters, respectively.
2. This command is used to address data on the screen using binary value addressing whenever the position on the screen being addressed is in columns 1 through 80.
3. Most ASCII displays attached to the ASCII workstation controller are supported as 24-row by 80-column screens. If the screen uses binary value addressing, but is being supported by the workstation controller as a screen with 132-column capability, this command is not used to address data beyond column 80. Instead the Extended Set Cursor Address command (specified by the XCSRADR tag) is used to address data in columns 81 through 132.
4. If the range of values used by the screen for its row and column address values had been different, the default row and column values specified for the Data value would still be set to represent a screen address of row 1 and column 1. For example, if the range of address values started at '1F'X instead of '20'X the data value in this example should have been specified as DATA='1B581F1F'X.

Whether supported by numeric character addressing or binary value addressing, specifying a Set Cursor Address command in the update screen table is a basic requirement for any ASCII display that is to be supported by the ASCII workstation controller. When customizing the update screen table for a supported ASCII display, this tag should not be removed. Likewise, when customizing this table for a new ASCII device type (one not supported by the ASCII workstation controller), the update screen table tags you specify should always include the CSRADR tag.

The reference manual for your ASCII display should describe the type of addressing supported by the display. You should set the ADDRMOD parameter to correspond with what your display supports.

Be aware that the way you set the ADDRMOD parameter determines how the ASCII workstation controller subsequently uses data that is specified for the CSRADR tag in the update screen table. The format of the data specified for the CSRADR tag should agree with the type of addressing mode set by the ADDRMOD parameter. After you have selected either numeric character or binary addressing, you should specify the data for the Set Cursor Address tag (CSRADR) in the update screen table to agree with the selected type of addressing. See "Set Cursor Address (CSRADR) Tag" on page 8-21 for a complete description of the format you should use for the CSRADR tag for the different types of addressing.

The Text Symbol Parameter (TEXTSYM): This parameter specifies whether or not the ASCII workstation controller displays text symbols on this ASCII display. Text symbols are graphic symbols representing control characters that may appear on a display when using the OfficeVision/400 text editor on the AS/400 system.

As an example, text symbols are supported by the ASCII workstation controller for the IBM 3151 displays. Table 8-2 shows an example of some of the graphic characters that are shown on a 3151 ASCII display for the listed control characters.

Table 8-2. Graphics for Text Control Characters

Text Control Character	IBM 3151 Displayed Graphic
Carrier Return	┘
Required Carrier Return	◄
Required Tab	►
Tab	┤

The characters shown in Table 8-2 are in the graphic character set for the 3151 ASCII display. The ASCII workstation controller, when required by the OfficeVision/400 editor, sends commands to the 3151 display to allow these graphic characters to be displayed.

If text symbols support is not specified (TEXTSYM=NOSUPPORT), the ASCII workstation controller never sends commands to display any text symbols. The OfficeVision/400 editor can still be used, but no graphic character appears for any of the text control characters.

If text symbols support is specified (TEXTSYM=SUPPORT), the display should support a special graphics character set (containing, for example, such graphics as line drawing characters) and a command for selecting this graphics character set. When you specify this support, the ASCII workstation controller uses the data specified for several of the other update screen table tags also. Therefore, these tags and data should be specified in your workstation customizing source when you indicate text symbol support. The affected tags are:

CARRTN
RQDCARRTN
HLFIDXUP
HLFIDXDN
PAGEND
RQDSPC
TAB
RQDTAB
STOPCODE
WORDUS
GCS

The last tag listed (set graphic character set, GCS) specifies the command that the ASCII workstation controller sends to an ASCII display to select the graphic character set of the display. The other tags listed specify the character code values in that graphic character set to which each of the text control symbols is mapped. See the descriptions of these

tags in "Update Screen Tags" on page 8-20 for more information about the specification of text symbols support.

Mapping Text Symbols: Of the update screen table tags shown in the previous list, the TAB (tab function) and GCS (graphic character set) tags do not indicate the way that text control characters are mapped. The remaining update screen table tags specify how text control characters are to be mapped by the ASCII workstation controller. For each of these tags, a single ASCII character code value should be specified in the data parameter. The ASCII character code value should correspond to the value assigned to a graphic character in an ASCII display's graphic character set. The ASCII workstation controller sends a command to select the graphic character set whenever it is going to display one of these text symbols. This command is specified by the GCS tag.

When the ASCII workstation controller selects the ASCII graphic character set, it assumes that the graphic character set is loaded into the upper range of an 8-bit ASCII character space (the range of ASCII codes from '80'X through 'FF'X). Data specified for the text symbol mapping tags should be within this range of values.

The data specified for the text symbol mapping tags is only used by the ASCII workstation controller when the text symbols parameter for the update screen table tag is set to indicate that text symbols are supported (TEXTSYM=SUPPORT).

The Automatic Scrolling Parameter (AUTOSCL): This parameter specifies whether the display automatically moves all the data on the screen up one line when data is written into the last screen position of the display.

If the ASCII display does not support this paging operation for the last screen position, this parameter should be set to NO. If the display does support paging, set the parameter to YES. When this parameter is set to YES, the ASCII workstation controller does not write data in the last screen position of the display.

The following sections describe the individual tags that you can specify following the DSCNTBL (update screen table) tag.

Update Screen Tags

The tags described in this section allow you to specify the ASCII control sequence for an individual update screen function for an ASCII display. The update screen tags must follow the DSCNTBL (update screen table) tag in your source.

The DSCNTBL tag and its entries define the update screen table for your ASCII display. This table is limited to 512 bytes in size. The hexadecimal values for the update screen tags that make up the entries for this table can be quite large. Although the size restriction for many of the individual update screen tags is a maximum of 255 bytes, when you specify values of this size for more than two update screen tags you can run out of space in the update screen table for your display.

With some exceptions, the update screen tags have the same general syntax. The tags that are exceptions to the general syntax are the ATRCMD (attribute command), SCNSIZE (screen size), CSRADR (set cursor address) and XCSRADR (extended set cursor address) tags. The syntax for these tags is described separately.

The general syntax for an update screen function tag is shown
in the following figure.

```
:xxxxx
        DATA = ASCII control sequence.
```

Figure 8-14. General Syntax for the Update Screen Tags

DATA
A required parameter. Specifies the ASCII control sequence for the screen function. The maximum length of this value is 255 bytes, unless otherwise noted. This data must be coded as a hexadecimal value.

ASCII control sequence
Hexadecimal values for mapping ASCII control sequences to ASCII display functions.

Table 8-3 lists the update screen function tags.

Table 8-3 (Page 1 of 2). Descriptions of the Update Screen Tags

Update Screen Tag	Description
ACS	Set ASCII character set
ALARM	Sound alarm command
CARRTN[1]	Carrier return
CLRSCN	Clear screen command
CSROFF	Set Cursor Display Off
CSRON	Set Cursor Display On
ENDBYP	End printer data bypass
GCS	Set graphic character set
HLFIDXDN[1]	Half index down
HLFIDXUP[1]	Half index up
INSCSR	Insert cursor command
NLCS	Set national language character set
PAGEND[1]	Page end
RQDCARRTN[1]	Required carrier return
RQDSPC[1]	Required space
RQDTAB[1]	Required tab
SETUP	Setup command
STOPCODE[1]	Stop code
STRBYP	Start printer data bypass

Table 8-3 (Page 2 of 2). Descriptions of the Update Screen Tags	
Update Screen Tag	**Description**
TAB[1]	Tab
WORDUS[1]	Word underscore
1 For these tags, the associated DATA value must be no more than 1 byte in length. See "Update Screen Tags" for information about coding these values.	

Clear Screen Command: This command is specified by the CLRSCN (clear screen) tag. It is one of the basic commands for setting up and using an unsupported ASCII display. The CLRSCN tag specifies a command that is sent to the ASCII display to clear or erase all data currently displayed on the screen.

In managing the appearance of data on an ASCII display, one of the operations most frequently done by the ASCII workstation controller is to clear the display screen. An ASCII display supported by the ASCII workstation controller depends on the existence of a valid command for clearing the screen in the update screen table so that the twinaxial device emulation works effectively. When customizing the update screen table for a display already supported by the ASCII work station controller, you should never delete the CLRSCRN tag and its associated hexadecimal data. When customizing the update screen table for a new or unsupported ASCII device type, the CLRSCN tag should always be present in the update screen table for a display.

Set Cursor Address Command: To use this command, you need to specify the CSRADR (set cursor address) tag in your workstation customizing source. This is one of the basic commands used for setting up and using an unsupported ASCII display. When this command is sent to an ASCII display to position the cursor or to set the screen address, all subsequent data received by the display appears beginning at that screen position.

The ASCII workstation controller supports two different formats for basic addressing of screen data on a 24-row by 80-column display screen. The format used is determined by the value you specify for the ADDRMOD parameter of the update screen table tag. Data for the CSRADR tag needs to be set differently depending on whether you select addressing by numeric characters or addressing by binary values. See "The Address Mode Parameter (ADDRMOD)" on page 8-17 for a more complete description of the two types of addressing and some examples of how the CSRADR tag could be specified for each.

Set Cursor Address (CSRADR) Tag: The CSRADR tag specifies the ASCII control sequence for the set cursor address command. The data you specify for this tag should provide the ASCII control sequence for positioning the cursor at row 1, column 1 on the screen. The ASCII workstation controller uses the values you specify here to calculate the offsets to the other screen addresses. Therefore, it is important that you specify the correct address information to be used as a base for calculating the offsets correctly. For an example showing how this tag is used, see "The Address Mode Parameter (ADDRMOD)" on page 8-17. The syntax for this tag is:

```
:CSRADR
      ROWTENS = row position tens byte
      ROWONES = row position ones byte
      COLHUNDS = column position hundreds byte
      COLTENS = column position tens byte
      COLONES = column position ones byte
      DATA = ASCII control sequence.
```

Figure 8-15. Syntax for the Set Cursor Address Tag

Note: If one or more of these values do not apply to the device you are customizing, specify those values as 0 (zero).

ROWTENS
A required parameter. Specifies the position of the Set Cursor Row (tens unit) byte in the control sequence. This value must be an integer. For binary value addressing, this specifies the position of the set cursor row byte in the control sequence.

ROWONES
A required parameter. Specifies the position of the Set Cursor Row (ones unit) byte in the control sequence. This value must be an integer.

COLHUNDS
A required parameter. Specifies the position of the Set Cursor Column (hundreds unit) byte in the control sequence. This value must be an integer. For binary value addressing, this specifies the position of the set cursor column byte in the control sequence.

COLTENS
A required parameter. Specifies the position of the Set Cursor Column (tens unit) byte in the control sequence. This value must be an integer.

COLONES
A required parameter. Specifies the position of the Set Cursor Column (ones unit) byte in the control sequence. This value must be an integer.

DATA
A required parameter. Specifies the ASCII control sequence for the set cursor address function. The maximum length for this value is 255 bytes. This parameter value must be coded as a hexadecimal value.

ASCII control sequence
Hexadecimal values for the ASCII control sequence for the set cursor address function.

Sound Alarm Command: You can specify a command for sounding the audible alarm on an ASCII display in the update screen table using the ALARM (alarm) tag. For most ASCII displays, this command is the single ASCII control character with a value of '07'X (BEL). If necessary, you can specify a multicharacter sequence.

The audible alarm command is sent to the display at various times by the ASCII workstation controller, depending on the types of operations being performed at the display. For example, the alarm command is sent to alert you when unrecognized keyboard data is entered, when an operator error condition occurs, or possibly when a message waiting condition becomes active.

The ALARM tag is not required in an update screen table. If this tag is deleted, the audible alarm is never sounded at that ASCII display.

Display Setup Commands: You can specify one or more setup commands using the SETUP update screen tag in your workstation customizing source. The SETUP tag allows you to specify a series of ASCII command sequences that are sent to an ASCII display to set the display to some initial state.

The control sequences you specify for this tag are sent to the display when the display is varied on. They are sent to the display again when the screen refresh function is called from the keyboard.

The data you specify for the setup command usually consists of ASCII display commands for initializing the state of the display, or for enabling or disabling display functions that you cannot otherwise set during the local setup of the ASCII display.

If a number of different ASCII display parameters must be initialized, you can set these different parameters by specifying the different ASCII command sequences for each command in consecutive hexadecimal data strings following the SETUP tag.

For example, separate commands are sent to an IBM 3151 display that allow the Reset key and the Print key to return data to the workstation controller when each key is pressed. These keys cannot be enabled during the device setup, but instead, must be enabled by sending command sequences to activate them. The command sequences to enable each key are as follows:

```
ENABLE RESET KEY:  ESC ( :  (1B 283A)
ENABLE PRINT KEY:  ESC ) :  (1B 293A)
```

Both of these commands can be specified in the setup tag as follows:

```
:SETUP DATA = '1B283A1B293A'X.
```

Note that the two different command sequences directly follow each other with no spaces in between. Additional command sequences to initialize other display conditions can be specified by simply adding the ASCII command sequences to the end of the data already specified for the setup tag.

When you specify more than one command sequence for the SETUP tag data, the ASCII workstation controller does not recognize the different command sequences specified as unique command sequences. The controller views the entire data string as a single command sequence and sends all the data to the display at the same time. Some ASCII displays support commands that require some delay before sending additional commands to the display. You should be aware that the display may not be able to handle all the individual commands specified in the SETUP data if one of the commands is one for which the display expects a delay before the next command is sent.

Set ASCII Character Set Command: You can specify the Set ASCII Character Set command using the ACS tag in the update screen table of your workstation customizing source.

This command selects the ASCII character set that is used for the lower range of an 8-bit ASCII character space. This covers the range of ASCII values for '00'X through '7F'X. For the IBM family of ASCII displays, this character space is defined as the G0 character space (see "ASCII Character Sets and Code Pages" on page 8-4 for the discussion of ASCII character spaces). When this command is specified, the ASCII workstation controller sends the command to the display to initialize the lower range of the character space when the device is varied on. This command is also sent when the screen refresh function is called from the keyboard.

Most ASCII displays already use the US ASCII character set for the lower range of an 8-bit character space, so in many cases, there is no need to specify this tag. However, you should specify this tag when the default character set in the lower range of the character space is required to be initialized for the display to function properly.

Set National Language Character Set Command: You can specify the Set National Language Character Set command using the NLCS (national language character set) tag in the update screen table section of your workstation customizing source. This command is sent to an ASCII display to select the national language character set for the upper range of an 8-bit ASCII character space ('80'X through 'FF'X.)

When you specify that text symbols are supported for a display, the ASCII workstation controller assumes that the upper range character space is to be shared between the national language character set and the special graphics character set. In this case, the national language character set is treated as the default character set for the upper range character space. When a text symbol needs to be displayed, the ASCII workstation controller replaces the national language character set in the upper range character space by sending the command to select the special graphics character set for the G1 space. After the commands to display the appropriate symbol are sent, the ASCII workstation controller sets the upper range character space back to the default by sending the command to select that national language character set.

When a display does not provide text symbol support, and the default character set used for the upper range character

space is the character set you want for the language you are using, you do not need to specify the NLCS tag after the DSCNTBL tag in your source.

Set Graphic Character Set Command: You can specify the Set Graphic Character Set command using the GCS (set graphic character set) tag in the update screen table of your workstation customizing source. This command is sent to an ASCII display to select a special graphics character set whenever text symbols are to be displayed.

The ASCII workstation controller assumes that when the special graphics character set is selected, it is loaded into the upper range of an 8-bit ASCII character space ('80'X through 'FF'X.)

Note: IBM ASCII displays refer to this character space as the G1 character space. See "ASCII Character Sets and Code Pages" on page 8-4 for more information about ASCII character spaces.

The ASCII display uses the support provided by the Set Graphic Character Set command when text symbol support is specified in the TEXTSYM parameter for the DSCNTBL tag. When this parameter is set and the GCS tag is specified, your ASCII display should support different ASCII character spaces in a manner similar to the IBM 3151 and 316x displays.

Set Cursor Display On and Set Cursor Display Off Commands: You can specify the Set Cursor On and Set Cursor Off commands by using the CSRON (set cursor on) and CSROFF (set cursor off) tags in the update screen table in your workstation customizing source. These commands are sent to an ASCII display to turn display of the cursor on or off.

Applications can define certain types of fields on the display screen where the cursor disappears when it is moved into that field. The ASCII workstation controller sends the command to turn the cursor off when such a field is entered. The command to turn it back on is sent when the cursor is moved out of the field.

When you specify the tags for these commands in an update screen table, you should be aware of the following:

- You should not specify a command to turn the cursor off without also specifying the command to turn it on. If the command to turn it off is the only one specified, the cursor disappears and does not come back after the first time the command to turn it off is sent.
- The command to turn the cursor on can actually be a command to set the cursor to a specific type, for example, a block cursor, an underscore cursor, or a blinking cursor. You should be aware that when such a command is sent to the ASCII display, it overrides any cursor style that you may have selected during the setup for the display. In this case, you should select a command for setting the cursor to a style that is not unpleasant to you when the cursor style override occurs.

Insert Cursor Command: You can specify the command to position the cursor by using the INSCSR (insert cursor) tag in the update screen table in your workstation customizing source. Some ASCII displays support the Insert Cursor command for positioning the cursor on the display separately from the command for positioning the data. For many ASCII displays, these commands are one and the same. IBM ASCII displays provide separate commands for positioning the cursor and positioning screen data. When the Insert Cursor command is sent to an ASCII display, it positions the cursor to the current screen address (as previously set by the Set Cursor Address command).

When the ASCII display positions the cursor to the same address that is set by the Set Cursor Address command, this tag does not need to be specified in an update screen table. If specified, the ASCII workstation controller uses this command for positioning the cursor. If specified for a device type other than an IBM ASCII display, the display should perform a function that is equivalent to the Insert Cursor command performed by the IBM displays.

Start and End Printer Data Bypass: You can specify the commands to start and end printer data bypass by using the STRBYP (start printer bypass) and ENDBYP (end printer bypass) tags in the update screen table in your workstation customizing source. These commands allow the ASCII workstation controller to support an auxiliary printer attached to an ASCII display. The workstation controller sends the command sequence specified on the STRBYP tag to the display before sending the data that is to be passed on to the printer. All subsequent data sent to the display should be sent to the printer until the workstation controller sends the command sequence to end printer bypass. When you specify these command sequences in your workstation customizing source, be sure you specify both the start and end command sequences; otherwise, your results could be unpredictable.

Display Attribute Commands: Applications sending data to a display attached to the AS/400 system also send attribute control characters that specify the way data for display on the screen is to be highlighted. These control characters are in the range of values from '20'X through '3F'X, and each value indicates a specific type of highlighting that would appear on a twinaxial display.

As part of its twinaxial device emulation function, the ASCII workstation controller maps these attribute control characters into ASCII command sequences that set the way data is to be highlighted on an ASCII display. This allows the workstation controller to format data on an ASCII display so that the highlighting is as close as possible to what it would look like on a twinaxial display.

You can map the attribute control characters by specifying the ATRCMD (attribute command) tag in the update screen table in your workstation customizing source. The ATRCMD tag allows you to specify the ASCII control sequence that is sent to set the highlighting for an ASCII display.

Attribute Command (ATRCMD) Tag: The ATRCMD (attribute command) tag specifies an ASCII control sequence to set a highlighting attribute. The syntax for this tag is:

```
:ATRCMD
        CTLCHAR = control character
        DATA = ASCII control sequence.
```

Figure 8-16. Syntax for the Attribute Command Tag

CTLCHAR
A required parameter. Specifies a control character for a highlighting attribute. (For a listing of the control characters for highlighting an ASCII display, see Table 8-4.)

control character
Hexadecimal value representing a highlighting attribute control character. The range of values is '20'X through '3F'X.

DATA
A required parameter. Specifies the ASCII control sequence for the function. The maximum length of this value is 255 bytes. This data must be coded as a hexadecimal value.

ASCII control sequence
Hexadecimal values for mapping a highlighting attribute control character to an ASCII control sequence.

The following table shows the highlighting codes that are associated with a twinaxial display:

Table 8-4 (Page 1 of 2). Control Characters for Highlighting an ASCII Display

Control Character	Monochrome Display	Color Display
'20'X	Normal	Green
'21'X	Reverse image	Green, Reverse image
'22'X	High Intensity	White
'23'X	Reverse image, High intensity	White, Reverse image
'24'X	Underscore	Green, Underscore
'25'X	Reverse image, Underscore	Green, Underscore, Reverse image
'26'X	High intensity, Underscore	White, Underscore
'27'X	Nondisplay	Nondisplay
'28'X	Blink	Red
'29'X	Reverse image, Blink	Red, Reverse image
'2A'X	High intensity, Blink	Red, Blink
'2B'X	Reverse image, High intensity, Blink	Red, Reverse image, Blink
'2C'X	Underscore, Blink	Red, Underscore
'2D'X	Reverse Image, Underscore, Blink	Red, Reverse Image, Underscore
'2E'X	High Intensity, Underscore, Blink	Red, Underscore, Blink
'2F'X	Nondisplay	Nondisplay
'30'X	Column Separators	Turquoise, Column separators
'31'X	Reverse image, Column separators	Turquoise, Reverse image, Column separators
'32'X	High intensity, Column separators	Yellow, Column separators
'33'X	Reverse image, High intensity, Column separators	Yellow, Reverse image, Column separators
'34'X	Underscore, Column separators	Turquoise, Underscore
'35'X	Reverse image, Underscore, Column separators	Turquoise, Reverse image, Underscore
'36'X	High Intensity, Underscore, Column separators	Yellow, Underscore
'37'X	Nondisplay	Nondisplay
'38'X	Blink, Column separators	Pink
'39'X	Reverse image, Blink, Column separators	Pink, Reverse image
'3A'X	High Intensity, Blink, Column separators	Blue
'3B'X	Reverse image, High intensity, Blink, Column separators	Blue, Reverse image
'3C'X	Underscore, Blink, Column separators	Pink, Underscore
'3D'X	Reverse image, Underscore, Blink, Column separators	Pink, Reverse image, Underscore

Table 8-4 (Page 2 of 2). Control Characters for Highlighting an ASCII Display		
Control Character	**Monochrome Display**	**Color Display**
'3E'X	High intensity, Underscore, Blink, Column separators	Blue, Underscore
'3F'X	Nondisplay	Nondisplay

The default attribute mappings that the ASCII workstation controller uses for each supported type of display map the control characters so that the appearance of data on an ASCII display is as close as possible to what it would be on a twinaxial display. These default mappings are contained in the workstation customizing source you retrieve from the system mapping tables. The workstation customizing functions allow you to change the attribute mappings, so that, if desired, the appearance of data on an ASCII display is quite different from what it would be on a twinaxial display.

Notes:

1. Different attribute control characters can be mapped into the same ASCII command sequence. This results in the same type of highlighting for different attribute control characters.
2. Attribute control character values of '27'X, '2F'X, '37'X, and '3F'X are special control characters that indicate the data following them is not displayed. The ASCII workstation controller performs special processing to ensure that the data associated with these control characters is not displayed. Mappings can be specified for these commands, but the ASCII workstation controller always forces data for these attributes not to be displayed.
3. Most ASCII displays do not support the exact same set of highlighting characteristics that may be supported on a twinaxial display. For example, most ASCII displays do not provide for a highlighting characteristic for column separators. You can substitute some other type of highlighting available on an ASCII display for showing such characteristics.
4. If you want to prevent the ASCII display from using an attribute command, the tags associated with those attributes need to be removed completely from the workstation customizing source. For example, the IBM 3101 display does not support attributes. When you retrieve the source for this display type, you can see that there are no ATRCMD tags in the source.

Set Screen Size Commands The ASCII workstation controller can send commands to an ASCII display to set the screen to different sizes, depending on the current requirements of the device emulation. You specify the commands for setting the screen to different sizes using the SCNSIZE (set screen size) tag in the update screen table in your workstation customizing source.

Set Screen Size (SCNSIZE) Tag: The SCNSIZE (set screen size) tag specifies the ASCII control sequence for setting a different screen size. You should set the screen size based on the emulation you want to use. The syntax for this tag is:

```
:SCNSIZE
      SIZE = 1920|2000|3564
      DATA = ASCII control sequence.
```

Figure 8-17. Syntax for the Set Screen Size Tag

SIZE
A required parameter. Specifies the size of the screen (in characters) for the ASCII display.

1920
The size of the screen is 1920 characters or 24 X 80 (lines X columns).

2000
The size of the screen is 2000 characters or 25 X 80.

3564
The size of the screen is 3564 characters or 27 X 132.

DATA
A required parameter. Specifies the ASCII control sequence for the screen sizing function. The maximum length of this value is 255 bytes. This data must be coded as a hexadecimal value.

ASCII control sequence
Hexadecimal values for mapping the ASCII control sequence for the screen size.

The ASCII workstation controller can set three screen sizes corresponding to the three values that you can specify for the SIZE parameter on the SCNSIZE tag. You can specify the SCNSIZE tag and its parameters up to three times in a workstation customizing source.

Notes:

1. An ASCII display does not have to support a command that sets the screen size to exactly 25x80 or exactly 27x132 for you to use these commands in the update screen table. However, for each SCNSIZE tag you specify, the display should be capable of showing at least as many rows and at least as many columns as are associated with the screen size specified. Therefore, a command that sets the screen to 26 rows by 80 columns could be used for the 25x80 screen size, or a command to set the screen size to 28 rows by 132 columns could be used for the 27x132 screen size. When this is the case, the ASCII workstation controller

only writes data on those areas of the screen associated with the sizes you select for the SCNSIZE tag.

2. The command to set the screen size to 24x80 is only sent if the twinaxial display being emulated is one that runs in 24x80 mode. In this case, the command is sent when you first vary on the display. Many ASCII displays always run in 24x80 mode. It is not necessary to specify this command in your update screen table if your display is one that always runs with this screen size.
3. If the twinaxial device emulation selected during device configuration is one that provides 132-column support, the ASCII workstation controller switches back and forth between displaying data in a 25x80 format and a 27x132 format. The ASCII workstation controller sends the two command sequences specified in the SCNSIZE tag for size values of 2000 and 3564 to set these sizes.
4. If the twinaxial device emulation selected provides 132-column support, the SCNSIZE tag should be specified in the update screen table with commands for both of the screen sizes: 25x80 and 27x132. You should not specify only the 27X132 screen size.

Extended Set Cursor Address Command: When you configure an IBM 3151 or 3162 ASCII display, you can select the twinaxial device emulation to be a twinaxial display that shows data for the screen in both the 25x80 and 27x132 formats. When you select this type of emulation, the ASCII workstation controller uses a separate command for addressing data in columns 80 through 132. This command is called the Extended Set Cursor Address command, which is specified in the update screen table by the XCSRADR (extended set cursor address) tag. This command is sent along with the command you specified for the CSRADR tag to address data on the ASCII display for this type of emulation.

Extended Set Cursor Address (XCSRADR) tag: The XCSRADR (extended set cursor address) tag specifies the ASCII control sequence for the extended set cursor address command. The syntax for this tag is:

```
:XCSRADR
      ROW = row position
      COLHIGH = column position high byte
      COLLOW = column position low byte
      DATA = ASCII control sequence.
```

Figure 8-18. Syntax for the Extended Set Cursor Address Tag

ROW
A required parameter. Specifies the position of the Extended Set Cursor Row byte in the control sequence. This value must be an integer.

COLHIGH
A required parameter. Specifies the position of the Extended Set Cursor Column (high) byte in the control sequence. This value must be an integer.

COLLOW
A required parameter. Specifies the position of the Extended Set Cursor Column (low) byte in the control sequence. This value must be an integer.

DATA
A required parameter. Specifies the ASCII control sequence for the extended set cursor address function. The maximum length of this value is 255 bytes. This parameter value must be coded as a hexadecimal value.

ASCII control sequence
Hexadecimal values for the ASCII control sequence for the extended set cursor address function.

If the XCSRADR tag and its parameters are present in an update screen table, the ASCII workstation controller uses a format that is similar to that of the Extended Set Cursor Address command for the IBM 3151 and 3162 displays. The Extended Set Cursor Address command for the 3151 and 3162 displays has the following general format:

```
ESC X  Prh Prl Pch Pcl
 1B 58  -   -   -   -

 Prh - Row Parameter, high byte
 Prl - Row Parameter, low byte
 Pch - Column Parameter, high byte
 Pcl - Column Parameter, low byte
```

The address parameter (either row or column) for the Extended Set Cursor Address command on these displays is a 2-byte parameter. The following scheme is used for the display in the specification of an address value for these 2 bytes.

Table 8-5. Address Parameter for the Extended Set Cursor Address Command

Decimal Address Value	High Byte	Low Byte
1 - 32	'20'X	'20'X - '3F'X
33 - 64	'21'X	'20'X - '3F'X
65 - 98	'22'X	'20'X - '3F'X
97 - 128	'23'X	'20'X - '3F'X
129 - 132	'24'X	'20'X - '3F'X

The 3151 and 3162 displays require certain bits within the last parameter byte of a command that contains multiple parameters to be set a certain way. This allows the display to determine which parameter is the last one in a multiple parameter command. `Pcl` is the last parameter byte in this case, and this requirement means that `Pcl` must always be set to a value that is '20'X greater than the value listed in the above table for a particular column address.

The XCSRADR tag for this particular example is specified as follows:

```
:XCSRADR
        ROW=4
        COLHIGH=5
        COLLOW=6
        DATA='1B5820202040'X
```

The default address specified in the data for this example corresponds to a screen address of row 1, column 1. The ROW, COLHIGH, and COLLOW parameters are each set to the offset value in the command for that address parameter. The XCSRADR tag does not provide a parameter to specify an index to the high byte of the ROW parameter in the command. The only rows that can ever be addressed on an ASCII display are in the range of 1 to 27. For a 2-byte addressing scheme, as described above, only one of the bytes ever changes value in the row address. The row parameter byte of the tag should point to the byte that changes value as different rows are specified. You should also note that the last byte specified for the data parameter in this example has a value of '40'X This value is specified to fulfill the previously described 3151 and 3162 requirement for marking the last parameter byte in a multiple parameter command.

The ASCII workstation controller uses this command only when the twinaxial display being emulated is one that can operate in 132-column mode.

Considerations for 132-Column Support: Some of the commands in the update screen table are used by the ASCII workstation controller only when the twinaxial device being emulated is one that supports a wide screen capability. The specific commands that are used only during this emulation are:

- Extended Set Cursor Address Command (XCSRADR tag)
- Set Screen Size to 25x80 command (SCNSIZE tag)
- Set Screen Size to 27x132 command (SCNSIZE tag)

These tags are described in more detail in previous sections in this chapter. The following information summarizes the different things you must keep in mind when customizing an ASCII display that is to be used in 132-column mode.

The only ASCII displays for which wide screen support is currently provided are the IBM 3151, with an appropriate expansion cartridge, and the IBM 3162. You can specify these commands that are specific to the 132-column support in any customized update screen table. However, the commands are used only when the configured device type for which the customized table is being used is a 3151 or a 3162 (and the twinaxial device type selected for the emulation is one that supports 132 columns).

Notes:

1. The command to set the screen size to 25x80 is sent any time the application data stream contains a command indicating that the default screen size (25x80) is to be set at the display.
2. The command to set the screen size to 27x132 is sent any time the application contains a command indicating that the display should be set to wide screen mode.
3. If the update screen table indicates that the display addresses data and the cursor using binary values (ADDRMOD=BINARY), the Set Cursor Address command is sent any time the screen position being addressed has a column value less than or equal to 80. When a position greater than column 80 is addressed, the Extended Set Cursor Address command is used.

   The Extended Set Cursor Address command specified in the update screen table in this case must conform to the format currently used by the IBM 3151 and 3162 ASCII displays. This is described in "Extended Set Cursor Address Command" on page 8-26.
4. If the update screen table indicates that the display addresses data and the cursor using decimal numeric characters (ADDRMOD=CHAR), the Set Cursor Address command is always sent to the display.

   Note that none of the currently supported ASCII displays that use decimal numeric character addressing are supported with the wide screen emulation. When you retrieve the source for an ASCII display that has numeric character addressing, the hundreds column (COLHUNDS) parameter for the CSRADR tag is always zero. To support wide screen emulation on a display that uses numeric character addressing, this parameter must be nonzero and be a valid command sequence for the hundreds unit.

Working with the EBCDIC-to-ASCII Code Mapping Table

The EBCDIC-to-ASCII mapping table handles the conversion of EBCDIC character data in the range '40'X through 'FF'X (from the ASCII workstation controller's internal screen image buffer) into ASCII character data. This table also provides the mappings for control character data in the range '00'X through '1F'X that can appear in the internal screen image buffer. To customize this table, you need to change the entries in your workstation customizing source associated with the EBCDIC-to-ASCII mapping table tag, DEBCTBL.

EBCDIC-to-ASCII Mapping Table (DEBCTBL) Tag:

The DEBCTBL (EBCDIC-to-ASCII mapping table) tag defines the EBCDIC-to-ASCII mapping table to be used for an ASCII display. The syntax for this tag is:

```
:DEBCTBL
        DATA = table data.
```

Figure 8-19. Syntax for EBCDIC-to-ASCII Mapping Table Tag

DATA

A required parameter. Specifies the EBCDIC-to-ASCII mapping table data for the ASCII display. The data are hexadecimal values used for the EBCDIC-to-ASCII translation of system data to data the ASCII display can use. The table data must be hexadecimal, and exactly 256 bytes in length.

table data

Hexadecimal values used for the EBCDIC to ASCII translation.

The EBCDIC-to-ASCII mapping table consists of 256 1-byte entries. Each entry corresponds to an EBCDIC value in the range '00'X through 'FF'X. Figure 8-20 illustrates the layout of this table in the retrieved source for an IBM 3151 display.

```
:DEBCTBL
DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '20202020202020202020202020202020'X      /* 0- */
     '20202020202020202020202A202020'X        /* 1- */
     '20202020202020202020202020202020'X      /* 2- */
     '20202020202020202020202020202020'X      /* 3- */
     '20206161616161636E5B2E3C282B21'X        /* 4- */
     '26656565656969696205D242A293B5E'X        /* 5- */
     '2D2F414141414141434E7C2C255F3E3F'X      /* 6- */
     '20454545454949494603A2340273D22'X        /* 7- */
     '20616263646566676869202020792020'X      /* 8- */
     '206A6B6C6D6E6F707172202020202020'X      /* 9- */
     '207E737475767778797A213F20202020'X      /* A- */
     '20202020202020202B20207C20202720'X      /* B- */
     '7B41424344454647484 92D6F6F6F6F6F'X      /* C- */
     '7D4A4B4C4D4E4F505152207575757579'X      /* D- */
     '5C20535455565758595A204F4F4F4F4F'X      /* E- */
     '30313233343536373839205555555520'X.     /* F- */
```

Figure 8-20. EBCDIC-to-ASCII Mapping Table Format

Notes:

1. The ASCII workstation controller does not verify whether or not the ASCII character code specified in an entry of this table corresponds to a code for a valid ASCII character. Entries from this table are accessed as they appear and are sent to the display.
2. Code values from '00'X through '1F'X are normally reserved for control characters on twinaxial displays. If any of these character codes appear in the EBCDIC screen image buffer for a display, they are mapped as specified in the EBCDIC-to-ASCII mapping table and then transmitted to the display. The EBCDIC-to-ASCII mapping table entries corresponding to these control characters are usually coded so that the characters are mapped into ASCII blanks ('20'X). One exception to this is the control character generated when you press the Dup key on the keyboard. The control character put into the screen image buffer for the Dup function is '1C'X. On twinaxial displays, this control character appears as an asterisk with a line above it. The EBCDIC-to-ASCII mapping tables usually map this control character into a substitute ASCII character (a plain asterisk).

   Some of the control characters from '00'X through '1F'X are significant to the text editing functions supported by the workstation controller, and the mappings for these control characters can be overridden if the display of text symbols is enabled (and the display supports these symbols).
3. Code values from '20'X through '3F'X are reserved for attribute control characters on twinaxial displays. These entries in the EBCDIC-to-ASCII mapping table are never used by the ASCII workstation controller.
4. A given ASCII character set often does not contain all the same characters as an EBCDIC character set. When possible, the existing EBCDIC-to-ASCII mapping tables map a character that is in an EBCDIC character set into a substitute character, when that exact same character does not exist in the corresponding ASCII character set. This is most often done for diacritic characters. For example, the character 'ä' is often mapped to the letter 'a'. If no substitute character is selected, the EBCDIC characters that are not in an ASCII character set are mapped to an ASCII space character.

When you customize the EBCDIC-to-ASCII mapping table, be sure to change any substitute mappings that are not appropriate for your display.

Using the Tags to Customize an ASCII Display Keyboard

The two mapping tables you can change to customize the keyboard for an ASCII display are:

ASCII to keyboard function mapping table,
ASCII-to-EBCDIC mapping table

The ASCII to keyboard function mapping table is set up to recognize the specific control codes and command sequences that are generated by the keyboard of an ASCII display.

The ASCII-to-EBCDIC mapping table allows the workstation controller to convert the ASCII character data entered from the keyboard to EBCDIC character data that an AS/400 application can understand.

Working with the ASCII to Keyboard Function Mapping Table

The ASCII to keyboard function mapping table handles the mapping of ASCII control characters and control character sequences to the following:

- Twinaxial keyboard function codes
- Local ASCII display functions
- Requests to set the display state for processing subsequent data received from the display

To change these mappings, you need to use the keyboard function tags that follow the DKBDTBL (ASCII to keyboard function mapping table) tag.

ASCII to Keyboard Function Mapping Table (DKBDTBL) Tag: The DKBDTBL (ASCII to keyboard function mapping table) tag defines an ASCII to keyboard function mapping table for an ASCII display. The syntax for this tag is:

```
:DKBDTBL.
```

Figure 8-21. Syntax for ASCII to Keyboard Function Mapping Table Tag

There are no keyword parameters associated with this tag. However, it is immediately followed by a number of individual keyboard function tags, which make up the table entries. See "Keyboard Function Tags" for more information about the keyboard function tags.

The size of the ASCII to keyboard function mapping table is limited. You need to keep this in mind when adding a large number of keyboard function tags that have long ASCII control sequences (hexadecimal data values).

The same keyboard function tag can be specified more than once in the same ASCII to keyboard function mapping table as long as the ASCII control sequences specified in each case are different. This allows the same keyboard function to be called by more than one ASCII keyboard mapping. When customizing an ASCII to keyboard function table, you should use caution when deciding whether to delete or omit particular keyboard function tags from a workstation customizing source.

The absence of a specific keyboard function tag after the DKBDTBL tag implies that particular keyboard function is not mapped. If the same keyboard function tag occurs multiple times (with different data values) following a DKBDTBL tag, the ASCII workstation controller uses all the occurrences of the tag in the source to map the function.

In general, keyboard function tags appearing in a workstation customizing source that has been retrieved should not be deleted. Although, it may be acceptable for functions that are not required for certain users (for example, some users will never have a need for the terminal disconnect function), deleting tags from the workstation customizing source can cause your display to be unusable or behave in an unpredictable way.

If a particular keyboard function tag does not appear in your retrieved source, you cannot use that keyboard function on your ASCII display. You can, however, try adding these functions by checking the tags that are available to add or change the function. Add the tag and the corresponding hexadecimal data to the source under the appropriate primary tag and then create and test the customizing object for that function.

Note: You cannot add a function that the workstation controller does not support for the type of device you are trying to customize.

When customizing an ASCII to keyboard function table, you should be sure that the ASCII control sequences you specify for the different keyboard function tags are unique.

Keyboard Function Tags: The tags described in this section allow you to specify the ASCII control sequence for an individual keyboard function of an ASCII display. The keyboard function tags must follow the DKBDTBL (ASCII to keyboard function mapping table) tag in the source. The same keyboard function tag can occur multiple times following the DKBDTBL tag, with different ASCII control sequences. Each occurrence results in a unique mapping entry in the final customizing object.

With one exception, all keyboard function tags have the same general syntax, as described in the following section. The exception to the general syntax is the FKEY tag, which is described separately.

The general syntax for a keyboard function tag is:

```
:xxxxx
        DATA = ASCII control sequence.
```

Figure 8-22. General Syntax for the Keyboard Function Tags

DATA

A required parameter. Specifies the ASCII control sequence for the keyboard function. The maximum length for this value is 31 bytes. This data must be coded as a hexadecimal value.

ASCII control sequence

Hexadecimal values for mapping ASCII control sequences to keyboard functions.

Note: A control sequence is only recognized if the first byte is in the range '00'X through '1F'X or is equal to '7F'X as shown in Figure 8-6 on page 8-10.

Table 8-6 lists the keyboard function tags.

Table 8-6 (Page 1 of 2). Descriptions of the Keyboard Function Tags

Keyboard Function Tag	Tag Description	Keyboard Function Tag	Tag Description
ATN	Attention	HELP	Help
BASE	Base	HEX	Hex
BOLD	Bold	HLFIDXDN	Half index down

Table 8-6 (Page 2 of 2). Descriptions of the Keyboard Function Tags

Keyboard Function Tag	Tag Description	Keyboard Function Tag	Tag Description
BOTPAG	Bottom of page	HLFIDXUP	Half index up
BSP	Backspace	HOME	Home
CARRTN	Carrier return	INSERT	Insert
CENTER	Center	LATINON	Latin language on
CLEAR	Clear	NEWLINE	New line
CLOSE	Close	NEXTSTOP	Next stop
CSRUP	Cursor up	NTLON	National language on
CSRDOWN	Cursor down	PAGDOWN	Page down
CSRLEFT	Cursor left	PAGUP	Page up
CSRRIGHT	Cursor right	PAGEND	Page end
CSRSEL	Cursor select	PA1	PA1
DISC	Terminal disconnect	PA2	PA2
DLT	Delete	PA3	PA3
DSPSYM	Display symbols	PRINT	Print
DUP	Duplicate	READSTS	Read Status
END	End	RQDSPC	Required space
ENDLINE	End of line	RQDTAB	Required tab
ENTER	Enter	RVS	Reverse
ERSINP	Erase input	SCNREFRESH	Screen refresh
ERSEOF	Erase EOF	SCNRVS	Screen reverse
ERRRESET	Error reset	SHIFTOUT	Set shift-out state
FCSRLEFT	Fast cursor left	SHIFTIN	Set shift-in state
FCSRRIGHT	Fast cursor right	STOPCODE	Stop code
FLDADV	Field advance	STRLINE	Beginning of line
FLDBSP	Field backspace	STRUS	Begin underscore
FLDEXIT	Field exit	SYSREQ	System request
FLDPLUS	Field plus	TOGIND	Toggle indicator
FLDMINUS	Field minus	TOPPAG	Top of page
FLDMRK	Field mark	TSTREQ	Test request
FWDTAB	Forward tab	WORDUS	Word underscore

Function Key (FKEY) tag: The function key tag, FKEY, specifies the ASCII control sequence to be mapped into a particular function key. The syntax for this tag is:

```
:FKEY
       KEY = F1|F2|F3|F4|F5|F6|F7|
             F8|F9|F10|F11|F12|
             F13|F14|F15|F16|F17|
             F18|F19|F20|F21|F22|
             F23|F24
       DATA = ASCII control sequence.
```

Figure 8-23. Syntax for the Function Key Tag

KEY
A required parameter. Specifies the function key to be mapped.

F1 - F24
The range of function keys you can set using this parameter.

DATA
A required parameter. Specifies the ASCII control sequence for the function key. The maximum length for this value is 31 bytes. This value must be a hexadecimal value.

ASCII control sequence
Hexadecimal values that represent the ASCII control sequence you want to map to the function key you specified for the KEY parameter.

For each keyboard function tag or function key tag (FKEY) that can be specified following a DKBDTBL tag, you can

specify data for the ASCII control sequence that you want to assign to that keyboard function.

When you specify the ASCII keystroke sequence for a given keyboard function tag, you need to be aware of the following programming considerations:

Notes:

1. You can specify a keystroke sequence that is anywhere from 1 to 31 bytes in length.
2. The control character values of '11'X and '13'X should never be specified anywhere in an ASCII control sequence. These values are reserved for the XON ('11'X) and XOFF ('13'X) control characters, which are used by ASCII devices to control the flow of data between devices. The ASCII workstation controller performs special processing when this character is received from a display. It does not map any ASCII control sequences containing this character into the specified keyboard function.
3. The first byte of a control sequence must have one of the following values: '00'X through '10'X, '12'X, '14'X to '1F'X, or '7F'X. The ASCII workstation controller will not recognize any ASCII control sequences beginning with values other than these.

   As an example, assume the cursor up (CSRUP) keyboard function tag is specified as follows:

   ```
   :CSRUP
           DATA = '221B49'X.
   ```

   The ASCII workstation controller will never map this control sequence into the cursor up keyboard function. Instead, the first character ('22'X) is treated as an ASCII character code and mapped into an EBCDIC code. The remaining bytes in the control sequence are ignored by the ASCII workstation controller.
4. If the same control sequence is specified for two or more different keyboard function tags, the ASCII workstation controller maps the specified ASCII input sequences into only one of the keyboard functions. The keyboard function into which it is mapped depends on the relative location of the keyboard function tags in the ASCII to keyboard function table.
5. If the ASCII control sequence specified for one keyboard function tag matches the beginning of a longer sequence specified in another entry, the workstation controller only recognizes the shorter sequence.

   For example, assume the following three keyboard function tags are specified:

   ```
   :CSRUP     DATA=X'1B32'
   :CSRLEFT   DATA=X'1B3203'
   :CLEAR     DATA=X'03'
   ```

   The ASCII workstation controller never maps a control sequence for the cursor left keyboard function in this example. If the control sequence assigned to cursor left ('1B3203'X) is received, the ASCII workstation controller performs the cursor up keyboard function and then subsequently performs the clear keyboard function. In this example, if no keyboard function is assigned the single value '03'X, the ASCII workstation controller still performs the cursor up function and then ignores the '03'X.

Working with the ASCII-to-EBCDIC Mapping Table

The ASCII-to-EBCDIC mapping table handles the conversion of ASCII character data received from a display to an EBCDIC character code. You can change the mapping of ASCII characters to EBCDIC characters by changing the data that follows the DASCTBL (ASCII-to-EBCDIC mapping table) tag.

ASCII-to-EBCDIC Mapping Table (DASCTBL) Tag:

The DASCTBL (ASCII-to-EBCDIC mapping table) tag defines the ASCII-to-EBCDIC mapping table for an ASCII display. This mapping table is formatted so that it begins at '20'X rather than '00'X; therefore, the size of this table is limited to 224 bytes. The syntax for this tag is:

```
:DASCTBL
        DATA = table data.
```

Figure 8-24. Syntax for the ASCII-to-EBCDIC Mapping Table Tag

DATA

A required parameter. Specifies the ASCII-to-EBCDIC mapping table data for the ASCII display. The data are hexadecimal values used for the translation of an ASCII character to an EBCDIC character. The table data must be hexadecimal, and exactly 224 bytes in length.

table data

Hexadecimal values used for the ASCII-to-EBCDIC translation.

The ASCII-to-EBCDIC mapping table consists of 224 1-byte entries. Each entry corresponds to an ASCII character code value in the range '20'X through 'FF'X. Each entry contains the EBCDIC character code to which a given ASCII character is mapped. Figure 8-25 on page 8-32 illustrates the layout of this table in your workstation customizing source.

```
:DASCTBL
DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '404F7F7B5B6C507D4D5D5C4E6B604B61'X      /* 2- */
     'F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F'X      /* 3- */
     '7CC1C2C3C4C5C6C7C8C9D1D2D3D4D5D6'X      /* 4- */
     'D7D8D9E2E3E4E5E6E7E8E94AE05A5F6D'X      /* 5- */
     '79818283848586878889919293949596'X      /* 6- */
     '979899A2A3A4A5A6A7A8A9C0BBD0A100'X      /* 7- */
     '00000000000000000000000000000000'X      /* 8- */
     '00000000000000000000000000000000'X      /* 9- */
     '00000000000000000000000000000000'X      /* A- */
     '00000000000000000000000000000000'X      /* B- */
     '00000000000000000000000000000000'X      /* C- */
     '00000000000000000000000000000000'X      /* D- */
     '00000000000000000000000000000000'X      /* E- */
     '00000000000000000000000000000000'X.     /* F- */
```

Figure 8-25. ASCII-to-EBCDIC Mapping Table Format

The EBCDIC character code values in this table should be in the range of valid EBCDIC characters ('40'X through 'FF'X) to be usable by the ASCII workstation controller. The value '00'X can be specified for ASCII code points that either have no ASCII graphic character assigned or for which there is no corresponding character in the EBCDIC character set.

For example, ASCII code points in the range from '80'X through '9F'X usually have no graphic character assigned to them. In the retrieved source, the entries in the ASCII-to-EBCDIC mapping table that correspond to these ASCII code points contain the value '00'X. When the ASCII workstation controller receives an ASCII character whose entry in the ASCII-to-EBCDIC mapping table has a value of '00'X, it is not mapped to an EBCDIC character and the received ASCII character is ignored.

You should not specify values in the range from '01'X through '3F'X for entries in the ASCII-to-EBCDIC mapping table. When a value in this range appears in a table entry and the ASCII code corresponding to this entry is received from the display, the ASCII workstation controller sends an error message to the display. This appears as a blinking 0002 on the bottom line of the display.

The ASCII workstation controller treats the ASCII code point, '7F'X as a control character. Although the entry for this ASCII code point is in the ASCII-to-EBCDIC mapping table, it is never used.

The ASCII-to-EBCDIC mapping table you retrieve when you customize an ASCII display comes from one of the AS/400 system ASCII-to-EBCDIC display mapping tables. There are a limited number of these source tables on the system, corresponding to the limited ASCII device type and language type combinations that can be configured when an ASCII display is not customized. ASCII-to-EBCDIC mapping tables that handle the mapping to every EBCDIC code page on the system do not exist. Default ASCII-to-EBCDIC mappings are provided in the retrieved source when you specify a device type and language combination that is not normally supported for ASCII displays. This default ASCII-to-EBCDIC mapping converts the US ASCII code page into EBCDIC code page 500. When you retrieve the source for such a device type and language combination, you need to be sure to change the ASCII-to-EBCDIC mapping data to suit the particular needs of your display.

Customizing a DEC VT-320 Display in VT-300 Mode

| In this example, you are customizing a DEC VT-320 ASCII
| display. The DEC VT-320 ASCII display is currently not sup-
| ported by the AS/400 system.

It is assumed that the VT-320 ASCII workstation is physically attached to the system, partly configured to work with the AS/400 system (appropriate controller and device descriptions created), and is powered on.

Step 1: Planning the Customizing

Familiarize yourself with the display by looking over the reference manual. Look at the functional capabilities and the character sets (languages) the display supports. Check the VT-220 characteristics listed in the *ASCII Work Station Reference and Example* against the VT-320 characteristics listed in the VT-320 reference manual. This comparison tells you that the VT-320 display is functionally similar to the DEC VT-220.

To begin, use the default setup for the display hardware. Because the VT-320 is an unsupported ASCII display, and its functions are similar to the VT-220, you need to change the device type to TYPE(V220) in the device description. Check the default values for the communications parameters used in the setup for the display and then specify these same values for the device description. For the VT-320, the following parameter values were specified:

Line speed	19200
Word length (Data Bits)	8
Parity	Even
Stop Bits	1

Vary on the device with the default mapping tables for a VT-220 display by specifying *NONE for the work station customizing object (WSCST) parameter in the device description.

The Sign On Screen appears and is clear. The tests provided by the DISPLAY TEST function for the VT-320 display also work correctly. This indicates that the commands for updating the screen match those supported by the display.

Using the work sheet for the ASCII update screen table, experiment with the display functions to verify the commands that are supported by the display are actually working. Be sure to look at those commands that have the most effect on the display, such as the values for the basic update screen table (DSCNTBL) tag, cursor addressing (CSRADR) tag, and

clearing the screen (CLRSCN) tag. Some degree of trial and error is required in the planning process for customizing. Use the recovery actions described in the messages in your job log to correct any errors that occur during experimentation.

The VT-320 display supports several different modes. The only major difference between the 320 mode of the VT-320 display and the VT-220 display is that the VT-220 display does not have function keys. The VT-320 function keys, F6 through F10 and F14, can be customized. (The other function keys cannot be customized because they provide local ASCII functions.)

Look at the VT-320 reference manual to find the hexadecimal code sequences sent by the F6, F7, and F8 keys and fill in the ASCII to keyboard function mapping table work sheet as shown in the following figure. This allows you to more easily make the changes to the workstation customizing source. (You can have more than one entry for each function key.)

Note: The VT-320 ASCII display provides a function that allows you to check the code generated by each key when you set the display to DISPLAY CONTROL mode.

ASCII to Keyboard Function Mapping Table (Function Keys)		
Function Key	**Key Sequence**	**Hexadecimal Data**
F1		
F2		
F3		
F4		
F5		
F6	F6	1B5B31377E
F7	F7	1B5B31387E
F8	F8	1B5B31397E
F16		

Figure 8-26. ASCII to Keyboard Function Mapping Table (Function Key)–Example Work Sheet

Step 2: Retrieving the Workstation Customizing Source

To create a workstation customizing source, you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command and specify the following:

Device type
DEVTYPE(V220)

Keyboard language type
KBDTYPE(USI)

Note: This value must match the keyboard language type you specified in the device description for the display; otherwise the display will not vary on.

Source member
SRCMBR(CSTV320)

Source file
SRCFILE(CSTSRC)

Library LIB(CSTLIB)

Text TEXT('WSCST source for VT-320 function keys')

Step 3: Changing the Source

After you retrieve the source, use the source entry utility (SEU) to change the function key definitions. The command is:

```
STRSEU SRCFILE(CSTLIB/CSTSRC) SRCMBR(CSTV320)
```

The following example source shows the code for the function keys that you want to map to your ASCII display keyboard. In the data parameter for each function key, type the ASCII hexadecimal data string for the command that the display sends when that key is pressed. (These are the hexadecimal values you wrote down on your work sheet in the planning part of the workstation customizing process.) This maps the display's code for a function key to the AS/400 function for that key.

```
   .
   .
:FKEY
      KEY = F6
      DATA = '1B5B31377E'X.
:FKEY
      KEY = F7
      DATA = '1B5B31387E'X.
:FKEY
      KEY = F8
      DATA = '1B5B31397E'X.
   .
   .
   .
```

Step 4: Creating the Workstation Customizing Object

After you change and save the source, create the workstation customizing object using the Create Work Station Customizing Object (CRTWSCST) command. Specify the following parameter values:

Workstation customizing object name
WSCST(CSTV320)

Library LIB(CSTLIB)

Source member
SRCMBR(CSTV320)

Text TEXT('Workstation Customizing Object for VT-320 function keys')

Source file
SRCFILE(CSTSRC)

Library LIB(CSTLIB)

If errors occur while the object is being created, messages are sent to the job log. Use the recovery actions provided in the help information for the message to correct any errors and try to create the customizing object again.

Step 5: Varying On the Device

Vary off the VT-320 display and change the device description specifying your customizing object, CSTV320, for the workstation customizing object (WSCST) parameter.

To activate the workstation customizing function, vary on the display again so that the customizing object is downloaded to the ASCII workstation controller. The F6, F7, and F8 function keys on the VT-320 keyboard now call the appropriate AS/400 functions for those keys.

Part 3. Printer Customization Technical Reference

Chapter 9. Customizing Printers

This chapter provides a brief overview of the workstation customizing functions for printers. It describes different methods you can use to customize the functions of ASCII printers. A description of the differences between the methods should help you choose the most appropriate method for customizing the functions of your ASCII printers.

Determining Whether an ASCII Printer Can Be Customized

To determine whether your ASCII printer should be customized, vary on the printer and print a small document or file. (Do not forget to start the printer writer and release the print job.) If your printout shows unusual characters or is missing highlighting or characters, you will probably want to customize the printer functions.

By using a process of experimentation and trial and error, you can determine whether or not an ASCII printer can be customized. Experiment with the printer after it has been connected and varied on. Use the information in this chapter to determine the method you should use to retrieve and customize the workstation customizing source for your ASCII printer. Workstation customizing functions support only the characters and functions that the device and the workstation controller, the display, or the host print transform function supports.

Understanding ASCII Printer Function Support

ASCII printers process data using an ASCII data stream. The AS/400 system produces an EBCDIC data stream. Therefore, the EBCDIC data stream from the system must be converted to an ASCII data stream to support ASCII printer functions.

Mapping tables are used to convert EBCDIC data streams into ASCII data streams. AS/400 EBCDIC data streams are converted into ASCII data streams for a printer. The way you choose to convert the EBCDIC data stream into an ASCII data stream determines which printer customization method you use. The various conversion methods are described as follows.

1. You can use the host print transform function on the AS/400 system to convert the EBCDIC data stream to an ASCII data stream for ASCII printers attached in the following ways:
   - ASCII printers attached to twinaxial displays
   - ASCII printers attached to a personal computer or Personal System/2* through 5250 emulation products (such as PC Support)
   - ASCII printers directly attached to the AS/400 system through a serial interface

   See Figure 9-1 on page 9-2 for an illustration of how ASCII printers that use the host print transform function can be attached to the AS/400 system.

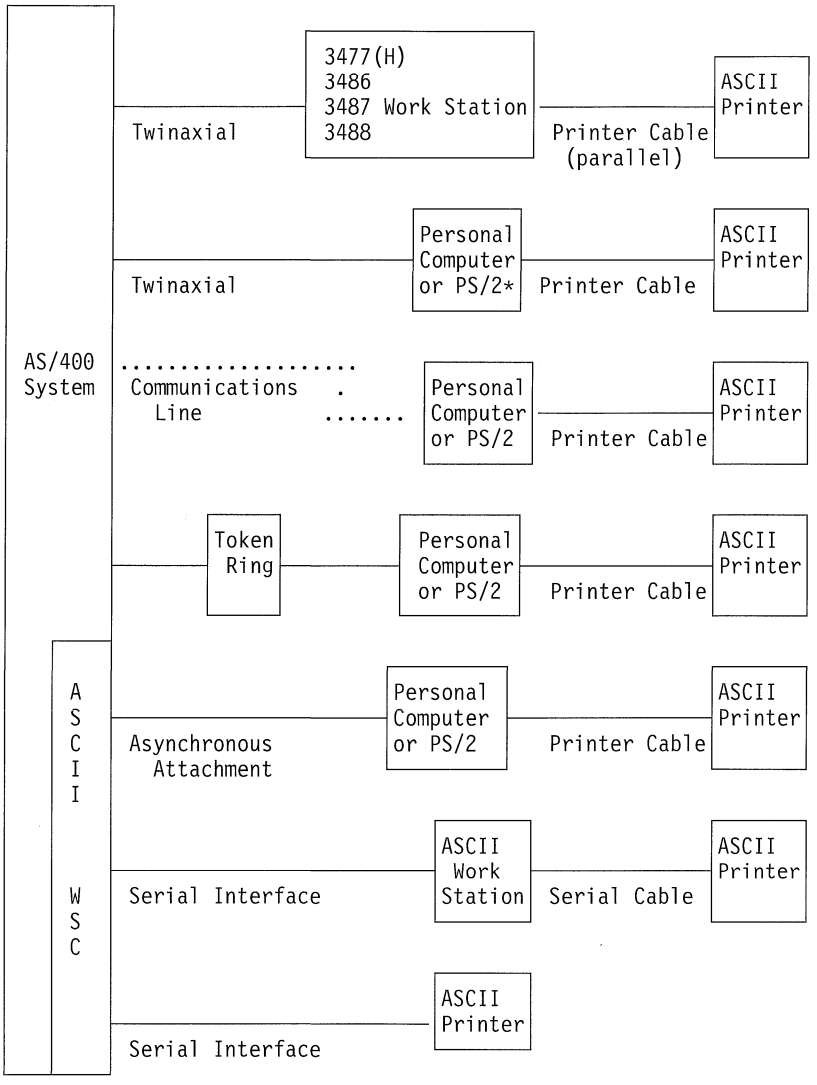


Figure 9-1. ASCII Printers That Use the Host Print Transform Function. This figure indicates the ways an ASCII printer that uses the host print transform function can be attached to the AS/400 system and can be customized. In this figure, WSC is an abbreviation for workstation controller.

When the host print transform function converts the data stream, the same ASCII data stream is always sent to a specific type of printer. The ASCII data stream sent to the printer does not depend on how the printer is attached to the AS/400 system.

The method of customizing printers that use the host print transform function provides more function, for more printer types, than other methods of customizing printers.

2. You can use an emulator on the display to convert the EBCDIC data stream to an ASCII data stream for printers attached to twinaxial displays. This conversion method may send a different ASCII data stream to a printer, depending on the type of display to which the printer is attached.

   Figure 9-2 on page 9-3 illustrates how ASCII printers that use the emulator on the display can be attached to the AS/400 system.

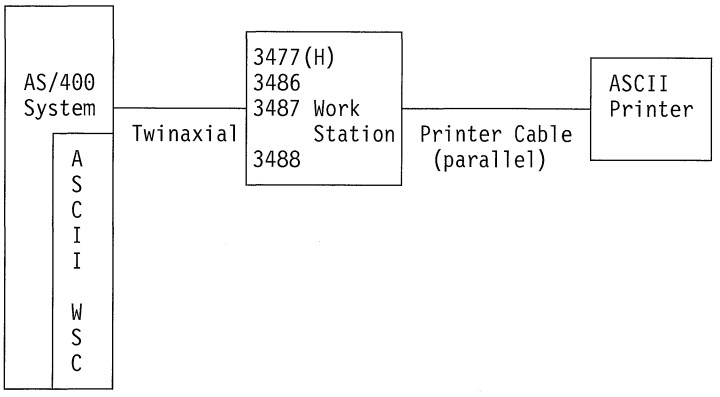


Figure 9-2. ASCII Printers That Use the Emulator on the Display. This figure indicates the way an ASCII printer that uses the emulator on the display can be attached to the AS/400 system and can be customized. In this figure, WSC is an abbreviation for workstation controller.

This conversion method maps to customizing printer functions for printers that use the emulator on the display. This customization method provides less function and consistency, for fewer printer types, than customizing printers that use host print transform function. The amount of printer function you can support with this method is determined by the type of display to which the printer is attached.

3. You can use an emulator on the workstation controller to convert the EBCDIC data stream to an ASCII data stream for directly attached printers. When an emulator on the local workstation controller converts the data stream, the same data stream is sent to a specific type of ASCII printer. It makes no difference which workstation controller attaches the printer to the AS/400 system. For example, the IBM 2637 ASCII Local Workstation Controller and IBM 6141 ASCII Local Workstation Controller send the same data stream to IBM 4019 LaserPrinters.

   Figure 9-3 on page 9-4 illustrates how ASCII printers that use the emulator on the workstation controller can be attached to the AS/400 system.

   This conversion method maps to customizing printer function for printers that use the emulator on the workstation controller. This method of customization provides less function, for fewer printer types, than customizing printers that use the host print transform function. Customizing printers that use the emulator on the workstation controller provides as much consistency, however, as customizing printers that use host print transform function.

Choosing a Method for Customizing Printer Functions

The OS/400* workstation customizing functions allow you to customize the mapping tables used to define the functions supported by the ASCII printer. You can customize ASCII printers supported by IBM and ASCII printers that are not supported by IBM.

To customize printer functions, retrieve the source for the mapping tables that define printer functions. Then add to or change the source to customize printer functions for your printer. The content of the source you retrieve is determined by the type of printer customized and the method of customizing printer functions you use. You must choose the method of customizing printer functions that most closely matches your customization needs.

The way you choose to convert the EBCDIC data stream to an ASCII data stream determines which of the printer function customization methods you use.

- For ASCII printers that use the host print transform function, you customize the printer by changing the source containing the transform table. The printer can be attached to the AS/400 system in any of the ways shown in Figure 9-1 on page 9-2.
- For ASCII printers that use the emulator on the display, you customize the printer by changing the source containing the printer definition table. The printer must be attached to a 3477 Model H, 3486, 3487, or 3488 twinaxial display station.
- For ASCII printers that use the emulator on the workstation controller, you customize the printer by changing the source containing:
  - The default printer EBCDIC-to-ASCII mapping table
  - The printer function table
  - The printer multilanguage EBCDIC-to-ASCII mapping table

  The printer must be attached directly to the AS/400 system by way of a local ASCII workstation controller or attached to an ASCII display station.

Printers that Use Host Print Transform Function

The printer data stream sent to the host print transform for ASCII printers attached to the AS/400 system is a data stream supported by twinaxial printers. A mapping table converts the twinaxial printer data stream into ASCII printer commands. The ASCII printer commands perform the same (or

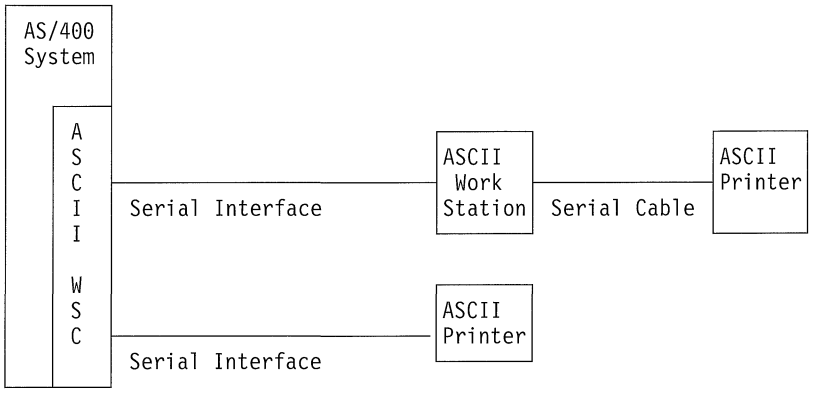


Figure 9-3. ASCII Printers That Use the Emulator on the Workstation Controller. This figure indicates the ways an ASCII printer that uses the emulator on the workstation controller can be attached to the AS/400 system and can be customized. In this figure, WSC is an abbreviation for workstation controller.

nearly the same) functions as those specified in the twinaxial printer data stream. It also converts character data specified in the application data stream as EBCDIC character code values into ASCII code values corresponding to those same characters.

To customize an ASCII printer that uses the host print transform function, you retrieve the source for the mapping table that defines printer functions. You need to know the printer manufacturer, type, and model to successfully retrieve the correct workstation customizing source for your printer.

The OS/400 workstation customizing functions provide source you can use when customizing printer functions for many printers sold by the following companies:

Epson** America Incorporated
Hewlett-Packard Company
IBM Corporation
NEC* Corporation
Okidata* Corporation
Panasonic Corporation

Note: Customizing printer functions for ASCII printers that use the host print transform function is described in detail in Chapter 10, "Customizing ASCII Printers That Use the Host Print Transform Function."

The Mapping Table (Transform Table): The host print transform function uses a specific mapping table (the transform table) to transform the printer data stream sent by the AS/400 system to a specific type of ASCII printer. Tags and commands that are part of the tag language for workstation customizing allow you to add the print characteristics that are not included in the system-provided transform table for the printer.

Customizing the Mapping Table: To begin customizing an ASCII printer, you must retrieve the source file member that contains the workstation customizing source for the characteristics of your printer. When retrieving the source, you specify the manufacturer, type, and model of your ASCII printer.

The source you retrieve contains tags only for the specified printer. You add or customize tags to change the functions supported by the specified printer. To remove a function for the specified printer, remove tags from the source. Exceptions are the EBCDIC-to-ASCII code page table and the font width values. These functions are read from default, system-provided mapping tables, even if you remove the tags from your source.

Advantages of Using the Host Print Transform Function: Choosing to customize an ASCII printer that uses the host print transform function has advantages. The host print transform function provides:

- Support for more ASCII printers than are supported by a particular device emulator.
- Support for more printer functions than are supported by a specific device emulator.
- Consistent function across all connection methods.

Limitations of Using the Emulator on the AS/400 System: When you consider customizing an ASCII printer, keep in mind the following limitation of the workstation customizing functions. Host print transform function cannot be used with double-byte character set (DBCS) printers, such as device types 5553 and 5583.

Printers that Use the Emulator on the Display

Printers that use the emulator on the display can be customized only if the twinaxial display is a 3477 Model H, 3486, 3487, or 3488. Applications send printer data streams to these twinaxial displays through workstation controllers. The printer data stream sent for ASCII printers attached to these displays is a data stream supported by twinaxial printers.

The twinaxial display uses one mapping table (the printer definition table) to convert the twinaxial printer data stream into ASCII printer commands. The ASCII printer commands perform the same (or nearly the same) functions as those specified in the twinaxial printer data stream. It also converts character data specified in the application data stream as EBCDIC character code values into ASCII code values corresponding to those same characters. Many IBM ASCII printers are currently supported by the twinaxial displays.

Note: Customizing printer function for ASCII printers that use the emulator on the display is described in detail in Chapter 11, "Customizing ASCII Printers That Use the Emulator on the Display."

The Mapping Table (Printer Definition Table): When you customize a printer definition table, it is downloaded from the workstation controller to the twinaxial display. It is then stored in the memory of the twinaxial display. When you set up the display, you specify a character set for the display. You also indicate that a specific type of ASCII printer is attached to the display.

When you vary on the display and printer, the character set and the printer definition table are used. They are used to convert the twinaxial printer data stream sent by an application to the ASCII characters and the ASCII function codes used by the printer.

The display station uses the printer definition table to convert the twinaxial printer data stream to an ASCII data stream for the printer. This table contains information on the functional capabilities of the printer. It also contains information on the command sequences that must be sent to the printer to perform specific print functions.

The printer definition table used by the twinaxial display for a particular printer depends on the type of printer and the setup in the display. The twinaxial displays that support the connection of an ASCII printer support only a subset of the available print functions. This is a limitation of the display.

Customizing the Mapping Table: The system provides you with a printer definition table for the device type you specify. It is this table that provides the source for an ASCII printer attached to a twinaxial display. Tags and commands that are part of the tag language for workstation customizing allow you to add the print characteristics that are not included in the system-provided printer definition table.

The primary tags for the workstation customizing source for an ASCII printer attached to a twinaxial display are the TKBDTBL (keyboard translation table) tag, and the PDFNTBL (ASCII printer definition table) tag. These tags denote the beginning of specific groups of tags that may occur in any order following the primary tags. Because the order and placement of the primary tags in each source file member are strictly enforced by the workstation customizing object compiler, the tag groups must remain intact. You add or customize tags to change the functions supported by the specified printer. You must set the ASCII data value to null ('0000'X) for a tag to remove a function for the specified printer.

If you remove a tag from the source, a value for the tag is read from system-provided tables when the device is varied on. Therefore, if you want to customize only printer functions, you can delete all tags for keyboard functions from your customized object.

Advantages of Using the Emulator on the Display: Choosing to customize an ASCII printer that uses the emulator on the display allows you to continue to use existing customizing source for printers. You may want to continue using existing customizing source instead of creating new customizing objects for the host print transform function.

Limitations of Using the Emulator on the Display: When you consider customizing an ASCII printer attached to a twinaxial display, keep in mind the following limitations of the workstation customizing functions.

- If you want to customize an ASCII printer attached to a 3477 display, the 3477 must be a Model H. If it is not a Model H, the 3477 display must be upgraded to a Model H. The workstation customizing functions do not support the earlier models of this display. To determine whether or not your 3477 is a Model H, see "Setting Up a 3477 Twinaxial Display" on page D-2.
- Depending on the device type and the print characteristics you need to customize, a lot of trial and error is involved in the customizing process. It could take anywhere from a few hours to a few days to perform a complete customization. The amount of time depends on the complexity of the printer (laser or line-oriented) and on the number of changes you need to make.
- The system-provided source for the printer definition table is for a line-oriented printer. If you are customizing to use a laser printer, you will need to make many changes to the retrieved source.

Printers that Use the Emulator on the Workstation Controller

The ASCII workstation controller supports ASCII printers as emulated twinaxial printers. The printer data stream sent to the workstation controller by an application is a data stream supported by twinaxial printers. Using twinaxial printer emulation, the ASCII workstation controller converts the twinaxial printer data stream into ASCII printer commands. The ASCII

printer commands perform the same (or nearly the same) functions as those specified in the twinaxial printer data stream. The workstation controller also maps character data specified in the application data stream as EBCDIC code values into their corresponding ASCII code values.

When you retrieve source to customize an ASCII printer that uses the emulator on the workstation controller, you retrieve customization source for the specified printer. To successfully retrieve the correct workstation customizing source for your printer, you must know the printer device type and the keyboard language type. The OS/400 workstation customizing functions provide mapping tables you can use as source for many IBM ASCII printers.

Note: Customizing printer function for ASCII printers that use the emulator on the workstation controller is described in detail in Chapter 12, "Customizing ASCII Printers That Use the Emulator on the Controller."

The Mapping Tables (Default EBCDIC-to-ASCII, Function, and Multilanguage EBCDIC-to-ASCII): The ASCII workstation controller uses these tables to map commands and data to an ASCII format for a specified printer:

1. Default printer EBCDIC-to-ASCII mapping table. The default EBCDIC-to-ASCII mapping table converts an EBCDIC character specified in an application data stream into a single ASCII character code value (for that same character).
2. Printer function table. The ASCII printer function table contains information used in the process of converting the twinaxial printer data stream into an ASCII data stream. This table provides the ASCII workstation controller with information on the functional capabilities of the printer. It also provides information on the command sequences that must be sent to the printer for particular print functions.

   Generally, each unique type of ASCII printer is supported by a unique printer function table. For a given type of printer, the same printer function table is used regardless of the language configured for the printer.
3. Printer multilanguage EBCDIC-to-ASCII mapping table. The multilanguage EBCDIC-to-ASCII mapping table contains EBCDIC-to-ASCII mapping information for all the EBCDIC code pages that an application sending data to an ASCII printer can select. The workstation controller uses information in this table (rather than information in the default EBCDIC-to-ASCII table) when the data stream sent to the printer contains a command to use a different code page.

These tables are downloaded to the workstation controller when the printer is varied on. The workstation controller downloads a particular multilanguage EBCDIC-to-ASCII mapping table when a printer is varied on, only if the same table has not previously been downloaded for another printer.

Customizing the Mapping Tables: When you retrieve source to customize a printer that uses the emulator on the workstation controller, you must do the following:

- Specify an ASCII printer for the device type.
- Specify the same language type as you did when you created the device description.

The source you retrieve contains three mapping tables for the specified printer. The three particular tables you see depend on the type of printer and the language configured for the printer.

After you retrieve the workstation customizing source, determine which of the mapping tables in the source you need to change. Verify that the printer is capable of printing the characters you need and performing any special functions you want to add. Then print a file from the application that is to use this printer for its output.

For example, if your organization uses the OfficeVision/400* program word processing function, you can print a single-page document. You should print the document from the printer you intend to use with the application. The document should include some of the formatting characteristics you want to add or change. Printing a document from the application allows you to determine which characters and functions are not usable from that application data stream. You can change the characters and the formatting characteristics that do not print correctly.

Use the reference manual for the printer and the tags in your retrieved source to change the existing printer mapping tables. By changing the existing printer mapping tables, you can add any new, supported print functions you need.

Advantages of Using the Emulator on the Workstation Controller: Choosing to customize an ASCII printer that uses the emulator on the workstation controller allows you to continue to use existing customizing source for printers. You may want to continue using existing customizing source instead of creating new customizing objects for the host print transform function.

Limitations of Using the Emulator on the Workstation Controller: When considering the customizing of a directly attached ASCII printer, keep in mind the following limitations of the OS/400 workstation customizing functions:

- Depending on the device type and the print characteristics you need to customize, a lot of trial and error is involved in the customizing process. It could take anywhere from a few hours to a few days to perform a complete customization. The amount of time you spend depends on whether or not the printer is a currently supported device. The amount of time also depends on the number of changes and additions you need to make.
- The workstation controller has limited storage for storing the mapping tables used for any device type. Many of the tags you can use to customize an ASCII printer require an ASCII control sequence (hexadecimal value)

for a parameter value. The maximum acceptable length of the hexadecimal value is 240 bytes. The maximum length allowed for a complete printer function table is 3584 bytes of functional data. Use a combination of the printer reference manual and trial and error to keep the ASCII control sequences to a reasonable length.

- You must specify the same language type for the Retrieve Work Station Customizing Object Source (RTVWSCST) command as you specified in the printer device description. Otherwise, the device cannot be varied on.

Chapter 10. Customizing ASCII Printers That Use the Host Print Transform Function

This chapter describes how you can use the workstation customizing functions to customize the functions of ASCII printers that use the host print transform function. Using the workstation customizing functions you can:

- Customize the functional characteristics of a supported ASCII printer
- Customize the functional characteristics and specify all necessary parameters required to support a normally unsupported ASCII printer

Use the following steps to customize the functional characteristics of an ASCII printer:

1. Prepare for the customization.
2. Retrieve the workstation customization source.
3. Change the workstation customization source.
4. Create a customizing object that contains the changed printer attributes.
5. Change the printer device description to specify the customizing object.

Preparing to Customize an ASCII Printer

You must gather source materials, complete printer set-up, and plan adequate time in your schedule to customize an ASCII printer.

Gathering Source Materials

Before you can begin customizing an ASCII printer, you must have information on the functions the ASCII printer supports. (You can only add or change printing functions that a printer supports.) You also need the hexadecimal values for these functions. The hexadecimal code information for your device is critical to the workstation customizing process. Often, the reference manual for the printer provides this information.

Completing Printer Setup

Before you begin printer customization, complete the following steps to set up both supported and unsupported printers:

- Set up all the necessary hardware to connect the printer to the AS/400 system.
- Set up any programmable features provided by the printer.

  This may involve some internal programming on the printer itself, setting device independent programming (DIP) switches, or selecting a printer to emulate. If you are using a non-IBM printer, check the reference manual to see if it emulates any IBM printers. If it does, set the emulation for the IBM printer. This may simplify the customizing process.
- Create the necessary controller and device descriptions, if they do not already exist. Some device descriptions for printers can be automatically created using automatic configuration.

After you set up and turn on the ASCII printer, use one of your usual applications to print a short test document. This is the starting point for workstation customizing.

Planning the Customization Schedule

Customizing an ASCII printer may involve a trial-and-error process. The amount of work required to customize a printer depends on:

The type of printer
Whether or not the printer is already printing
The completeness of the manual for the printer

You should plan anywhere from 1 to 5 days to complete a successful ASCII printer customization.

Customizing Unsupported ASCII Printers

To customize an unsupported ASCII printer, ask the following questions:

- Can the printer emulate a supported ASCII printer?

  If so, set it up to use the emulation. It could make your customizing easier.
- What printer functions or characteristics and national characters do I want this printer to support?

  Write these down so that you can answer the next question.
- Does the printer itself support the functions I need?

  Check the manual to determine this. If the printer cannot support the functions you need, you cannot customize the printer to provide these functions.

Retrieving the Workstation Customizing Source

To begin customizing an ASCII printer that uses the host print transform function, you must retrieve the source file member that contains the workstation customizing source for changing your printer's characteristics. When retrieving the source, specify the manufacturer, type, and model of the ASCII printer you want to customize. The source you retrieve is a copy of the transform table for the printer manufacturer, type, and model you specify.

Understanding the Transform Table

The host print transform function uses a transform table to transform the printer data stream sent by the AS/400 system to an attached ASCII printer. Some of the functional characteristics within the transform table include:

- Line spacing
- Pitch (characters per inch)
- Page size
- Highlighting characteristics (bold, underline)
- Draft, letter, or text quality printing
- Subscripting and superscripting
- Initialization and reset sequences

Choosing the Customizing Source

When you decide to customize an ASCII printer, you need to retrieve the correct source for your printer. Use the Retrieve Work Station Customizing Object Source (RTVWSCST) command to retrieve the source.

1. Type RTVWSCST and press the Enter key.
2. Specify a device type of *TRANSFORM and press the Enter key.
3. Specify the printer manufacturer, type, and model (MFRTYPMDL parameter). To see a complete list of ASCII printers supported, press F4 (Prompt) on the *Manufacturer type and model* field.

   Note: If you are customizing a laser printer that is not listed, you may want to select the *HPII value. Many laser printers have printer functions similar to those provided by the Hewlett-Packard LaserJet** Series II.
4. Specify a name for the source file member to be created for the transform table you want to retrieve. This should be a name you can easily remember.
5. Specify a library and source file name in which to store the source file member you specified in the previous step. The library you specify must exist.
6. Specify a text description for the source file member if it does not already have one. This description should be unique.

The system provides you with a transform table for a specific ASCII printer.

Changing the Source

You use printer function tags to change the ASCII control sequence for an individual printer function. You change, add, or delete the printer function tags by editing the source file member for the retrieved source using the source entry utility (SEU). See "Using Source Entry Utility to Change the Tags and Keywords" on page 3-2 for complete instructions on using SEU to change the source file member.

The source file member has the following format:

```
:WSCST DEVCLASS=TRANSFORM.
    :TRNSFRMTBL.
       .
       .
      tag or comment
       .
       .
:EWSCST.
```

Transform Table (TRNSFRMTBL Tag)

The TRNSFRMTBL (transform table) tag defines a transform table for an ASCII printer that uses the host print transform function. The syntax for this tag is:

```
:TRNSFRMTBL.
```

There are no keyword parameters associated with this tag. However, it is immediately followed by a number of separate printer function tags that make up the table entries. See "Using the Tags" for more information about the individual printer function tags.

Using the Tags

Following are descriptions of the valid printer functions and corresponding tags for a device class of TRANSFORM. These tags must follow the transform table tag (TRNSFRMTBL) in your source.

The syntax for each tag is shown, following the tag description. Some tags have only a data parameter. Other tags have more than one parameter to describe variables within the functions.

The printer function tags are divided into the following general tasks for customizing:

- Printer controls (such as sounding the bell)
- Printer data stream
- Print media size
- Highlighting
- Horizontal spacing and relative movement
- Vertical spacing and relative movement
- Indexing
- Color
- No print border
- Page length
- Paper drawer selection
- Paper orientation
- Print quality
- Fonts
- Code page support

Programming Considerations

In general, when you customize a transform table, the host print transform function uses only tags defined in your customizing object. You can delete function tags from the retrieved source when creating your customizing object. The host print transform function then assumes a null value for the ASCII control sequence for that function.

The absence of a specific printer function tag after the TRNSFRMTBL tag implies that particular printer function is not mapped and cannot be used. If the same printer function tag occurs multiple times following a TRNSFRMTBL tag, warning messages are not sent to the job log. The customizing object uses the last occurrence of the tag in the source to map the printer function.

Programming considerations for specific tags, or specific tag sets, are included in the description of those tags.

Customizing Printer Controls

You can support the following general printer control functions:

- Bell
- Carrier return
- Initialize printer
- Reset printer
- Jog output tray

You can also support the following printer controls that specify on which side of the paper printing occurs:

- Set duplex printing
- Select next side printing in duplex
- Set simplex printing
- Set tumble duplex printing

Bell (BELL Tag)

The BELL (bell) tag defines the ASCII control sequence for the bell function for an ASCII printer. The bell function sounds the printer bell or beeper briefly. The syntax for this tag is:

```
:BELL
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the bell function. This must be a hexadecimal value.

Carrier Return (CARRTN Tag)

The CARRTN (carrier return) tag defines the ASCII control sequence for the carrier return function for an ASCII printer. The CARRTN function returns the carrier to the left margin, without advancing a line. The syntax for this tag is:

```
:CARRTN
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the carrier return function. This must be a hexadecimal value.

Initialize Printer (INITPRT Tag)

The INITPRT (initialize printer) tag defines the ASCII control sequence for the initialize printer function for an ASCII printer. For example, an INITPRT tag is used by the host print transform function at the beginning of each print job that uses the SNA character set. The syntax for this tag is:

```
:INITPRT
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the initialize printer function. This must be a hexadecimal value.

Reset Printer (RESETPRT Tag)

The RESETPRT (reset printer) tag resets an ASCII printer after every print job. The syntax for this tag is:

```
:RESETPRT
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the reset printer function. This must be a hexadecimal value.

Jog Output Tray (JOGOUTTRAY Tag)

The JOGOUTTRAY (jog output tray) tag defines the ASCII control sequence for the job output tray function for an ASCII printer. The syntax for this tag is:

```
:JOGOUTTRAY
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the jog output tray function. This must be a hexadecimal value.

Set Duplex Printing (DUPXPRT Tag)

The DUPXPRT (duplex printing) tag defines the ASCII control sequence for the duplex printing function for an ASCII printer. The DUPXPRT control prints on both sides of a sheet of paper, from left to right. (Contrast the DUPXPRT control with the TUMDUPXPRT control.) The syntax for this tag is:

```
:DUPXPRT
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the duplex printing function. This must be a hexadecimal value.

Select Next Side Printing in Duplex (NXTDUPXPRT Tag)

The NXTDUPXPRT (select next side printing in duplex) tag defines the ASCII control sequence for the select next side printing in duplex function for an ASCII printer. If you do not define this tag, but the host print transform function requests it, a form feed is substituted for the NXTDUPXPRT tag. The syntax for this tag is:

```
:NXTDUPXPRT
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the select next side printing in duplex function. This must be a hexadecimal value.

Set Simplex Printing (SMPXPRT Tag)

The SMPXPRT (set simplex printing) tag defines the ASCII control sequence for the set simplex printing function for an ASCII printer. The SMPXPRT tag sets the printer to print on one side of the paper. The syntax for this tag is:

```
:SMPXPRT
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the set simplex printing function. This must be a hexadecimal value.

Set Tumble Duplex Printing (TUMDUPXPRT Tag)

The TUMDUPXPRT (set tumble duplex printing) tag defines the ASCII control sequence for the set tumble duplex printing function for an ASCII printer. The TUMDUPXPRT control prints on both sides of a sheet of paper, from top-to-bottom. (Contrast the TUMDUPXPRT control with the DUPXPRT control.) The syntax for this tag is:

```
:TUMDUPXPRT
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the set tumble duplex printing function. This must be a hexadecimal value.

Customizing Printer Data Stream (PRTDTASTRM Tag)

The PRTDTASTRM (printer data stream) tag identifies the printer data stream supported by the ASCII printer. The syntax for this tag is:

```
:PRTDTASTRM
        DATASTREAM = NULL|IBMPPDS1|IBM3812|
                     HPPCL4|IBMPPDS2|EPSON|
                     NEC|IBMGRAPHICS|HPPCL5.
```

DATASTREAM
A required parameter. Defines the type of printer data stream supported.

EPSON
The Epson printer data stream is supported.

HPPCL4
The Hewlett-Packard PCL4 printer data stream is supported.

HPPCL5
The Hewlett-Packard PCL5 printer data stream is supported.

IBMGRAPHICS
The IBM graphics printer data stream is supported.

IBMPPDS1
The IBM page printer data stream level 1 is supported.

IBMPPDS2
The IBM page printer data stream level 2 is supported.

IBM3812
The IBM 3812 printer data stream is supported.

NEC
The NEC printer data stream is supported.

NULL
The printer data stream supported is not known.

Customizing Print Media Size

The workstation customizing functions allow you to customize print media size for an ASCII printer. You can:

Customize envelope size
Customize paper size

Customizing Envelope Size

You can customize printing for various sizes of envelopes. You begin support for various envelope sizes with an ENVSIZXFM tag, and end the support with an EENVSIZXFM tag. The ENVSIZE tags fall between the start and end tags to define each different envelope size.

Using the envelope definition tags, your source could look something like this for a printer supporting two different envelope sizes:

```
:ENVSIZXFM.
     :ENVSIZE....
     :ENVSIZE....
:EENVSIZXFM.
```

Following is a description of each of the tags used to support different envelope sizes for an ASCII printer.

Set Envelope Size for Transform (ENVSIZXFM Tag): The ENVSIZXFM (set envelope size for transform) tag begins a group of different envelope size entry tags. The ENVSIZXFM tag must be followed by one or more envelope size entry tags (ENVSIZE) containing the envelope sizes. The syntax for the ENVSIZXFM tag is:

```
:ENVSIZXFM.
```

Envelope Size Entry (ENVSIZE Tag): The ENVSIZE (envelope size entry) tag defines an ASCII control sequence to set one envelope size for the transform table. One or more of these tags follow an ENVSIZXFM tag. A group of one or more of these tags must be followed by an EENVSIZXFM tag. The syntax for the ENVSIZE tag is:

```
:ENVSIZE
        ENVWTH = envelope width
        ENVLEN = envelope length
        DATA = ASCII control sequence.
```

ENVWTH
A required parameter. Specifies the envelope width in 1/1440-inch increments. This value must be an integer. The width is the left-to-right dimension when the envelope is in its normal orientation.

ENVLEN
A required parameter. Specifies the envelope length in 1/1440-inch increments. This value must be an integer. The length is the top-to-bottom dimension when the envelope is in its normal orientation.

DATA
A required parameter. Specifies the ASCII control sequence for the specified envelope size entry. This must be a hexadecimal value.

End Set Envelope Size for Transform (EENVSIZXFM Tag): The EENVSIZXFM (end set envelope size for transform) tag ends a group of envelope size entries. The syntax for this tag is:

```
:EENVSIZXFM.
```

Customizing Paper Size

You can customize printing for various sizes of paper. You begin support for various paper sizes with a PAGSIZXFM tag, and end the support with an EPAGSIZXFM tag. The PAGSIZE tags fall between the start and end tags to define each different page size.

Using the page size definition tags, your source could look something like this for a printer supporting five separate page sizes:

```
:PAGSIZXFM.
     :PAGSIZE....
     :PAGSIZE....
     :PAGSIZE....
     :PAGSIZE....
     :PAGSIZE....
:EPAGSIZXFM.
```

Following is a description of each of the tags used to support different page sizes for an ASCII printer.

Set Page Size for Transform (PAGSIZXFM Tag): The PAGSIZXFM (set page size for transform) tag defines the ASCII control sequences for a group of different page sizes. The PAGSIZXFM tag must be followed by one or more PAGSIZE (page size entry) tags containing the page sizes. The syntax for the PAGSIZXFM tag is:

```
:PAGSIZXFM.
```

Page Size Entry (PAGSIZE Tag): The PAGSIZE (page size entry) tag defines the ASCII control sequence to set one page size. One or more of these tags must follow the PAGSIZXFM (set page size for transform) tag. A group of PAGSIZE tags must be followed by an EPAGSIZXFM (end set page size for transform) tag. The syntax for this tag is:

```
:PAGSIZE
        PAGWTH = page width
        PAGLEN = page length
        DATA = ASCII control sequence.
```

PAGWTH
A required parameter. Specifies the page width in 1/1440-inch increments. This value must be an integer. The width is the left-to-right dimension of a piece of paper when it is in its normal orientation.

PAGLEN
A required parameter. Specifies the page length in 1/1440-inch increments. This value must be an integer. The length is the top-to-bottom dimension of a piece of paper when it is in its normal orientation.

DATA
A required parameter. Specifies the ASCII control sequence for selecting a specific page size entry. This must be a hexadecimal value.

End Set Page Size for Transform (EPAGSIZXFM Tag): The EPAGSIZXFM (end set page size for transform) tag ends a group of page size entries in the customizing source. The syntax for this tag is:

```
:EPAGSIZXFM.
```

Customizing Highlighting

You can support highlighting functions, such as underlining and bold printing for an ASCII printer.

Start Bold Printing (STRBOLD Tag)

The STRBOLD (start bold printing) tag defines the ASCII control sequence for starting the bold printing function for an ASCII printer. The syntax for this tag is:

```
:STRBOLD
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the start bold printing function. This must be a hexadecimal value.

End Bold Printing (ENDBOLD Tag)

The ENDBOLD (end bold printing) tag defines the ASCII control sequence for ending the bold printing function for an ASCII printer. The syntax for this tag is:

```
:ENDBOLD
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the end bold printing function. This must be a hexadecimal value.

Start Underscore Function (STRUS Tag)

The STRUS (start underscore) tag defines the ASCII control sequence for starting the underscore function for an ASCII printer. The syntax for this tag is:

```
:STRUS
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the start underscore function. This must be a hexadecimal value.

End Underscore (ENDUS Tag)

The ENDUS (end underscore) tag defines the ASCII control sequence for ending the underscore function for an ASCII printer. The syntax for this tag is:

```
:ENDUS
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the end underscore function. This must be a hexadecimal value.

Customizing Horizontal Spacing and Relative Movement

You can support the following horizontal spacing and relative movement functions:

Backspacing
Setting characters per inch
Setting characters per inch in computer output reduction (COR) mode
Adjusting horizontal relative movement
Starting and ending proportional spacing
Setting spacing

Backspace (BSP Tag)

The BSP (backspace) tag defines the ASCII control sequence for the backspace function for an ASCII printer. The syntax for this tag is:

```
:BSP
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the backspace function. This must be a hexadecimal value.

Characters Per Inch

You can set the number of characters per horizontal inch of printing. You can set the number of characters per inch in normal printing mode or in computer output reduction (COR) mode. Computer output reduction consists of:

Rotating the paper to a landscape right orientation
Reducing the character size
Changing the line density
Changing the left and top margins

Following is a description of the tags used to customize horizontal character spacing.

Customizing Characters per Inch in Normal Print Mode (CPI Tag): The CPI (set characters per inch) tag defines the ASCII control sequence for setting the number of characters per inch. The syntax for this tag is:

```
:CPI
        CPI = 5|10|12|133|15|166|
              171|20|25|27
        DATA = ASCII control sequence.
```

CPI
A required parameter. Defines the number of characters per inch.

5	5 characters per inch
10	10 characters per inch
12	12 characters per inch
133	13.3 characters per inch
15	15 characters per inch
166	16.6 characters per inch

171	17.1 characters per inch
20	20 characters per inch
25	25 characters per inch
27	27 characters per inch

DATA

A required parameter. Specifies the ASCII control sequence for setting the number of characters per inch. This must be a hexadecimal value.

Customizing Characters per Inch in COR Mode (CPICOR Tag):

The CPICOR (set characters per inch in computer output reduction (COR) mode) tag defines the control sequence for setting the number of characters per inch while in COR mode. For example, you can define all print jobs that normally use a 10-pitch type style use a particular 17-pitch type style when COR (reduction) is applied. The syntax for this tag is:

```
:CPICOR
        CPI = 10|12|15
        ASCIIFNT = font identifier number (integer)
        FNTWTH = font width (integer)
        FNTATR = font attribute (integer)
        DATA = ASCII control sequence.
```

CPI

A required parameter. This is the current number of characters per inch before the reduction is applied. For example, to define the type style to use for COR in a 10-pitch print job, specify CPI=10.

10	10 characters per inch
12	12 characters per inch
15	15 characters per inch

ASCIIFNT

A required parameter. Specifies the ASCII type style (the global font ID (FGID)) to be used when reduction is applied. This value must be an integer.

For example, the FGID for a Courier 17-pitch font (17 characters per inch) is 254. Therefore, you specify ASCIIFNT=254 for a Courier 17-pitch font. For information on global font IDs, see the *Guide to Programming for Printing*.

FNTWTH

A required parameter. Specifies the width, in 1/1440-inch increments, of the font used when reduction is applied. This value must be an integer.

For example, if you use a 17-pitch font when reduction is applied, divide the increment by the number of characters per inch (1440 divided by 17). This gives you a font width of 85 (rounded off) for the substituted font. Therefore, you specify FNTWTH=85.

FNTATR

A required parameter. Specifies the attribute value for the substituted font. This value must be an integer. In most cases, you specify a fixed-pitch type style (FNTATR=1) for the substitute font attribute value.

Integer	Attribute value
01	Fixed-pitch font
02	Proportional-spaced font
04	Typographic font

DATA

An optional parameter. Specifies the ASCII control sequence to select the substituted font. This must be a hexadecimal value. You find the ASCII control sequence in the technical reference manual for your printer under the description of selecting type styles. If there is no control sequence, ''X must be specified. When a control sequence of ''X is specified, the values specified for ASCIIFNT, FNTWTH, and FNTATR are used to select the COR font.

Horizontal Relative Movement (HORRMOV Tag)

The HORRMOV (horizontal relative movement) tag adjusts the print position backward or forward relative to the current print position. You may find the horizontal relative movement function referred to as horizontal positioning, when looking it up in your printer manual.

You can have up to two HORRMOV tags in the same source file. If you specify two HORRMOV tags in the same source file, one must have a direction of forward and the other a direction of backward. The syntax of this tag is:

```
:HORRMOV
        DIRECTION = FWD|BCK|FWDBCK
        VAROFFSET = variable offset in
                    control sequence
        VARLEN = variable length
        VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
                  CHRHEX|CHRAN
        CNVNUM = conversion ratio
                 numerator
        CNVDEN = conversion ratio
                 denominator
        DATA = ASCII control sequence.
```

DIRECTION

A required parameter. Defines the direction of the relative movement command.

FWD

Defines the horizontal relative movement forward.

BCK

Defines the horizontal relative movement backward.

FWDBCK

Defines the horizontal relative movement command forward and backward.

VAROFFSET

A required parameter. Defines the offset for the variable portion of the control sequence. This value must be an integer.

Note: This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that

the variable starts in the first byte of the control sequence.

VARLEN
A required parameter. Defines the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

VARTYPE
A required parameter. Defines the type of variable used with this category of printer function tags.

HIGHLOW
The byte order of the variable is in high-low order. The most significant byte is first.

LOWHIGH
The byte order of the variable is in low-high order. The most significant byte is last.

CHRDEC
The variable is in character decimal format with no byte order consideration. All characters are in the range from 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.

CHRHEX
The variable is in character hexadecimal format with no byte order consideration. All characters are in the range from 0 to 9, A to F.

CHRAN
The variable is in character alphanumeric format with no byte order consideration. All characters are in the range from 0 to 9, A to Z.

CNVNUM
A required parameter. Defines the numerator of the conversion ratio. This value must be an integer. The most commonly used value for CVNUM is 1.

CNVDEN
A required parameter. Defines the denominator of the conversion ratio. This value must be a nonzero integer.

CNVDEN defines the units to be used for the variable portion. For example, if movement is in 1/300ths, CNVNUM should be defined as 1 and CNVDEN should be defined as 300.

DATA
A required parameter. Specifies the ASCII control sequence for the horizontal relative movement printer function. This must be a hexadecimal value.

Start Proportional Space (STRPROP Tag)

The STRPROP (start proportional space) tag defines the ASCII control sequence for starting the proportional space mode for an ASCII printer. The syntax for this tag is:

```
:STRPROP
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the start proportional space function. This must be a hexadecimal value.

End Proportional Space (ENDPROP Tag)

The ENDPROP (end proportional space) tag defines the ASCII control sequence for ending the proportional space mode for an ASCII printer. The syntax for this tag is:

```
:ENDPROP
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the end proportional space function. This must be a hexadecimal value.

Space (SPACE Tag)

The SPACE (space) tag defines the ASCII control sequence for the space control function for an ASCII printer. The syntax for this tag is:

```
:SPACE
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the space function. This must be a hexadecimal value.

Customizing Vertical Spacing and Relative Movement

You can support the following vertical spacing and relative movement functions:

Form feed
Half line feed
Line feed
Relative vertical movement
Reverse half line feed
Reverse line feed
Vertical line spacing

Form feed (FORMFEED Tag)

The FORMFEED (form feed) tag defines the ASCII control sequence for the form feed function for an ASCII printer. The FORMFEED control advances the paper to the top of the next page. The syntax for this tag is:

```
:FORMFEED
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the form feed function. This must be a hexadecimal value.

Half Line Feed (HLFLINEFEED Tag)

The HLFLINEFEED (half line feed) tag defines the ASCII control sequence for the half line feed function for an ASCII printer. The HLFLINEFEED control advances the paper one half of a line. The syntax for this tag is:

```
:HLFLINEFEED
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the half line feed function. This must be a hexadecimal value.

Line Feed (LINEFEED Tag)

The LINEFEED (line feed) tag defines the ASCII control sequence for the line feed function for an ASCII printer. The line feed function advances the paper one line. The syntax for this tag is:

```
:LINEFEED
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the line feed function. This must be a hexadecimal value.

Vertical Relative Movement (VERRMOV Tag)

The VERRMOV (vertical relative movement) tag is used to move the print position upward or downward relative to the current print position. You may find the relative movement function referred to as vertical positioning, when looking it up in your printer manual.

You can have up to two VERRMOV tags in the same source file. If you specify two VERRMOV tags in the same source file, one must have a direction of upward and the other a direction of downward. The syntax of this command is:

```
:VERRMOV
        DIRECTION = UP|DOWN|UPDOWN
        VAROFFSET = variable offset in
                    control sequence
        VARLEN = variable length
        VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
                  CHRHEX|CHRAN
        CNVNUM = conversion ratio
                 numerator
        CNVDEN = conversion ratio
                 denominator
        DATA = ASCII control sequence.
```

DIRECTION
Defines the direction of the relative movement command. This is a required parameter.

UP
Defines the vertical relative movement upward.

DOWN
Defines the vertical relative movement downward.

UPDOWN
Defines the vertical relative movement command upward and downward.

VAROFFSET
A required parameter. Defines the offset for the variable portion of the control sequence. This value must be an integer.

Note: This offset is relative to the beginning of the control sequence. Therefore, a value of zero (0) implies that the variable starts in the first byte of the control sequence.

VARLEN
A required parameter. Defines the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

VARTYPE
A required parameter. Defines the type of variable used with this category of printer function tags.

HIGHLOW
The byte order of the variable is in high-low order. The first byte is the most significant.

LOWHIGH
The byte order of the variable is in low-high order. The last byte is the most significant.

CHRDEC
The variable is in character decimal format with no byte order consideration. All characters are in the range from 0 to 9. Many Hewlett-Packard printers use this variable type.

CHRHEX
The variable is in character hexadecimal format with no byte order consideration. All characters are in the range from 0 to 9, A to F.

CHRAN
The variable is in character alphanumeric format with no byte order consideration. All characters are in the range from 0 to 9, A to Z.

CNVNUM
A required parameter. Defines the numerator of the conversion ratio. This value must be an integer. The most common value is 1.

CNVDEN
A required parameter. Defines the denominator of the conversion ratio. This value must be a nonzero integer. Defines the units of measure to be used for the variable portion. For example, if movement is in 1/300ths, specify CNVNUM as 1 and CNVDEN as 300.

DATA
A required parameter. Specifies the ASCII control sequence for the printer function. This must be a hexadecimal value.

Reverse Half Line Feed (RVSHLFLINEFEED Tag)

The RVSHLFLINEFEED (reverse half line feed) tag defines the ASCII control sequence for the reverse half line feed function for an ASCII printer. The reverse half line feed function moves the paper back up one half line. The syntax for this tag is:

```
:RVSHLFLINEFEED
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the reverse half line feed function. This must be a hexadecimal value.

Reverse Line Feed (RVSLINEFEED Tag)

The RVSLINEFEED (reverse line feed) tag defines the ASCII control sequence for the reverse line feed function for an ASCII printer. The reverse line feed function moves the paper back up one line. The syntax for this tag is:

```
:RVSLINEFEED
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the reverse line feed function. This must be a hexadecimal value.

Vertical Line Spacing

You can set the distance between printed lines in one of two ways. You can specify the number of lines per inch or you can set line spacing to a variable value. Following is a description of the tags used to customize vertical line spacing.

Customizing Lines per Inch (LPI Tag): The LPI (set lines per inch) tag defines the control sequence for setting the number of lines per inch you want to print. Variable line spacing, if defined, overrides fixed-pitch line spacing. The syntax for this tag is:

```
:LPI
        LPI = 3|4|6|8
        DATA = ASCII control sequence.
```

LPI
A required parameter. Defines the number of lines per inch. The valid values for this parameter are 3, 4, 6, and 8. The most common vertical spacing is 6 or 8 lines per inch.

DATA
A required parameter. Specifies the ASCII control sequence for setting lines per inch. This must be a hexadecimal value.

Customizing Variable Line Spacing (VARLSPC Tag): The VARLSPC (variable line spacing) tag is used to set variable line spacing on the printer. Variable line spacing, if defined, overrides fixed-pitch line spacing. If you do not define variable line spacing, the customizing source is searched for a fixed pitch tag, for example, 8 lines per inch. The syntax for this tag is:

```
:VARLSPC
        VAROFFSET = variable offset in
                    control sequence
        VARLEN = variable length
        VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
                  CHRHEX|CHRAN
        CNVNUM = conversion ratio
                 numerator
        CNVDEN = conversion ratio
                 denominator
        DATA = ASCII control sequence.
```

VAROFFSET
A required parameter. Defines the offset for the variable portion of the control sequence. This value must be an integer.

Note: This offset is relative to the beginning of the control sequence. Therefore, a value of zero (0) implies that the variable starts in the first byte of the control sequence.

VARLEN
A required parameter. Defines the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

VARTYPE
A required parameter. Defines the type of variable used with the set page length in inches tag.

HIGHLOW
The byte order of the variable is in high-low order. The first byte is the most significant.

LOWHIGH
The byte order of the variable is in low-high order. The last byte is the most significant.

CHRDEC
The variable is in character decimal format with no byte order consideration. All characters are in the range from 0 to 9. Many Hewlett-Packard printers use this variable type.

CHRHEX
The variable is in character hexadecimal format with no byte order consideration. All characters are in the range from 0 to 9, A to F.

CHRAN
The variable is in character alphanumeric format with no byte order consideration. All characters are in the range from 0 to 9, A to Z.

CNVNUM
A required parameter. Defines the numerator of the conversion ratio. This value must be an integer. The most common value for CNVNUM is 1.

CNVDEN
A required parameter. Defines the denominator of the conversion ratio. This value must be a nonzero integer. Defines the units of measure used for the variable portion.

DATA
A required parameter. Specifies the ASCII control sequence for variable line spacing. This must be a hexadecimal value.

Customizing Indexing

You can support superscript and subscript controls for an ASCII printer. On some printers, these controls move the print position up or down 1/2 line spaces. Other printers create superscripts and subscripts by compressing the height of the characters.

Start Subscript Function (STRSUBS Tag)

The STRSUBS (start subscript) tag defines the ASCII control sequence for starting the subscript function for an ASCII printer. The syntax for this tag is:

```
:STRSUBS
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the start subscript function. This must be a hexadecimal value.

End Subscript Function (ENDSUBS Tag)

The ENDSUBS (end subscript) tag defines the ASCII control sequence for ending the subscript function for an ASCII printer. The syntax for this tag is:

```
:ENDSUBS
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the end subscript function. This must be a hexadecimal value.

Start Superscript Function (STRSUPS Tag)

The STRSUPS (start superscript) tag defines the ASCII control sequence for starting the superscript function for an ASCII printer. The syntax for this tag is:

```
:STRSUPS
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the start superscript function. This must be a hexadecimal value.

End Superscript Function (ENDSUPS Tag)

The ENDSUPS (end superscript) tag defines the ASCII control sequence for ending the superscript function for an ASCII printer. The syntax for this tag is:

```
:ENDSUPS
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the end superscript function. This must be a hexadecimal value.

Customizing Color Selection (FOREGRND Tag)

The FOREGRND (foreground color) tag defines the ASCII control sequence for setting different colors when the printer supports color. The syntax for this tag is:

```
:FOREGRND
        COLOR = BLACK|BLUE|RED|PINK|GREEN|
                CYAN|YELLOW|WHITE|DRKBLUE|
                ORANGE|PURPLE|DRKGREEN|
                TURQ|MUSTARD|GREY|BROWN
        DATA = ASCII control sequence.
```

COLOR
A required parameter. Defines the color to be used by the printer when it supports different colors for printing.

BLACK	DRKBLUE (dark blue)	MUSTARD	RED
BLUE	DRKGREEN (dark green)	ORANGE	TURQ (turquoise)
BROWN	GREEN	PINK	WHITE
CYAN	GREY	PURPLE	YELLOW

DATA
A required parameter. Specifies the ASCII control sequence for setting the foreground color. This must be a hexadecimal value.

Customizing the No-Print Border (NOPRTBDR Tag)

The NOPRTBDR (no-print border) tag allows you to define the physical area on which printing cannot be done. The no-print border size is based on the physical restrictions of your printer. You may find the no-print border referred to as the "unprintable area," when looking it up in your printer manual.

If the no-print border is defined, it is included in the top, bottom, left, and right margins used for an AS/400 print job. For example, if you specify a value of 1/2 inch for the top, portrait no print border, and the AS/400 document sets a top margin of 1 inch, the host print transform only advances your ASCII printer position 1/2 inch. The effect is that the printed document has a 1-inch top margin.

The syntax for this tag is:

```
:NOPRTBDR
        OPTION = TOP|LEFT|RIGHT|BOTTOM
        ORIENT = PORTRAIT|LANDSCAPE
        DATA = no print border size in
               1440ths of an inch (integer).
```

OPTION
A required parameter. Defines the border type.

TOP
Sets no-print border at the top of the paper.

LEFT
Sets no-print border on the left side of the paper.

RIGHT
Sets no-print border on the right side of the paper.

BOTTOM
Sets no-print border at the bottom of the paper.

ORIENT
A required parameter. Defines the type of paper orientation. Some printers have different unprintable areas based on the orientation of the printed output.

PORTRAIT
The no-print border specified is for portrait-oriented print.

LANDSCAPE
The no-print border specified is for landscape-oriented print.

DATA
A required parameter. Specifies the size of the no print in 1/1440-inch increments. This value must be an integer.

Customizing Page Length

You can set page length in lines or in inches. Following is a description of the tags used to customize page length.

Set Page Length in Inches (PAGLENI Tag)

The PAGLENI (set page length in inches) tag sets the page length in terms of inches. The syntax for this tag is:

```
:PAGLENI
        VAROFFSET = variable offset in
                    control sequence
        VARLEN = variable length
        VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
                  CHRHEX|CHRAN
        CNVNUM = conversion ratio
                 numerator
        CNVDEN = conversion ratio
                 denominator
        DATA = ASCII control sequence.
```

VAROFFSET
A required parameter. Defines the offset for the variable portion of the control sequence. This value must be an integer.

Note: This offset is relative to the beginning of the control sequence. Therefore, a value of zero (0) implies that the variable starts in the first byte of the control sequence.

VARLEN
A required parameter. Defines the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

VARTYPE
A required parameter. Defines the type of variable used with the set page length in inches tag.

HIGHLOW
The byte order of the variable is in high-low order. The first byte is the most significant.

LOWHIGH
The byte order of the variable is in low-high order. The last byte is the most significant.

CHRDEC
The variable is in character decimal format with no byte order consideration. All characters are in the range from 0 to 9. Many Hewlett-Packard printers use this variable type.

CHRHEX
The variable is in character hexadecimal format with no byte order consideration. All characters are in the range from 0 to 9, A to F.

CHRAN
The variable is in character alphanumeric format with no byte order consideration. All characters are in the range from 0 to 9, A to Z.

CNVNUM
A required parameter. Defines the numerator of the conversion ratio. This value must be an integer. The most common value for CNVNUM is 1.

CNVDEN
A required parameter. Defines the denominator of the conversion ratio. This value must be a nonzero integer. The most common value for CNVDEN is 1.

DATA
A required parameter. Specifies the ASCII control sequence for setting the page length in inches. This must be a hexadecimal value.

Set Page Length in Lines (PAGLENL Tag)

The PAGLENL (set page length in lines) tag sets the page length in terms of the number of lines. The number of lines is carried as a variable in the control sequence for page length. The syntax for this tag is:

```
:PAGLENL
        VAROFFSET = variable offset in
                    control sequence
        VARLEN = variable length
        VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
                  CHRHEX|CHRAN
        DATA = ASCII control sequence.
```

VAROFFSET
A required parameter. Defines the offset for the variable portion of the control sequence. This value must be an integer.

Note: This offset is relative to the beginning of the control sequence. Therefore, a value of zero (0) implies that the variable starts in the first byte of the control sequence.

VARLEN
A required parameter. Defines the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

VARTYPE
A required parameter. This attribute defines the type of variable. Possible values are :

HIGHLOW
The byte order of the variable is in high-low order. The first byte is the most significant.

LOWHIGH
The byte order of the variable is in low-high order. The last byte is the most significant.

CHRDEC
The variable is in character decimal format with no byte order consideration. All characters are in the range 0 to 9. Many Hewlett-Packard printers use this variable type.

CHRHEX
The variable is in character hexadecimal format with no byte order consideration. All characters are in the range 0 to 9, A to F.

CHRAN
The variable is in character alphanumeric format with no byte order consideration. All characters are in the range 0 to 9, A to Z.

DATA
A required parameter. Specifies the ASCII control sequence for setting the page length in lines. This must be a hexadecimal value.

Customizing Paper Drawer Selection (DWRSLT Tag)

The DWRSLT (drawer selection) tag defines the control sequences available for paper drawer selection. The syntax for this tag is:

```
:DWRSLT
        DRAWER = PAPER|ENVELOPE|
                 DRAWER1|DRAWER2
        DATA = ASCII control sequence.
```

DRAWER
A required parameter. Defines the drawer selection.

PAPER
The drawer selection is manual paper feed.

ENVELOPE
The drawer selection is envelope.

DRAWER1
The drawer selection is drawer 1.

DRAWER2
The drawer selection is drawer 2.

DATA
A required parameter. Specifies the ASCII control sequence for paper drawer selection. This must be a hexadecimal value.

Customizing Paper Orientation (PRTORIENT Tag)

The PRTORIENT (paper orientation) tag defines the control sequence for setting different paper orientations. The syntax for this tag is:

```
:PRTORIENT
        ORIENT = PORTRAIT|LANDSCAPE|
                 RTT180|RTT270
        DATA = ASCII control sequence.
```

ORIENT
A required parameter. The orientation in which a job prints.

PORTRAIT
The print job prints in an orientation rotated 0 degrees.

LANDSCAPE
The print job prints in an orientation rotated 90 degrees.

RTT180
The print job prints in an orientation rotated 180 degrees.

RTT270
The print job prints in an orientation rotated 270 degrees.

DATA
A required parameter. Specifies the ASCII control sequence for setting the paper orientation for the printer. This must be a hexadecimal value.

Customizing Print Quality (PRTQLTY Tag)

The PRTQLTY (print quality) tag defines the control sequence for selecting the level of print quality (such as draft or letter) an ASCII printer provides. The syntax for this tag is:

```
:PRTQLTY
        QLTYTYPE = DRAFT|LETTER|TEXT
        DATA = ASCII control sequence.
```

QLTYTYPE
A required parameter. Defines the quality of print.

DRAFT
The print quality is draft quality. This is equivalent to the *DRAFT type used in the OS/400 printer file commands.

LETTER
The print quality is letter quality. This is equivalent to the *NLQ type used in the OS/400 printer file commands.

TEXT
The print quality is text quality. This is equivalent to the *STD type used in the OS/400 printer file commands.

DATA
A required parameter. Specifies the ASCII control sequence for the function. This is a hexadecimal value.

Customizing Fonts

You can change the appearance of characters printed by an ASCII printer by selecting and customizing fonts. You can choose a range of fonts to be used by the printer. You can also customize individual fonts that are used by the printer.

When customizing fonts, consider the following:

- Individual font definitions are read before group font definitions.
- The first group definition that satisfies the font request is used.
- If you do not specify font-width data, a system-supplied font width is used.
- You do not have to define font widths for fixed-pitch type styles unless the font identifier you specify is outside the normal range of font identifiers for a particular pitch.

  For example, the normal range of font identifiers is from 1 to 65 for a 10-pitch type style. If you specify a font identifier greater than 65 for a 10-pitch type style, you must also define a font width for that type style.
- If you specify font-width data, the data values must be either 256 or 512 bytes in length. A maximum width of 255/1440ths of an inch per character can fit in a single byte. If the font widths of all characters can fit in a single byte, the data values are 256 bytes in length. If the font width of any one character exceeds 255/1440ths of an inch in width, use a 2-byte value for each character. The data values are then 512 bytes in length.
- To ensure your print jobs look similar to print jobs generated from the host, you must specify font-width data equal to, or less than, the font-width data used by the host application.

  For example, if you use the justify function with a proportional-spaced font in an OfficeVision/400 document, the OfficeVision/400 program uses its own internal font widths. If the OfficeVision/400 font widths are narrower, on average, than the font width defined in the workstation customizing object, your document may not justify properly.
- For fixed-pitch fonts, character per inch (CPI) definitions are read last.
- Type style requests are automatically generated for IBM page printer data stream level 2, Hewlett-Packard PCL4, and Hewlett-Packard PCL5 tables.
- For proportional and typographic fonts, start proportional mode tags and end proportional mode tags are read last.

Following is a description of the tags used to customize fonts.

Font Groups

You can choose font groups used by an ASCII printer. Using the font group tags, your source could look something like this for a printer supporting five separate font groups:

```
:FNTGRP.
     :FNTGRPE....
     :FNTGRPE....
     :FNTGRPE....
     :FNTGRPE....
     :FNTGRPE....
:EFNTGRP.
```

Following is a description of each of the tags used to support font groups for an ASCII printer.

Font Group (FNTGRP Tag): The FNTGRP (font group) tag defines the beginning of one or more font group entry (FNTGRPE) tags. It must be followed by one or more FNTGRPE tags. The syntax for the FNTGRP tag is:

```
:FNTGRP.
```

Font Group Entry (FNTGRPE Tag): The FNTGRPE (font group entry) tag defines a range of fonts. The FNTGRPE tag must follow either a FNTGRP (font group) tag or another FNTGRPE tag in your source. A group of one or more of these tags must be followed by an EFNTGRP tag. The syntax for this tag is:

```
:FNTGRPE
         MINFID = font identifier (integer)
         MAXFID = font identifier (integer)
         FNTSTR = start font ASCII control
                  sequence
         FNTEND = end font ASCII control
                  sequence
         FNTWTH = character width data of
                  the font.
```

MINFID
A required parameter. Defines the smallest font identifier in a group. This value must be an integer.

MAXFID
A required parameter. Defines the largest font identifier in a group. This value must be an integer.

FNTSTR
A required parameter. Defines the ASCII control sequence to start a font request.

FNTEND
An optional parameter. Defines the ASCII control sequence to end a font request. If no ASCII control sequence is defined to end a font request, ''X must be entered.

FNTWTH
An optional parameter. Specifies the individual character widths in 1/1440-inch increments for the font group range. If specified, this must be either a 256-byte or 512-byte hexadecimal value. If no font-width data is defined, ''X must be entered.

Notes:

1. The ranges specified for MINFID and MAXFID are not validated for different FNTGRPE tags to ensure they do not overlap.
2. Validation is not performed to ensure individual fonts (INDFNTE) are not defined within a range of fonts specified in a FNTGRPE tag.
3. For information on font identifiers, see the *Guide to Programming for Printing*.

End Font Group (EFNTGRP Tag): The EFNTGRP (end font group) tag ends the font group definition for a transform table. The syntax for this tag is:

```
:EFNTGRP.
```

Individual Fonts

You can customize individual fonts used by an ASCII printer. Using the individual font tags, your source could look something like:

```
:INDFNT.
     :INDFNTE....
     :INDFNTE....
     :INDFNTE....
     :INDFNTE....
     :INDFNTE....
:EINDFNT.
```

Following is a description of each of the tags used to support individual fonts for an ASCII printer.

Individual Font (INDFNT Tag): The INDFNT (individual font) tag defines the beginning of one or more INDFNTE tags. The INDFNT tag must be followed by one or more individual font entry (INDFNTE) tags. The syntax for this tag is:

```
:INDFNT.
```

Individual Font Entry (INDFNTE Tag): The INDFNTE (individual font entry) tag defines an individual font. One or more individual font entry tags can be defined, but must follow the INDFNT tag. The INDFNTE tag or tags must be followed by the EINDFNT tag. The syntax for this tag is:

```
:INDFNTE
         FID = font identifier (integer)
         POINTSIZE = font point size
                     (integer)
         FNTSTR = start font ASCII control
                  sequence
         FNTEND = end font ASCII control
                  sequence
         FNTWTH = character width data of
                  the font.
```

FID
A required parameter. Identifies the individual font. This value must be an integer. For information on font identifiers, see the *Guide to Programming for Printing*.

POINTSIZE
Specifies the point size of the individual font in 1/72-inch increments. This value must be an integer. If the font point size is not required (for example, if you are defining a fixed-pitch font), 0 (zero) must be entered.

FNTSTR
A required parameter. The ASCII control sequence for the start of the font.

FNTEND
The ASCII control sequence for the end of the individual font. If there is no end font ASCII control sequence, ''X must be specified.

FNTWTH
Specifies the individual character widths in 1/1440-inch increments for the individual font. This must be specified as a 256-byte or 512-byte hexadecimal value. If there is no font-width data, ''X must be specified.

Notes:

1. The FID and POINTSIZE parameter pairs are not validated to ensure they are not duplicated.
2. The FID parameter is not checked to see if it is already defined in a font group (FNTGRPE) tag.

End Individual Font (EINDFNT Tag): The EINDFNT (end individual font) tag defines the end of one or more INDFNTE tags. The syntax for this tag is:

```
:EINDFNT.
```

Customizing Code Page Support

The workstation customizing functions allow you to customize code page support for an ASCII printer. You can:

Customize EBCDIC-to-ASCII code page mapping
Support additional ASCII code pages
Override the default ASCII code page

When you specify code page information in a customizing object, the customizing object information takes precedence over information in system-supplied code page tables. If you remove code page tags from a customizing object, code page information continues to be read from the system-supplied tables.

Customizing EBCDIC-to-ASCII Code Page Mapping

EBCDIC-to-ASCII mapping tables convert an EBCDIC character specified in an application data stream into an ASCII character code value (for that same character). The EBCDIC-to-ASCII mapping table used for a given ASCII printer depends on the manufacturer, type, and model configured for the printer. Different types of ASCII printers support different ASCII code pages. The code page in use at any time is determined by a command to select the ASCII code page.

You can customize the EBCDIC-to-ASCII mapping that should be used for an ASCII printer. You can customize the mapping to replace the entire ASCII code page mapped to an EBCDIC code page. For example, the U.S. English EBCDIC code page 037 normally maps to the ASCII code page 437. You could use these tags to map EBCDIC code page 037 to an ASCII code page other than 437 (such as ASCII code page 850).

The following code page mapping tables are available on the AS/400 system for use by the host print transform function. The default mapping table is the mapping table used if you do not override the default table using a customizing object.

EBCDIC Code Page Used	Default ASCII Code Page	Alternate ASCII Code Page	Alternate ASCII Code Page	Alternate ASCII Code Page	Alternate ASCII Code Page	Alternate ASCII Code Page	Alternate ASCII Code Page
037	437	850	860	863	1051		
273	850	437	1051				
277	865	850	1051				
278	865	437	850	1051			
280	850	437	1051				
282	850	860	1051				
284	850	437	1051				
285	850	437	1051				
297	850	437	1051				
420	864	1051					
423	851	1051					
424	856	862	1051				
500	850	437	860	861	863	865	1051
838	874	1051					
870	852	1051					
871	850	437	861	1051			
875	869	1051					
880	850	1051					
905	857	1051					
1025	850	1051					
1026	857	1051					

In addition, you can map the EBCDIC symbols code page (code page 259) to an ASCII code page. If you configure a printer to support the ASCII symbols code page (code page 899), the complete mapping from EBCDIC code page 259 to ASCII code page 899 is used. Otherwise, a partial mapping is done from EBCDIC code page 259 to the currently defined ASCII code page. The partial mapping can be done from EBCDIC code page 259 to any of the following ASCII code pages:

437 862
850 863
851 864
852 865
856 869
857 874
860 899
861 1051

For more information on specific code pages, see the *National Language Support Planning Guide*. For information on generating your own code page mapping table in QUSRSYS, see the Create Table (CRTTBL) command in the *CL Reference*.

You can also customize EBCDIC-to-ASCII mapping for an individual code point within an EBCDIC code page. When you customize the mapping for an individual code point, though, you must indicate data values for every code point within the code page. You cannot just indicate the data value for the code point you want customized.

You begin support for various mapping tables with an EBCASCTBL tag, and end the support with an EEBCASCTBL tag. The EBCASCTBLE tags fall between the start and end tags to define the EBCDIC-to-ASCII mapping using the DATA parameter.

Using the EBCDIC-to-ASCII definition tags, your source could look something like this:

```
:EBCASCTBL.
     :EBCASCTBLE...
                .
                .
     :EBCASCTBLE...
                .
                .
:EEBCASCTBL.
```

Following is a description of each of the tags used to support EBCDIC-to-ASCII mapping for an ASCII printer.

EBCDIC-to-ASCII Mapping Table (EBCASCTBL Tag): You use the EBCASCTBL tag to begin a group of one or more EBCASCTBLE tags. This tag must be followed by one or more mapping table entry (EBCASCTBLE) tags. There are no parameters on this tag. The syntax for this tag is:

```
:EBCASCTBL.
```

EBCDIC-to-ASCII Mapping Table Entry (EBCASCTBLE Tag): You specify the EBCDIC-to-ASCII conversion table for an ASCII printer using the DATA parameter of the EBCASCTBLE (EBCDIC-to-ASCII mapping table entry) tag. The EBCASCTBLE tag must follow an EBCASCTBL tag. The syntax for this tag is:

```
:EBCASCTBLE
        EBCDICCP = EBCDIC code page
                   identifier (integer)
        ASCIICP = ASCII code page
                  identifier (integer)
        DATA = EBCDIC to ASCII table data.
```

EBCDICCP

A required parameter. Specifies the EBCDIC code page identifier.

code page ID

A registered identifier used to specify a particular assignment of code points to graphic characters.

ASCIICP

A required parameter. Specifies the ASCII code page identifier.

code page ID

A registered identifier used to specify a particular assignment of code points to graphic characters.

DATA

A required parameter. Specifies the hexadecimal data that is used to map EBCDIC codes from the AS/400 system to the ASCII codes needed by the ASCII printer.

When you customize to replace the entire ASCII code page mapped to an EBCDIC code page, you set the DATA value to ''X (null). For example, EBCDIC code page 277 maps to ASCII code page 865 by default. If you prefer to map EBCDIC code page 277 to ASCII code page 1051 (Roman 8), enter the following in your customizing source:

```
:EBCASCTBLE
          EBCDICCP = 277
          ASCIICP = 1051
          DATA = ''X.
```

When you customize for an individual code point within an ASCII code page, you must indicate data values for every code point within the code page. You cannot just indicate the data value for the code point you want customized. The data must be hexadecimal, and exactly 192 bytes in length. For example, EBCDIC code page 277 maps to ASCII code page 865 by default. You like this EBCDIC-to-ASCII code page mapping, except for the mapping of one character. To change the code page mapping for that character, you must indicate data values for every code point within the code page, not just for the code point you want changed.

End EBCDIC-to-ASCII Mapping Table (EEBCASCTBL Tag): You use the EEBCASCTBL (end EBCDIC-to-ASCII mapping table) tag to end the EBCDIC-to-ASCII mapping customization. The syntax for this tag is:

```
:EEBCASCTBL.
```

Supporting Additional ASCII Code Pages

Support for each ASCII code page begins with an ASCCPINFO tag and ends with an EASCCPINFO tag. The following tags fall between the start and end tags to fully define the support of additional code pages:

CODEPAGE
ASCIICTL

Using the ASCII code page support tags, your source could look something like this for a printer supporting five separate code pages:

```
:ASCCPINFO.
     :CODEPAGE....
        :ASCIICTL....
        :ASCIICTL....
     :CODEPAGE....
        :ASCIICTL....
     :CODEPAGE....
     :CODEPAGE....
     :CODEPAGE....
        :ASCIICTL....
        :ASCIICTL....
        :ASCIICTL....
:EASCCPINFO.
```

Following is a description of each of the tags used to support code pages for an ASCII printer.

ASCII Code Page Information (ASCCPINFO Tag): The ASCCPINFO (ASCII code page information) tag defines the beginning of a group of different ASCII code points within a specified ASCII code page. The ASCCPINFO tag must immediately precede a CODEPAGE tag in your source. The syntax for this tag is:

```
:ASCCPINFO.
```

Set Code Page (CODEPAGE Tag): The CODEPAGE (set code page) tag defines the ASCII control sequence for selecting the ASCII code page. The CODEPAGE tag must follow the ASCCPINFO tag, another CODEPAGE tag, or an ASCIICTL tag in your source. You can use more than one CODEPAGE tag, but these tags must come between the ASCCPINFO and EASCCPINFO tags. The syntax for this tag is:

```
:CODEPAGE
        CODEPAGE = ASCII code page (integer)
        DATA = ASCII control sequence.
```

CODEPAGE
A required parameter. Specifies the identifier (ID) of the ASCII code page to be selected. Any integer value is accepted.

DATA
A required parameter. Specifies the ASCII control sequence for selecting the ASCII code page on the ASCII printer. This must be a hexadecimal value. The ASCII control sequence for selecting an ASCII code page is sent to the printer when that code page is requested in the data stream.

ASCII Control Code Mapping (ASCIICTL Tag): The ASCIICTL (ASCII control code mapping) tag defines the ASCII control sequence for an ASCII control code. The ASCIICTL tag must immediately follow the CODEPAGE tag or another ASCIICTL tag in your source. You can use more than one ASCIICTL tag, but these tags must follow a CODEPAGE tag. The syntax for this tag is:

```
:ASCIICTL
        ASCII = control code
        DATA = ASCII control sequence.
```

ASCII
A required parameter. Specifies an ASCII control code. This must be a hexadecimal value from '01'X to 'FF'X.

DATA
A required parameter. Specifies the ASCII control sequence that will be substituted for the ASCII control code you want to map. This must be a hexadecimal value.

Specifying the ASCIICTL tag allows you to convert a single-byte control code to a multiple-byte control sequence. For example, you could use this code to have the printer print characters that are not normally in the range of printable characters (such as a heart or a diamond shape). You could also use this tag to allow the printer to change to a different code page to print a single character. Then you can have the printer change back to the original code page.

End ASCII Code Page Information (EASCCPINFO Tag): The EASCCPINFO (end ASCII code page information) tag ends a group of CODEPAGE and ASCIICTL tags defining ASCII code page support for an ASCII printer. This tag must come after an ASCCPINFO tag, and immediately following either a CODEPAGE or an ASCIICTL tag in your source. The syntax for this tag is:

```
:EASCCPINFO.
```

Overriding the Default ASCII Code Page (DFTASCCP Tag)

The DFTASCCP (default ASCII code page) tag allows you to override the default ASCII code page for all EBCDIC code page values. You might use this tag to replace an IBM-defined code page with the code page defined by another manufacturer for a specific printer. For example, Hewlett-Packard printers have code pages defined that differ from the code pages defined by IBM. With the DFTASCCP tag, you can support a Hewlett-Packard defined code page. The DFTASCCP tag is used only when searching the system-supplied tables. The syntax for this tag is:

```
:DFTASCCP
        ASCIICP = default ASCII code page
                  identifier (integer).
```

ASCIICP
A required parameter. Specifies the ASCII code page identifier that should be used in place of the recommended ASCII code page for all EBCDIC code page values. The identifier must be an integer. The code page identifier is not validated to ensure you have entered a valid code page.

Creating the Workstation Customizing Object

After you complete your changes to the workstation customizing source table in the source file member, use the Create Workstation Customizing Object (CRTWSCST) command. The CRTWSCST command creates a customizing object. For complete instructions on compiling and creating the workstation customizing object, see "Compiling and Creating the Workstation Customizing Object" on page 4-1.

Specifying the Workstation Customizing Object

After you create the workstation customizing object, you need to specify the customizing object in the device description for a printer. Vary off the ASCII printer you want to customize and change the device description. Use the Change Device Description (Printer) (CHGDEVPRT) command to specify the customizing object in the device description for a printer.

1. Enable the host print transform function by specifying *Yes in *Host print transform* prompt of the device description.
2. Specify *WSCST for the *Manufacturer type and model* prompt in the MFRTYPMDL parameter.
3. Specify the name of the object for the *Workstation customizing object* prompt. Specifying the name of the object in the device description links the object to the device.

For complete instructions on changing the device description, see Chapter 5, "Testing the Customizing Object."

Vary on the ASCII printer. The host print transform function uses the customized source to map the data sent by the AS/400 system to the printer:

- When you start the printer writer (using the STRPRTWTR command), for spooled print jobs
- When you print the job, for nonspooled print jobs

Deleting the Workstation Customizing Object

Because customization involves a process of trial and error, you may need to delete customizing objects you have created. Use the Delete Workstation Customizing Object (DLTWSCST) command to delete a customized object that did not work as expected or that is no longer needed.

Customizing a Hewlett-Packard LaserJet 4 Printer

This example shows the steps for customizing a Hewlett-Packard LaserJet 4 ASCII printer that uses the host print transform function. The objective for this example is to have the printer use the Roman 8 symbol set. The Roman 8 symbol set uses the code page normally supported by the Hewlett-Packard LaserJet 4 printer.

It is assumed that your printer is physically attached to the system and configured to work with the AS/400 system. The device type for the Hewlett-Packard printer is set to 3812, which is a comparable IBM laser printer.

Setting up the Hewlett-Packard LaserJet 4 Printer: Set up the Hewlett-Packard LaserJet 4 ASCII printer using the following setup values:

Device type	3812
Device model	0
Host print transform	*YES
Manufacturer type and model	*HP4
Paper source 1	*MFRTYPMDL
Paper source 2	*MFRTYPMDL
Envelope source	*MFRTYPMDL
ASCII code page 899 support	*NO

When this is done, you can vary on the printer.

Step 1: Planning the Customizing

For this example, you specifically want to support the Roman 8 symbol set. Customizing an ASCII printer involves trial and error. You may have to go through the workstation customizing procedure several times before the characteristics you want to add or change are working correctly.

To support the Roman 8 symbol set, you find that you need to set the default ASCII code page to Roman 8. You then need to define the ASCII command sequence to select the Roman 8 code page at the printer.

Using the *National Language Support Planning Guide*, you determine that the IBM code page identifier (ID) for the Roman 8 symbol set is code page 1051. Next, using "Overriding the Default ASCII Code Page (DFTASCCP Tag)" on page 10-19, you determine that you change the default code page by specifying the 1051 code page ID in the DFTASCCP tag. In addition, you determine that you define the ASCII command sequence to select the Roman 8 code page at the printer using the CODEPAGE tag as described in "Set Code Page (CODEPAGE Tag)" on page 10-18.

Table 10-1. Printer Function Tags–Example Work Sheet

Tag	Description	Hexadecimal Data
DFTASCCP	Overriding the default code page	ASCIICP=1051
CODEPAGE	Support for a code page	CODEPAGE=1051 DATA='1B283855'X.

Step 2: Retrieving the Workstation Customizing Object Source

To create the workstation customizing source, you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command and specify the following:

Device type	*TRANSFORM
Manufacturer type and model	*HP4
Source member	SRCMBR(CSTHP4)
Source file	SRCFILE(CSTSRC)
Library[1]	LIB(CSTLIB)
Text	TEXT('Customizing source for HP 4 LaserJet Printer')

Note: [1] You must specify an existing library.

Step 3: Changing the Source

After you retrieve the source, use the source entry utility (SEU) to change the printer characteristics and add the special character support. The command looks like the following:

```
STRSEU SRCFILE(CSTLIB/CSTSRC) SRCMBR(CSTHP4)
```

When you edit the file, only make changes, do not delete any of the tags even if no change is made. If tags are deleted, the functions and characteristics associated with those tags are no longer mapped and do not work anymore. This step and the next two steps are performed several times. Each time a few more changes are made to the workstation customizing source, which is then compiled and the device and controller are varied off and on. Print the test document each time you vary on the printer and start the printer writer to see if the changes are effective.

The following listing shows part of the source that you changed to customize the printer. For an example of retrieved source, see Appendix B, "Source Code Examples."

```
:WSCST DEVCLASS=TRANSFORM.

 :TRNSFRMTBL.
        .
        .
        .
/* Add tag here to override the default code page */

    :DFTASCCP
     ASCIICP=1051.

/* End of new tag. */

/* ASCII Code Page */
    :ASCCPINFO.

      :CODEPAGE
        CODEPAGE=437
        DATA='1B28313055'X.

      :CODEPAGE
        CODEPAGE=850
        DATA='1B28313255'X.

/* Add tag here to define the Roman 8 code page */

      :CODEPAGE
        CODEPAGE=1051
        DATA='1B283855'X.

/* End of new tag. */

    :EASCCPINFO.


:EWSCST.
```

Step 4: Creating the Workstation Customizing Object

After you change and save the source, create the workstation customizing object using the Create Work Station Customizing Object (CRTWSCST) command. Follow these steps:

1. Type CRTWSCST on any command line and press F4 (Prompt).
2. Specify a name for the customizing object. This name must be unique to the device for which the object is being created. In this example, you might name the customizing object `MYHP4`.
3. Press F10 (Additional parameters) to display the remaining parameters.
4. Specify the name of the source file member you created and changed. In this example, you specified:
   Source member SRCMBR(CSTHP4)
   Source file SRCFILE(CSTSRC)
   Library LIB(CSTLIB)
5. Specify the authority you want to grant to users who do not have specific authority to the object.
6. Indicate whether or not this customizing object should replace an existing customizing object of the same name. In this example, you would indicate `*NO` in the *Replace* prompt.
7. Specify a description of the customizing object. For this example, you might specify `'Customizing object for HP 4 LaserJet Printer'` for the *Text* prompt.
8. Press Enter.
9. Press F10 (Include detailed messages) on the Command Entry display to verify the command completed successfully.

Step 5: Specifying the Workstation Customizing Object

After you create the workstation customizing object, use the CHGDEVPRT command to specify the customizing object in the device description for the printer. Follow these steps:

1. Vary off the ASCII printer you want to customize.
2. Type CHGDEVPRT on any command line and press F4 (Prompt).
3. Change the *Manufacturer type and model* prompt from `*HP4` to `*WSCST`.
4. Press F10 (Additional parameters) and page through the information until the *Workstation customizing object* prompt is shown. At the *Workstation customizing object* prompt change the value to the unique name you create for the object. In this example, you would change the value to `MYHP4`.

Step 6: Varying On the Device

To activate the workstation customizing function, vary off and then vary on the printer. Then start the printer writer for the printer. Print the test document on the Hewlett-Packard printer. The document is printed using the Roman 8 symbol set.

Chapter 11. Customizing ASCII Printers That Use the Emulator on the Display

This chapter describes the workstation customizing source you use to customize an ASCII printer that does not use the host print transform function. This ASCII printer must be attached to one of the following twinaxial displays:

3477 Model H
3486
3487
3488

The OS/400 workstation customizing functions for ASCII printers attached to twinaxial displays allow you to do the following:

- Customize the functional characteristics of a supported ASCII printer attached to a twinaxial display
- Customize the functional characteristics and specify all the necessary parameters required to support a normally unsupported ASCII printer attached to a twinaxial display

To perform these customizations, the OS/400 licensed program provides a way to change the mapping tables used by the twinaxial display to support an ASCII printer. You must have the reference manual for both the twinaxial display (to set it up) and the ASCII printer in order to provide the new values for the printer mapping tables.

Note: This particular type of customizing may involve a lot of trial-and-error, depending on the type of printer, whether or not the device is already printing, and the completeness of the reference manual for the printer. You may spend anywhere from one to five days to complete a successful ASCII printer customization.

Supported ASCII Printers Attached to Twinaxial Displays

Many IBM ASCII printers are currently supported by twinaxial displays. If the printer supports a character or function, you can add to or change the printing characteristics of these printers. To do this, make changes to a source member that contains the mapping tables used by the display to convert the AS/400 data stream to an ASCII data stream.

Customizing Unsupported ASCII Printers Attached to Twinaxial Displays

To customize an ASCII printer not currently supported by IBM that is attached to a twinaxial display, ask the following questions:

- What printer functions or characteristics and national characters do I want this printer to support?

  Write these down so that you can answer the next question.

- Does the printer itself support the functions I need?

  Check the reference manual to determine this. If neither the printer nor the twinaxial display can support the functions you need, you cannot customize the printer to provide these functions.

- Does the printer emulate or support the emulation of an IBM printer?

  If so, set it up to use the emulation because it could make your customizing easier.

- Is it a laser printer or a line/listing printer?

  Laser printers attached to twinaxial displays are somewhat more difficult to customize and the customizing may require more time than you expect.

The reference manual is especially important when you customize an unsupported device. The hexadecimal code information provided by the reference manual for your device is critical to the workstation customizing process. Without the reference manual you cannot customize the device.

After you have answered the previous questions, you also need to do the following:

- Set up all the necessary hardware to connect the printer to the display

  This is usually a parallel connection.

- Set up any programmable features provided by the printer

  This is the time to set up IBM emulation (if supported), as well as any DIP switches and other parameters that affect the printer.

- Create the necessary controller and device descriptions if needed

  The controller description is for a twinaxial controller and the device description for the printer can be automatically created using automatic configuration.

After you have set up and turned on the display and printer, do not forget to set up for the display to acknowledge the printer. When the display setup is complete, use one of your usual applications, such as OfficeVision/400, to print a short test document. This is the starting point for workstation customizing.

Printer Mapping Table for ASCII Printers Attached to Twinaxial Displays

When you customize a printer definition table, it is downloaded from the workstation controller to the twinaxial display. It is then stored in the memory of the twinaxial display. The workstation controller does not download a particular EBCDIC-to-ASCII mapping table when a printer is varied on if the same table has previously been downloaded for another printer.

Note: For ASCII printers attached to 3477 Model H displays, you must sign off and turn off the display when you replace a printer definition table. This allows a new printer definition table to be downloaded to the display station.

ASCII Printer Definition Table

The following printer support characteristics and capabilities can be specified within a printer definition table:

- Line spacing
- Pitch (characters per inch)
- Page length
- Highlighting characteristics (bold, underscore)
- Draft, letter, or text quality printing
- Paper feed capabilities
- Subscripting and superscripting
- Initialization and reset sequences
- Set global fonts
- Relative forward and backward movement

At first glance, the source for the printer definition table does not look like it supports all the functions listed previously. However, the tags and commands that are part of the tag language for workstation customizing allow you to add the print characteristics that are not included in the default printer definition table. The system provides you with a default printer definition table. It is this table which provides the source for an ASCII printer attached to a twinaxial display. This table is based on a line-oriented printer that has limited printing capabilities.

Determining Which ASCII Printer Tables to Use

When you decide to customize an ASCII printer that is attached to a twinaxial display, you need to be sure you have retrieved the correct source for your twinaxial display and printer using the Retrieve Work Station Customizing Object Source (RTVWSCST) command. When you use the RTVWSCST command to retrieve a source file for twinaxial displays, you also retrieve the printer definition table for attached ASCII printers. Even though the default source for this printer definition table is for a line-oriented printer, you can still use this source as a basis for customizing a laser printer. (See "Customizing a Hewlett-Packard LaserJet Series IIP Printer Attached to a 3477 Twinaxial Display" on page 11-13 for an example showing the customizing of a laser printer using this table.) You can customize both your display and ASCII printer using this workstation customizing source.

Working with the Tag Language for ASCII Printers Attached to Twinaxial Displays

When you specify a twinaxial display for the device type parameter on the Retrieve Work Station Customizing Object Source (RTVWSCST) command, the device class specified in the retrieved source is `DEVCLASS = TWINAXPRT`. The basic structure of the source file member looks like the following figure:

```
:WSCST DEVCLASS=TWINAXPRT.

  :TKBDTBL with parameters.

    :TKSTATE with parameters.
    /*one keyboard state table*/
                  .
                  .

   /*from 1 - 5 additional state tables */
   /* depending on keyboard type */
                  .
                  .

  :PDFNTBL.  /*no parameters*/

                .
                .
       printer function tag  /*tag with parameters*/
                .
                .
                .
                .


:EWSCST.
```

Figure 11-1. Source Structure for ASCII Printers Attached to Twinaxial Displays

When using a workstation customizing object in which either a TKBDTBL or PDFNTBL tag is not present, the default system tables are used when the device is varied on. For more information about customizing the keyboards for twinaxial displays, see the description of the TKBDTBL tag in "Keyboard Translation Table (TKBDTBL) Tag" on page 7-15. This section provides the information you need to change the keyboard state tables required for specific combinations of parameters.

The following sections describe the additional tags that appear in the source you retrieve to customize an ASCII printer attached to a twinaxial display.

Using the Tags to Customize ASCII Printers Attached to Twinaxial Displays

When you retrieve a source file for customizing an ASCII printer attached to a twinaxial display, specify the device type of the twinaxial display. Be sure you specify 3477, 3486, or 3487 for the device type on the Retrieve Work Station Customizing Object Source (RTVWSCST) command.

Note: When you retrieve a source file for customizing an ASCII printer attached to a 3488 display, specify 3487 as the device type. The OS/400 customizing functions support a 3488 display station as a 3487 display station.

ASCII Printer Definition Table (PDFNTBL) Tag

The PDFNTBL (ASCII printer definition table) tag defines an ASCII printer definition table for an ASCII printer attached to a twinaxial display. The syntax for this tag is:

```
:PDFNTBL.
```

Figure 11-2. Syntax for the ASCII Printer Definition Table Tag

There are no keyword parameters associated with this tag. However, it is immediately followed by a number of separate printer function tags, which make up the table entries. See "Printer Function Tags" for more information about the individual printer function tags.

If you delete a printer function tag from your customizing source, the value for that tag is read from the default ASCII printer definition table stored in the emulator. If the same printer function tag occurs multiple times following a PDFNTBL tag, warning messages are sent to the job log. The customizing object uses the last occurrence of the tag in the source to map the printer function.

Printer Function Tags

Printer function tags allow you to specify the ASCII control sequence for an individual printer function. The following sections describe the valid functions and corresponding tags for the device class of ASCII printer attached to a twinaxial display (DEVCLASS=TWINAXPRT). These tags must follow an ASCII printer definition table tag (PDFNTBL) in your source. Although some of these tags are valid for the device class of ASCII printer (DEVCLASS=ASCIIPRT), the parameters and values may differ. If you are customizing an ASCII printer that is directly attached to the ASCII workstation controller, use the tags described in Chapter 12, "Customizing ASCII Printers That Use the Emulator on the Controller."

Printer Function Tags for 3477 Model H, 3486, 3487, and 3488 Displays

The following printer function tags work for all 3477 Model H, 3486, 3487, and 3488 displays. The syntax for each tag is shown, following the tag description. Some tags have only a data parameter. Other tags have more than one parameter to describe variables within functions.

Printer Function Tags with Variable and Relative Movement

The following table shows the printer function tags that have a variable and relative movement for the ASCII printer attached to a twinaxial display. Following the table is a description of the syntax for these tags and an example of how to change the values associated with the tags.

Table 11-1. Printer Function Tags with a Variable and Relative Movement

Tag	Description
FWDRMOV	Forward Relative Movement
BCKRMOV	Backward Relative Movement
PAGLENI	Set Page Length in Inches
VARLSPC	Variable Line Spacing

The general syntax for a printer function tag with a variable and relative movement is:

```
:xxxxxxxxxx
         VAROFFSET = variable offset in
                     control sequence
         VARLEN = variable length
         VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
                   CHRHEX|CHRAN
         VARMAX = maximum variable value
         ADJUST = adjustment
           CNVNUM = conversion ratio numerator
           CNVDEN = conversion ratio denominator
           DATA = ASCII control sequence.
```

VAROFFSET
A required parameter. Defines the offset for the variable portion of the control sequence. This value must be an integer.

Note: This offset is relative to the beginning of the control sequence. Therefore, a value of zero (0) implies that the variable starts in the first byte of the control sequence.

VARLEN
A required parameter. Defines the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

VARTYPE
A required parameter. Defines the type of variable used with this category of printer function tags.

HIGHLOW
The byte order of the variable is in high-low order. The most significant byte is first.

LOWHIGH
The byte order of the variable is in low-high order. The most significant byte is last.

CHRDEC
The variable is in character decimal format with no byte order consideration. All characters are in the range from 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.

CHRHEX
The variable is in character hexadecimal format with no byte order consideration. All characters are in the range from 0 to 9, A to F.

CHRAN
The variable is in character alphanumeric format with no byte order consideration. All characters are in the range from 0 to 9, A to Z.

VARMAX
A required parameter. Defines the maximum variable value. This value must be an integer.

ADJUST
A required parameter. Defines the adjustment value for the variable. This value must be a signed integer.

CNVNUM
A required parameter. Defines the numerator of the conversion ratio. This value must be an integer. The most commonly used value for CVNUM is 1.

CNVDEN
A required parameter. Defines the denominator of the conversion ratio. This value must be an integer other than zero.

CNVDEN defines the units to be used for the variable portion. For example, if movement is in 1/300ths, CNVNUM should be defined as 1 and CNVDEN should be defined as 300.

DATA
A required parameter. Specifies the ASCII control sequence for the printer function. The maximum length of this value is 240 bytes. This value is the hexadecimal string associated with the function that is defined in the reference manual for the device.

Backspace (BSP) Tag

The BSP (backspace) tag defines the ASCII control sequence for the backspace function for an ASCII printer. The syntax for this tag is:

```
:BSP
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the backspace function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Bell (BELL) Tag

The BELL (bell) tag defines the ASCII control sequence for the bell function for an ASCII printer. The bell function sounds the printer bell or beeper briefly. The syntax for this tag is:

```
:BELL
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the bell function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Start Bold Printing Function (STRBOLD) Tag

The STRBOLD (start bold printing) tag defines the ASCII control sequence for the start bold printing function for an ASCII printer. The syntax for this tag is:

```
:STRBOLD
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the start bold printing function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

End Bold Printing (ENDBOLD) Tag

The ENDBOLD (end bold printing) tag defines the ASCII control sequence for the end bold printing function for an ASCII printer. The syntax for this tag is:

```
:ENDBOLD
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the end bold printing function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Carrier Return (CARRTN) Tag

The CARRTN (carrier return) tag defines the ASCII control sequence for the carrier return function for an ASCII printer. The CARRTN function returns the carrier to the left margin, without advancing a line. The syntax for this tag is:

```
:CARRTN
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the carrier return function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Set Characters per Inch (CPI) Tag

The CPI (set characters per inch) tag defines the control sequence for setting the number of characters per inch. The CPI tag must follow a PDFNTBL tag in your source. The syntax for this tag is:

```
:CPI
        CPI = 5|6|855|10|12|15|171
        DATA = ASCII control sequence.
```

CPI
A required parameter. Defines the number of characters per inch.

5	5 characters per inch
6	6 characters per inch
855	8.55 characters per inch
10	10 characters per inch
12	12 characters per inch
15	15 characters per inch
171	17.1 characters per inch

DATA
A required parameter. Specifies the ASCII control sequence for setting the number of characters per inch. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Note: See "Set Characters per Inch (CPI) Tag" on page 11-11 for additional parameter values for this tag when used with 3486, 3487, and 3488 displays.

Set Code Page (CODEPAGE) Tag

The CODEPAGE (set code page) tag defines the ASCII control sequence for setting different code pages. The CODEPAGE tag must follow a PDFNTBL tag in your source. The syntax for this tag is:

```
:CODEPAGE
        CODEPAGE = 850|899
        DATA = ASCII control sequence.
```

CODEPAGE
A required parameter. Specifies the identifier (ID) of the code page to be set.

850	ASCII code page value
899	ASCII code page value

DATA
A required parameter. Specifies the ASCII control sequence for setting the code pages for an ASCII printer attached to a twinaxial display. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Set Double Character Height (DBLCHRH) Tag

The DBLCHRH (set double character height) tag defines the ASCII control sequence for doubling the character height for an ASCII printer. The syntax for this tag is:

```
:DBLCHRH
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the set double character height function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Start Double-Wide Continuous (STRWIDE) Tag

The STRWIDE (start double-wide continuous) tag defines the ASCII control sequence for starting double-wide character spacing for an ASCII printer. The syntax for this tag is:

```
:STRWIDE
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the start double-wide continuous spacing function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

End Double-Wide Continuous (ENDWIDE) Tag

The ENDWIDE (end double-wide continuous) tag defines the ASCII control sequence for ending double-wide character spacing for an ASCII printer. The syntax for this tag is:

```
:ENDWIDE
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the end double-wide continuous spacing function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Drawer Selection (DWRSLT) Tag

The DWRSLT (drawer selection) tag defines the control sequence for drawer selection. Drawer selection refers to the slots on the printer in which various sizes (or types) of paper are stored for use by the printer. The DWRSLT tag must follow the PDFNTBL tag in your source. The syntax for this tag is:

```
:DWRSLT
        DRAWER = PAPER|ENVELOPE|
                 DRAWER1|DRAWER2
        DATA = ASCII control sequence.
```

DRAWER
A required parameter. Defines the drawer or slot from which paper is to be used for printing.

PAPER
Selects manual paper feed as the source of paper for documents that are to be printed.

ENVELOPE
Selects the envelope drawer as the source of paper for documents that are to be printed.

DRAWER1
Selects drawer 1 as the source of paper for documents that are to be printed.

DRAWER2
Selects drawer 2 as the source of paper for documents that are to be printed.

DATA
A required parameter. Specifies the ASCII control sequence for the function. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Global Fonts for Printer Definition Table (FNTGPDT) Tag

The FNTGPDT (global fonts for printer definition table (PDT)) tag defines the beginning of the global font ranges for a printer definition table. The FNTGPDT tag must follow the PDFNTBL tag in your source. It must be followed by at least one FNTGRNG (global font range) tag. The FNTGRNG tag must in turn be followed by an EFNTGPDT (end global fonts for printer definition table) tag. The syntax for the FNTGPDT tag is:

```
:FNTGPDT.
```

End Global Fonts for PDT (EFNTGPDT) Tag

The EFNTGPDT (end global fonts for printer definition table (PDT)) tag ends the set of global font ranges for a PDT. The syntax for this tag is:

```
:EFNTGPDT.
```

Global Font Range (FNTGRNG) Tag

The FNTGRNG (global font range) tag defines a range of global fonts for a printer definition table (PDT). The FNTGRNG tag must follow either a FNTGPDT (global fonts for printer definition table) tag or another FNTGRNG tag in your source. You can define a maximum of five fonts for use with the 3477 Model H display. The syntax for this tag is:

```
:FNTGRNG
        MINFID = font id
        MAXFID = font id
        DATA = ASCII control sequence.
```

MINFID
A required parameter. Defines the smallest global font ID in a group. This value must be an integer.

MAXFID
A required parameter. Defines the largest global font ID in a group. This value must be an integer.

DATA
A required parameter. Specifies the ASCII control sequence for the global font range. This must be a hexadecimal value. The maximum length of this value is 37 bytes.

Foreground Color (FOREGRND) Tag

The FOREGRND (foreground color) tag defines the ASCII control sequence for setting different print or ink colors when the printer supports color. The FOREGRND tag must follow the PDFNTBL tag in your source. The syntax for this tag is:

```
:FOREGRND
        COLOR = BLACK|BLUE|RED|PINK|GREEN|
                CYAN|YELLOW|WHITE|DRKBLUE|
                ORANGE|PURPLE|DRKGREEN|
                TURQ|MUSTARD|GREY|BROWN
        DATA = ASCII control sequence.
```

COLOR
A required parameter. Defines the color of the ink or print to be used by the printer when it supports different colors for printing.

BLACK	DRKBLUE (dark blue)	MUSTARD	RED
BLUE	DRKGREEN (dark green)	ORANGE	TURQ (turquoise)
BROWN	GREEN	PINK	WHITE
CYAN	GREY	PURPLE	YELLOW

DATA
A required parameter. Specifies the ASCII control sequence for setting the foreground color. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Form feed (FORMFEED) Tag

The FORMFEED (form feed) tag defines the ASCII control sequence for the form feed function for an ASCII printer. The FORMFEED control advances the paper to the top of the next page. The syntax for this tag is:

```
:FORMFEED
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the form feed function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Initialize Printer (INITPRT) Tag

The INITPRT (initialize printer) tag defines the ASCII control sequence for the initialize printer function for an ASCII printer. The syntax for this tag is:

```
:INITPRT
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the initialize printer function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Line Feed (LINEFEED) Tag

The LINEFEED (line feed) tag defines the ASCII control sequence for the line feed function for an ASCII printer. The line feed function advances the paper one line. The syntax for this tag is:

```
:LINEFEED
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the line feed function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Set Lines per Inch (LPI) Tag

The LPI (set lines per inch) tag defines the control sequence for setting the number of lines per inch you want to print. The LPI tag must follow the PDFNTBL tag in your source. The syntax for this tag is:

```
:LPI
        LPI = 3|4|6|8
        DATA = ASCII control sequence.
```

LPI
A required parameter. Defines the number of lines per inch. The valid values for this parameter are 3, 4, 6, and 8.

DATA
A required parameter. Specifies the ASCII control sequence for the function. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Set Page Length in Lines (PAGLENL) Tag

The PAGLENL (set page length in lines) tag sets the page length in terms of the number of lines. The number of lines is carried as a variable in the control sequence for page length. The PAGLENL tag must follow the PDFNTBL tag in your source. The syntax for this tag is:

```
:PAGLENL
        VAROFFSET = variable offset in
                    control sequence
        VARLEN = variable length
        VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
                  CHRHEX|CHRAN
        VARMAX = maximum variable value
        ADJUST = adjustment
        DATA = ASCII control sequence.
```

VAROFFSET
A required parameter. Defines the offset for the variable in the control sequence. This value must be an integer.

Note: This offset is relative to the beginning of the control sequence. Therefore, a value of zero (0) implies that the variable starts in the first byte of the control sequence.

VARLEN
A required parameter. Defines the length of the variable. This value must be an integer.

VARTYPE
A required parameter. This attribute defines the type of variable. Possible values are :

HIGHLOW
The byte order of the variable is in high-low order.

LOWHIGH
The byte order of the variable is in low-high order.

CHRDEC
The variable is in characters with no byte order consideration. All characters are in the range 0 to 9.

CHRHEX
The variable is in characters with no byte order consideration. All characters are in the range 0 to 9, A to F.

CHRAN
The variable is in characters with no byte order consideration. All characters are in the range 0 to 9, A to Z.

VARMAX
A required parameter. Defines the maximum variable value. This value must be an integer.

ADJUST
A required parameter. Defines the adjustment value to the variable. This value must be a signed integer.

DATA
A required parameter. Specifies the ASCII control sequence for the printer function. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Paper Feed (PRTFEED) Tag

The PRTFEED (paper feed) tag defines the control sequence for different types of paper feeding. The PRTFEED tag must follow the PDFNTBL tag in your source. The syntax for this tag is:

```
:PRTFEED
        FEEDTYPE = EJECT|MANUAL|AUTO
        DATA = ASCII control sequence.
```

FEEDTYPE
A required parameter. Defines the type of paper feeding.

EJECT
The paper is ejected from the printer

MANUAL
The paper is fed to the printer manually (by hand).

AUTO
The paper is fed to the printer automatically.

DATA
A required parameter. Specifies the ASCII control sequence for the function. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Paper Orientation (PRTORIENT) Tag

The PRTORIENT (paper orientation) tag defines the control sequence for setting different paper orientations. The PRTORIENT tag must follow the PDFNTBL tag in your source. The syntax for this tag is:

```
:PRTORIENT
        ORIENT = PORTRAIT|LANDSCAPE
        DATA = ASCII control sequence.
```

ORIENT
A required parameter. Defines the orientation in which a job prints.

PORTRAIT
The print job prints in an orientation rotated 0 degrees.

LANDSCAPE
The print job prints in an orientation rotated 90 degrees.

DATA
A required parameter. Specifies the ASCII control sequence for setting the paper orientation for the printer. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Note: For additional parameter values for this tag when used with 3486, 3487, and 3488 displays, see "Paper Orientation (PRTORIENT) Tag" on page 11-12.

Print Quality (PRTQLTY) Tag

The PRTQLTY (print quality) tag defines the print quality control sequence. The PRTQLTY tag must follow the PDFNTBL tag in your source. The syntax for this tag is:

```
:PRTQLTY
        QLTYTYPE = DRAFT|LETTER|TEXT
        DATA = ASCII control sequence.
```

QLTYTYPE
A required parameter. Defines the quality of print.

DRAFT
The print quality is draft quality. This is equivalent to the *DRAFT type used in the OS/400 printer file commands.

LETTER
The print quality is letter quality. This is equivalent to the *NLQ type used in the OS/400 printer file commands.

TEXT
The print quality is text quality. This is equivalent to the *STD type used in the OS/400 printer file commands.

DATA
A required parameter. Specifies the ASCII control sequence for the function. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Start Proportional Space (STRPROP) Tag

The STRPROP (start proportional space mode) tag defines the ASCII control sequence for the start proportional space mode for an ASCII printer. The syntax for this tag is:

```
:STRPROP
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the start proportional space mode function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

End Proportional Space (ENDPROP) Tag

The ENDPROP (end proportional space mode) tag defines the ASCII control sequence for the end proportional space mode function for an ASCII printer. The syntax for this tag is:

```
:ENDPROP
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the end proportional space mode function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Quality Font Download (SETQLTY) Tag

The SETQLTY (quality font download) tag defines the ASCII control sequence that sets the print quality for specific fonts. The SETQLTY tag must follow the PDFNTBL tag in your source. The syntax for this tag is:

```
:SETQLTY
        QLTYTYPE = DRAFT|LETTER
        DATA = ASCII control sequence.
```

QLTYTYPE
A required parameter. Defines the quality for the font to be downloaded.

DRAFT
The font is a draft quality font.

LETTER
The font is a letter quality font.

DATA
A required parameter. Specifies the ASCII control sequence for downloading the quality font. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Space (SPACE) Tag

The SPACE (space) tag defines the ASCII control sequence for the space control function for an ASCII printer. The syntax for this tag is:

```
:SPACE
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the space function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Set Standard Character Height (STDCHRH) Tag

The STDCHRH (set standard character height) tag defines the ASCII control sequence for the set standard character height function for an ASCII printer. The syntax for this tag is:

```
:STDCHRH
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the set standard character height function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Start Subscript (STRSUBS) Tag

The STRSUBS (start subscript) tag defines the ASCII control sequence for the start subscript function for an ASCII printer. The syntax for this tag is:

```
:STRSUBS
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the start subscript function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

End Subscript (ENDSUBS) Tag

The ENDSUBS (end subscript) tag defines the ASCII control sequence for the end subscript function for an ASCII printer. The syntax for this tag is:

```
:ENDSUBS
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the end subscript function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Start Superscript (STRSUPS) Tag

The STRSUPS (start superscript) tag defines the ASCII control sequence for the start superscript function for an ASCII printer. The syntax for this tag is:

```
:STRSUPS
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the start superscript function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

End Superscript (ENDSUPS) Tag

The ENDSUPS (end superscript) tag defines the ASCII control sequence for the end superscript function for an ASCII printer. The syntax for this tag is:

```
:ENDSUPS
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the end superscript function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Table Name (TBLNAME) Tag

The TBLNAME (table name) tag specifies a name for the printer definition table. The TBLNAME tag must follow the PDFNTBL tag in your source. The syntax for this tag is:

```
:TBLNAME
        DATA = name.
```

DATA
A required parameter. Specifies a table name for the printer definition table.

name
A 1- to 8-character string used to name the printer definition table. Only characters defined in code page 500 may appear in the name.

Translation Printer Definition Table (TRNEBCDIC) Tag

The TRNEBCDIC (translation printer definition table) tag defines the beginning of the translation mappings for a printer definition table (PDT). The TRNEBCDIC tag must follow the PDFNTBL tag in your source. It must be followed by at least one TRNEBCDICE (translation entry) tag, which must be followed by an ETRNEBCDIC (end translation PDT) tag. The syntax for this tag is:

```
:TRNEBCDIC.
```

End Translation Printer Definition Table (ETRNEBCDIC) Tag

The ETRNEBCDIC (end translation printer definition table (PDT)) tag ends the set of translation mappings for a PDT. The syntax for this tag is:

```
:ETRNEBCDIC.
```

Translation Entry (TRNEBCDICE) Tag

The TRNEBCDICE (translation entry) tag defines the ASCII control sequence for mapping one EBCDIC character to one ASCII character. The TRNEBCDICE tag must follow either a TRNEBCDIC (translation printer definition table (PDT)) tag or another TRNEBCDICE tag in your source. The syntax for this tag is:

```
:TRNEBCDICE
        EBCDIC = code
        DATA = ASCII control sequence.
```

EBCDIC
A required parameter. Specifies the EBCDIC code you want to map. This must be a hexadecimal value.

DATA
A required parameter. Specifies the ASCII control sequence for the EBCDIC character to ASCII character mapping. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Start Underscore (STRUS) Tag

The STRUS (start underscore) tag defines the ASCII control sequence for the start underscore function for an ASCII printer. The syntax for this tag is:

```
:STRUS
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the start underscore function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

End Underscore (ENDUS) Tag

The ENDUS (end underscore) tag defines the ASCII control sequence for the end underscore function for an ASCII printer. The syntax for this tag is:

```
:ENDUS
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the end underscore function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Set Vertical Units (VERUNT) Tag

The VERUNT (set vertical units) tag defines the ASCII control sequence for the set vertical units function for an ASCII printer. The syntax for this tag is:

```
:VERUNT
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the set vertical units function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Additional Tags for Use with 3486, 3487, and 3488 Displays

The following tags, or additional parameters described with tags, work with 3486, 3487, and 3488 displays. These tags or additional parameters do not work for 3477 Model H displays.

The following tags define more printer functions and must also follow the PDFNTBL tag in your workstation source.

Adjust Horizontal Origin (ADJHRZORG) Tag

The ADJHRZORG (adjust horizontal origin) tag specifies the horizontal print head positioning. The ADJHRZORG tag must follow an ASCII printer definition table (PDFNTBL) tag in your source. The syntax for this tag is:

```
:ADJHRZORG
        ADJUST = adjustment.
```

ADJUST
A required parameter. Specifies the size of the adjustment for horizontal print head positioning in 1/1440-inch increments. This value must be a signed integer.

adjustment
Integer value for the adjustment size for horizontal print head positioning.

Adjust Vertical Origin (ADJVERORG) Tag

The ADJVERORG (adjust vertical origin) tag specifies the vertical print head positioning. This is the vertical position on the physical page where printing should begin. The ADJVERORG tag must follow an ASCII printer definition table tag (PDFNTBL) in your source. The syntax for this tag is:

```
:ADJVERORG
        ADJUST = adjustment.
```

ADJUST
A required parameter. Specifies the size of the adjustment for vertical print head positioning in 1/1440-inch increments. This value must be a signed integer.

adjustment
Integer value for the adjustment size for vertical print head positioning.

Set Characters per Inch (CPI) Tag

The CPI (set characters per inch) tag defines the control sequence for setting the number of characters per inch. The CPI tag must follow a PDFNTBL tag in your source. The syntax for this tag is:

```
:CPI
        CPI = 5|6|855|10|12|15|171|20|25
        DATA = ASCII control sequence.
```

CPI
A required parameter. Defines the number of characters per inch.

5 5 characters per inch
6 6 characters per inch
855 8.55 characters per inch
10 10 characters per inch
12 12 characters per inch
15 15 characters per inch
171 17.1 characters per inch
20 20 characters per inch
25 25 characters per inch

Note: The 20 CPI and 25 CPI values are supported only when your ASCII printer is attached to a 3486, 3487, or 3488 twinaxial display.

DATA
A required parameter. Specifies the ASCII control sequence for setting the number of characters per inch. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Set Characters per Inch in COR Mode (CPICOR) Tag

The CPICOR (set characters per inch in computer output reduction (COR) mode) tag defines the control sequence for setting the number of characters per inch while in COR mode. For example, you can define all print jobs that normally use a 10-pitch type style use a particular 17-pitch type style when COR (reduction) is applied.

The CPICOR tag must follow the PDFNTBL tag in your source. The syntax for this tag is:

```
:CPICOR
        CPI = 10|12|15
        ASCIIFNT = type style
        FNTWTH = integer
        FNTATR = integer
        DATA = ASCII control sequence.
```

CPI
A required parameter. This is the current number of characters per inch before the reduction is applied. For example, to define the type style to use for COR in a 10-pitch print job, specify CPI=10.

10 10 characters per inch
12 12 characters per inch
15 15 characters per inch

ASCIIFNT
A required parameter. Specifies the ASCII type style (the global font ID (FGID)) to be used when reduction is applied. This value must be an integer.

For example, the FGID for a Courier 17-pitch font (17 characters per inch) is 254. Therefore, you specify ASCIIFNT=254 for a Courier 17-pitch font. For information on global font IDs, see the *Guide to Programming for Printing*.

FNTWTH
A required parameter. Specifies the width, in 1/1440-inch increments, of the font used when reduction is applied, This value must be an integer.

For example, if you use a 17-pitch font when reduction is applied, divide the increment by the number of characters per inch (1440/17). This gives you a font width of 85 (rounded off) for the substituted font. Therefore, you specify FNTWTH=85.

FNTATR
A required parameter. Specifies the attribute value for the substituted font. This value must be an integer.

For example, the technical reference manual for the IBM 4029 printer lists 0 through 4 as possible attribute values for the substituted font. In most cases, you specify a fixed-pitch type style (FNTATR=1) for the substitute font attribute value.

DATA
A required parameter. Specifies the ASCII control sequence to select the substituted font. This must be a hexadecimal value. The maximum length of this value is 240 bytes. You find the ASCII control sequence in the technical reference manual for your printer under the description of selecting type styles.

For example, the control sequence for selecting type styles for an IBM 4029 printer is in the 4029 technical reference manual under the description of the Select Font Global command. To select a font group ID of 254 for an IBM 4029 printer, you specify DATA='1B5B49050000FE005401'X.

Duplex Printing (DUPXPRT) Tag

The DUPXPRT (duplex printing) tag defines the ASCII control sequence for the duplex printing function for an ASCII printer. The DUPXPRT control prints on both sides of a sheet of paper. The syntax for this tag is:

```
:DUPXPRT
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the duplex printing function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Half Line Feed (HLFLINEFEED) Tag

The HLFLINEFEED (half line feed) tag defines the ASCII control sequence for the half line feed function for an ASCII printer. The HLFLINEFEED control advances the paper one half of a line. The syntax for this tag is:

```
:HLFLINEFEED
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the half line feed function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Jog Output Tray (JOGOUTTRAY) Tag

The JOGOUTTRAY (jog output tray) tag defines the ASCII control sequence for the jog output tray function for an ASCII printer. The syntax for this tag is:

```
:JOGOUTTRAY
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the jog output tray function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Select Next Side Printing in Duplex (NXTDUPXPRT) Tag

The NXTDUPXPRT (select next side printing in duplex) tag defines the ASCII control sequence for the select next side printing in duplex function for an ASCII printer. The syntax for this tag is:

```
:NXTDUPXPRT
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the select next side printing in duplex function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Paper Orientation (PRTORIENT) Tag

The PRTORIENT (paper orientation) tag defines the control sequence for setting different paper orientations. The PRTORIENT tag must follow the PDFNTBL tag in your source. The syntax for this tag is:

```
:PRTORIENT
        ORIENT = PORTRAIT|LANDSCAPE|
                 RTT180|RTT270
        DATA = ASCII control sequence.
```

ORIENT
A required parameter. The orientation in which a print job prints.

PORTRAIT
The print job prints in an orientation rotated 0 degrees.

LANDSCAPE
The print job prints in an orientation rotated 90 degrees.

RTT180
The print job prints in an orientation rotated 180 degrees.

RTT270
The print job prints in an orientation rotated 270 degrees.

Note: The RTT180 and RTT270 values are supported only when your ASCII printer is attached to a 3486, 3487, or 3488 twinaxial display.

DATA
A required parameter. Specifies the ASCII control sequence for setting the paper orientation for the printer.

This must be a hexadecimal value. The maximum length of this value is 240 bytes.

Reverse Half Line Feed (RVSHLFLINEFEED) Tag

The RVSHLFLINEFEED (reverse half line feed) tag defines the ASCII control sequence for the reverse half line feed function for an ASCII printer. The reverse half line feed function moves the paper back up one half line. The syntax for this tag is:

```
:RVSHLFLINEFEED
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the reverse half line feed function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Reverse Line Feed (RVSLINEFEED) Tag

The RVSLINEFEED (reverse line feed) tag defines the ASCII control sequence for the reverse line feed function for an ASCII printer. The reverse line feed function moves the paper back up one line. The syntax for this tag is:

```
:RVSLINEFEED
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the reverse line feed function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Set Simplex Printing (SMPXPRT) Tag

The SMPXPRT (set simplex printing) tag defines the ASCII control sequence for the set simplex printing function for an ASCII printer. The SMPXPRT tag sets the printer to print on one side of the paper. The syntax for this tag is:

```
:SMPXPRT
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the set simplex printing function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Set Tumble Duplex Printing (TUMDUPXPRT) Tag

The TUMDUPXPRT (set tumble duplex printing) tag defines the ASCII control sequence for the set tumble duplex printing function for an ASCII printer. The syntax for this tag is:

```
:TUMDUPXPRT
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the set tumble duplex printing function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Set Vertical Units in Half (VERUNTHLF) Tag

The VERUNTHLF (set vertical units in half) tag defines the ASCII control sequence for the set vertical units in half function for an ASCII printer. The syntax for this tag is:

```
:VERUNTHLF
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the set vertical units in half function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

Customizing a Hewlett-Packard LaserJet Series IIP Printer Attached to a 3477 Twinaxial Display

This example shows the steps for the customizing of a Hewlett-Packard LaserJet Series IIP ASCII printer attached to a 3477 Model H twinaxial display with a 122-key typewriter keyboard. The objective, for this example, is having the printer work effectively with simple OfficeVision/400 documents.

It is assumed that your twinaxial workstation is physically attached to the system, configured to work with the AS/400 system (appropriate controller and device descriptions are created), and is varied on. It is also assumed that the AS/400 system has automatically configured the attached printer (system value QAUTOCFG = 1). When you do this, the device type for the Hewlett-Packard printer is set to 5219, which is a comparable IBM laser printer.

Setting up the 3477 Model H Display: First, set up the 3477 display station to handle a printer. Select the following setup values to accommodate the Hewlett-Packard LaserJet Series IIP ASCII printer.

Terminal mode	Display-Printer
Display address	0
Printer address	1
Printer character set	Multilingual4
Printer emulation	5219
Keyboard type	Standard
ASCII printer type	User-Defined
ASCII printer ID	01 (The ASCII printer ID cannot be set to 00 because of limitations of the twinaxial display.)

Next, change the setup for the printer so that the connection to the 3477 display is parallel. You should also check the DIP switch settings and the other setup values that are programmed on the printer itself. When this is done, you can vary on the printer.

Step 1: Planning the Customizing

Customizing an ASCII printer attached to a 3477 Model H display involves trial-and-error. You may have to go through the workstation customizing procedure several times before all the characteristics you want to add or change are working correctly.

First, create a simple one-page document using the OfficeVision/400 editor as shown in Figure 11-3. Print this document to test the characteristics that you expect from your printer. For this example, you specifically want to support the following print characteristics:

- Bold printing
- Centering
- Underlined printing
- Line spacing
- Portrait/Landscape printing

You may need to change more than just these characteristics, but you do not know that until you have tried customizing and seen what the results are.

```
Example Showing Printer Functions Before and After Customizing

This is a test for bold printing.


This is a test for underscored printing.


This is a test for centering.


The following two lines will be separated by a 6-lines-per-inch
spacing.  The following two lines will be separated by a
6-lines-per-inch spacing.


The following two lines will be separated by an 8-lines-per-inch
spacing.  The following two lines will be separated by an
8-lines-per-inch spacing.


This is a test for print orientation in portrait.


This is a test for print orientation in landscape. (Rotate 90)


This is a test to show the customized special characters that
would not print without customizing:
(not used for ASCII printer attached to 3477 test)

ä ö ü Ä Ö Ü
```

Figure 11-3. OfficeVision/400 Printer Test Document

The first time you print this document, extra pages are generated, some strange characters are printed on the top of the page, the title appears alone on the second page, and none of the other print characteristics are apparent except for centering. The strange character string at the top of the page looks like this:


RV2H486-0

The last page of the test document when printed the first time looked something like Figure 11-4 on page 11-16.

To remove the strange characters, you investigate the printer initialization sequence. You find that you need to set the variable line spacing, page length in lines and inches, forward and backward relative movement, paper feed to the printer, and drawer selection.

After you know what needs to be changed, check to see if the printer itself can support these functions. If so, then you need to fill in the printer customizing work sheet showing the printer characteristic and the associated hexadecimal value from the printer reference manual. Remember, if the device does not support a character or function, then workstation customizing cannot provide it either.

After checking the reference manual for the Hewlett-Packard LaserJet Series IIP printer, you find that subscripts and superscripts are not supported at all (with the basic font cartridge). As you are going through the manual, you write down the hexadecimal codes that the printer uses to call the other functions and characteristics you want to support. The list is shown in Table 11-2, Table 11-3 on page 11-15, and Table 11-4 on page 11-17.

You also find that Hewlett-Packard uses some terminology that is different from IBM terminology when referring to some of the printer characteristics. For example, IBM uses the terms code page and character set to describe different sets of national language and graphics characters. Hewlett-Packard provides support for different character sets, but refers to them as symbol sets. Knowing that there are some differences in terminology helps you find some of the hexadecimal codes needed for customizing.

Work Sheet

Table 11-2. Printer Function Tags with Only a Data Parameter – Example

Tag	Description	Hexadecimal Data
INITPRT	Initialize Printer	DATA='1B5B4B03000031011B5B5C020090001B5B5405000000035200'X
ENDBOLD	End Bold Printing	DATA='1B46'X
ENDUS	End Underscore	DATA='1B2D00'X
STRBOLD	Start Bold Printing	DATA='1B45'X
STRUS	Start Underscore	DATA='1B2D01'X

Movement – Example Work Sheet

Table 11-3 (Page 1 of 2). Printer Function Tags with Variable and Relative

Tag	Description	Data (see Note)
FWDRMOV	Forward Relative Movement	VAROFFSET= 2 VARLEN= 2 VARTYPE= **LOWHIGH** VARMAX= 32767 ADJUST= 0 CNVNUM= 12 CNVDEN= 1 DATA='1B640F00'X
BCKRMOV	Backward Relative Movement	VAROFFSET= 2 VARLEN= 2 VARTYPE= **LOWHIGH** VARMAX=32767 ADJUST=0 CNVNUM=12 CNVDEN=1 DATA='1B650F00'X

Table 11-3 (Page 2 of 2). Printer Function Tags with Variable and Relative

Tag	Description	Data (see Note)
PAGLENI	Set Page Length in Inches	VAROFFSET= 3 VARLEN= 1 VARTYPE= **LOWHIGH** VARMAX= 255 ADJUST= 0 CNVNUM= 1440 CNVDEN= 1 DATA='1B43000B'X
VARLSPC	Variable Line Spacing	VAROFFSET= 11 VARLEN= 1 VARTYPE= **LOWHIGH** VARMAX= 255 ADJUST= 0 CNVNUM= 10 CNVDEN= 1 DATA='1B5B5C040090009 0001B41101B32'X.

Note: Values in the Data column can be hexadecimal or integer.

```
?This is a test for bold printing.
?
?
?This is a test for underscored printing.
?
?
?                     This is a test for centering.
?
?
?The following two lines will be separated by a 6-lines-per-inch
?spacing.  The following two lines will be separated by a
?6-lines-per-inch spacing.
?
?
?The following two lines will be separated by an 8-lines-per-inch
?spacing.  The following two lines will be separated by an
?8-lines-per-inch spacing.
?
?
?This is a test for print orientation in portrait.
?
?
?This is a test for print orientation in landscape. (Rotate 90)
?
?
?This is a test to show the customized special characters that
?would not print without customizing:
?(not used for ASCII printer attached to 3477 test)
?
?
?ä ö ü Ä Ö Ü
```

Figure 11-4. OfficeVision/400 Test Document First Printout

Table 11-4. Other Printer Function Tags – Example Work Sheet

Tag	Description	Data
DWRSLT	Drawer selection	DRAWER= **DRAWER1** DATA='1B5B460300030101'X
DWRSLT	Drawer selection	DRAWER= **DRAWER2** DATA='1B5B460300030102'X
DWRSLT	Drawer selection	DRAWER= **ENVELOPE** DATA='1B5B460300030200'X
PAGLENL	Set page length in lines	VAROFFSET= 2 VARLEN= 1 VARTYPE= **LOWHIGH** VARMAX= 255 ADJUST= 0 DATA='1B4342'X
PRTFEED	Paper feed	FEEDTYPE= **MANUAL** DATA='1B5B460300010100'X
PRTORIENT	Paper orientation	ORIENT= **PORTRAIT** DATA='1B266C304F'X
PRTORIENT	Paper orientation	ORIENT= **LANDSCAPE** DATA='1B266C314F'X

Step 2: Retrieving the Workstation Customizing Object Source

To create the workstation customizing source, you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command and specify the following:

Device type
DEVTYPE(3477)

Keyboard language type
KBDTYPE(USI) (This value must match the keyboard language type you specified in the device description for the display; otherwise, the display will not vary on.)

Source member
SRCMBR(CST3477P)

Keyboard attached
KBD(*TYPE122)

Source file
SRCFILE(CSTSRC)

Library LIB(CSTLIB)

Text TEXT('Workstation customizing source for 3477 with attached ASCII Printer')

Step 3: Changing the Source

After you retrieve the source, use the source entry utility (SEU) to change the printer characteristics and add the special character support. The command looks like the following:

```
STRSEU SRCFILE(CSTLIB/CSTSRC) SRCMBR(CST3477P)
```

| This step and the next two steps are performed several
| times. Each time a few more changes are made to the work-
| station customizing source, which is then compiled and the
| device and controller are varied off and on. Print the test
| document each time you vary on the printer to see if the
| changes are effective.

| If you delete a printer function tag from your customizing
| source, the value for that tag is read from the default ASCII
| printer definition table stored in the emulator. If you perform
| a hexadecimal trace, you may see escape sequences for
| tags that are read from the default table, in addition to the
| tags read from the customizing source. Therefore, to remove
| support for printer functions, you must specify the tags for
| those functions in your customizing source. You then assign
| a hexadecimal value of null (''X) to the tags to remove
| support for those functions.

The following listing shows part of the source that you changed to customize the printer. For an example of the original retrieved source, see Appendix B, "Source Code Examples."

```
      .
      .
      .
:PDFNTBL.
      .
      .
      .
:VARLSPC
   VAROFFSET=   11
   VARLEN=    1
   VARTYPE=LOWHIGH
   VARMAX=  255
   ADJUST= 0
   CNVNUM=   10
   CNVDEN=    1
   DATA ='1B5B5C0400900090001B41101B32'X.
:PAGLENI
   VAROFFSET=    3
   VARLEN=    1
   VARTYPE=LOWHIGH
```

```
   VARMAX=  255
   ADJUST= 0
   CNVNUM= 1440
   CNVDEN=    1
   DATA ='1B43000B'X.
:PAGLENL
   VAROFFSET=    2
   VARLEN=    1
   VARTYPE=LOWHIGH
   VARMAX=  255
   ADJUST= 0
   DATA ='1B4342'X.
:FWDRMOV
   VAROFFSET=    2
   VARLEN=    2
   VARTYPE=LOWHIGH
   VARMAX=32767
   ADJUST= 0
   CNVNUM=   12
   CNVDEN=    1
   DATA ='1B640F00'X.
:BCKRMOV
   VAROFFSET=    2
   VARLEN=    2
   VARTYPE=LOWHIGH
   VARMAX=32767
   ADJUST= 0
   CNVNUM=   12
   CNVDEN=    1
   DATA ='1B650F00'X.
:STRBOLD
   DATA ='1B45'X.
:ENDBOLD
   DATA ='1B46'X.
:PRTFEED
   FEEDTYPE=MANUAL
   DATA ='1B5B460300010100'X.
:DWRSLT
   DRAWER=DRAWER1
   DATA ='1B5B460300030101'X.
:DWRSLT
   DRAWER=DRAWER2
   DATA ='1B5B460300030102'X.
:DWRSLT
   DRAWER=ENVELOPE
   DATA ='1B5B460300030200'X.
:PRTORIENT
   ORIENT=PORTRAIT
   DATA ='1B266C304F'X.
:PRTORIENT
   ORIENT=LANDSCAPE
   DATA ='1B266C314F'X.
:STRUS
   DATA ='1B2D01'X.
:ENDUS
   DATA ='1B2D00'X.
:STRSUPS
   DATA ='1B5300'X.
:ENDSUPS
   DATA ='1B54'X.
:STRSUBS
   DATA ='1B5301'X.
:ENDSUBS
   DATA ='1B54'X.
:INITPRT
   DATA ='1B5B4B030000031011B5B5C0200
         90001B5B5405000000035200'X.
      .
      .
      .
```

Step 4: Creating the Workstation Customizing Object

After you change and save the source, create the workstation customizing object using the Create Work Station Customizing Object (CRTWSCST) command, specify the following parameter values:

Workstation customizing object
WSCST(CST3477P)

Library LIB(CSTLIB)

Source member
SRCMBR(CST3477P)

Source file
SRCFILE(CSTSRC)

Library LIB(CSTLIB)

Text TEXT('Workstation customizing object for 3477 with ASCII printer')

Note: This step is performed several times.

Step 5: Varying On the Device

Finally, you change the device description for the 3477 Model H display and specify your customizing object CST3477P for the workstation customizing object (WSCST) parameter.

To activate the workstation customizing function, vary off and then vary on the display so that the customizing object is downloaded to the workstation controller and then to the display. Then print the test document on the Hewlett-Packard printer. As shown in Figure 11-5 on page 11-19, the document is printed showing the characteristics you customized for, as well as the removal of the strange characters that were appearing at the top of each printed page.

Note: This step is performed several times.

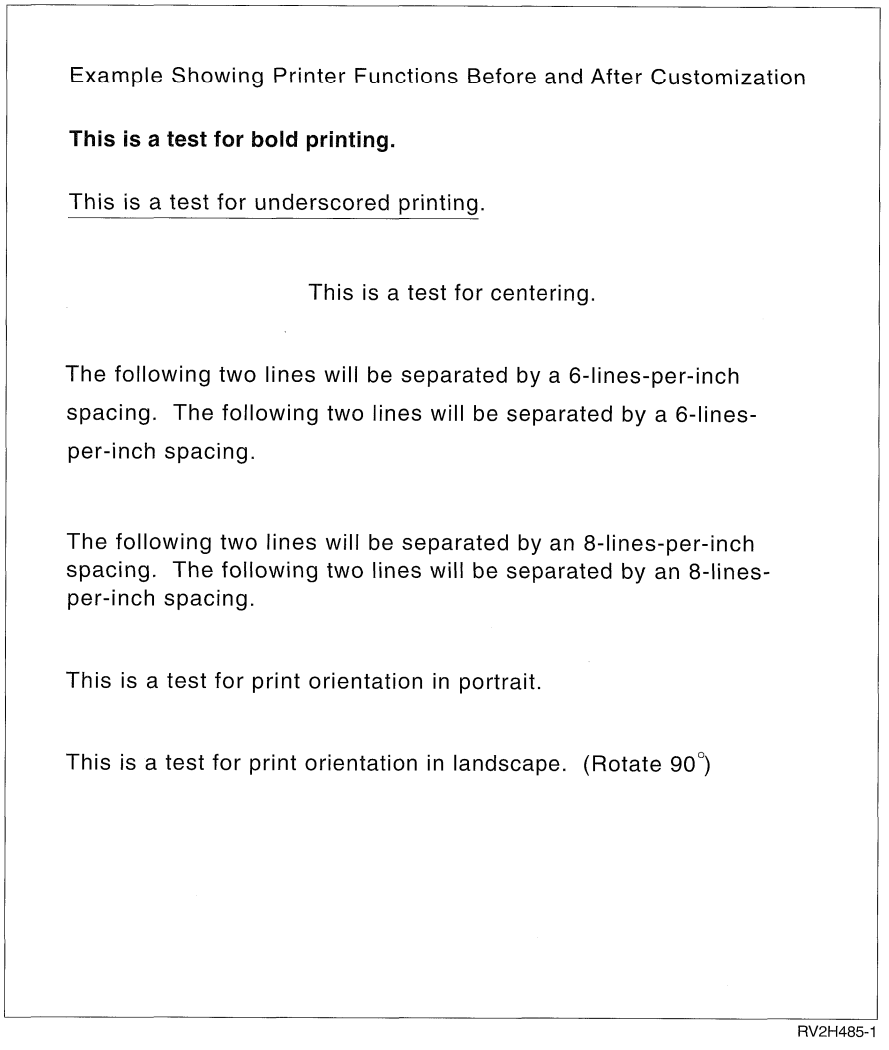


Figure 11-5. Final OfficeVision/400 Printer Test Document

Chapter 12. Customizing ASCII Printers That Use the Emulator on the Controller

This chapter describes the mapping tables and the tags you use to customize the support for an ASCII printer directly attached to the AS/400 system by way of the ASCII workstation controller. This ASCII printer does not use the host print transform function.

The workstation customizing functions for directly attached ASCII printers provide you with the following capabilities:

- Allow you to customize the functional characteristics of a supported ASCII printer
- Allow you to customize the functional characteristics and specify all the necessary parameters required to support a normally unsupported ASCII printer.

To perform these customizations, the OS/400 licensed program provides a way to change one or more of the various mapping tables used by the ASCII workstation controller to support an ASCII printer.

Supported ASCII Printers

Many IBM ASCII printers are supported by the workstation customizing functions. If the printer and the ASCII workstation controller support a character or function, you can add to or change the printing characteristics of these printers. All you need to do is retrieve the workstation customizing source for your ASCII printer type and then add the printer function tag and the corresponding hexadecimal value (or change the existing hexadecimal value).

The workstation customizing procedure for directly attached ASCII printers may involve some trial and error. You may need to add or change more than one tag to add or change the printer functions. If you have a lot of additions or changes to make, you may want to do this by making one or two changes to the source at a time, creating the customizing object, and then testing your changes. Doing this can help you avoid compiling errors and unpredictable results from the printer.

If you are customizing a supported ASCII printer, you can skip the next section and go on to the technical overview about the ASCII printer mapping tables. The remaining sections in this chapter provide more detailed technical information about the printer mapping tables and the tags you can use to customize your printer.

Customizing Unsupported ASCII Printers

When you want to customize an ASCII printer that is not currently supported by IBM, you need to retrieve a source file member based on a supported IBM device type and language type. If your ASCII printer is not an IBM printer and it provides a function for emulating an IBM printer, use this setting. Be sure to specify the emulated printer in the device description for the printer.

To find out which IBM device type is most like your unsupported printer, you need to answer the following questions:

- What printer functions or characteristics and national characters do I want this printer to support?

  Write these down so that you can answer the next question.

- Does the printer itself support the functions I need?

  Check the reference manual to determine this. If the printer cannot support the functions you need, you cannot customize the printer to provide these functions.

- Does the printer emulate or support the emulation of an IBM printer?

  If so, set it up to use the emulation because it could make your customizing process easier.

- Is it a laser printer or a line/listing printer?

  Laser printers are more complex and can be somewhat more difficult to customize. The customization may require more time than you expect.

- Which supported ASCII printer has similar print characteristics to my unsupported ASCII printer?

  To find out more about the characteristics of the printers supported by IBM and the AS/400 system, you can do one or more of the following:

  - Check the reference manuals for the supported printers. These may be available through your technical support specialist or marketing representative. You may already be using one or more of the supported printers and have a manual on hand, or you could contact the manufacturer of the supported printer and ask for a list of the print characteristics or the manual for the supported printer.
  - The print characteristics for many of the ASCII printers supported by the AS/400 system are listed in the *Guide to Programming for Printing*.

When you know or have an idea which supported printer your printer is most like, you can use the device type for the supported printer to retrieve a workstation customizing source file member to be the basis of your workstation customizing object.

After you have answered the previous questions, you also need to do the following:

- Set up all the necessary hardware to connect the printer to the ASCII workstation controller

  This is usually a serial connection.

- Set up any programmable features provided by the printer

  This is the time to set up IBM emulation (if supported), as well as any switches and other parameters that affect the printer.

- Create the necessary controller and device descriptions if needed

  The controller description is for the ASCII workstation controller and may already exist. The device description is for the ASCII printer.

After you have set up and turned on the printer, try to print a test document using one of your usual applications, such as OfficeVision/400. This is the starting point for workstation customizing.

The workstation customizing procedure for directly attached ASCII printers that are not currently supported can involve a lot of trial-and-error. For an unsupported printer, you usually have a lot of additions or changes to make. To avoid compiling errors, you should do this by making one or two changes to the source at a time and then creating the workstation customizing object. Test the object (vary off the printer, specify the workstation customizing object in the device description, vary the printer back on, and print a test document) to see if the results are what you expected. If so, then make the next changes; otherwise, go back to your source and the printer reference manual to determine the cause of any problems with these changes before going further with the workstation customizing procedure.

The remaining sections in this chapter provide more detailed technical information about the printer mapping tables and the tags you can use to customize your printer.

Mapping Tables for ASCII Printers

The ASCII workstation controller uses three tables to map commands and data to an ASCII format for a given printer. These tables are:

- Default Printer EBCDIC-to-ASCII Mapping Table
- Printer Function Table
- Printer Multilanguage EBCDIC-to-ASCII Mapping Table

Default EBCDIC-to-ASCII Mapping Table

The default EBCDIC-to-ASCII mapping table converts an EBCDIC character specified in an application data stream into a single ASCII character code value (for that same character).

The EBCDIC-to-ASCII mapping table used for a given printer depends on the device type and the language configured for the printer. Different types of printers support different ASCII code pages. Also, certain ASCII printers are capable of supporting different national-language-dependent ASCII code pages (the code page in use at any time is determined by a command to select the ASCII code page). The following list shows the values and value ranges you can enter in the EBCDIC-to-ASCII mapping table.

00	Nonprintable character
01–1E	Converts the ASCII code to a multiple-byte code
20–FF	Character data

Note: See "ASCII Control Code Mapping (ASCIICTL) Tag" on page 12-11 for more information about using the values in the range 01-1E for converting the ASCII code to a multiple-byte code. The AS/400 system provides a separate EBCDIC-to-ASCII mapping table for each unique combination of EBCDIC code page and ASCII character set based on the device type and the language configured for the device. The following example shows the source retrieved for the default EBCDIC-to-ASCII mapping table for an IBM 4219 printer that is configured with a language type of USI.

```
      .
      .
      .
:PDFTMAPTBL.
  :PDFTEBCTBL
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '2020838485A0C68687A45B2E3C282B21'X      /* 4- */
     '268288898AA18C8B8DE15D242A293B5E'X      /* 5- */
     '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
     '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
     '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
     'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
     'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
     'BD9CBE169FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
     '7B41424344454647 48492D939495A2E4'X     /* C- */
     '7D4A4B4C4D4E4F505152FB968197A398'X      /* D- */
     '5C20535455565758595AFDE299E3E0E5'X      /* E- */
     '3031323334353637 3839FCEA9AEBE900'X.    /* F- */
  :PFNTWTH
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '0A0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
     '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
     '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
     '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
     '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
     '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
     '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
     '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
     '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
     '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
     '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
     '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */
:EPDFTMAPTBL.
      .
      .
      .
```

Figure 12-1. Source for the Default EBCDIC-to-ASCII Mapping Table

This table cannot be used to map EBCDIC characters into ASCII data when an application data stream being sent to the printer contains a command indicating that a different EBCDIC code page should be used for character conversion. When this occurs, the ASCII workstation controller uses the EBCDIC-to-ASCII mapping data contained in the multilanguage EBCDIC-to-ASCII mapping table. The controller goes back to using the default EBCDIC-to-ASCII mapping table if a command is received to set the printer to initial conditions or when the printer is varied off and then varied back on.

The PFNTWTH (font width) portion of the table contains the font width information for each EBCDIC character being mapped into ASCII code. Font widths are normally specified in 1/120-inch units. The only exception to this occurs when you specify the PRTCTL (print control flag) tag with the value, '80'X. When you do this, the font widths should be specified in 1/1440-inch units.

The ASCII workstation controller uses the font width data to calculate the character width when proportional spacing mode is used. For example, the value 20 in the EBCDIC-to-ASCII mapping table in Figure 12-1 is mapped to a font width of '0A'X. The ASCII character for the value 20 is the space character (blank). This mapping indicates that when the printer is printing in proportional space mode, the space character (blank) is (10 x 1/120) = 1/12 inch wide.

ASCII Printer Function Table

The following are among the printer characteristics and capabilities that you can specify within a printer function table:

- Line spacing
- Pitch (characters per inch)
- Form length
- Highlighting characteristics (bold, underscore)
- Draft, letter, or text quality printing
- Paper feed capabilities
- Subscripting and superscripting
- Initialization and reset sequences
- Selecting character sets
- Selecting fonts
- Relative forward and backward movement
- Margins
- Font quality
- Type styles

More information about the content and format of the ASCII printer function table is available in "Printer Function Tags" on page 12-8.

Printer Multilanguage EBCDIC-to-ASCII Mapping Table

The multilanguage EBCDIC-to-ASCII mapping table contains EBCDIC-to-ASCII mapping information for all the EBCDIC code pages that an application sending data to an ASCII printer can select. The ASCII workstation controller uses information in this table (rather than information in the default EBCDIC-to-ASCII table) when the data stream sent to the printer contains a command to use a different code page.

As shown in the following figure, the multilanguage EBCDIC-to-ASCII mapping table is actually a series of EBCDIC-to-ASCII mapping tables, each one in the series having the same format as a default EBCDIC-to-ASCII mapping table (consisting of character mapping and font width data).

```
      .
      .
      .
:PMLGMAPTBL.

  :PMLGEBCTBL
    EBCDICCP=   29
    ASCIICP=  850
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '2020838485A0C68687A4E82E3C282B21'X      /* 4- */
     '268288898AA18C8B8DE192242A293B99'X      /* 5- */
     '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
     '9B90D2D3D4D6D7D8DED03A23D1273D22'X      /* 7- */
     '9D6162636465666676869AEAF60EC7BF1'X     /* 8- */
     'F86A6B6C6D6E6F707172A6A77DF75DCF'X      /* 9- */
     'E6947374757677787797AADA840ED5BA9'X     /* A- */
     'BD9CBEFAB8F5F4ACABF3AA7CEEF95C9E'X      /* B- */
     'E7414243444546474849F0937E95A2E4'X      /* C- */
     '914A4B4C4D4E4F505152FB968197A398'X      /* D- */
     'EFF65354555657585 95AFDE25EE3E0E5'X     /* E- */
     '3031323334353637 3839FCEA9AEBE900'X.    /* F- */
  :PFNTWTH
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '0A0A0A0A0A0A0A0A0C0E0A0A0A0A0A'X        /* 4- */
     '0C0A0A0A0A060606060C0E0A0A0A0A0E'X      /* 5- */
     '0A0A0E0E0E0E0E0E0E0A0A0A0A0A0A'X        /* 6- */
     '0A0C0C0C0C080808080C0A0A0E060A0A'X      /* 7- */
     '0E0A0C0A0C0A080C0C060E0E0A0C0A0A'X      /* 8- */
     '0A060C060E0C0A0C0C0A0A0A0A0A0A0A'X      /* 9- */
     '0C0A0A080C0C0E0C0C0A0A0A0A0E0A0A'X      /* A- */
     '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
     '0C0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
     '0E0A0E0C0E0E0E0C0E0E060C0C0C0C0C0C'X    /* D- */
     '0A0A0C0E0E0E0E0E0E0C0A0E0A0E0E0E'X      /* E- */
     '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

  :PMLGEBCTBL
    EBCDICCP=   30
    ASCIICP=  853
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '2020838485A000917BA4802E3C282B21'X      /* 4- */
     '268288898AA18C8B8DE1A6982A293B5E'X      /* 5- */
     '2D2FB686B7B500925BA5AD2C255F3E3F'X      /* 6- */
     '0090D2D3D4D6D7D8DED53A99B8273D9A'X      /* 7- */
     'F461626364656667686 9E886C7ED007C'X     /* 8- */
     'F86A6B6C6D6E6F707172A99B9FF700CF'X      /* 9- */
     'E6947374757677787797AE78FC6EC0040'X     /* A- */
     'FA9CBE7DBD155D00AB24A89DACF9EF9E'X      /* B- */
     '8741424344454647484 9F0937E95A2E5'X     /* C- */
     'A74A4B4C4D4E4F50515260965C81A300'X      /* D- */
     '81F6535455565758595AFDE223E3E0E4'X      /* E- */
     '3031323334353637 3839FCEA22EBE900'X.    /* F- */
  :PFNTWTH
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '0A0A0A0A0A0A0A0A0C0E0A0A0A0A0A'X        /* 4- */
     '0C0A0A0A0A060606060C0E080A0A0A0A'X      /* 5- */
     '0A0A0E0E0E0E0E0E0A0E0A0A0A0A0A'X        /* 6- */
     '0A0C0C0C0C080808080606 0A0E0C060A0E'X   /* 7- */
     '0A0A0C0A0C0A080C0C060C0A0A0C0C0A'X      /* 8- */
     '0A060C060E0C0A0C0C0A0C0C060A0E0A'X      /* 9- */
     '0C0A0A080C0C0E0C0C0A0E0E0C0E0E0A'X      /* A- */
     '0A0A0A0A0C0A0A0A0A0A0E0E0C0E0A0A'X      /* B- */
     '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0C'X      /* C- */
     '0C0A0E0C0E0E0E0C0E0E0A0C0A0C0C0C'X      /* D- */
     '0C0A0C0E0E0E0E0E0E0C0A0E0A0E0E0E'X      /* E- */
     '0A0A0A0A0A0A0A0A0A0A0A0E0A0E0E0A'X.     /* F- */
      .
      .
      .
```

Figure 12-2. Source for the Multilanguage EBCDIC-to-ASCII Mapping Table

Besides containing EBCDIC-to-ASCII mapping and character font width information, the multilanguage table also contains the following:

- ASCII character set ID values that are sent to an ASCII printer when a new ASCII code page is selected (a different ASCII code page may need to be activated in the printer when the application switches to a different EBCDIC code page).
- A table indicating which EBCDIC code pages are valid when selected by the application.

There are separate multilanguage EBCDIC-to-ASCII mapping tables for printers attached to the ASCII workstation controller. Which multilanguage EBCDIC-to-ASCII mapping table the ASCII workstation controller uses for a given printer depends on which multinational ASCII character set is supported by that printer. The following are the main multinational character sets for printers that attach to the ASCII workstation controller:

- Character Set 437 (Personal Computer Character Set 2), used on IBM 4201-2, 4202-1, 4202-2, 4207-1, and 4208-1 printers.
- Character Set 256 (International Number 1 Character Set), used on IBM 4224 printers.
- Character Set 850 (Personal Computer Multinational Character Set), used on IBM 5204, 4201-3, 4202-3, 4207-2, 4208-2, 4216-10, 4234-13, 4019, and 6252-A58 printers.

Determining Which ASCII Printer Tables to Use

| The three ASCII printer tables described in "Mapping Tables for ASCII Printers" on page 12-2 are downloaded to the ASCII workstation controller when a printer is varied on. The particular tables that are downloaded when a device is varied on depend on the type of printer and the language configured for the printer. The system determines which set of tables to download for a given printer by searching a table of tables. This table contains the names of each of the tables that must be used for a given type of printer configured for a particular language.

| When you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command to create your workstation customizing source, you must specify an ASCII printer for the device type. You must also specify the same language type as you did when you created the device description. This provides the correct mapping tables to use as a base and the correct printer functions for your specific ASCII printer.

| If you adding support for a device that is not supported by IBM, specify a device type that is similar to your unsupported device. For example, if you are trying to customize an unsupported ASCII laser printer, you might choose a device type of 4019 on the RTVWSCST command. The 4019 is a supported ASCII laser printer with some similar functional characteristics to non-IBM laser printers.

Use the following list as a guideline to help you determine which tables in your workstation customizing source you want to customize.

- Characters do not appear or are not the characters that are expected.

  To change the characters that are supported for the printer's default code page and character set, you need to change one or more entries in your default EBCDIC-to-ASCII mapping table, which is denoted in your workstation customizing source by the PDFTEBCTBL (EBCDIC-to-ASCII mapping table) tag.

- Adding or changing the print characteristics.

  If you want to add or change the printing characteristics of your ASCII printer, such as the margins, line and page spacing, paper drawer selection, paper feed, paper orientation, and so on, you need to add to or change the existing printer function tags in your source. The printer function table in a retrieved workstation customizing source begins with the PFCNTBL (ASCII printer function table) tag. This tag is followed by a number of individual printer function tags with their parameters and data values that, when compiled, make up the actual printer function table. See "Printer Function Tags" on page 12-8 for more information about the individual printer function tags.

- The data stream from your application contains commands that tell the printer to use a different code page.

  If you know that your application is capable of producing documents that contain commands to use a different code page, you need to change the entries in one or more of the multilanguage EBCDIC-to-ASCII mapping tables that are denoted by the PMLGMAPTBL (multilanguage EBCDIC-to-ASCII mapping table) tag in your source. This tag is followed by the PMLGEBCTBL (EBCDIC-to-ASCII mapping table) tag and the corresponding PFNTWTH (font width mapping table) tag and its entries. There must be at least 30 of these matched pairs of mapping tables in a single workstation customizing source, each containing the EBCDIC-to-ASCII mapping and its corresponding font width mapping for a different code page. See "Multilanguage EBCDIC-to-ASCII Mapping Table (PMLGMAPTBL) Tag" on page 12-7 for more information about these mapping table pairs.

Using the reference manual for the printer and the tags in your retrieved source, you can now change the existing printer mapping tables and add any new, supported print functions you need.

After you have made your changes to the source, you should create the customizing object from the retrieved source using the Create Work Station Customizing Object (CRTWSCST) command. Use the Work with Configuration Status (WRKCFGSTS) command to vary the printer off and then on again. Now, try printing your test document again.

Working with the Tag Language for Directly Attached ASCII Printers

When you specify an ASCII printer for the device type on the Retrieve Work Station Customizing Object Source (RTVWSCST) command, the device class is specified in the source as an ASCII printer (`DEVCLASS = ASCIIPRT`). The source structure for this device class looks like the following:

```
:WSCST DEVCLASS=ASCIIPRT.

 :PDFTMAPTBL. /*default EBCDIC-to-ASCII mapping table*/


   :PDFTEBCTBL with parameters. /*mapping table entries*/

   :PFNTWTH with parameters. /*font width map table entries*/


 :EPDFTMAPTBL.

 :PMLGMAPTBL. /*multilanguage EBCDIC-to-ASCII mapping table*/

   :PMLGEBCTBL with parameters.  (mapping table entries)

   :PFNTWTH with parameters. /*font width map table entries*/

                .
                .
       ( PMLGEBCTBL/PFNTWTH pairs)
                .
                .

   :PMLGEBCTBL with parameters. /*mapping table entries*/

   :PFNTWTH with parameters. /*font width map table entries*/


 :EPMLGMAPTBL.

 :PFCNTBL. /*printer function table*/
              .
              .
      printer function tag   /*tag with parameters*/
              .
              .
              .
              .
:EWSCST.
```

Figure 12-3. Source Structure for ASCII Printers

The source structure provides the basic outline of a source file member retrieved for an ASCII printer.

The primary tags for the ASCII printer source structure are the following:

PDFTMAPTBL	Default EBCDIC-to-ASCII mapping table tag
PMLGMAPTBL	Multilanguage EBCDIC-to-ASCII mapping table tag
PFCNTBL	ASCII printer function table tag

The order and placement of the primary tags in each source file member is strictly enforced by the workstation customizing object compiler. When a primary tag is missing in the source, the system default table associated with the missing

primary tag may be used in place of a customized table. The default table chosen is based on the device type and national language type you specified when you used the RTVWSCST command. When you make changes to your workstation customizing source, you should leave the tags in the order they were in when you retrieved the source. This helps to avoid possible errors and unpredictable results when you create and use the workstation customizing object.

Following the PMLGMAPTBL tag, the number of PMLGEBCTBL (EBCDIC-to-ASCII mapping table) and PFNTWTH (font width mapping table) tag pairs determines the size of the customized multilanguage mapping table. Each tag pair represents one code page to character set mapping table. For a complete description of the PMLGEBCTBL tag, see "EBCDIC-to-ASCII Mapping Table (PMLGEBCTBL) Tag" on page 12-7.

When a PFCNTBL (ASCII printer function table) tag is present in the source, but has no printer function tags following it, the system default printer function table is not used. Instead, no printer functions are mapped when this customizing object is used by the workstation controller. This can cause your printer to produce unpredictable results.

Using the Tags to Customize ASCII Printers

The following sections describe the tags in the source you retrieved for customizing an ASCII printer. If you have not already retrieved the workstation customizing source for customizing an ASCII printer, be sure to specify a supported ASCII printer for the device type when you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command.

Default EBCDIC-to-ASCII Mapping Table (PDFTMAPTBL) Tag

The PDFTMAPTBL (default EBCDIC-to-ASCII mapping table) tag defines a default EBCDIC-to-ASCII mapping table for an ASCII printer. The syntax for this tag is:

```
:PDFTMAPTBL.
```

There are no keyword parameters associated with this tag. However, it must be immediately followed by an PDFTEBCTBL (EBCDIC-to-ASCII mapping table) tag and a PFNTWTH (font width mapping table) tag. This pair of tags indicates that the contents of the tables are to be used for customizing the ASCII printer. An example showing these tags and source for these tables is shown in Figure 12-1 on page 12-3.

End Default EBCDIC-to-ASCII Mapping Table (EPDFTMAPTBL) Tag

The EPDFTMAPTBL (end default EBCDIC-to-ASCII mapping table) tag ends the default EBCDIC-to-ASCII mapping table. This tag follows the PDFTMAPTBL, PDFTEBCTBL, and PFNTWTH tags. The syntax for this tag is:

```
:EPDFTMAPTBL.
```

EBCDIC-to-ASCII Mapping Table (PDFTEBCTBL) Tag

The PDFTEBCTBL (EBCDIC-to-ASCII mapping table) tag defines the mapping table entries for the default EBCDIC-to-ASCII mapping table for an ASCII printer. The PDFTEBCTBL tag must always follow a PDFTMAPTBL tag and it must always be paired with an accompanying PFNTWTH (font width mapping table) tag. The syntax for this tag is:

```
:PDFTEBCTBL
        DATA = table data.
```

DATA
A required parameter. Specifies the hexadecimal EBCDIC-to-ASCII mapping table data that is used to map EBCDIC codes from the AS/400 system to the ASCII code needed by the ASCII printer. The table data must be hexadecimal, and exactly 192 bytes in length.

table data
Hexadecimal values for mapping EBCDIC codes to ASCII codes for the ASCII printer.

Font Width Mapping Table (PFNTWTH) Tag

The PFNTWTH (font width mapping table) tag defines the font width mapping table for an ASCII printer. The font width mapping table tag must follow either a PDFTMAPTBL tag or a PMLGMAPTBL tag. Also, it must always be paired with an accompanying PDFTEBCTBL or PMLGEBCTBL tag. The syntax for this tag is:

```
:PFNTWTH
        DATA = table data.
```

DATA
A required parameter. Specifies the hexadecimal font width mapping codes. The table data must be hexadecimal, and exactly 192 bytes in length.

table data
Hexadecimal values for mapping the necessary font widths for the ASCII printer.

The ASCII workstation controller uses the font width data to calculate the character width when proportional spacing mode is used. Font widths are normally specified in 1/120-inch units. The only exception to this occurs when you specify the PRTCTL (print control flag) tag with the value,

'80'X. When you do this, the font widths should be specified in 1/1440-inch units.

ASCII Printer Function Table (PFCNTBL) Tag

The PFCNTBL (ASCII printer function table) tag defines the printer function table to be used for an ASCII printer. The syntax for this tag is:

```
:PFCNTBL.
```

There are no keyword parameters for this tag. However, it must be immediately followed by a number of individual printer function tags, which make up the function table entries. See "Printer Function Tags" on page 12-8 for more information about the individual printer function tags. The absence of a specific printer function tag following the PFCNTBL tag implies that particular printer function is not mapped. When the same printer function tag occurs multiple times following a PFCNTBL tag, the workstation controller uses the last occurrence of the tag in the source to map the printer function.

Multilanguage EBCDIC-to-ASCII Mapping Table (PMLGMAPTBL) Tag

The PMLGMAPTBL (multilanguage EBCDIC-to-ASCII mapping table) tag defines the multilanguage EBCDIC-to-ASCII mapping table for customizing an ASCII printer. The syntax for this tag is:

```
:PMLGMAPTBL.
```

There are no keywords associated with this tag. However, it is immediately followed by one or more occurrences of a PMLGEBCTBL (EBCDIC-to-ASCII mapping table) tag and a PFNTWTH (font width mapping table) tag. These pairs of tags specify the table contents. The maximum number of pairs allowed is 150. The minimum number of pairs required is 30. See "EBCDIC-to-ASCII Mapping Table (PMLGEBCTBL) Tag" for a description of the PMLGEBCTBL tag and more information on the required pairs.

End Multilanguage EBCDIC-to-ASCII Mapping Table (EPMLGMAPTBL) Tag

The EPMLGMAPTBL (end multilanguage EBCDIC-to-ASCII mapping table) tag ends the multilanguage EBCDIC-to-ASCII mapping table. The syntax for this tag is:

```
:EPMLGMAPTBL.
```

EBCDIC-to-ASCII Mapping Table (PMLGEBCTBL) Tag

The PMLGEBCTBL (EBCDIC-to-ASCII mapping table) tag defines an EBCDIC-to-ASCII mapping table for an ASCII printer. The PMLGEBCTBL tag must follow a PMLGMAPTBL tag. Also, it must always be paired with an accompanying PFNTWTH tag. The syntax for this tag is:

```
:PMLGEBCTBL
        EBCDICCP = code page ID
        ASCIICP = code page ID
        DATA = table data.
```

EBCDICCP
A required parameter. Specifies the EBCDIC code page identifier. The value for this parameter is an integer value, from 1 to 32767 without leading zeros.

code page ID
A 5-digit registered identifier used to specify a particular assignment of code points to graphic characters. The code page ID is the second part of the QCHRID system value or the CHRID parameter value.

ASCIICP
A required parameter. Specifies the ASCII code page identifier. The value for this parameter is an integer value, from 1 to 32767 without leading zeros.

code page ID
A 5-digit registered identifier used to specify a particular assignment of code points to graphic characters. The code page ID is the second part of the QCHRID system value or the CHRID parameter value.

Note: For unsupported printers, this value is tied to the variable type parameter (VARTYPE) on the CODEPAGVAR (code page variable) tag.

DATA
A required parameter. Specifies the hexadecimal table data that is used to map EBCDIC codes from the AS/400 system to the ASCII code needed by the ASCII printer. The table data must be hexadecimal, and exactly 192 bytes in length.

table data
Hexadecimal values for mapping EBCDIC codes to ASCII codes needed by the ASCII printer.

A minimum set of thirty PMLGEBCTBL/PFNTWTH tag pairs is required. This is because the workstation controller uses an offset value to select the code page for a given data stream. One each of the EBCDICCP (code page) values shown in Table 12-1 on page 12-8 must be present on the PMLGEBCTBL tags of this minimum set. There is no requirement for the PMLGEBCTBL tags to be in any particular order, only that the required ones must each exist once, and only once, between the PMLGMAPTBL tag and the EPMLGMAPTBL tag.

Note: The code page values supported by the workstation controller differ from those supported for twinaxial-connected displays, as shown in Table 7-21 on page 7-16. It is strongly recommended that you do not delete the original 30 pairs you find in the retrieved source. Unpredictable results may appear on your printed output. You are free to add as many new pairs of PMLGEBCTBL/PFNTWTH tags as you need.

Table 12-1. Required Code Page Identifiers

EBCDICCP Parameter Value for PMLGEBCTBL Tag	EBCDIC Code Page
037	United States
256	International
273	Austria/Germany
274	Belgium
275	Brazil
276	Canadian French
277	Denmark/Norway
278	Finland/Sweden
280	Italy
281	Japan English
282	Portugal
283	Spain
284	Spanish Speaking
285	United Kingdom
290	Japan Katakana
297	France
330	Languages of the former Yugoslavia
340	OCR
420	Arabic
423	Greece
424	Hebrew
500	International
870	ROECE/Latin 2
871	Icelandic
880	ROECE/Cyrillic
892	OCR-A
893	OCR-B
905	Turkey
259	Map to ASCII character set 899 **Note:** You must have the ASCIICP set to 899 for the PMLGEBCTBL tag.
259	Map to ASCII character set 850, or ASCII character set 437 **Note:** You must have set the ASCIICP set to 850 or 437 for the PMLGEBCTBL tag.

Printer Function Tags

These tags allow you to specify the ASCII control sequence for an individual printer function. The tags for these functions must follow a PFCNTBL (ASCII printer function table) tag in your source.

Most of the printer function tags fall into one of following general formats.

- Tags with only the data parameter
- Tags that provide additional parameters to describe variables within functions
- Tags for functions that support relative movement combined with the first two tag formats

The remaining printer functions that are described are functions with specialized parameters that do not meet any of the general formats.

Syntax for Printer Function Tags with the Data Parameter Only:

The following table lists the tags that have only a data parameter that you can use to set or change printer characteristics.

Table 12-2 (Page 1 of 2). Printer Function Tags with Data Parameter Only

Tag	Description
BELL	Bell
BSP	Backspace
CARRTN	Carrier Return
ENDBOLD	End Bold Printing
ENDSUBS	End Subscript
ENDSUPS	End Superscript
ENDUS	End Underscore
FNTGPFT	Set Global Font for PFT
FORMFEED	Form Feed
INITPRT	Initialize Printer
INITVON	Initialization at Vary On
LINEFEED	Line Feed
PRTCTL	Printer Control Flags Valid values are: '00'X No flags set '01'X Printer response allowed '02'X Page length change in middle of page not allowed '30'X Inhibit computer output reduction (COR) mode '80'X Font widths in 1/1440-inch units
RVSIDX	Reverse Index
SPACE	Space
STRBOLD	Start Bold Printing
STRPROP	Start Proportional Space Mode

Table 12-2 (Page 2 of 2). Printer Function Tags with Data Parameter Only	
Tag	**Description**
STRSUBS	Start Subscript
STRSUPS	Start Superscript
STRUS	Start Underscore

The general syntax for a printer function tag that uses only the data parameter is:

```
:xxxxx
        DATA = ASCII control sequence.
```

DATA
A required parameter. Specifies the ASCII control sequence for the printer function. This data must be coded as a hexadecimal value. The maximum length of this value is 240 bytes.

ASCII control sequence
Hexadecimal values for mapping the ASCII control sequence for the printer function.

The FNTGPFT (set global font for PFT) tag allows you to set the base font for printing. It is this font that is used for all printed characters until the printer receives a command to change to another font.

The printer control flags (PRTCTL tag) are bit-mapped flags that allow you to control certain print characteristics, such as the font width units for the FNTGPFT (set global font) tag. The hexadecimal values for the PRTCTL tag that are defined in Table 12-2 on page 12-8 can all be specified on a single occurrence of the tag. To do this you can use a logical OR operation to combine the values. For example, when you combine the values '01'X. and '02'X. using a logical OR operation, the result is '03'X. Specifying '03'X. for the PRTCTL tag allows you to set both the flag for allowing a printer response and the flag for not allowing a page length change in mid-page.

Syntax for Printer Function Tags with a Variable

The following table lists the tags that you can use to set or change printer characteristics that involve a variable.

Table 12-3. Printer Function Tags with a Variable	
Tag	**Description**
CODPAGVAR	Set Code Page
FNTTYPE	Font Type
PRISPCM	Set Primary Spacing Mode
PRISTYLE	Set Primary Style

The general syntax for a printer function tag with a variable is:

```
:xxxxx
        VAROFFSET = variable offset in
                    control sequence
        VARLEN - variable length
        VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
                  CHRHEX|CHRAN
        DATA = ASCII control sequence.
```

VAROFFSET
A required parameter. Defines the offset of the variable in the control sequence. This value must be an integer.

Note: This offset is relative to the beginning of the control sequence. Therefore, a value of zero (0) implies that the variable starts in the first byte of the control sequence.

VARLEN
A required parameter. Defines the length of the variable. This value must be an integer.

VARTYPE
A required parameter. Defines the type of variable. The choices are:

HIGHLOW
The byte order of the variable is in high-low order.

LOWHIGH
The byte order of the variable is in low-high order.

CHRDEC
The variable is in characters and there is no byte order consideration. All characters are in the range from 0 to 9.

CHRHEX
The variable is in characters and there is no byte order consideration. All characters are in the range from 0 to 9, A to F.

CHRAN
The variable is in characters and there is no byte order consideration. All characters are in the range from 0 to 9, A to Z.

DATA
A required parameter. Specifies the ASCII control sequence for the printer function. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

ASCII control sequence
Hexadecimal values for mapping the ASCII control sequence for the printer function.

The PRISPCM (primary spacing mode) tag and the PRISTYLE (set primary style) tag, in combination with the PRICHRH (set primary character height) tag are used for font selection when an ASCII printer does not support global fonts or provide font quality and characters per inch (CPI) support.

Setting the primary spacing mode allows you to indicate whether the spacing should be fixed or proportional.

Primary style is related to the font map identifier. When the value of a font mapping table entry is greater than 255, the high order byte of the font map identifier is used to set the print to a particular style within the specified font. For example, when an ASCII printer is set up to print in a Gothic font, the primary style might be italic or upright print within the Gothic font.

Syntax for Printer Function Tags with Variable and Relative Movement

The following table lists the tags that you can use to set or change printer characteristics that involve a variable and some relative movement.

Table 12-4. Printer Function Tags with Variable and Relative Movement

Tag	Description
BCKRMOV	Backward Relative Movement
FWDRMOV	Forward Relative Movement
PAGLENI	Set Page Length in Inches
PRICHRH	Set Primary Character Height
VARCPI	Variable Characters Per Inch
VARLSPC	Variable Line Spacing

The general syntax for a printer function tag with a variable is:

```
:xxxxxxxxxx
        VAROFFSET = variable offset in
                    control sequence
        VARLEN = variable length
        VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
                  CHRHEX|CHRAN
          VARMAX = maximum variable value
          ADJUST = adjustment
          CNVNUM = conversion ratio numerator
          CNVDEN = conversion ratio denominator
          DATA = ASCII control sequence.
```

VAROFFSET
A required parameter. Defines the offset position of the variable in the control sequence. This value must be an integer.

Note: This offset position is relative to the beginning of the control sequence. Therefore, a value of zero (0) implies that the variable starts in the first byte of the control sequence.

VARLEN
A required parameter. Defines the length of the variable. This value must be an integer.

VARTYPE
A required parameter. Defines the type of variable used with this category of printer function tags.

HIGHLOW
The byte order of the variable is in high-low order.

LOWHIGH
The byte order of the variable is in low-high order.

CHRDEC
The variable is in characters and there is no byte order consideration. All characters are in the range from 0 to 9.

CHRHEX
The variable is in characters and there is no byte order consideration. All characters are in the range from 0 to 9, A to F.

CHRAN
The variable is in characters and there is no byte order consideration. All characters are in the range from 0 to 9, A to Z.

VARMAX
A required parameter. Defines the maximum variable value. This value must be an integer.

ADJUST
A required parameter. Defines the adjustment value for the variable. This value must be a signed integer.

CNVNUM
A required parameter. Defines the numerator of the conversion ratio. This value must be an integer.

CNVDEN
A required parameter. Defines the denominator of the conversion ratio. This value must be a nonzero integer.

DATA
A required parameter. Specifies the ASCII control sequence for the printer function. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

ASCII control sequence
Hexadecimal values for mapping the ASCII control sequence for the printer function.

For variable line spacing (VARLSPC tag), the conversion ratio specified converts line spacing parameter values in the application data stream from units of 1/1440 inch to whatever units are used by the ASCII printer. For example, if the line spacing units used by an ASCII printer in its variable spacing command are in units of 1/72 inch, then the conversion ratio specified in the command header should be 20.

For setting the page length, the ASCII workstation controller only recognizes the first occurrence of the PAGLENI tag. If more than one of these tags is specified, the values in the first occurrence are used and all other occurrences are ignored.

ASCII Control Code Mapping (ASCIICTL) Tag

The ASCIICTL (ASCII control code mapping) tag defines the ASCII control sequence for an ASCII control code. The ASCIICTL tag must follow the PFCNTBL tag in your source. The syntax for this tag is:

```
:ASCIICTL
        ASCII = control code
        DATA = ASCII control sequence.
```

ASCII
A required parameter. Specifies an ASCII control code. This must be a hexadecimal value from '01'X to '1E'X.

DATA
A required parameter. Specifies the ASCII control sequence for the ASCII control code you want to map. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

ASCII control sequence
Hexadecimal value for mapping an ASCII control sequence to an ASCII control code.

Specifying the ASCIICTL tag allows you to convert a single-byte character code to a multiple-byte sequence. For example, you could use this code to have the printer print control characters, such as © and §, that are not normally in its range of printable characters. You could also use this tag to allow the printer to change to a different code page to print a single character and then change back to the original code page.

Collate Width (COLLATE) Tag

The COLLATE (collate width) tag sets the collate width. Collate width is the printer offset between overstruck characters when a type-over method is used to create bold highlighting for printed data. The printer offset referred to here is the offset between the first and second character when the printer reverses the print direction to print over previously printed characters and there is a slight space between them. The COLLATE tag must follow the PFCNTBL tag in your source. The syntax for this tag is:

```
:COLLATE
        DATA = collate width.
```

DATA
A required parameter. Defines the collate width. This value must be an integer denoting the number of 1/1440-inch increments.

Collate width
Integer value for defining the collate width to the ASCII printer.

Set Characters per Inch (CPI) Tag

The CPI (set characters per inch) tag defines the control sequence for setting the number of characters per inch. The CPI tag must follow the PFCNTBL tag in your source. You can specify this tag more than one time in a workstation customizing source when you specify different hexadecimal values for the different spacings. The syntax for this tag is:

```
:CPI
        CPI = 5|855|10|12|133|15|
              171|20|27
        DATA = ASCII control sequence.
```

Note: Use the STRPROP tag for proportional space mode.

CPI
A required parameter. Defines the number of characters per inch. Possible values include:

5	5 characters per inch
855	8.55 characters per inch
10	10 characters per inch
12	12 characters per inch
133	13.3 characters per inch
15	15 characters per inch
171	17.1 characters per inch
20	20 characters per inch
27	27 characters per inch

DATA
A required parameter. Specifies the ASCII control sequence for setting the number of characters per inch. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

ASCII control sequence
Hexadecimal value for mapping an ASCII control sequence to define the number of characters that are to be printed per inch of paper

Default Font ID (DFTFNTID) Tag

The DFTFNTID (default font ID) tag defines the ASCII printer function table default font IDs for different fonts. The DFTFNTID tag must follow the PFCNTBL tag in your source. The syntax for this tag is:

```
:DFTFNTID
        CPI = 5|855|10|12|133|15|171|
              20|27|PROP
        DEFAULT = font ID.
```

CPI
A required parameter. Defines the font in characters per inch using the values shown in the following list:

5	5 characters per inch
855	8.55 characters per inch
10	10 characters per inch
12	12 characters per inch
133	13.3 characters per inch
15	15 characters per inch

171 17.1 characters per inch
20 20 characters per inch
27 27 characters per inch
PROP Proportional spacing

DEFAULT
A required parameter. Specifies the default font ID. This font ID must be one that is recognized by the AS/400 system. This value must be an integer.

Drawer Selection Tag (DWRSLT)

The DWRSLT (drawer selection) tag defines the control sequence for drawer selection. A drawer selection defines the source of paper to be used for printing. The DWRSLT tag must follow the PFCNTBL tag in your source. The syntax for this tag is:

```
:DWRSLT
        DRAWER = ENVELOPE|DRAWER1|DRAWER2
        DATA = ASCII control sequence.
```

DRAWER
A required parameter. Defines the drawer from which paper is to be selected.

ENVELOPE
Selects the envelope drawer as the source of paper for documents that are to be printed.

DRAWER1
Selects drawer 1 as the source of paper for documents that are to be printed.

DRAWER2
Selects drawer 2 as the source of paper for documents that are to be printed.

DATA
A required parameter. Specifies the ASCII control sequence for the drawer selection function. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

ASCII control sequence
Hexadecimal value for mapping an ASCII control sequence to define the drawer selection for the ASCII printer.

Font ID Mapping (FNTMAP) Tag

The FNTMAP (font ID mapping) tag defines the ASCII printer function table font ID mappings. The FNTMAP tag must follow the PFCNTBL tag in your source. It must also be followed by one or more FNTMAPE (font mapping table entry) tags containing the font mapping pairs. These pairs are immediately followed by an EFNTMAP (end font ID mapping) tag. The syntax for the FNTMAP tag is:

```
:FNTMAP
        CPI = 5|855|10|12|133|15|171|
              20|27|PROP
        DEFAULT = default font id.
```

CPI
A required parameter. Defines the font in characters per inch. Possible values include:

5 5 characters per inch
855 8.55 characters per inch
10 10 characters per inch
12 12 characters per inch
133 13.3 characters per inch
15 15 characters per inch
171 17.1 characters per inch
20 20 characters per inch
27 27 characters per inch
PROP Proportional spacing

DEFAULT
A required parameter. Defines the default mapped font ID. This value must be an integer. (See the reference manual for your device to find this value.)

End Font ID Mapping (EFNTMAP) Tag

The EFNTMAP (end font ID mapping) tag ends the ASCII printer function table font ID mappings. The syntax for this tag is:

```
:EFNTMAP.
```

Font Mapping Table Entry (FNTMAPE) Tag

The FNTMAPE (font mapping table entry) tag defines one pair of font ID mappings, which are placed into the ASCII printer function table font ID mapping table. One or more of these tags follow a FNTMAP (font ID mapping) tag, and a group of these tags must be followed by an EFNTMAP (end font ID mapping) tag. The syntax for this tag is:

```
:FNTMAPE
        IBMFNT = font ID
        ASCIIFNT = type style.
```

IBMFNT
A required parameter. Specifies an IBM printer font ID. This value must be an integer.

ASCIIFNT
A required parameter. Specifies an ASCII type style. This value must be an integer.

Font Quality (FONTQLTY) Tag

The FONTQLTY (font quality) tag defines the control sequence for setting different font qualities. The FONTQLTY tag must follow the PFCNTBL tag in your source. The syntax for this tag is:

```
:FONTQLTY
        FONTCPI = 10|12|15|171|PROP
        QLTYTYPE = DRAFT|LETTER|TEXT
        DATA = ASCII control sequence.
```

FONTCPI
A required parameter. Defines the type of font in characters per inch.

10	10 characters per inch
12	12 characters per inch
15	15 characters per inch
171	17.1 characters per inch
PROP	Proportional spacing

Note: You cannot specify both 15 and 17.1 for the same ASCII printer.

QLTYTYPE
A required parameter, and Defines the quality of the printing for the ASCII printer.

DRAFT
The print quality is draft quality

LETTER
The print quality is letter quality

TEXT
The print quality is text quality

DATA
A required parameter. Specifies the ASCII control sequence for the font quality of the ASCII printer. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

ASCII control sequence
Hexadecimal value for mapping an ASCII control sequence to define the font quality for the ASCII printer.

Set Margin (MARGIN) Tag

The MARGIN (set margin) tag allows you to set the margin size for different types of paper and paper orientation. The MARGIN tag must follow the PFCNTBL tag in your source. The syntax for this tag is:

```
:MARGIN
        OPTION = TOP|LEFT|RIGHT|BOTTOM
        ORIENT = PORTRAIT|LANDSCAPE
        DATA = margin size.
```

OPTION
A required parameter. Defines the margin type.

TOP
Set the margin for the top of the page

LEFT
Set the margin for the left side of the page

RIGHT
Set the margin for the right side of the page

BOTTOM
Set the margin for the bottom of the page

ORIENT
A required parameter. Defines the type of paper orientation.

PORTRAIT
The paper is vertically aligned in the printer

LANDSCAPE
The paper is horizontally aligned in the printer

DATA
A required parameter. Specifies the size of the margin and denotes the number of 1/1440-inch increments. This value must be an integer.

data
Integer value for the margin size

Set Page Length in Lines (PAGLENL) Tag

The PAGLENL (set page length in lines) tag sets the page length in terms of the number of lines per page. The number of lines is carried as a variable in the control sequence for page length. The PAGLENL tag must follow the PFCNTBL tag in your source. The syntax for this tag is:

```
:PAGLENL
        VAROFFSET = variable offset in
                    control sequence
        VARLEN = variable length
        VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
                  CHRHEX|CHRAN
        VARMAX = maximum variable value
        ADJUST = adjustment
        DATA = ASCII control sequence.
```

VAROFFSET
A required parameter. Defines the offset of the variable in the control sequence. This value must be an integer.

Note: This offset is relative to the beginning of the control sequence. Therefore, a value of zero (0) implies that the variable starts in the first byte of the control sequence.

VARLEN
A required parameter. Defines the length of the variable. This value must be an integer.

VARTYPE
A required parameter. This attribute defines the type of variable. Possible values are:

HIGHLOW
The byte order of the variable is in high-low order.

LOWHIGH
The byte order of the variable is in low-high order.

CHRDEC
The variable is in characters and there is no byte order consideration. All characters are in the range 0 to 9.

CHRHEX
The variable is in characters and there is no byte order consideration. All characters are in the range 0 to 9, A to F.

CHRAN
The variable is in characters and there is no byte order consideration. All characters are in the range 0 to 9, A to Z.

VARMAX
A required parameter. Defines the maximum variable value. This value must be an integer.

ADJUST
A required parameter. Defines the adjustment value to the variable. This value must be a signed integer.

DATA
A required parameter. Specifies the ASCII control sequence for the printer function. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

ASCII control sequence
Hexadecimal values for mapping the ASCII control sequence for the printer function.

Page Size for Printer Function Table (PAGSIZPFT) Tag

The PAGSIZPFT (page size for printer function table (PFT)) tag defines the page size for different types of paper. The PAGSIZPFT tag must follow the PFCNTBL tag in your source. The syntax for this tag is:

```
:PAGSIZPFT
        PAGWTH = page width
        PAGLEN = page length
        PAPER = MANUAL|DRAWER1|DRAWER2|
                ENVELOPE
```

PAGWTH
A required parameter. Defines the width of the page in the printer. This value must be an integer denoting the number of 1/1440-inch increments.

PAGLEN
A required parameter. Defines the length of the page in the printer. This value must be an integer denoting the number of 1/1440-inch increments.

PAPER
A required parameter. Defines the type of paper to be used by the printer.

MANUAL
Paper is fed to the printer manually (by hand).

DRAWER1
Paper is fed to the printer from the first drawer.

DRAWER2
Paper is fed to the printer from the second drawer.

ENVELOPE
Paper is fed to the printer from the envelope drawer.

Paper Feed (PRTFEED) Tag

The PRTFEED (paper feed) tag defines the control sequence for feeding different types of paper to the ASCII printer. The PRTFEED tag must follow the PFCNTBL tag in your source. Most printers do not support all the paper feed characteristics that can be specified using the parameters for this tag. If your printer does not support a certain characteristic, you should set the corresponding parameter value to '00'X. (zero).

The syntax for this tag is:

```
:PRTFEED
        FEEDTYPE = EJECT|MANUAL|AUTO
        DATA = ASCII control sequence.
```

FEEDTYPE
A required parameter. Defines the type of paper feeding.

EJECT
The paper is ejected from the printer

MANUAL
The paper is fed to the printer manually (by hand)

AUTO
The paper is fed to the printer automatically

DATA
A required parameter. Specifies the ASCII control sequence for the function. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

ASCII control sequence
Hexadecimal value for mapping an ASCII control sequence to define the way paper is fed to the printer.

Paper Orientation (PRTORIENT) Tag

The PRTORIENT (paper orientation) tag defines the control sequence for setting different paper orientations, such as portrait and landscape (rotate 90 degrees). The PRTORIENT tag must follow the PFCNTBL tag in your source. The syntax for this tag is:

```
:PRTORIENT
        ORIENT = PORTRAIT|LANDSCAPE|
                 RTT180|RTT270
        DATA = ASCII control sequence.
```

ORIENT
A required parameter. Defines the orientation of the paper in the printer.

PORTRAIT
The paper is vertically aligned in the printer

LANDSCAPE
The paper is horizontally aligned in the printer

RTT180
The paper is rotated 180 degrees

RTT270
The paper is rotated 270 degrees

DATA
A required parameter. Specifies the ASCII control sequence for setting the paper orientation for the printer. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

ASCII control sequence
Hexadecimal value for mapping an ASCII control sequence to define the orientation for the paper to be used with printer.

Print Quality (PRTQLTY) Tag

The PRTQLTY (print quality) tag defines the control sequence for setting the print quality. The PRTQLTY tag must follow the PFCNTBL tag in your source. The syntax for this tag is:

```
:PRTQLTY
        QLTYTYPE = DRAFT|LETTER|TEXT
        DATA = ASCII control sequence.
```

QLTYTYPE
A required parameter. Defines the quality of print.

DRAFT
The print quality is draft quality. This is equivalent to the *DRAFT type used in the OS/400 printer file commands.

LETTER
The print quality is letter quality. This is equivalent to the *NLQ type used in the OS/400 printer file commands.

TEXT
The print quality is text quality. This is equivalent to the *STD type used in the OS/400 printer file commands.

DATA
A required parameter. Specifies the ASCII control sequence for the function. This must be a hexadecimal value. The maximum length of this value is 240 bytes.

ASCII control sequence
Hexadecimal value for mapping an ASCII control sequence to define the print quality for the ASCII printer.

Tag Considerations for Customizing a Directly Attached ASCII Printer

In some cases there may be tags in your source that do not affect the printer or may cause unpredictable results when paired with other tags. The following sections describe these known error conditions and provide information to help you determine which tags you should use together.

Using the Superscript and Subscript Tags

If your ASCII printer supports the superscript and subscript functions, you can use the workstation customizing functions in one of two ways to allow them to work properly.

When your printer is set up to emulate an IBM 3812 printer, you need to define the reverse index (RVSIDX) tag so that your printer receives the correct instructions from the application data stream to move the paper up or down a half line and print. When your printer does not emulate the IBM 3812 printer, you need to use the STRSUPS-ENDSUPS and STRSUBS-ENDSUBS tags in pairs to turn on the superscript and subscript functions. Figure 12-4 shows these considerations in a flow chart format.

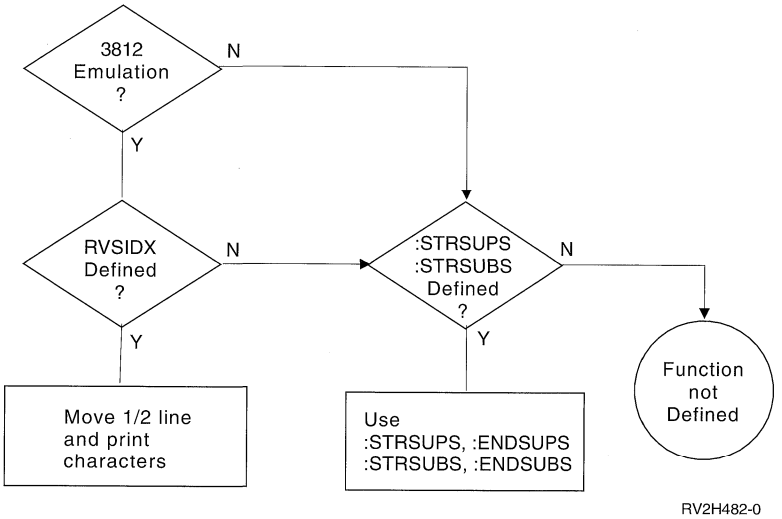


Figure 12-4. Decision Structure for Selecting the Subscript and Superscript Tags

Using the Set Page Length Tag

There are two tags provided by the workstation customizing functions for setting the page length for the printer. PAGLENL sets the page length by the number of lines that should be on a single page. PAGLENI sets the page length by number of inches. In general, you need to determine whether the number of lines on a page exceeds 255 and then check the reference manual for the printer to determine which measurement is used. In most cases, if the page length exceeds 255 lines, you should use the PAGLENI tag and its parameters to set your page length. Figure 12-5 shows a flow chart to help you determine which of these two tags to use.

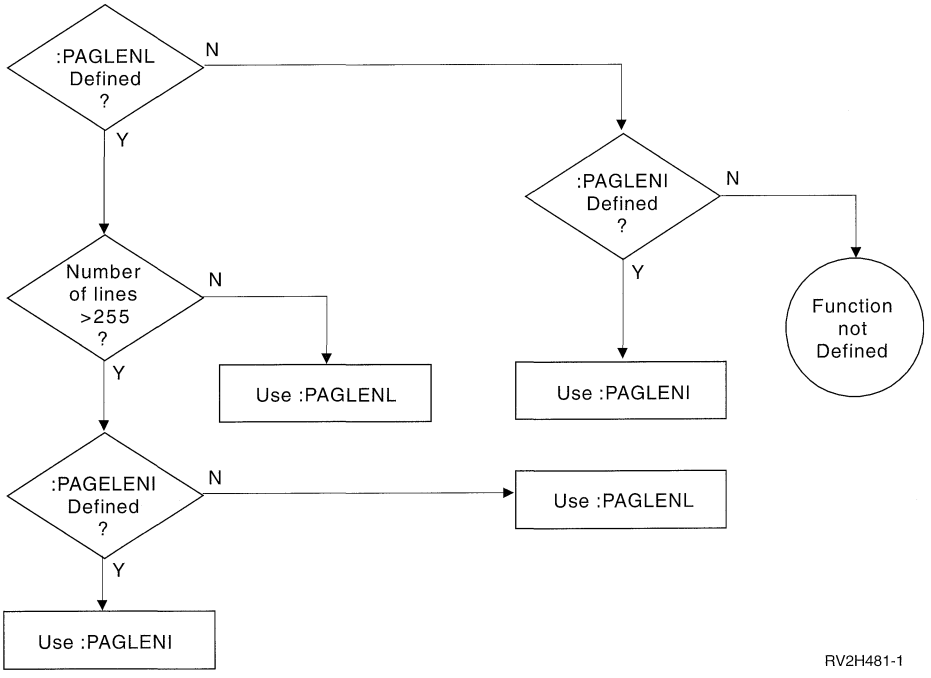


Figure 12-5. Decision Structure for Selecting the Correct Page Length Tags

Using the Font Selection Tags

The tags for font selection that are in your workstation customizing source are based on the device type you specified when you retrieved the source. If you are customizing a supported ASCII printer, you should use the font selection tags that are already in your source. You can add to or change the data that is associated with the type of tags that are retrieved; however, you need to be careful that you do not specify tags for font selection that do not work together. If you are customizing an unsupported ASCII printer, you need to check the font selection tags that are used in the workstation customizing source you retrieved and then compare the tags and their values to the font selection information in the reference manual for the printer. Verify that the method the printer uses to select a font is supported by the type of font selection tags that you retrieved in your source.

Figure 12-6 on page 12-17 shows a flow chart that provides a hierarchy for choosing the correct font selection tags for your ASCII printer. This chart is most helpful when you are not sure how fonts are selected for an unsupported printer. A list following the flow chart briefly describes the tag hierarchy for font selection.

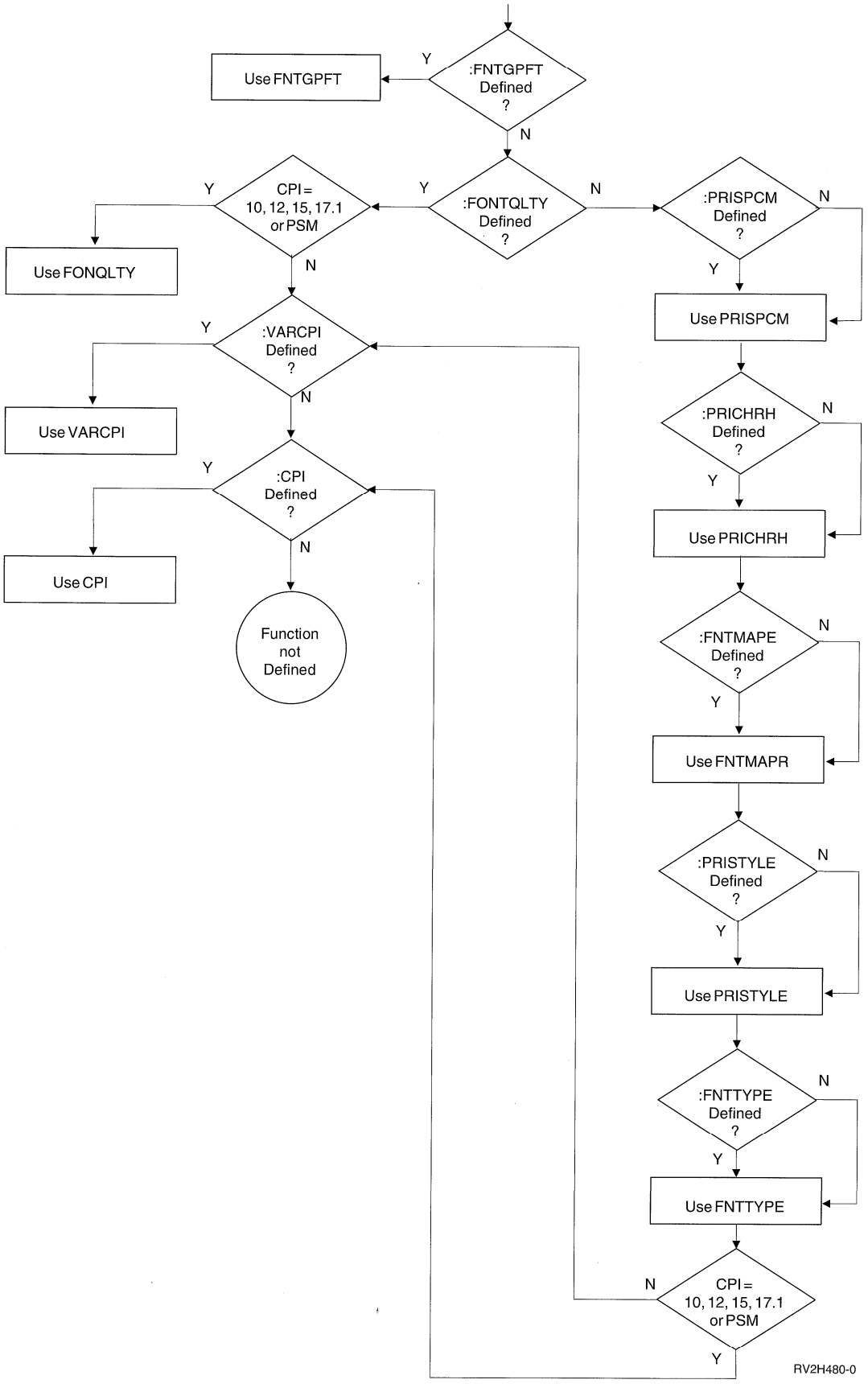


Figure 12-6. Decision Structure for Selecting the Correct Font Selection Tags

- If your printer supports using the Set Global Font for PFT tag, FNTGPFT, then you need specify no other tags for font selection.
- If you use the PRISTYLE (primary style), PRICHRH (primary character height), PRISPCM (primary spacing mode), and FNTTYPE (font type) tags, you should not use any of the other tags for font selection except for the CPI (characters per inch) tag.
- If your printer supports variable spacing, you should use the FONTQLTY (font quality) tag and the VARCPI (variable characters per inch) tag.

  Note: If variable characters per inch is not supported, you can use the CPI tag to set the characters per inch that are supported along with the FONTQLTY tag.
- If it is not apparent which method of font selection your printer uses, you should begin by setting the appropriate CPI tags and then try the other combinations of tags with the CPI settings.

Customizing a Hewlett-Packard LaserJet Series III Printer that Uses the Emulator on the Workstation Controller

This example shows the steps for customizing a Hewlett-Packard LaserJet Series III ASCII printer attached to the ASCII workstation controller. The objective for this example is to have the printer work effectively with simple OfficeVision/400 documents. This customization involves a lot of trial and error.

To begin the workstation customizing process you need to read the reference manual for the printer. You then set up the printer using the setup instructions and hints provided in the manual. You should look at the following setup items:

- Character set: PC 850 (Hewlett-Packard calls this a symbol set)
- Physical connection: Serial (parallel or serial)
- DIP switch settings
- Printer panel settings

After the printer is physically connected to the AS/400 system, and the appropriate controller and device descriptions are created, you should go back and set the controller description parameters shown below to the values indicated. These are the recommended settings for this type of printer.

ROBUST XON(ON)
DTR POLARITY(HI)

The printer should be powered on and varied on.

Use the following parameter values to create the device description for the Hewlett-Packard LaserJet Series III ASCII printer:

Device type	4019
Device model	*ASCII
Emulated twinaxial device	3812
Physical attachment	*WIRE4
Line speed	19200
Word length	8
Parity	NONE
Stop bits	1

Step 1: Planning the Customizing

To begin customizing this ASCII printer, you need to see whether or not you can get a document to print. Create a simple one-page document using the OfficeVision/400 editor. Use this document to test the characteristics that you expect from your printer.

When you try to print the test document, special characters are at the top of the page on the initial printout. Some characters do not show at all, or are not the characters you expected to see. Some printer functions do not work, including the following:

- Printer initialization sequences
- Bold printing
- Underlined printing
- Margins
- Landscape printing
- Font selection

In addition, the data stream from your application contains commands that tell the printer to use a different code page. Using this guide, you determine you must change all of the mapping tables to customize this printer.

Some characters are not shown or are not the characters you expected to see. Therefore, you need to change the default EBCDIC-to-ASCII mapping table. Use the reference manual for the printer to find the hexadecimal values you should specify for the default EBCDIC-to-ASCII mapping table. Your customized default EBCDIC-to-ASCII mapping table looks like this:

```
:PDFTEBCTBL
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '2020838485A0C68687A4BD2E3C282B7C'X      /* 4- */
   '26828889BAA18C8B8DE121242A293BAA'X      /* 5- */
   '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
   '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
   '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
   'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
   'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
   '5E9CBEFA9FF5F4ACABF35B5DEEF9EFF2'X      /* B- */
   '7B41424344454647 4849F0939495A2E4'X     /* C- */
   '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
   '5C2053545556575859 5AFDE299E3E0E5'X     /* E- */
   '3031323334353637 3839FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '48486A6A6A6A6A6A6A787852F052D378'X      /* 4- */
   'BB6A6A6A6A434343437 85278785252D3'X     /* 5- */
   '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
   '7895959595525252527 85278D352D36E'X     /* 7- */
   'AD6A786A786A5278784360 60787878D3'X     /* 8- */
   '78437843BB787878785678789E78D378'X      /* 9- */
   '86785B437878AD78786A526AB2AD8B78'X      /* A- */
   '787878AD787878D3D3D3D37878787878'X      /* B- */
   '6AAD95A3B29590ADB252527878787878'X      /* C- */
   '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
   '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
   '7878787878787878787856B2B2B2B278'X.     /* F- */
:EPDFTMAPTBL.
```

You also want to add and change printer functions. Therefore, you need to change the ASCII printer function table. Use the reference manual for the printer to find the hexadecimal values you should specify for the ASCII printer function table. You need to find the hexadecimal values you should specify for the following printer functions and characteristics:

- Printer initialization. Use these values to change the INITPRT and INITVON tags in the customizing source as follows:

```
:INITVON
  DATA ='1B451B283132551B266C3831411B
         266C32411B266C303045'X.
:INITPRT
  DATA ='1B451B283132551B266C303045'X.
```

- Bold highlighting. Use these values to change the STRBOLD and ENDBOLD tags in the customizing source as follows:

```
:STRBOLD
  DATA ='1B28733342'X.
:ENDBOLD
  DATA ='1B28733042'X.
```

- Underlining. Use these values to change the STRUS and ENDUS tags in the customizing source as follows:

```
:STRUS
  DATA ='1B26643044'X.
:ENDUS
  DATA ='1B266440'X.
```

- Margin setting. Use these values to change the MARGIN tags in the customizing source as follows:

```
:MARGIN
  OPTION=TOP
  ORIENT=PORTRAIT
  DATA =  720.
:MARGIN
  OPTION=LEFT
  ORIENT=PORTRAIT
  DATA =  240.
:MARGIN
  OPTION=RIGHT
  ORIENT=PORTRAIT
  DATA =  240.
:MARGIN
  OPTION=BOTTOM
  ORIENT=PORTRAIT
  DATA =  240.
:MARGIN
  OPTION=TOP
  ORIENT=LANDSCAPE
  DATA =  720.
:MARGIN
  OPTION=LEFT
  ORIENT=LANDSCAPE
  DATA =  240.
:MARGIN
  OPTION=RIGHT
  ORIENT=LANDSCAPE
  DATA =  240.
:MARGIN
  OPTION=BOTTOM
  ORIENT=LANDSCAPE
  DATA =  240.
```

- Portrait and landscape printing orientation. Use these values to change the PRTORIENT tags in the customizing source as follows:

```
:PRTORIENT
  ORIENT=PORTRAIT
  DATA ='1B266C304F'X.
:PRTORIENT
  ORIENT=LANDSCAPE
  DATA ='1B266C314F'X.
:PRTORIENT
  ORIENT=RTT180
  DATA ='1B266C324F'X.
:PRTORIENT
  ORIENT=RTT270
  DATA ='1B266C334F'X.
```

- Page length set in lines. Use this value to change the PAGLENL tag in the customizing source as follows:

```
:PAGLENL
  VAROFFSET=    3
  VARLEN=    3
  VARTYPE=CHRDEC
  VARMAX=  255
  ADJUST= 0
  DATA ='1B266C30303050'X.
```

  You want to set page length in lines, even if the number of lines is greater than 255 per page. Therefore, delete the PAGLENI tag from the customizing source.
- ASCII control code mapping. To remove the existing ASCII control code mapping, delete the ASCIICTL tags.
- Font selection. Use these values to add tags to select the correct fonts, as illustrated on Figure 12-6 on page 12-17. To select the correct fonts, you must do the following:

- Delete the FNTGPFT and FONTQLTY tags from the customizing source.
- Add PRISPCM, PRICHRH, FNTMAP, FNTMAPE, EFNTMAP, PRISTYLE, FNTTYPE, and VARCPI tags to the customizing source.

Specify the added tags as follows:

```
:FNTTYPE
  VAROFFSET=    3
  VARLEN=    3
  VARTYPE=CHRDEC
  DATA ='1B287330303054'X.
:PRISTYLE
  VAROFFSET=    3
  VARLEN=    1
  VARTYPE=CHRDEC
  DATA ='1B28733053'X.
:PRISPCM
  VAROFFSET=    3
  VARLEN=    1
  VARTYPE=CHRDEC
  DATA ='1B28733050'X.
:PRICHRH
  VAROFFSET=    3
  VARLEN=    2
  VARTYPE=CHRDEC
  VARMAX=65535
  ADJUST= 0
  CNVNUM=   20
  CNVDEN=    1
  DATA ='1B2873303056'X.
:VARCPI
  VAROFFSET=    3
  VARLEN=    2
  VARTYPE=CHRDEC
  VARMAX=255
  ADJUST= 0
  CNVNUM= 1440
  CNVDEN= 1440
  DATA ='1B2873303548'X.
:FNTMAP
  CPI=5
  DEFAULT=    11.
:FNTMAPE
  IBMFNT=   240
  ASCIIFNT=   11.
:EFNTMAP.
:FNTMAP
  CPI=855
  DEFAULT=    11.
:FNTMAPE
  IBMFNT=   260
  ASCIIFNT=   11.
:EFNTMAP.
:FNTMAP
  CPI=10
  DEFAULT=     3.
:FNTMAPE
  IBMFNT=      3
  ASCIIFNT=  110.
:FNTMAPE
  IBMFNT=      5
  ASCIIFNT=   10.
:FNTMAPE
  IBMFNT=     11
  ASCIIFNT=    3.
:FNTMAPE
  IBMFNT=     12
  ASCIIFNT=    8.
:FNTMAPE
  IBMFNT=     18
  ASCIIFNT=  259.
:FNTMAPE
  IBMFNT=     19
  ASCIIFNT=  104.
:FNTMAPE
  IBMFNT=     30
  ASCIIFNT=    8.
:FNTMAPE
  IBMFNT=     38
  ASCIIFNT=   10.
:FNTMAPE
  IBMFNT=     39
  ASCIIFNT=    6.
:FNTMAPE
  IBMFNT=     40
  ASCIIFNT=    6.
:FNTMAPE
  IBMFNT=     41
  ASCIIFNT=    5.
:FNTMAPE
  IBMFNT=     46
  ASCIIFNT=    3.
:FNTMAPE
  IBMFNT=     60
  ASCIIFNT=    8.
:EFNTMAP.
:FNTMAP
  CPI=PROP
  DEFAULT=    23.
:FNTMAPE
  IBMFNT=    160
  ASCIIFNT=   23.
:FNTMAPE
  IBMFNT=    155
  ASCIIFNT=  308.
:FNTMAPE
  IBMFNT=    157
  ASCIIFNT=   52.
:FNTMAPE
  IBMFNT=    158
  ASCIIFNT=    4.
:FNTMAPE
  IBMFNT=    159
  ASCIIFNT=   52.
:FNTMAPE
  IBMFNT=    162
  ASCIIFNT=  279.
:FNTMAPE
  IBMFNT=    187
  ASCIIFNT=   52.
:FNTMAPE
  IBMFNT=    188
  ASCIIFNT=  279.
:FNTMAPE
  IBMFNT=    189
  ASCIIFNT=  279.
:FNTMAPE
  IBMFNT=    194
  ASCIIFNT=  279.
:FNTMAPE
  IBMFNT=    195
  ASCIIFNT=  279.
:EFNTMAP.
:FNTMAP
  CPI=        12
  DEFAULT=     6.
:FNTMAPE
  IBMFNT=     85
  ASCIIFNT=    3.
:FNTMAPE
  IBMFNT=     66
  ASCIIFNT=    6.
:FNTMAPE
```

```
      IBMFNT=     68
      ASCIIFNT=  262.
    :FNTMAPE
      IBMFNT=     84
      ASCIIFNT=    7.
    :FNTMAPE
      IBMFNT=     86
      ASCIIFNT=    8.
    :FNTMAPE
      IBMFNT=     91
      ASCIIFNT=  262.
    :FNTMAPE
      IBMFNT=     92
      ASCIIFNT=  259.
    :FNTMAPE
      IBMFNT=    108
      ASCIIFNT=    3.
    :FNTMAPE
      IBMFNT=    109
      ASCIIFNT=  262.
    :FNTMAPE
      IBMFNT=    110
      ASCIIFNT=    6.
    :FNTMAPE
      IBMFNT=    111
      ASCIIFNT=    8.
    :FNTMAPE
      IBMFNT=    112
      ASCIIFNT=  264.
    :EFNTMAP.
    :FNTMAP
      CPI=       133
      DEFAULT=     6.
    :FNTMAPE
      IBMFNT=    204
      ASCIIFNT=    6.
    :EFNTMAP.
    :FNTMAP
      CPI=        15
      DEFAULT=     6.
    :FNTMAPE
      IBMFNT=    230
      ASCIIFNT=    6.
    :FNTMAPE
      IBMFNT=    231
      ASCIIFNT=    6.
    :EFNTMAP.
    :FNTMAP
      CPI=       171
      DEFAULT=     3.
    :FNTMAPE
      IBMFNT=    253
      ASCIIFNT=    3.
    :FNTMAPE
      IBMFNT=    254
      ASCIIFNT=    3.
    :FNTMAPE
      IBMFNT=    252
      ASCIIFNT=    3.
    :FNTMAPE
      IBMFNT=    255
      ASCIIFNT=    3.
    :EFNTMAP.
    :FNTMAP
      CPI=        20
      DEFAULT=     6.
    :FNTMAPE
      IBMFNT=    281
      ASCIIFNT=    6.
    :EFNTMAP.
```

- Superscript printing. Use these values to change the STRSUPS and ENDSUPS tags in the customizing source as follows:

```
:STRSUPS
  DATA ='1B26612D2E3552'X.
:ENDSUPS
  DATA ='1B3D'X.
```

- Subscript printing. Use these values to change the STRSUBS and ENDSUBS tags in the customizing source as follows:

```
:STRSUBS
  DATA ='1B3D'X.
:ENDSUBS
  DATA ='1B26612D2E3552'X.
```

- Forward and backward relative movement. Use these values to change the FWDRMOV and BCKRMOV tags in the customizing source as follows:

```
:FWDRMOV
  VAROFFSET=    4
  VARLEN=    5
  VARTYPE=CHRDEC
  VARMAX=32767
  ADJUST= 0
  CNVNUM=    2
  CNVDEN=    1
  DATA ='1B26612B303030303048'X.
:BCKRMOV
  VAROFFSET=    4
  VARLEN=    5
  VARTYPE=CHRDEC
  VARMAX=32767
  ADJUST= 0
  CNVNUM=    2
  CNVDEN=    1
  DATA ='1B26612D303030303048'X.
```

- Default font identifiers for different fonts. Use these values to change the DFTFNTID tags in the customizing source as follows:

```
:DFTFNTID
  CPI=855
  DEFAULT=  265.
:DFTFNTID
  CPI=10
  DEFAULT=   11.
:DFTFNTID
  CPI=12
  DEFAULT=   85.
:DFTFNTID
  CPI=15
  DEFAULT=  230.
:DFTFNTID
  CPI=171
  DEFAULT=  253.
:DFTFNTID
  CPI=PROP
  DEFAULT=  159.
:DFTFNTID
  CPI=5
  DEFAULT=  244.
:DFTFNTID
  CPI=133
  DEFAULT=  204.
:DFTFNTID
  CPI=20
  DEFAULT=  281.
```

- Number of characters per inch. Use these values to change the CPI tags in the customizing source as follows:

```
:CPI
  CPI=855
  DATA ='1B2873303848'X.
:CPI
  CPI=10
  DATA ='1B2873313048'X.
:STRPROP
  DATA ='1B2873303150'X.
:CPI
  CPI=12
  DATA ='1B2873313248'X.
:CPI
  CPI=15
  DATA ='1B2873313548'X.
:CPI
  CPI=171
  DATA ='1B266B325300'X.
```

- Control sequence for feeding different types of paper to the printer. Use these values to change the PRTFEED tags in the customizing source as follows:

```
:PRTFEED
  FEEDTYPE=MANUAL
  DATA ='1B266C3248'X.
```

- Control sequence for paper drawer selection. Use these values to change the DWRSLT tags in the customizing source as follows:

```
:DWRSLT
  DRAWER=DRAWER1
  DATA ='1B266C3148'X.
:DWRSLT
  DRAWER=DRAWER2
  DATA ='1B266C3448'X.
:DWRSLT
  DRAWER=ENVELOPE
  DATA ='1B266C3648'X.
```

- Reverse index. Use this value to change the RVSIDX tag in the customizing source. as follows:

```
:RVSIDX
  DATA ='1B26612D3152'X.
```

- Printer control flags. The printer control flag values come from this guide, instead of the printer reference manual. Use the values indicated in this guide to change the PRTCTL tag in the customizing source. You set printer control flags to '82'X to indicate:
  - Page length changes in the middle of the page are not allowed.
  - Font widths are indicated in 1/1440-inch units.
- Page size for different types of paper. Use these values to change the PAGSIZPFT tags in the customizing source. For legal size (8.5-inch by 14-inch) paper, manually fed, specify:

```
:PAGSIZPFT
  PAGWTH=12240
  PAGLEN=20160
  PAPER=MANUAL.
```

  For letter size (8.5-inch by 11-inch) paper, in paper drawer 1, specify:

```
:PAGSIZPFT
  PAGWTH=12240
  PAGLEN=15840
  PAPER=DRAWER1.
```

  For letter size (8.5-inch by 11-inch) paper, in paper drawer 2, specify:

```
:PAGSIZPFT
  PAGWTH=12240
  PAGLEN=15840
  PAPER=DRAWER2.
```

  For commercial 10 (9.5-inch by 4.125-inch) envelopes, in the envelope feed, specify:

```
:PAGSIZPFT
  PAGWTH=12240
  PAGLEN= 5760
  PAPER=ENVELOPE.
```

- Code page. If your printer uses more than one ASCII code page symbol set, use this value to change the :CODPAGVAR tag in your customizing source. If your printer uses only one ASCII code page, delete the CODPAGVAR tag from your customizing source.

```
:CODPAGVAR
  VAROFFSET=    2
  VARLEN=    3
  VARTYPE=CHRAN
  DATA ='1B28313255'X.
```

- Print quality. Use these values to add PRTQLTY tags to the customizing source as follows:

```
:PRTQLTY
  QLTYTYPE=LETTER
  DATA ='1B28733053'X.
:PRTQLTY
  QLTYTYPE=DRAFT
  DATA ='1B28733053'X.
:PRTQLTY
  QLTYTYPE=TEXT
  DATA ='1B28733153'X.
```

The data stream from your application contains commands that tell the printer to use a different code page. Therefore, you need to change the printer multi language EBCDIC-to-ASCII mapping table. Use the reference manual for the printer to find the hexadecimal values you should indicate for the printer multi language EBCDIC-to-ASCII mapping table.

If your applications use only U.S. English and U.S. international code pages (code page 037 and code page 038), the changed portion of the printer multi language EBCDIC-to-ASCII mapping table looks like this:

```
/* Following is the code page mapping for */
/* EBCDIC code page 037 (USB) to ASCII    */
/* code page 3157 (PC 850)                */

:PMLGEBCTBL
  EBCDICCP=   37
  ASCIICP=  3157
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '2020838485A0C68687A4BD2E3C282B7C'X      /* 4- */
   '26828889BAA18C8B8DE121242A293BAA'X      /* 5- */
   '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
   '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
   '9D6162636465666769AEAFD0ECE7F1'X        /* 8- */
   'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
   'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
   '5E9CBEFA9FF5F4ACABF35B5DEEF9EFF2'X      /* B- */
   '7B414243444546474849F0939495A2E4'X      /* C- */
   '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
   '5C20535455565758595AFDE299E3E0E5'X      /* E- */
   '303132333435363738 39FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
   'BB6A6A6A6A434343437 85278785252D3'X     /* 5- */
   '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
   '7895959595525252527 85278D352D36E'X     /* 7- */
   'AD6A786A786A5278784360 60787878D3'X     /* 8- */
   '78437843BB787878785678789E78D378'X      /* 9- */
   '86785B437878AD78786A526AB2AD8B78'X      /* A- */
   '787878AD787878D3D3D3D37878787878'X      /* B- */
   '6AAD95A3B29590ADB252527878787878'X      /* C- */
   '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
   '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
   '7878787878787878787856B2B2B2B278'X.     /* F- */

/* Following is the code page mapping for */
/* EBCDIC code page 038 (USI) to ASCII    */
/* code page 3157 (PC 850).               */

:PMLGEBCTBL
  EBCDICCP=   38
  ASCIICP=  3157
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '2020838485A0C68687A45B2E3C282B21'X      /* 4- */
   '268288898AA18C8B8DE15D242A293B5E'X      /* 5- */
   '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
   '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
   '9D6162636465666769AEAFD0ECE7F1'X        /* 8- */
   'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
   'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
   'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
   '7B414243444546474849 2D939495A2E4'X     /* C- */
   '7D4A4B4C4D4E4F505152FB968197A398'X      /* D- */
   '5C20535455565758595AFDE299E3E0E5'X      /* E- */
   '303132333435363738 39FCEA9AEBE900'X.    /* F- */

:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '48486A6A6A6A6A6A785252F052D352'X        /* 4- */
   'BB6A6A6A6A434343437 8527878525278'X     /* 5- */
   '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
   '7895959595525252527 85278D352D36E'X     /* 7- */
   'AD6A786A786A5278784360 60787878D3'X     /* 8- */
   '78437843BB787878785678789E78D378'X      /* 9- */
   '86785B437878AD78786A526AB2AD8B78'X      /* A- */
   '787878AD787878D3D3D3D37878787878'X      /* B- */
   '6AAD95A3B29590ADB252527878787878'X      /* C- */
   '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
   '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
   '7878787878787878787856B2B2B2B278'X.     /* F- */
```

If your applications use additional code pages, you must change the printer multi-language EBCDIC-to-ASCII mapping tables for each additional code page used.

Step 2: Retrieving the Workstation Customizing Source

To create a workstation customizing source, you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command and specify the following:

Device type
DEVTYPE(4019)

Keyboard language type
KBDTYPE(USI) This value must match the language type you specified in the device description for the printer; otherwise it will not vary on.

Source member
SRCMBR(CSTHPDA)

Source file
SRCFILE(CSTSRC)

Library LIB(CSTLIB)

Text TEXT('Workstation customizing source for HPDA ASCII Printer')

Step 3: Changing the Source

After you retrieve the source, use the source entry utility (SEU) to change the printer characteristics. The command looks like the following:

```
STRSEU SRCFILE(CSTLIB/CSTSRC) SRCMBR(CSTHPDA)
```

The copy of the source you retrieve contains all the mapping tables for the printer. These mapping tables are downloaded to the workstation controller when the printer is varied on. When you edit the file, functions and characteristics associated with tags you delete are no longer mapped and do not work anymore.

Make a few changes at a time to the workstation customizing source. Then compile the source. Vary off the printer and then vary it on. Print the test document each time you vary on the printer to see if your changes are effective.

Note: When you vary the device on and off many times, the workstation controller storage can fill up. When this occurs, the device no longer varies on and an error message is sent to the QSYSOPR message queue. To correct this situation, it is necessary to vary off the workstation controller, and then vary it on again specifying RESET(*YES). This clears the storage in the workstation controller and allows the mapping tables in the new workstation customizing object to be loaded.

The following listing shows the customized source, after you have made all changes. For an example of the original retrieved source, see Appendix B, "Source Code Examples."

```
:WSCST DEVCLASS=ASCIIPRT.

  :PDFTMAPTBL.
    :PDFTEBCTBL
      DATA =
     /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
       '2020838485A0C68687A4BD2E3C282B7C'X      /* 4- */
       '26828889BAA18C8B8DE121242A293BAA'X      /* 5- */
       '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
       '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
       '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
       'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
       'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
       '5E9CBEFA9FF5F4ACABF35B5DEEF9EFF2'X      /* B- */
       '7B41424344454646474849F0939495A2E4'X    /* C- */
       '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
       '5C20535455565758595AFDE299E3E0E5'X      /* E- */
       '30313233343536373839FCEA9AEBE900'X.     /* F- */
    :PFNTWTH
      DATA =
     /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
       '48486A6A6A6A6A6A6A787852F052D378'X      /* 4- */
       'BB6A6A6A6A6A434343437852785252D3'X      /* 5- */
       '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
       '78959595955252525278 5278D352D36E'X     /* 7- */
       'AD6A786A786A52787843606078787 8D3'X     /* 8- */
       '78437843BB787878785678789E78D378'X      /* 9- */
       '86785B437878AD78786A526AB2AD8B78'X      /* A- */
       '787878AD787878D3D3D3D37878787878'X      /* B- */
       '6AAD95A3B29590ADB252527878787878'X      /* C- */
       '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
       '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
       '7878787878787878787856B2B2B2B278'X.     /* F- */
  :EPDFTMAPTBL.

    :PFCNTBL.
    :DFTFNTID
      CPI=855
      DEFAULT=  265.
    :DFTFNTID
      CPI=10
      DEFAULT=   11.
    :DFTFNTID
      CPI=12
      DEFAULT=   85.
    :DFTFNTID
      CPI=15
      DEFAULT=  230.
    :DFTFNTID
      CPI=171
      DEFAULT=  253.
    :DFTFNTID
      CPI=PROP
      DEFAULT=  159.
    :DFTFNTID
      CPI=5
      DEFAULT=  244.
    :DFTFNTID
      CPI=133
      DEFAULT=  204.
    :DFTFNTID
      CPI=20
      DEFAULT=  281.
    :VARLSPC
      VAROFFSET=    3
      VARLEN=    2
      VARTYPE=CHRDEC
      VARMAX=  255
      ADJUST= 0
      CNVNUM=   30
      CNVDEN=    1
      DATA ='1B266C30302E303043'X.

/* Removed the PAGLENI tag here */

    :PAGLENL
      VAROFFSET=    3
      VARLEN=    3
      VARTYPE=CHRDEC
      VARMAX=  255
      ADJUST= 0
      DATA ='1B266C30303050'X.
    :CODPAGVAR
      VAROFFSET=    2
      VARLEN=    3
      VARTYPE=CHRAN
      DATA ='1B28313255'X.

/* Removed the FNTGPFT tag here */

    :FWDRMOV
      VAROFFSET=    4
      VARLEN=    5
      VARTYPE=CHRDEC
      VARMAX=32767
      ADJUST= 0
      CNVNUM=    2
      CNVDEN=    1
      DATA ='1B26612B303030303048'X.
    :BCKRMOV
      VAROFFSET=    4
      VARLEN=    5
      VARTYPE=CHRDEC
      VARMAX=32767
      ADJUST= 0
      CNVNUM=    2
      CNVDEN=    1
      DATA ='1B26612D303030303048'X.

/* Add the FNTTYPE, PRISTYLE, PRISPCM,   */
/* PRICHRH, and VARCPI tags here.        */

    :FNTTYPE
      VAROFFSET=    3
      VARLEN=    3
      VARTYPE=CHRDEC
      DATA ='1B287330303054'X.
    :PRISTYLE
      VAROFFSET=    3
      VARLEN=    1
      VARTYPE=CHRDEC
      DATA ='1B28733053'X.
    :PRISPCM
      VAROFFSET=    3
      VARLEN=    1
      VARTYPE=CHRDEC
      DATA ='1B28733050'X.
    :PRICHRH
      VAROFFSET=    3
      VARLEN=    2
      VARTYPE=CHRDEC
      VARMAX=65535
      ADJUST= 0
      CNVNUM=   20
      CNVDEN=    1
      DATA ='1B2873303056'X.
    :VARCPI
      VAROFFSET=    3
      VARLEN=    2
      VARTYPE=CHRDEC
```

```
      VARMAX=255
      ADJUST= 0
      CNVNUM= 1440
      CNVDEN= 1440
      DATA ='1B2873303548'X.

/* End of new tags. */

    :CPI
      CPI=855
      DATA ='1B2873303848'X.
    :CPI
      CPI=10
      DATA ='1B2873313048'X.
    :STRPROP
      DATA ='1B2873303150'X.
    :CPI
      CPI=12
      DATA ='1B2873313248'X.
    :CPI
      CPI=15
      DATA ='1B2873313548'X.
    :CPI
      CPI=171
      DATA ='1B266B325300'X.

/* Removed the FONTQLTY tags here. */

    :STRBOLD
      DATA ='1B28733342'X.
    :ENDBOLD
      DATA ='1B28733042'X.

/* Add PRTQLTY tags here. */

    :PRTQLTY
      QLTYTYPE=LETTER
      DATA ='1B28733053'X.
    :PRTQLTY
      QLTYTYPE=DRAFT
      DATA ='1B28733053'X.
    :PRTQLTY
      QLTYTYPE=TEXT
      DATA ='1B28733153'X.

/* End of new tags. */

:PRTFEED
  FEEDTYPE=MANUAL
  DATA ='1B266C3248'X.
:DWRSLT
  DRAWER=DRAWER1
  DATA ='1B266C3148'X.
:DWRSLT
  DRAWER=DRAWER2
  DATA ='1B266C3448'X.
:DWRSLT
  DRAWER=ENVELOPE
  DATA ='1B266C3648'X.
:PRTORIENT
  ORIENT=PORTRAIT
  DATA ='1B266C304F'X.
:PRTORIENT
 ORIENT=LANDSCAPE
 DATA ='1B266C314F'X.
:PRTORIENT
  ORIENT=RTT180
  DATA ='1B266C324F'X.
:PRTORIENT
  ORIENT=RTT270
  DATA ='1B266C334F'X.
:STRUS
  DATA ='1B26643044'X.
:ENDUS
  DATA ='1B266440'X.
:STRSUPS
  DATA ='1B26612D2E3552'X.
:ENDSUPS
  DATA ='1B3D'X.
:STRSUBS
  DATA ='1B3D'X.
:ENDSUBS
  DATA ='1B26612D2E3552'X.
:RVSIDX
  DATA ='1B26612D3152'X.
:INITVON
 DATA ='1B451B283132551B266C3831411B266C32411B266C303045'X.
:INITPRT
  DATA ='1B451B283132551B266C303045'X.
:SPACE
  DATA ='20'X.
:BELL
  DATA ='07'X.
:CARRTN
  DATA ='0D'X.
:LINEFEED
  DATA ='0A'X.
:FORMFEED
  DATA ='0C'X.
:PRTCTL
  DATA ='82'X.
:BSP
  DATA ='08'X.
:COLLATE
  DATA =   12.
:PAGSIZPFT
  PAGWTH=12240
  PAGLEN=20160
  PAPER=MANUAL.
:PAGSIZPFT
  PAGWTH=12240
  PAGLEN=15840
  PAPER=DRAWER1.
:PAGSIZPFT
  PAGWTH=12240
  PAGLEN=15840
  PAPER=DRAWER2.
:PAGSIZPFT
  PAGWTH=12240
  PAGLEN= 5760
  PAPER=ENVELOPE.
:MARGIN
  OPTION=TOP
  ORIENT=PORTRAIT
  DATA =  720.
:MARGIN
  OPTION=LEFT
  ORIENT=PORTRAIT
  DATA =  240.
:MARGIN
  OPTION=RIGHT
  ORIENT=PORTRAIT
  DATA =  240.
:MARGIN
  OPTION=BOTTOM
  ORIENT=PORTRAIT
  DATA =  240.
:MARGIN
  OPTION=TOP
  ORIENT=LANDSCAPE
  DATA =  720.
:MARGIN
  OPTION=LEFT
  ORIENT=LANDSCAPE
```

```
   DATA =  240.
:MARGIN
   OPTION=RIGHT
   ORIENT=LANDSCAPE
   DATA =  240.
:MARGIN
   OPTION=BOTTOM
   ORIENT=LANDSCAPE
   DATA =  240.

/* Removed the ASCIICTL tags here.             */

/* Add FNTMAP, FNTMAPE, and EFNTMAP tags here. */

     :FNTMAP
       CPI=5
       DEFAULT=    11.
     :FNTMAPE
       IBMFNT=   240
       ASCIIFNT=   11.
     :EFNTMAP.
     :FNTMAP
       CPI=855
       DEFAULT=    11.
     :FNTMAPE
       IBMFNT=   260
       ASCIIFNT=   11.
     :EFNTMAP.
     :FNTMAP
       CPI=10
       DEFAULT=     3.
     :FNTMAPE
       IBMFNT=      3
       ASCIIFNT=  110.
     :FNTMAPE
       IBMFNT=      5
       ASCIIFNT=   10.
     :FNTMAPE
       IBMFNT=     11
       ASCIIFNT=    3.
     :FNTMAPE
       IBMFNT=     12
       ASCIIFNT=    8.
     :FNTMAPE
       IBMFNT=     18
       ASCIIFNT=  259.
     :FNTMAPE
       IBMFNT=     19
       ASCIIFNT=  104.
     :FNTMAPE
       IBMFNT=     30
       ASCIIFNT=    8.
     :FNTMAPE
       IBMFNT=     38
       ASCIIFNT=   10.
     :FNTMAPE
       IBMFNT=     39
       ASCIIFNT=    6.
     :FNTMAPE
       IBMFNT=     40
       ASCIIFNT=    6.
     :FNTMAPE
       IBMFNT=     41
       ASCIIFNT=    5.
     :FNTMAPE
       IBMFNT=     46
       ASCIIFNT=    3.
     :FNTMAPE
       IBMFNT=     60
       ASCIIFNT=    8.
     :EFNTMAP.
     :FNTMAP
       CPI=PROP
       DEFAULT=    23.
     :FNTMAPE
       IBMFNT=    160
       ASCIIFNT=   23.
     :FNTMAPE
       IBMFNT=    155
       ASCIIFNT=  308.
     :FNTMAPE
       IBMFNT=    157
       ASCIIFNT=   52.
     :FNTMAPE
       IBMFNT=    158
       ASCIIFNT=    4.
     :FNTMAPE
       IBMFNT=    159
       ASCIIFNT=   52.
     :FNTMAPE
       IBMFNT=    162
       ASCIIFNT=  279.
     :FNTMAPE
       IBMFNT=    187
       ASCIIFNT=   52.
     :FNTMAPE
       IBMFNT=    188
       ASCIIFNT=  279.
     :FNTMAPE
       IBMFNT=    189
       ASCIIFNT=  279.
     :FNTMAPE
       IBMFNT=    194
       ASCIIFNT=  279.
     :FNTMAPE
       IBMFNT=    195
       ASCIIFNT=  279.
     :EFNTMAP.
     :FNTMAP
       CPI=         12
       DEFAULT=     6.
     :FNTMAPE
       IBMFNT=     85
       ASCIIFNT=    3.
     :FNTMAPE
       IBMFNT=     66
       ASCIIFNT=    6.
     :FNTMAPE
       IBMFNT=     68
       ASCIIFNT=  262.
     :FNTMAPE
       IBMFNT=     84
       ASCIIFNT=    7.
     :FNTMAPE
       IBMFNT=     86
       ASCIIFNT=    8.
     :FNTMAPE
       IBMFNT=     91
       ASCIIFNT=  262.
     :FNTMAPE
       IBMFNT=     92
       ASCIIFNT=  259.
     :FNTMAPE
       IBMFNT=    108
       ASCIIFNT=    3.
     :FNTMAPE
       IBMFNT=    109
       ASCIIFNT=  262.
     :FNTMAPE
       IBMFNT=    110
       ASCIIFNT=    6.
     :FNTMAPE
       IBMFNT=    111
       ASCIIFNT=    8.
```

```
    :FNTMAPE
      IBMFNT=    112
      ASCIIFNT=  264.
    :EFNTMAP.
    :FNTMAP
      CPI=       133
      DEFAULT=     6.
    :FNTMAPE
      IBMFNT=    204
      ASCIIFNT=    6.
    :EFNTMAP.
    :FNTMAP
      CPI=        15
      DEFAULT=     6.
    :FNTMAPE
      IBMFNT=    230
      ASCIIFNT=    6.
    :FNTMAPE
      IBMFNT=    231
      ASCIIFNT=    6.
    :EFNTMAP.
    :FNTMAP
      CPI=       171
      DEFAULT=     3.
    :FNTMAPE
      IBMFNT=    253
      ASCIIFNT=    3.
    :FNTMAPE
      IBMFNT=    254
      ASCIIFNT=    3.
    :FNTMAPE
      IBMFNT=    252
      ASCIIFNT=    3.
    :FNTMAPE
      IBMFNT=    255
      ASCIIFNT=    3.
    :EFNTMAP.
    :FNTMAP
      CPI=        20
      DEFAULT=     6.
    :FNTMAPE
      IBMFNT=    281
      ASCIIFNT=    6.
    :EFNTMAP.

/* End of new tags. */


  :PMLGMAPTBL.

/* Following are the changed code page mappings.  */


    :PMLGEBCTBL
      EBCDICCP=   29
      ASCIICP=  3157
      DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '2020838485A0C68687A4E82E3C282B21'X      /* 4- */
     '268288898AA18C8B8DE192242A293B99'X      /* 5- */
     '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
     '9B90D2D3D4D6D7D8DED03A23D1273D22'X      /* 7- */
     '9D6162636465666676869AEAF60EC7BF1'X     /* 8- */
     'F86A6B6C6D6E6F707172A6A77DF75DCF'X      /* 9- */
     'E6947374757677787 97AADA840ED5BA9'X     /* A- */
     'BD9CBEFAB8F5F4ACABF3AA7CEEF95C9E'X      /* B- */
     'E7414243444546474849F0937E95A2E4'X      /* C- */
     '914A4B4C4D4E4F505152FB968197A398'X      /* D- */
     'EFF65354555657585 95AFDE25EE3E0E5'X     /* E- */
     '3031323334353637 3839FCEA9AEBE900'X.    /* F- */
    :PFNTWTH
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
     'BB6A6A6A6A434343437 85278785252D3'X     /* 5- */
     '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
     '7895959595525252527 85278D352D36E'X     /* 7- */
     'AD6A786A786A52787843606078787 8D3'X     /* 8- */
     '78437843BB787878785678789E78D378'X      /* 9- */
     '86785B437878AD78786A526AB2AD8B78'X      /* A- */
     '787878AD787878D3D3D3D37878787878'X      /* B- */
     '6AAD95A3B29590ADB252527878787878'X      /* C- */
     '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
     '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
     '7878787878787878787856B2B2B2B278'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=   30
      ASCIICP= 3157
      DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '2020838485A000917BA4802E3C282B21'X      /* 4- */
     '268288898AA18C8B8DE1A6982A293B5E'X      /* 5- */
     '2D2FB686B7B500925BA5AD2C255F3E3F'X      /* 6- */
     '0090D2D3D4D6D7D8DED53A99B8273D9A'X      /* 7- */
     'F46162636465666768 69E886C7ED007C'X     /* 8- */
     'F86A6B6C6D6E6F707172A99B9FF700CF'X      /* 9- */
     'E6947374757677787 97AE78FC6EC0040'X     /* A- */
     'FA9CBE7DBDF55D00AB24A89DACF9EF9E'X      /* B- */
     '8741424344454647 4849F0937E95A2E5'X     /* C- */
     'A74A4B4C4D4E4F50515260965C81A300'X      /* D- */
     '81F6535455565758595AFDE223E3E0E4'X      /* E- */
     '3031323334353637 3839FCEA22EBE900'X.    /* F- */
    :PFNTWTH
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
     'BB6A6A6A6A434343437 85278785252D3'X     /* 5- */
     '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
     '7895959595525252527 85278D352D36E'X     /* 7- */
     'AD6A786A786A52787843606078787 8D3'X     /* 8- */
     '78437843BB787878785678789E78D378'X      /* 9- */
     '86785B437878AD78786A526AB2AD8B78'X      /* A- */
     '787878AD787878D3D3D3D37878787878'X      /* B- */
     '6AAD95A3B29590ADB252527878787878'X      /* C- */
     '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
     '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
     '7878787878787878787856B2B2B2B278'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=   37
      ASCIICP= 3157
      DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '2020838485A0C68687A4BD2E3C282B7C'X      /* 4- */
     '26828889BAA18C8B8DE121242A293BAA'X      /* 5- */
     '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
     '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
     '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
     'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
     'E67E7374757677787 97AADA8D1EDE8A9'X     /* A- */
     '5E9CBEFA9FF5F4ACABF35B5DEEF9EFF2'X      /* B- */
     '7B4142434445464748 49F0939495A2E4'X     /* C- */
     '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
     '5C2053545556575859 5AFDE299E3E0E5'X     /* E- */
     '3031323334353637 3839FCEA9AEBE900'X.    /* F- */
    :PFNTWTH
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
     'BB6A6A6A6A434343437 85278785252D3'X     /* 5- */
     '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
     '7895959595525252527 85278D352D36E'X     /* 7- */
     'AD6A786A786A52787843606078787 8D3'X     /* 8- */
```

```
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=   38
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A0C68687A45B2E3C282B21'X      /* 4- */
      '268288898AA18C8B8DE15D242A293B5E'X      /* 5- */
      '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
      '9D616263646566676869AEAFD0ECE7F1'X      /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
      'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
      '7B41424344445464748 49F0939495A2E4'X    /* C- */
      '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
      '5C20535455565758595AFDE299E3E0E5'X      /* E- */
      '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852787852 52D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X       /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078787 8D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=   45
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838461A0C79F87865B2E3C282B21'X      /* 4- */
      '2682A98985A18C9692E15D242A293B5E'X      /* 5- */
      '2D2FB68E41B5C6AC808FD82C255F3E3F'X      /* 6- */
      'F390A8D3DED6D79591603A2340273D22'X      /* 7- */
      'F461626364656667686998E5D0ECFDAD'X      /* 8- */
      'F86A6B6C6D6E6F70717288E4E7F7F2CF'X      /* 9- */
      'A5F1737475767778797AB8D5D1EDFCB8'X      /* A- */
      'FA4CBEDDBDF5A7ABA68D9DE3E6F9EFEE'X      /* B- */
      'D4414243444546474849F09394EAA28B'X      /* C- */
      'D24A4B4C4D4E4F505152B7FB819CA379'X      /* D- */
      'A420535455565758595A32E299E8E08A'X      /* E- */
      '3031323334353637383933EB9A9BE900'X.     /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852787852 52D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X       /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078787 8D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  256
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A0C68687A45B2E3C282B21'X      /* 4- */
      '268288898AA18C8B8DE15D242A293B5E'X      /* 5- */
      '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
      '9D616263646566676869AEAFD0ECE7F1'X      /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
      'BD9CBE169FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
      '7B41424344454 6474849F0939495A2E4'X     /* C- */
      '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
      '5C20535455565758595AFDE299E3E0E5'X      /* E- */
      '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852787852 52D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X       /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078787 8D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  257
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '202080828486888A8C8E5B2E3C282B21'X      /* 4- */
      '2690929496989A9EEF815D242A293B5E'X      /* 5- */
      '2D2F838587898B8D8F917C2C255F3E3F'X      /* 6- */
      '9395972D999B9CA0A2603A2340273D22'X      /* 7- */
      '9A616263646566676869A6A8AAACB5B7'X      /* 8- */
      'BD6A6B6C6D6E6F707172C6D0D2D4D6D8'X      /* 9- */
      'DE7E737475767778797AE1E3E5E7E9EB'X      /* A- */
      'EDF1F3F5F7F9FB9E9DA1A3A5A7A9ABAD'X      /* B- */
      '7B41424344454 6474849B6B8BEC7D1D3'X     /* C- */
      '7D4A4B4C4D4E4F505152D5D7DDE0E2E4'X      /* D- */
      '5CCF535455565758595AE6E8EAECEEF2'X      /* E- */
      '3031323334353637383 9F4F6F8FAFC00'X.    /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852787852 52D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X       /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078787 8D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  258
       ASCIICP=  3157
       DATA =
```

```
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '20208384EEA0C79F87865B2E3C282B21'X      /* 4- */
      '2682A98985A18C9692E15D242A293B5E'X      /* 5- */
      '2D2FB68EF1B5C6AC808F7C2C255F3E3F'X      /* 6- */
      'F390A8D3DED6D79591603A2340273D22'X      /* 7- */
      'F4616263646566667686998E5D0ECFDAD'X     /* 8- */
      'F86A6B6C6D6E6F70717288E4E7F7F2CF'X      /* 9- */
      'A57E737475767778797AB8D5D1EDFCB8'X      /* A- */
      'FAA4BEDDBDF5A7ABA68D9DE3E6F9EF9E'X      /* B- */
      '7B41424344454646474849F09394EAA28B'X    /* C- */
      '7D4A4B4C4D4E4F505152B7FB819CA3DB'X      /* D- */
      '5CF6535455565758595AD4E299E8E08A'X      /* E- */
      '3031323334353637383 9D2EB9A9BE900'X.    /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A4343434378527878525 2D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X       /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078787 8D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
      EBCDICCP=  259
      ASCIICP=  2125
      DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '20203D2D2B247044A22F00CE3C280021'X      /* 4- */
      '5CB0FCF90000400000CF00CB5D290000'X      /* 5- */
      '2D5EC6C5E9D7FCF9FD002700005F3ED5'X      /* 6- */
      'B6B5BABBC00000C2000025FE23272200'X      /* 7- */
      'B06162777A65706B6769E0D2E1F5E2E4'X      /* 8- */
      '00796A786C6D6F716368B32BB400F2F4'X      /* 9- */
      'EF0072736E2A64767466F0D1F1EC0000'X      /* A- */
      '00000000000000000000B400CC000000'X      /* B- */
      'E359245755A4504B00A1EC000000F600'X      /* C- */
      'F30000585A00A3004348000000B0E8EA'X      /* D- */
      'FB2052A24E26447D547C3FC80000E5E7'X      /* E- */
      '3031323334353637383D800000000000'X.     /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A6A787852F052D378'X      /* 4- */
      'BB6A6A6A6A4343434378527878525 2D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X       /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078787 8D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
      EBCDICCP=  260
      ASCIICP=  3157
      DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A0C68687A45B2E3C282B21'X      /* 4- */
      '268288898AA18C8B8DE15D242A293B5E'X      /* 5- */
      '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
      'F86A6B6C6D6E6F70717288E4E7F7F2CF'X      /* 9- */
      'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
      'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
      '7B41424344454646474849F0939495A2E4'X    /* C- */
      '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
      '5C20535455565758595AFDE299E3E0E5'X      /* E- */
      '3031323334353637383FCEA9AEBE900'X.      /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A4343434378527878525 2D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X       /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078787 8D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
      EBCDICCP=  273
      ASCIICP=  3157
      DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020837B85A0C68687A48E2E3C282B21'X      /* 4- */
      '26828889BAA18C8B8D7E9A242A293B5E'X      /* 5- */
      '2D2FB65BB7B5C78F80A5942C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE603A23F5273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      'E6E1737475767778797AADA8D1EDE8A9'X      /* A- */
      'BD9CBEFA9F40F4ACABF3AA7CEEF9EFF2'X      /* B- */
      '8441424344454646474849F093D095A2E4'X    /* C- */
      '814A4B4C4D4E4F505152D5967D97A398'X      /* D- */
      '9920535455565758595AFDE25CE3E0E5'X      /* E- */
      '3031323334353637383FCEA5DEBE900'X.      /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A4343434378527878525 2D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X       /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078787 8D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
      EBCDICCP=  274
      ASCIICP=  3157
      DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838440A0C6865CA45B2E3C282B21'X      /* 4- */
      '267B88897DA18C8B8DE15D242A293B5E'X      /* 5- */
      '2D2FB68EB7B5C78F80A5972C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE603A2385273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      'E6F973747576778797AADA8D1EDE8A9'X       /* A- */
      'BD9CBEFA9FF5F4ACABF3AA7CEE7EEFF2'X      /* B- */
      '8241424344454646474849F0939495A2E4'X    /* C- */
      '8A4A4B4C4D4E4F505152D5968197A398'X      /* D- */
      '8720535455565758595AFDE299E3E0E5'X      /* E- */
```

```
      '30313233343536373839FCEA9AEBE900'X.     /* F- */
    :PFNTWTH
     DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X      /* 4- */
      'BB6A6A6A6A434343437785278785252D3'X      /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '78959595955252525278527 8D352D36E'X      /* 7- */
      'AD6A786A786A52787843606078787 8D3'X      /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '78787878787878787856B2B2B2B278'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=  275
      ASCIICP=  3157
      DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '2020838485A06086D0A4902E3C282B21'X      /* 4- */
      '267B8889BAA18C8B8DE124802A293B5E'X      /* 5- */
      '2D2FB68EB7B5408F5DA5872C255F3E3F'X      /* 6- */
      '9B5BD2D3D4D6D7D8DEC63AE5C7273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X      /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
      'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
      'E4414243444546474849F0939495A27B'X      /* C- */
      '824A4B4C4D4E4F505152D5968197A398'X      /* D- */
      '5C20535455565758595AFDE299E3E023'X      /* E- */
      '30313233343536373839FCEA9AEBE900'X.     /* F- */
    :PFNTWTH
     DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X      /* 4- */
      'BB6A6A6A6A434343437785278785252D3'X      /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '78959595955252525278527 8D352D36E'X      /* 7- */
      'AD6A786A786A52787843606078787 8D3'X      /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '78787878787878787856B2B2B2B278'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=  276
      ASCIICP=  3157
      DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '202083845BA0C68687A4852E3C282B21'X      /* 4- */
      '267B88897DA18C8B8DE1EF242A293B5E'X      /* 5- */
      '2D2FB68EB7B5C78F80A5972C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X      /* 8- */
      'F86A6B6C6D6E6F707172A6A7915C92CF'X      /* 9- */
      'E6F97374757677787 97AADA8D1EDE8A9'X      /* A- */
      'BD9CBEFA9FF5F4ACABF3AA7CEE7E5DF2'X      /* B- */
      '824142434445464748 49F0939495A2E4'X      /* C- */
      '8A4A4B4C4D4E4F505152D5968197A398'X      /* D- */
      'F720535455565758595AFDE299E3E0E5'X      /* E- */
      '30313233343536373839FCEA9AEBE900'X.     /* F- */
    :PFNTWTH
     DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X      /* 4- */
      'BB6A6A6A6A434343437785278785252D3'X      /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '78959595955252525278527 8D352D36E'X      /* 7- */
      'AD6A786A786A52787843606078787 8D3'X      /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '78787878787878787856B2B2B2B278'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=  277
      ASCIICP=  3157
      DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '2020838485A0C67B87A4232E3C282B21'X      /* 4- */
      '26828889BAA18C8B8DE1CF8F2A293B5E'X      /* 5- */
      '2D2FB68EB7B5C72480A59B2C255F3E3F'X      /* 6- */
      'D090D2D3D4D6D7D8DE603A929D273D22'X      /* 7- */
      '4061626364656666768 69AEAFD0ECE7F1'X      /* 8- */
      'F86A6B6C6D6E6F707172A6A77BF75B5D'X      /* 9- */
      'E6817374757677787 97AADA8D1EDE8A9'X      /* A- */
      'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
      '914142434445464748 49F0939495A2E4'X      /* C- */
      '864A4B4C4D4E4F505152D5967E97A398'X      /* D- */
      '5C20535455565758595AFDE299E3E0E5'X      /* E- */
      '30313233343536373839FCEA9AEBE900'X.     /* F- */
    :PFNTWTH
     DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X      /* 4- */
      'BB6A6A6A6A434343437785278785252D3'X      /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '78959595955252525278527 8D352D36E'X      /* 7- */
      'AD6A786A786A52787843606078787 8D3'X      /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '78787878787878787856B2B2B2B278'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=  278
      ASCIICP=  3157
      DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020837B85A0C67D87A4F52E3C282B21'X      /* 4- */
      '26608889BAA18C8B8DE1CF8F2A293B5E'X      /* 5- */
      '2D2FB623B7B5C72480A5942C255F3E3F'X      /* 6- */
      '9B5CD2D3D4D6D7D8DE823A8E99273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X      /* 8- */
      'F86A6B6C6D6E6F707172A6A791F7925D'X      /* 9- */
      'E6817374757677787 97AADA8D1EDE8A9'X      /* A- */
      'BD9CBEFA9F5BF4ACABF3AA7CEEF9EFF2'X      /* B- */
      '844142434445464748 49F093D095A2E4'X      /* C- */
      '864A4B4C4D4E4F505152D5967E97A398'X      /* D- */
      '902053545556575859 5AFDE240E3E0E5'X      /* E- */
      '30313233343536373839FCEA9AEBE900'X.     /* F- */
    :PFNTWTH
     DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X      /* 4- */
      'BB6A6A6A6A434343437785278785252D3'X      /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '78959595955252525278527 8D352D36E'X      /* 7- */
      'AD6A786A786A52787843606078787 8D3'X      /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
```

```
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=  279
      ASCIICP=  3157
      DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838440A0C6865CA4F82E3C282B21'X      /* 4- */
      '267B88897DA18C8B8DE1F5242A293B5E'X      /* 5- */
      '2D2FB68EB7B5C78F80A5972C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DEE63A9C85273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
      '5B6A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      '60F97374757677 8797AADA8D1EDE8A9'X      /* A- */
      'BD23BEFA9F5DF4ACABF3AA7CEE7EEFF2'X      /* B- */
      '824142434444546474849F0939495A2E4'X     /* C- */
      '8A4A4B4C4D4E4F505152D59681D0A398'X      /* D- */
      '8720535455565758595AFDE299E3E0E5'X      /* E- */
      '303132333435363738 39FCEA9AEBE900'X.    /* F- */
    :PFNTWTH
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852 78785252D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X       /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078 78D3'X       /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=  280
      ASCIICP=  3157
      DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '202083847BA0C6865CA4F82E3C282B21'X      /* 4- */
      '265D88897DA18C8B7EE182242A293B5E'X      /* 5- */
      '2D2FB68EB7B5C78F80A5952C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE973A9CF5273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
      '5B6A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      'E68D737475767778 797AADA8D1EDE8A9'X     /* A- */
      'BD23BEFA9F40F4ACABF3AA7CEEF9EFF2'X      /* B- */
      '854142434445464748 49F09394D0A2E4'X     /* C- */
      '8A4A4B4C4D4E4F505152D5968160A398'X      /* D- */
      '8720535455565758595AFDE299E3E0E5'X      /* E- */
      '303132333435363738 39FCEA9AEBE900'X.    /* F- */
    :PFNTWTH
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852 78785252D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X       /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078 78D3'X       /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=  281
      ASCIICP=  3157
      DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A0C68687A49C2E3C282B7C'X      /* 4- */
      '26828889BAA18C8B8DE121BE2A293BAA'X      /* 5- */
      '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      'E6EE737475767778 797AADA8D1EDE8A9'X     /* A- */
      'BD5B5CFA9FF5F4ACABF35E5D7EF9EFF2'X      /* B- */
      '7B414243444546474849F0939495A2E4'X      /* C- */
      '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
      '242053545556575 8595AFDE299E3E0E5'X     /* E- */
      '303132333435363738 39FCEA9AEBE900'X.    /* F- */
    :PFNTWTH
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852 78785252D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X       /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078 78D3'X       /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=  282
      ASCIICP=  3157
      DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A07B867EA45B2E3C282B21'X      /* 4- */
      '26828889BAA18C8B8DE15D242A293B5E'X      /* 5- */
      '2D2FB68EB7B5238F5CA5E42C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE603AC7E5273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      'E6877374757677 8797AADA8D1EDE8A9'X      /* A- */
      'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
      'C641424344454 6474849F0939495A2D0'X     /* C- */
      'EF4A4B4C4D4E4F505152D5968197A398'X      /* D- */
      '8020535455565758595AFDE299E3E040'X      /* E- */
      '303132333435363738 39FCEA9AEBE900'X.    /* F- */
    :PFNTWTH
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852 78785252D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X       /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078 78D3'X       /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=  283
      ASCIICP=  3157
      DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A0C68687D05B2E3C282B7C'X      /* 4- */
      '26828889BAA18C8B8DE15D502A293BAA'X      /* 5- */
```

```
      '2D2FB68EB7B5C78F8023A42C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE603AA540273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      'E6F97374757677 8797AADA8D1EDE8A9'X      /* A- */
      'BD9CBEFA9FF5F4ACABF35E21EE7EEFF2'X      /* B- */
      '7B41424344454 6474849F0939495A2E4'X     /* C- */
      '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
      '5C20535455565758595AFDE299E3E0E5'X      /* E- */
      '303132333435363738 39FCEA9AEBE900'X.    /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437 85278785252D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '7895959595525252 52785278D352D36E'X     /* 7- */
      'AD6A786A786A5278784360 60787878D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  284
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A0C68687D05B2E3C282B7C'X      /* 4- */
      '26828889BAA18C8B8DE15D242A293BAA'X      /* 5- */
      '2D2FB68EB7B5C78F8023A42C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE603AA540273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      'E6F97374757677 8797AADA8D1EDE8A9'X      /* A- */
      'BD9CBEFA9FF5F4ACABF35E21EE7EEFF2'X      /* B- */
      '7B41424344454 6474849F0939495A2E4'X     /* C- */
      '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
      '5C20535455565758595AFDE299E3E0E5'X      /* E- */
      '303132333435363738 39FCEA9AEBE900'X.    /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437 85278785252D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '7895959595525252 52785278D352D36E'X     /* 7- */
      'AD6A786A786A5278784360 60787878D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  285
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A0C68687A4242E3C282B7C'X      /* 4- */
      '26828889BAA18C8B8DE1219C2A293BAA'X      /* 5- */
      '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      'E6EE7374757677 8797AADA8D1EDE8A9'X      /* A- */
      'BD5BBEFA9FF5F4ACABF35E5D7EF9EFF2'X      /* B- */
      '7B41424344454 6474849F0939495A2E4'X     /* C- */
      '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
      '5C20535455565758595AFDE299E3E0E5'X      /* E- */
      '303132333435363738 39FCEA9AEBE900'X.    /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437 85278785252D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '7895959595525252 52785278D352D36E'X     /* 7- */
      'AD6A786A786A5278784360 60787878D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  290
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '20A1A2A3A4A5A6A7A8A9E12E3C282B7C'X      /* 4- */
      '26AAABACADAEAF00B000215C2A293BE3'X       /* 5- */
      '2D2F000000000000000000002C255F3E3F'X     /* 6- */
      '0000000000000000000003A2340273D22'X      /* 7- */
      '00B1B2B3B4B5B6B7B8B9BA00BBBCBDBE'X       /* 8- */
      'BFC0C1C2C3C4C5C6C7C8C90000CACBCC'X       /* 9- */
      '007ECDCECFD0D1D2D3D4D500D6D7D8D9'X       /* A- */
      '0000000000000000000000DADBDCDDDEDF'X     /* B- */
      '7B41424344454647484900000000000000'X     /* C- */
      '004A4B4C4D4E4F505152000000000000'X       /* D- */
      '2420535455565758595A000000000000'X       /* E- */
      '303132333435363738390000000000000'X.     /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437 85278785252D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '7895959595525252 52785278D352D36E'X     /* 7- */
      'AD6A786A786A5278784360 60787878D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  293
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A0C68687A45B2E3C282B21'X      /* 4- */
      '268288898AA18C8B8DE15D242A293B5E'X      /* 5- */
      '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
      'E67E737475767778 797AADA8D1EDE8A9'X     /* A- */
      'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
      '7B41424344454 6474849F0939495A2E4'X     /* C- */
      '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
      '5C20535455565758595AFDE299E3E0E5'X      /* E- */
      '303132333435363738 39FCEA9AEBE900'X.    /* F- */
     :PFNTWTH
     DATA =
```

```
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X      /* 4- */
      'BB6A6A6A6A4343434378527878525 2D3'X   /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X     /* 6- */
      '7895959595525252527852 78D352D36E'X   /* 7- */
      'AD6A786A786A52787843606078787 8D3'X   /* 8- */
      '78437843BB787878785678789E78D378'X     /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X     /* A- */
      '787878AD787878D3D3D3D37878787878'X     /* B- */
      '6AAD95A3B29590ADB252527878787878'X     /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X     /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X     /* E- */
      '7878787878787878787856B2B2B2B278'X.    /* F- */

     :PMLGEBCTBL
       EBCDICCP=  297
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838440A0C6865CA4F82E3C282B21'X     /* 4- */
      '267B88897DA18C8B8DE1F5242A293B5E'X     /* 5- */
      '2D2FB68EB7B5C78F80A5972C255F3E3F'X     /* 6- */
      '9B90D2D3D4D6D7D8DEE63A9C85273D22'X     /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X    /* 8- */
      '5B6A6B6C6D6E6F707172A6A791F792CF'X     /* 9- */
      '60F9737475767778797AADA8D1EDE8A9'X     /* A- */
      'BD23BEFA9F5DF4ACABF3AA7CEE7EEFF2'X     /* B- */
      '8241424344454646474849F0939495A2E4'X   /* C- */
      '8A4A4B4C4D4E4F505152D59681D0A398'X     /* D- */
      '8720535455565758595AFDE299E3E0E5'X     /* E- */
      '3031323334353637383 9FCEA9AEBE900'X.   /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X       /* 4- */
      'BB6A6A6A6A4343434378527878525 2D3'X    /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '7895959595525252527852 78D352D36E'X    /* 7- */
      'AD6A786A786A52787843606078787 8D3'X    /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  310
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A0C68687A45B2E3C282B21'X     /* 4- */
      '268288898AA18C8B8DE15D242A293B5E'X     /* 5- */
      '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X     /* 6- */
      '9B90D2D3D4D6D7D8DE603A2340273D22'X     /* 7- */
      '9D6162636465666768 69AEAFD0ECE7F1'X    /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X     /* 9- */
      'E67E737475767778797AADA8D1EDE8A9'X     /* A- */
      'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X     /* B- */
      '7B4142434445464748 49F0939495A2E4'X    /* C- */
      '7D4A4B4C4D4E4F505152D5968197A398'X     /* D- */
      '5C20535455565758595AFDE299E3E0E5'X     /* E- */
      '3031323334353637383 9FCEA9AEBE900'X.   /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X       /* 4- */
      'BB6A6A6A6A4343434378527878525 2D3'X    /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '7895959595525252527852 78D352D36E'X    /* 7- */
      'AD6A786A786A52787843606078787 8D3'X    /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  330
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838461A0C79F87865B2E3C282B21'X     /* 4- */
      '2682A98985A18C9692E15D242A293B5E'X     /* 5- */
      '2D2FB68E41B5C6AC808FD82C255F3E3F'X     /* 6- */
      'F390A8D3DED6D79591603A2340273D22'X     /* 7- */
      'F4616263646566676869 8E5D0ECFDAD'X     /* 8- */
      'F86A6B6C6D6E6F70717288E4E7F7F2CF'X     /* 9- */
      'A5F1737475767778797AB8D5D1EDFCB8'X     /* A- */
      'FA4CBEDDBDF5A7ABA68D9DE3E6F9EFEE'X     /* B- */
      'D4414243444546474849F09394EAA28B'X     /* C- */
      'D24A4B4C4D4E4F505152B7FB819CA379'X     /* D- */
      'A4205354555657585 95A32E299E8E08A'X    /* E- */
      '3031323334353637383933EB9A9BE900'X.    /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X       /* 4- */
      'BB6A6A6A6A4343434378527878525 2D3'X    /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '7895959595525252527852 78D352D36E'X    /* 7- */
      'AD6A786A786A52787843606078787 8D3'X    /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  340
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A0C68687A45B2E3C282B21'X     /* 4- */
      '268288898AA18C8B8DE15D242A293B5E'X     /* 5- */
      '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X     /* 6- */
      '9B90D2D3D4D6D7C4DE603A2340273D22'X     /* 7- */
      'A7616263646566676869 AEAFD0ECE7F1'X    /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X     /* 9- */
      'E67E737475767778797AADA8D1EDE8A9'X     /* A- */
      'BD9C9DFAB8F5F4ACABF3AA7CEEF9EFDB'X     /* B- */
      '7B4142434445464748492D609495A2E4'X     /* C- */
      '7D4A4B4C4D4E4F505152FB7E8197A35E'X     /* D- */
      '5CF6535455565758595AFD5F99E3E0E5'X     /* E- */
      '3031323334353637383 9FCEA9AEBE900'X.   /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X       /* 4- */
      'BB6A6A6A6A4343434378527878525 2D3'X    /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '7895959595525252527852 78D352D36E'X    /* 7- */
      'AD6A786A786A52787843606078787 8D3'X    /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
```

```
      '78787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  420
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020F1F0E09FC1C2A2C3C02E3C282B7C'X     /* 4- */
      '26A5C40000C6C7A8A9C821242A293BDC'X     /* 5- */
      '2D2FC9AACAABCBADCCAE7C2C255F3E3F'X     /* 6- */
      'CDAFCECFD0D1D2BCD3AC3A2340273D22'X     /* 7- */
      'BD6162636465666768696D4BED5EBD6D7'X    /* 8- */
      'D86A6B6C6D6E6F707172DFC5D9ECEEED'X     /* 9- */
      'DADD737475767778797AF7BAE1F8E2FC'X     /* A- */
      'E3FBF9FA999A00009D9EE4EFE5F2E6F3'X     /* B- */
      'BB41424344454647484 9A1E700F400E8'X    /* C- */
      'BF4A4B4C4D4E4F505152E9F5FDF6EAB0'X     /* D- */
      'DEF6535455565758595AB1B200B3B4B5'X     /* E- */
      '3031323334353637383900B6B7B8B900'X.    /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X       /* 4- */
      'BB6A6A6A6A4343434378527878 5252D3'X    /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '78959595955252525278 5278D352D36E'X    /* 7- */
      'AD6A786A786A52787843606078787 8D3'X    /* 8- */
      '78437843BB787878785678789E78D378'X     /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X     /* A- */
      '787878AD787878D3D3D3D37878787878'X     /* B- */
      '6AAD95A3B29590ADB252527878787878'X     /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X     /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X     /* E- */
      '78787878787878787856B2B2B2B278'X.      /* F- */

     :PMLGEBCTBL
       EBCDICCP=  423
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '20A4A5A6A7A8A9AAACAD5B2E3C282B21'X     /* 4- */
      '26B5B6B7B8BDBEC6C7CF5D242A293B5E'X     /* 5- */
      '2D2FD0D1D2D3D4D500007C2C255F3E3F'X     /* 6- */
      '00868D8F0090929598603A9CF5273D22'X     /* 7- */
      '8E6162636465666768696D6D7D8DDDEE0'X    /* 8- */
      '996A6B6C6D6E6F707172E1E2E3E4E5E6'X     /* 9- */
      '9AF9737475767778797AE7E8E9EAEBEC'X     /* A- */
      '009B9D9EA09FA2A3FBFDEDEEF2F3F4F6'X     /* B- */
      'F7414243444546474849F0FA83858488'X     /* C- */
      'EF4A4B4C4D4E4F505152F1828A898C8B'X     /* D- */
      'F820535455565758595AAB9493969781'X     /* E- */
      '3031323334353637383979 8780000000'X.   /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X       /* 4- */
      'BB6A6A6A6A4343434378527878 5252D3'X    /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '78959595955252525278 5278D352D36E'X    /* 7- */
      'AD6A786A786A52787843606078787 8D3'X    /* 8- */
      '78437843BB787878785678789E78D378'X     /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X     /* A- */
      '787878AD787878D3D3D3D37878787878'X     /* B- */
      '6AAD95A3B29590ADB252527878787878'X     /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X     /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X     /* E- */
      '78787878787878787856B2B2B2B278'X.      /* F- */

     :PMLGEBCTBL
       EBCDICCP=  424
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2080818283848586878 9B2E3C282B7C'X     /* 4- */
      '26898A8B8C8D8E8F909121242A293BAA'X     /* 5- */
      '2D2F92939495969798997C2C255F3E3F'X     /* 6- */
      '009A0000200000005F603A2340273D22'X     /* 7- */
      '00616263646566676869AEAF000000F1'X     /* 8- */
      'F86A6B6C6D6E6F707172000000000000'X     /* 9- */
      'E67E737475767778797A000000000052'X     /* A- */
      '5E9C9DF900F5F4ACAB005B5D2D000078'X     /* B- */
      '7B41424344454647 8492D00000000000'X    /* C- */
      '7D4A4B4C4D4E4F505152000000000000'X     /* D- */
      '5CF6535455565758595AFD0000000000'X     /* E- */
      '3031323334353637383900000000000000'X.  /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X       /* 4- */
      'BB6A6A6A6A4343434378527878 5252D3'X    /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '78959595955252525278 5278D352D36E'X    /* 7- */
      'AD6A786A786A52787843606078787 8D3'X    /* 8- */
      '78437843BB787878785678789E78D378'X     /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X     /* A- */
      '787878AD787878D3D3D3D37878787878'X     /* B- */
      '6AAD95A3B29590ADB252527878787878'X     /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X     /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X     /* E- */
      '78787878787878787856B2B2B2B278'X.      /* F- */

     :PMLGEBCTBL
       EBCDICCP=  500
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A0C68687A45B2E3C282B21'X     /* 4- */
      '268288898AA18C8B8DE15D242A293B5E'X     /* 5- */
      '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X     /* 6- */
      '9B90D2D3D4D6D7D8DE603A2340273D22'X     /* 7- */
      '9D616263646566676869AEAFD0ECE7F1'X     /* 8- */
      'F86A6B6C6D6E6F707172A6A791F792CF'X     /* 9- */
      'E67E737475767778797AADA8D1EDE8A9'X     /* A- */
      'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X     /* B- */
      '7B41424344454647 849F0939495A2E4'X     /* C- */
      '7D4A4B4C4D4E4F505152D5968197A398'X     /* D- */
      '5C2053545565758595AFDE299E3E0E5'X      /* E- */
      '3031323334353637383 9FCEA9AEBE900'X.   /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X       /* 4- */
      'BB6A6A6A6A4343434378527878 5252D3'X    /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '78959595955252525278 5278D352D36E'X    /* 7- */
      'AD6A786A786A52787843606078787 8D3'X    /* 8- */
      '78437843BB787878785678789E78D378'X     /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X     /* A- */
      '787878AD787878D3D3D3D37878787878'X     /* B- */
      '6AAD95A3B29590ADB252527878787878'X     /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X     /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X     /* E- */
      '78787878787878787856B2B2B2B278'X.      /* F- */

     :PMLGEBCTBL
       EBCDICCP=  870
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '20208384EEA0C79F87865B2E3C282B21'X     /* 4- */
      '2682A98985A18C9692E15D242A293B5E'X     /* 5- */
      '2D2FB68EF1B5C6AC808F7C2C255F3E3F'X     /* 6- */
      'F390A8D3DED6D79591603A2340273D22'X     /* 7- */
      'F461626364656667686998E5D0ECFDAD'X     /* 8- */
```

```
      'F86A6B6C6D6E6F70717288E4E7F7F2CF'X      /* 9- */
      'A57E7374757677787 97AB8D5D1EDFCB8'X     /* A- */
      'FAA4BEDDBDF5A7ABA68D9DE3E6F9EF9E'X      /* B- */
      '7B414243444546474849F09394EAA28B'X      /* C- */
      '7D4A4B4C4D4E4F505152B7FB819CA3DB'X      /* D- */
      '5CF6535455565758595AD4E299E8E08A'X      /* E- */
      '30313233343536373839D2EB9A9BE900'X.     /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852787852 52D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078787 8D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  871
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A0C68687A4E82E3C282B21'X      /* 4- */
      '268288898AA18C8B8DE192242A293B99'X      /* 5- */
      '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
      '9B90D2D3D4D6D7D8DED03A23D1273D22'X      /* 7- */
      '9D6162636465666676869AEAF60EC7BF1'X     /* 8- */
      'F86A6B6C6D6E6F707172A6A77DF75DCF'X      /* 9- */
      'E6947374757677 8797AADA840ED5BA9'X      /* A- */
      'BD9CBEFAB8F5F4ACABF3AA7CEEF95C9E'X      /* B- */
      'E7414243444546474849F0937E95A2E4'X      /* C- */
      '914A4B4C4D4E4F505152FB968197A398'X      /* D- */
      'EFF6535455565758595AFDE25EE3E0E5'X      /* E- */
      '30313233343536373839FCEA9AEBE900'X.     /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852787852 52D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078787 8D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  880
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '202080828486888A8C8E5B2E3C282B21'X      /* 4- */
      '2690929496989A9EEF815D242A293B5E'X      /* 5- */
      '2D2F838587898B8D8F917C2C255F3E3F'X      /* 6- */
      '9395972D999B9CA0A2603A2340273D22'X      /* 7- */
      '9A6162636465666768 69A6A8AAACB5B7'X     /* 8- */
      'BD6A6B6C6D6E6F707172C6D0D2D4D6D8'X      /* 9- */
      'DE7E737475767778797AE1E3E5E7E9EB'X      /* A- */
      'EDF1F3F5F7F9FB9E9DA1A3A5A7A9ABAD'X      /* B- */
      '7B414243444546474849B6B8BEC7D1D3'X      /* C- */
      '7D4A4B4C4D4E4F505152D5D7DDE0E2E4'X      /* D- */
      '5CCF535455565758595AE6E8EAECEEF2'X      /* E- */
      '3031323334353637383 9F4F6F8FAFC00'X.    /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852787852 52D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078787 8D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  892
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020000000000000000005B2E3C282B21'X     /* 4- */
      '260000000000000000005D242A293B00'X      /* 5- */
      '2D2F008E0000008F00A5002C25003E3F'X      /* 6- */
      '00000000000000C400003A2340273D22'X      /* 7- */
      'A7616263646566676869000000000000'X      /* 8- */
      '006A6B6C6D6E6F707172000000009200'X      /* 9- */
      '0000737475767778797A000000000000'X      /* A- */
      '009C9D000000000000000007C000000DB'X     /* B- */
      '7B414243444546474849006000000000'X      /* C- */
      '7D4A4B4C4D4E4F505152007E0000005E'X      /* D- */
      '5C2053545556575859 5A005F99000000'X     /* E- */
      '3031323334353637383900009A000000'X.     /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852787852 52D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '7895959595525252527852 78D352D36E'X     /* 7- */
      'AD6A786A786A52787843606078787 8D3'X     /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '7878787878787878787856B2B2B2B278'X.     /* F- */

     :PMLGEBCTBL
       EBCDICCP=  893
       ASCIICP=  3157
       DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020008400000086000 05B2E3C282B21'X     /* 4- */
      '260000000000000000E15D242A293B5E'X      /* 5- */
      '2D2F008E0000008F00A5002C255F3E3F'X      /* 6- */
      'A60000000000A9C400603A2340273D22'X      /* 7- */
      'A7616263646566676869000000000B9BA'X     /* 8- */
      '006A6B6C6D6E6F707172000091F792CF'X      /* 9- */
      '007E737475767778797A000000000000'X      /* A- */
      '009C9D0000F500000000007C00F9EFDB'X      /* B- */
      '7B414243444546474849000094000000'X      /* C- */
      '7D4A4B4C4D4E4F50515200008100005E'X      /* D- */
      '5C2053545556575859A000099000000'X       /* E- */
      '3031323334353637383900009A000000'X.     /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A787852F052D378'X        /* 4- */
      'BB6A6A6A6A434343437852787852 52D3'X     /* 5- */
```

```
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '78959595955252525278278D352D36E'X       /* 7- */
      'AD6A786A786A527878436060787878D3'X      /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '78787878787878787856B2B2B2B278'X.       /* F- */

     :PMLGEBCTBL
      EBCDICCP=  905
      ASCIICP=  3157
      DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '2020838485A000917BA4802E3C282B21'X      /* 4- */
      '268288898AA18C8B8DE1A6982A293B5E'X      /* 5- */
      '2D2FB686B7B500925BA5AD2C255F3E3F'X      /* 6- */
      '0090D2D3D4D6D7D8DED53A99B8273D9A'X      /* 7- */
      'F461626364656676869E886C7ED007C'X       /* 8- */
      'F86A6B6C6D6E6F707172A99B9FF700CF'X      /* 9- */
      'E6947374757677787797AE78FC6EC0040'X     /* A- */
      'FA9CBE7DBDF55D00AB24A89DACF9EF9E'X      /* B- */
      '8741424344454647484 9F0937E95A2E5'X     /* C- */
      'A74A4B4C4D4E4F50515260965C81A300'X      /* D- */
      '81F6535455565758595AFDE223E3E0E4'X      /* E- */
      '3031323334353637383 9FCEA22EBE900'X.    /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A6A787852F052D378'X      /* 4- */
      'BB6A6A6A6A434343437 85278785252D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '78959595955252525278278D352D36E'X       /* 7- */
      'AD6A786A786A527878436060787878D3'X      /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '78787878787878787856B2B2B2B278'X.       /* F- */

     :PMLGEBCTBL
      EBCDICCP=  259
      ASCIICP=  3157
      DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '20200000000000001A2F5B003C28B800'X      /* 4- */
      '00EE3C3E0000000000005DFA0029F95E'X      /* 5- */
      '2D000000BA003C3E0000EF0C0B5F3E00'X      /* 6- */
      '00000000001C00000060F6F1F82722A9'X      /* 7- */
      '0000E100000000000000DAC3C0B30000'X      /* 8- */
      '00000000E600000000000C2C5C1000000'X     /* 9- */
      '007E0000009E00000000BFB4D9C40000'X      /* A- */
      '00000000000000000000000079CCFBE'X       /* B- */
      '7B000000001B0000F418F00400007C00'X      /* C- */
      '7D00F50000001900000000FE00EE0000'X      /* D- */
      '5C20001A000000000000000AA101E0000'X     /* E- */
      '00FBFDFC0000000000009D0000000000'X.     /* F- */
     :PFNTWTH
     DATA =
    /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
      '48486A6A6A6A6A6A6A787852F052D378'X      /* 4- */
      'BB6A6A6A6A434343437 85278785252D3'X     /* 5- */
      '5252ADADADADADADA3B27852D378F06A'X      /* 6- */
      '78959595955252525278278D352D36E'X       /* 7- */
      'AD6A786A786A527878436060787878D3'X      /* 8- */
      '78437843BB787878785678789E78D378'X      /* 9- */
      '86785B437878AD78786A526AB2AD8B78'X      /* A- */
      '787878AD787878D3D3D3D37878787878'X      /* B- */
      '6AAD95A3B29590ADB252527878787878'X      /* C- */
      '6A5BA895D8B2AD8BAD9E567878787878'X      /* D- */
      '52488295B2ADE2ADAD9A56ADADADADAD'X      /* E- */
      '78787878787878787856B2B2B2B278'X.       /* F- */

   :EPMLGMAPTBL.
 :EWSCST.
```

Step 4: Creating the Workstation Customizing Object

After you change and save the source, create the workstation customizing object using the Create Work Station Customizing Object (CRTWSCST) command. Specify the following parameter value:

Workstation customizing object name
WSCST(CSTHPDA)

Library LIB(CSTLIB)

Source member
SRCMBR(CSTHPDA)

Source file
SRCFILE(CSTSRC)

Library LIB(CSTLIB)

Text TEXT('Workstation customizing object for HPDA ASCII Printer')

Note: This step was performed several times.

Step 5: Varying On the Device

Change the device description for the ASCII printer and specify your customizing object CSTHPDA for the workstation customizing object (WSCST) parameter.

To activate the workstation customizing function, vary off and then vary on the printer so that the customizing object is downloaded to the workstation controller. When you print the test document on the Hewlett-Packard printer, it shows the characteristics for which you customized. In addition, the strange characters that appeared at the top of the page are gone.

Note: This step was performed several times. When you vary the device on and off many times as you do when you test a new workstation customizing object, the workstation controller storage can fill up. When this occurs, the device no longer varies on and an error message is sent to the QSYSOPR message queue. To correct this situation, it is necessary to vary off the workstation controller, and then vary it on again specifying RESET(*YES). This clears the storage in the workstation controller and allows the mapping tables in the new workstation customizing object to be loaded.

Appendix A. Twinaxial Keyboard Layouts

This appendix provides illustrations of the five basic keyboard types that are used with twinaxial displays. The core area of each keyboard is not labeled because the intention is to show the physical layout of the keyboard and not necessarily the positions of individual keys.

The keyboard layouts shown are for a United States English (USB) layout. For other national languages, slight differences may exist in terms of the number of keys in certain rows of the core areas of the keyboards. In most cases, it is the core area for each keyboard layout that is functionally different from one national language to the next. This area contains the alphanumeric characters and some function keys. The location of the non-alphanumeric keys is almost always the same from one national language to the next.

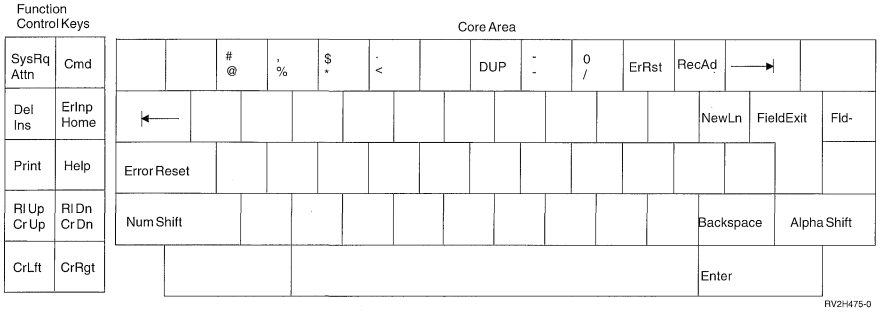
Use the following keyboard illustrations to help you select the correct keyboard type when you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command to retrieve the source for your twinaxial display. The following list matches the values for the keyboard attached parameter to the name of the actual keyboard.

- Specify *DATA5250 for the 5250 data entry keyboard
- Specify *TYPE5250 for the 5250 typewriter keyboard
- Specify *DATA122 for the 122-key data entry keyboard
- Specify *TYPE122 for the 122-key typewriter keyboard
- Specify *ENHANCED for the Enhanced style keyboard

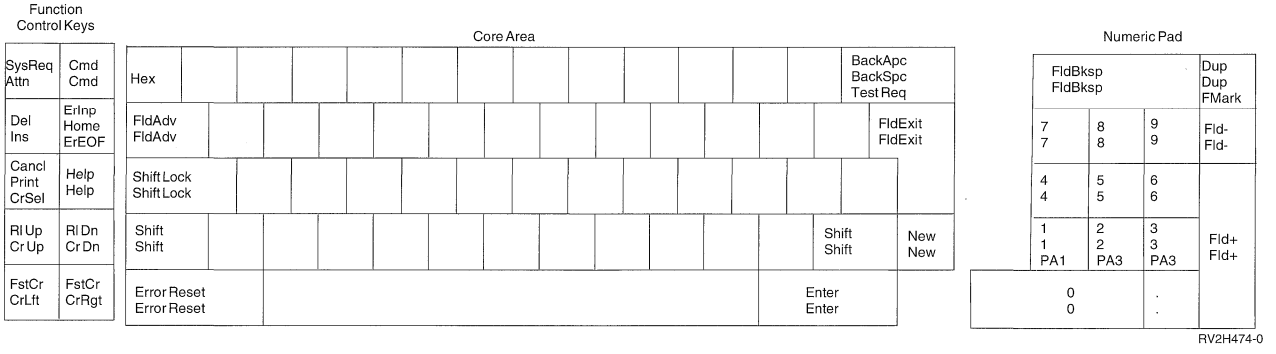
Notes:

1. Not all the different keyboard styles are supported for all the different keyboard languages the AS/400 system can support. For example, there are very few national languages that have a data entry layout defined for the 122-key keyboard.
2. The 5250 data entry keyboard has no numeric key pad area.
3. The number of keys on an Enhanced keyboard varies from one national language to another. The Enhanced keyboard shown in this appendix has 102 keys. Other Enhanced keyboards may have 102 keys, 103 keys, 105 keys, or 107 keys. The layout of the keyboard is the same; however, some of the larger keys, such as the Shift key above the Enter key, may be split into two keys rather than one large key.

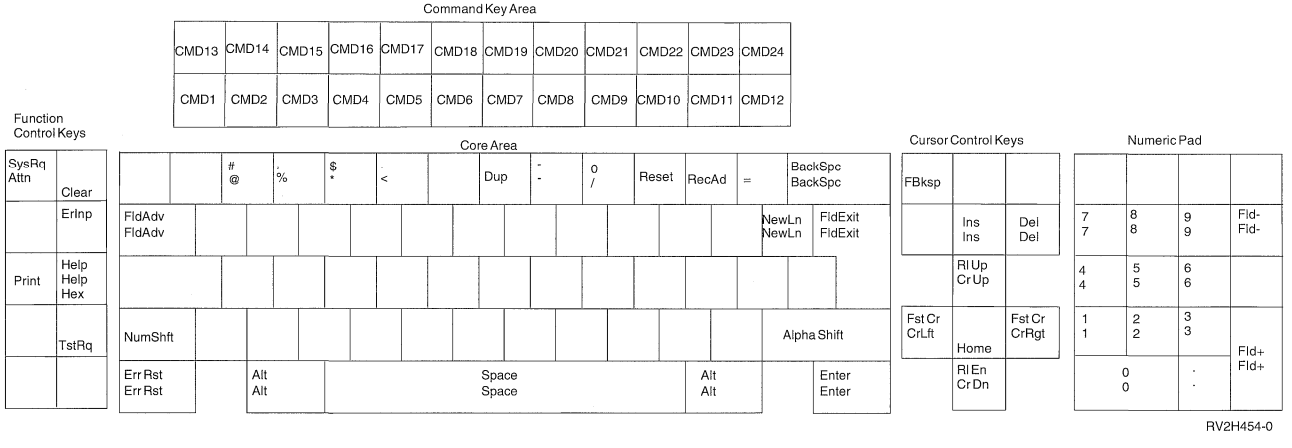
5250 Data Entry Keyboard Layout



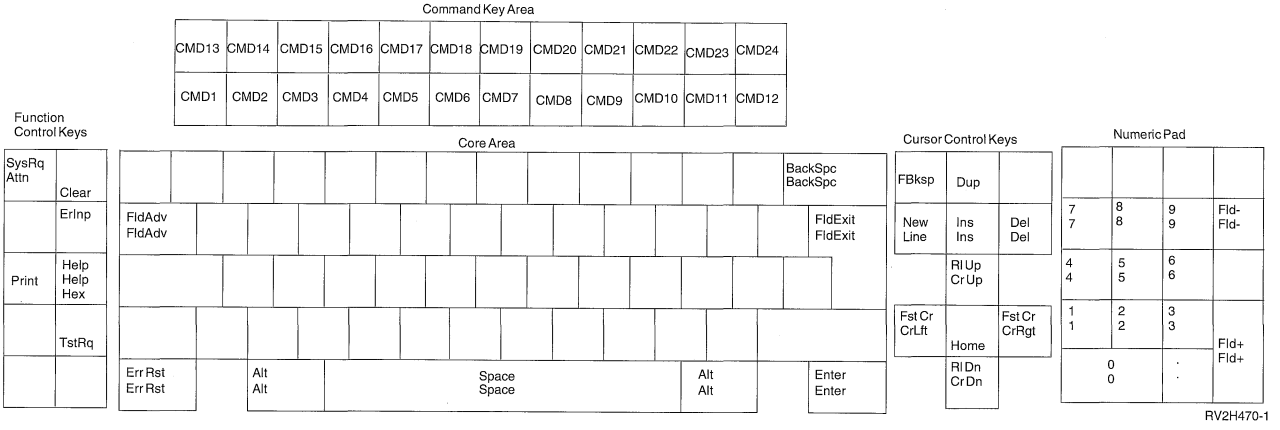
5250 Typewriter Keyboard Layout



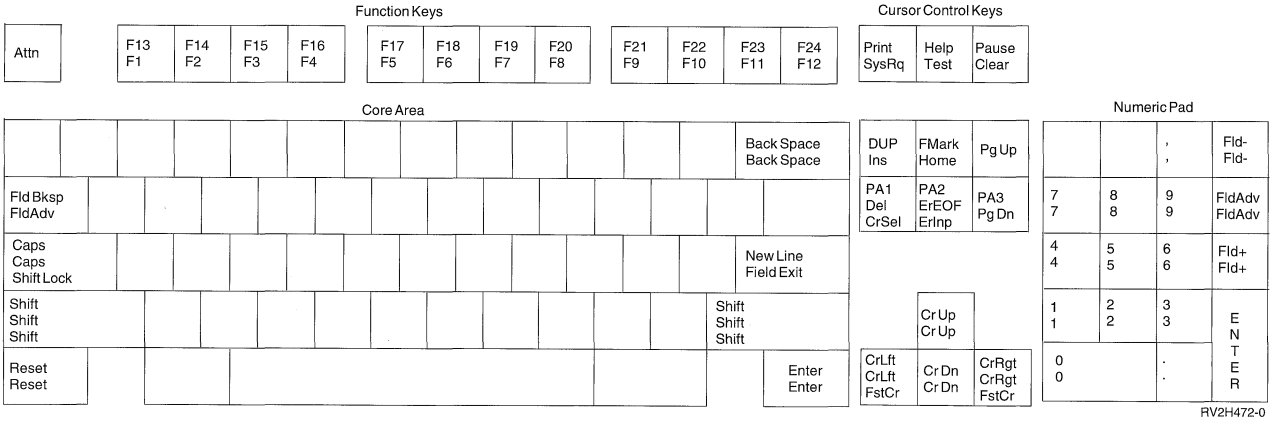
122-Key Data Entry Keyboard Layout



122-Key Typewriter Keyboard Layout



Enhanced Keyboard Layout



Appendix B. Source Code Examples

You can retrieve copies of source code to customize workstations. The following examples show what the retrieved source code looks like for each of these types of devices:

- Twinaxial display
- Twinaxial display with an attached ASCII printer that uses the emulator on the display
- ASCII display
- ASCII printer that uses the host print transform function
- ASCII printer that uses the emulator on the workstation controller

The first example is for both the twinaxial display and the twinaxial display with an attached printer that uses the emulator on the display.

Note: The retrieved source code for twinaxial displays (other than the 3477 Model H, 3486, 3487, and 3488) does not contain the code for the printer definition table. The remaining examples are for:

- The ASCII display
- The ASCII printer that uses the host print transform function
- The ASCII printer attached to the ASCII workstation controller that uses the emulator on the workstation controller

Use these examples to help you understand and work with the values for the tags used in the coding for workstation customizing.

Source Code for 3477 Twinaxial Display

The source for this example is created when you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command specifying the following parameters:

Device type
DEVTYPE(3477)

Keyboard language type
KBDTYPE(USI)

Keyboard attached
KBD(*TYPE122)

Note: Device type 3477 is one of three device types that you can specify to customize an ASCII printer attached to a twinaxial display. The other two device types you can specify to include the printer definition table in your source are the 3486 and the 3487.

```
:WSCST DEVCLASS=TWINAXPRT.

  :TKBDTBL
      LANGTYPE = USI
      KBDTYPE = TYPE122   .
    :TKSTATE
      MODE = NONE   SHIFT = UNSHIFT
      DATA =
        '2300'X      /* 00 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0AA9'X      /* 01 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0AA7'X      /* 02 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A83'X      /* 03 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0AA5'X      /* 04 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A82'X      /* 05 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A95'X      /* 06 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A94'X      /* 07 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '106B'X      /* 08 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '104B'X      /* 09 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0B61'X      /* 0A */  /*    MODE = NONE   SHIFT = UNSHIF */
        '2300'X      /* 0B */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0101'X      /* 0C */  /*    MODE = NONE   SHIFT = UNSHIF */
        '2300'X      /* 0D */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0B4C'X      /* 0E */  /*    MODE = NONE   SHIFT = UNSHIF */
        '1040'X      /* 0F */  /*    MODE = NONE   SHIFT = UNSHIF */
        '2300'X      /* 10 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A81'X      /* 11 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0AA2'X      /* 12 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A84'X      /* 13 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A86'X      /* 14 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A87'X      /* 15 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A88'X      /* 16 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A91'X      /* 17 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A92'X      /* 18 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A93'X      /* 19 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0B5E'X      /* 1A */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0B7D'X      /* 1B */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0BC0'X      /* 1C */  /*    MODE = NONE   SHIFT = UNSHIF */
        '010E'X      /* 1D */  /*    MODE = NONE   SHIFT = UNSHIF */
        '2200'X      /* 1E */  /*    MODE = NONE   SHIFT = UNSHIF */
        '2300'X      /* 1F */  /*    MODE = NONE   SHIFT = UNSHIF */
        '010B'X      /* 20 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A98'X      /* 21 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0AA6'X      /* 22 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A85'X      /* 23 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A99'X      /* 24 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0AA3'X      /* 25 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0AA8'X      /* 26 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0AA4'X      /* 27 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A89'X      /* 28 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A96'X      /* 29 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0A97'X      /* 2A */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0BB0'X      /* 2B */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0BE0'X      /* 2C */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0201'X      /* 2D */  /*    MODE = NONE   SHIFT = UNSHIF */
        '2300'X      /* 2E */  /*    MODE = NONE   SHIFT = UNSHIF */
        '2300'X      /* 2F */  /*    MODE = NONE   SHIFT = UNSHIF */
        '2300'X      /* 30 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0DF1'X      /* 31 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0DF2'X      /* 32 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0DF3'X      /* 33 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0DF4'X      /* 34 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0DF5'X      /* 35 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0DF6'X      /* 36 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0DF7'X      /* 37 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0DF8'X      /* 38 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0DF9'X      /* 39 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0DF0'X      /* 3A */  /*    MODE = NONE   SHIFT = UNSHIF */
        '1060'X      /* 3B */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0B7E'X      /* 3C */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0108'X      /* 3D */  /*    MODE = NONE   SHIFT = UNSHIF */
        '2602'X      /* 3E */  /*    MODE = NONE   SHIFT = UNSHIF */
        '2300'X      /* 3F */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0EF0'X      /* 40 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0EF1'X      /* 41 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0EF2'X      /* 42 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0EF3'X      /* 43 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0EF4'X      /* 44 */  /*    MODE = NONE   SHIFT = UNSHIF */
        '0EF5'X      /* 45 */  /*    MODE = NONE   SHIFT = UNSHIF */
```

```
      '0EF6'X      /* 46 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0EF7'X      /* 47 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0EF8'X      /* 48 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0EF9'X      /* 49 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '104B'X      /* 4A */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0109'X      /* 4B */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0204'X      /* 4C */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0203'X      /* 4D */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0202'X      /* 4E */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2200'X      /* 4F */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2200'X      /* 50 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '1703'X      /* 51 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 52 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 53 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2001'X      /* 54 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2200'X      /* 55 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2003'X      /* 56 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2002'X      /* 57 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 58 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 59 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 5A */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 5B */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 5C */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 5D */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 5E */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 5F */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2200'X      /* 60 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 61 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '030B'X      /* 62 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 63 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 64 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 65 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '030C'X      /* 66 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '030A'X      /* 67 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0604'X      /* 68 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 69 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 6A */  /*    MODE = NONE   SHIFT = UNSHIF */
      '1703'X      /* 6B */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0301'X      /* 6C */  /*    MODE = NONE   SHIFT = UNSHIF */
      '010D'X      /* 6D */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0601'X      /* 6E */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0310'X      /* 6F */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0107'X      /* 70 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0106'X      /* 71 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0102'X      /* 72 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0103'X      /* 73 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 74 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2200'X      /* 75 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 76 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 77 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2200'X      /* 78 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 79 */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X      /* 7A */  /*    MODE = NONE   SHIFT = UNSHIF */
      '1703'X      /* 7B */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0402'X      /* 7C */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0607'X      /* 7D */  /*    MODE = NONE   SHIFT = UNSHIF */
      '0401'X      /* 7E */  /*    MODE = NONE   SHIFT = UNSHIF */
      '2300'X.     /* 7F */  /*    MODE = NONE   SHIFT = UNSHIF */
:TKSTATE
  MODE = NONE   SHIFT = UPPER
  DATA =
      '2300'X      /* 00 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09E9'X      /* 01 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09E7'X      /* 02 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09C3'X      /* 03 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09E5'X      /* 04 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09C2'X      /* 05 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09D5'X      /* 06 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09D4'X      /* 07 */  /*    MODE = NONE   SHIFT = UPPER  */
      '106B'X      /* 08 */  /*    MODE = NONE   SHIFT = UPPER  */
      '104B'X      /* 09 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B6F'X      /* 0A */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 0B */  /*    MODE = NONE   SHIFT = UPPER  */
      '0101'X      /* 0C */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 0D */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B6E'X      /* 0E */  /*    MODE = NONE   SHIFT = UPPER  */
      '1040'X      /* 0F */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 10 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09C1'X      /* 11 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09E2'X      /* 12 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09C4'X      /* 13 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09C6'X      /* 14 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09C7'X      /* 15 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09C8'X      /* 16 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09D1'X      /* 17 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09D2'X      /* 18 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09D3'X      /* 19 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B7A'X      /* 1A */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B7F'X      /* 1B */  /*    MODE = NONE   SHIFT = UPPER  */
      '0BD0'X      /* 1C */  /*    MODE = NONE   SHIFT = UPPER  */
      '010E'X      /* 1D */  /*    MODE = NONE   SHIFT = UPPER  */
      '2200'X      /* 1E */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 1F */  /*    MODE = NONE   SHIFT = UPPER  */
      '010C'X      /* 20 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09D8'X      /* 21 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09E6'X      /* 22 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09C5'X      /* 23 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09D9'X      /* 24 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09E3'X      /* 25 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09E8'X      /* 26 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09E4'X      /* 27 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09C9'X      /* 28 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09D6'X      /* 29 */  /*    MODE = NONE   SHIFT = UPPER  */
      '09D7'X      /* 2A */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B4F'X      /* 2B */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B6A'X      /* 2C */  /*    MODE = NONE   SHIFT = UPPER  */
      '0201'X      /* 2D */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 2E */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 2F */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 30 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0BBB'X      /* 31 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B7C'X      /* 32 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B7B'X      /* 33 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B5B'X      /* 34 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B6C'X      /* 35 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0BBA'X      /* 36 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B50'X      /* 37 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B5C'X      /* 38 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B4D'X      /* 39 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B5D'X      /* 3A */  /*    MODE = NONE   SHIFT = UPPER  */
      '0B6D'X      /* 3B */  /*    MODE = NONE   SHIFT = UPPER  */
      '0F4E'X      /* 3C */  /*    MODE = NONE   SHIFT = UPPER  */
      '0108'X      /* 3D */  /*    MODE = NONE   SHIFT = UPPER  */
      '2603'X      /* 3E */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 3F */  /*    MODE = NONE   SHIFT = UPPER  */
      '0EF0'X      /* 40 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0EF1'X      /* 41 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0EF2'X      /* 42 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0EF3'X      /* 43 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0EF4'X      /* 44 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0EF5'X      /* 45 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0EF6'X      /* 46 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0EF7'X      /* 47 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0EF8'X      /* 48 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0EF9'X      /* 49 */  /*    MODE = NONE   SHIFT = UPPER  */
      '104B'X      /* 4A */  /*    MODE = NONE   SHIFT = UPPER  */
      '0608'X      /* 4B */  /*    MODE = NONE   SHIFT = UPPER  */
      '0609'X      /* 4C */  /*    MODE = NONE   SHIFT = UPPER  */
      '0203'X      /* 4D */  /*    MODE = NONE   SHIFT = UPPER  */
      '0202'X      /* 4E */  /*    MODE = NONE   SHIFT = UPPER  */
      '2200'X      /* 4F */  /*    MODE = NONE   SHIFT = UPPER  */
      '2200'X      /* 50 */  /*    MODE = NONE   SHIFT = UPPER  */
      '1703'X      /* 51 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 52 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 53 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2001'X      /* 54 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2200'X      /* 55 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2003'X      /* 56 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2002'X      /* 57 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 58 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 59 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 5A */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 5B */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 5C */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 5D */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 5E */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 5F */  /*    MODE = NONE   SHIFT = UPPER  */
      '2200'X      /* 60 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 61 */  /*    MODE = NONE   SHIFT = UPPER  */
      '060A'X      /* 62 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 63 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 64 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2300'X      /* 65 */  /*    MODE = NONE   SHIFT = UPPER  */
      '2200'X      /* 66 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0303'X      /* 67 */  /*    MODE = NONE   SHIFT = UPPER  */
      '0604'X      /* 68 */  /*    MODE = NONE   SHIFT = UPPER  */
```

```
    '2300'X      /* 69 */  /*    MODE = NONE   SHIFT = UPPER  */
    '2300'X      /* 6A */  /*    MODE = NONE   SHIFT = UPPER  */
    '1703'X      /* 6B */  /*    MODE = NONE   SHIFT = UPPER  */
    '0302'X      /* 6C */  /*    MODE = NONE   SHIFT = UPPER  */
    '0303'X      /* 6D */  /*    MODE = NONE   SHIFT = UPPER  */
    '0304'X      /* 6E */  /*    MODE = NONE   SHIFT = UPPER  */
    '0310'X      /* 6F */  /*    MODE = NONE   SHIFT = UPPER  */
    '0603'X      /* 70 */  /*    MODE = NONE   SHIFT = UPPER  */
    '0602'X      /* 71 */  /*    MODE = NONE   SHIFT = UPPER  */
    '0104'X      /* 72 */  /*    MODE = NONE   SHIFT = UPPER  */
    '0105'X      /* 73 */  /*    MODE = NONE   SHIFT = UPPER  */
    '2300'X      /* 74 */  /*    MODE = NONE   SHIFT = UPPER  */
    '2200'X      /* 75 */  /*    MODE = NONE   SHIFT = UPPER  */
    '2300'X      /* 76 */  /*    MODE = NONE   SHIFT = UPPER  */
    '2300'X      /* 77 */  /*    MODE = NONE   SHIFT = UPPER  */
    '2200'X      /* 78 */  /*    MODE = NONE   SHIFT = UPPER  */
    '2300'X      /* 79 */  /*    MODE = NONE   SHIFT = UPPER  */
    '2300'X      /* 7A */  /*    MODE = NONE   SHIFT = UPPER  */
    '1703'X      /* 7B */  /*    MODE = NONE   SHIFT = UPPER  */
    '0403'X      /* 7C */  /*    MODE = NONE   SHIFT = UPPER  */
    '0607'X      /* 7D */  /*    MODE = NONE   SHIFT = UPPER  */
    '0401'X      /* 7E */  /*    MODE = NONE   SHIFT = UPPER  */
    '2300'X.     /* 7F */  /*    MODE = NONE   SHIFT = UPPER  */

:PDFNTBL.
:SPACE
  DATA ='20'X.
:BELL
  DATA ='07'X.
:BSP
  DATA ='08'X.
:LINEFEED
  DATA ='0A'X.
:FORMFEED
  DATA ='0C'X.
:CARRTN
  DATA ='0D'X.
:STRUS
  DATA ='1B2D31'X.
:ENDUS
  DATA ='1B2D30'X.
:PAGLENL
  VAROFFSET=    2
  VARLEN=    1
  DATA ='1B4300'X
  VARTYPE=HIGHLOW
  VARMAX= 0
  ADJUST= 0.
:PAGLENI
  VAROFFSET=    3
  VARLEN=    1
  DATA ='1B4300000'X
  VARTYPE=HIGHLOW
  CNVNUM=    1
  CNVDEN=    1
  VARMAX= 0
  ADJUST= 0.
:VARLSPC
  VAROFFSET=    2
  VARLEN=    1
  DATA ='1B4100'X
  VARTYPE=HIGHLOW
  CNVNUM=    1
  CNVDEN=   96
  VARMAX= 0
  ADJUST= 0.
:LPI
  LPI=6
  DATA ='1B32'X.
:LPI
  LPI=8
  DATA ='1B30'X.
:PRTQLTY
  QLTYTYPE=DRAFT
  DATA ='1B48'X.
:PRTQLTY
  QLTYTYPE=LETTER
  DATA ='1B47'X.
:PRTQLTY
  QLTYTYPE=TEXT
  DATA ='1B47'X.
:CPI
  CPI=5
    DATA ='0E12'X.
:CPI
  CPI=6
  DATA ='0E12'X.
:CPI
  CPI=855
  DATA ='0E0F'X.
:CPI
  CPI=10
  DATA ='1412'X.
:CPI
  CPI=12
  DATA ='1412'X.
:CPI
  CPI=15
  DATA ='0F14'X.
:CPI
  CPI=171
  DATA ='0F14'X.
:STRBOLD
  DATA ='1B45'X.
:ENDBOLD
  DATA ='1B46'X.
:TRNEBCDIC.
:TRNEBCDICE
  EBCDIC='42'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='43'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='44'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='45'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='46'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='47'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='48'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='49'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='51'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='52'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='53'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='54'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='55'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='56'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='57'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='58'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='59'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='60'X
  DATA ='2D'X.
:TRNEBCDICE
  EBCDIC='62'X
  DATA ='20'X.
:TRNEBCDICE
  EBCDIC='63'X
```

```
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='64'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='65'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='66'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='67'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='68'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='69'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='6A'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='70'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='71'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='72'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='73'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='74'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='75'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='76'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='77'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='78'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='80'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='8A'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='8B'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='8C'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='8D'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='8E'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='8F'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='90'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='9A'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='9B'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='9C'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='9D'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='9E'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='9F'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='A0'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='AA'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='AB'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='AC'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='AD'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='AE'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='AF'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='B0'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='B1'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='B2'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='B3'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='B4'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='B5'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='B6'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='B7'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='B8'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='B9'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='BA'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='BB'X
   DATA ='7C'X.
 :TRNEBCDICE
   EBCDIC='BC'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='BD'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='BE'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='BF'X
   DATA ='20'X.
 :TRNEBCDICE
   EBCDIC='CA'X
   DATA ='2D'X.
 :TRNEBCDICE
   EBCDIC='CB'X
   DATA ='20'X.
```

```
    :TRNEBCDICE
      EBCDIC='CC'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='CD'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='CE'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='CF'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='DA'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='DB'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='DC'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='DD'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='DE'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='DF'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='E1'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='EA'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='EB'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='EC'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='ED'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='EE'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='EF'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='FA'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='FB'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='FC'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='FD'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='FE'X
      DATA ='20'X.
    :TRNEBCDICE
      EBCDIC='FF'X
      DATA ='20'X.
    :ETRNEBCDIC.
:EWSCST.
```

Source Code for 3151 ASCII Display

The source for this example is created when you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command specifying the following parameters:

Device type

DEVTYPE(3151)

Keyboard language type

KBDTYPE(USB)

```
:WSCST DEVCLASS=ASCIIDSP.

  :DASCTBL
    DATA =
     /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
       '405A7F7B5B6C507D4D5D5C4E6B604B61'X      /* 2- */
       'F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F'X      /* 3- */
       '7CC1C2C3C4C5C6C7C8C9D1D2D3D4D5D6'X      /* 4- */
       'D7D8D9E2E3E4E5E6E7E8E9BAE0BBB06D'X      /* 5- */
       '798182838485868788889919293949596'X      /* 6- */
       '979899A2A3A4A5A6A7A8A9C04FD0A100'X      /* 7- */
       '00000000000000000000000000000000'X      /* 8- */
       '00000000000000000000000000000000'X      /* 9- */
       '00000000000000000000000000000000'X      /* A- */
       '00000000000000000000000000000000'X      /* B- */
       '00000000000000000000000000000000'X      /* C- */
       '00000000000000000000000000000000'X      /* D- */
       '00000000000000000000000000000000'X      /* E- */
       '00000000000000000000000000000000'X.     /* F- */

  :DEBCTBL
    DATA =
     /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
       '20202020202020202020202020202020'X      /* 0- */
       '20202020202020202020202A202020'X      /* 1- */
       '20202020202020202020202020202020'X      /* 2- */
       '20202020202020202020202020202020'X      /* 3- */
       '20206161616161636E202E3C282B7C'X      /* 4- */
       '26656565656969696921242A293B20'X      /* 5- */
       '2D2F41414141414434E7C2C255F3E3F'X      /* 6- */
       '20454545454949494603A2340273D22'X      /* 7- */
       '20616263646566676869202020792020'X      /* 8- */
       '206A6B6C6D6E6F707172202020202020'X      /* 9- */
       '207E737475767778797A213F20202020'X      /* A- */
       '5E2020202020202B205B5D20202720'X      /* B- */
       '7B41424344454647484492D6F6F6F6F6F'X      /* C- */
       '7D4A4B4C4D4E4F505152207575757579'X      /* D- */
       '5C2053545556575859 5A204F4F4F4F4F'X      /* E- */
       '3031323334353637383920555555552 0'X.     /* F- */

  :DSCNTBL
      CHARATR = CHAR
      ADDRMOD = BINARY
      TEXTSYM = SUPPORT
      AUTOSCL = NO.
    :CARRTN
      DATA = 'EA'X.
    :HLFIDXDN
      DATA = 'F6'X.
    :HLFIDXUP
      DATA = 'F7'X.
    :PAGEND
      DATA = 'BE'X.
    :RQDCARRTN
      DATA = 'BD'X.
    :RQDSPC
      DATA = 'BF'X.
    :RQDTAB
      DATA = 'BC'X.
    :STOPCODE
      DATA = 'FE'X.
    :TAB
      DATA = 'F5'X.
    :WORDUS
      DATA = 'ED'X.

    :ATRCMD
      CTLCHAR = '20'X
      DATA = '1B3440'X.
    :ATRCMD
      CTLCHAR = '21'X
      DATA = '1B3441'X.
    :ATRCMD
      CTLCHAR = '22'X
      DATA = '1B3448'X.
    :ATRCMD
```

```
    CTLCHAR = '23'X
    DATA = '1B3449'X.
  :ATRCMD
    CTLCHAR = '24'X
    DATA = '1B3442'X.
  :ATRCMD
    CTLCHAR = '25'X
    DATA = '1B3443'X.
  :ATRCMD
    CTLCHAR = '26'X
    DATA = '1B344A'X.
  :ATRCMD
    CTLCHAR = '27'X
    DATA = '1B3440'X.
  :ATRCMD
    CTLCHAR = '28'X
    DATA = '1B3444'X.
  :ATRCMD
    CTLCHAR = '29'X
    DATA = '1B3445'X.
  :ATRCMD
    CTLCHAR = '2A'X
    DATA = '1B344C'X.
  :ATRCMD
    CTLCHAR = '2B'X
    DATA = '1B344D'X.
  :ATRCMD
    CTLCHAR = '2C'X
    DATA = '1B3446'X.
  :ATRCMD
    CTLCHAR = '2D'X
    DATA = '1B3447'X.
  :ATRCMD
    CTLCHAR = '2E'X
    DATA = '1B344E'X.
  :ATRCMD
    CTLCHAR = '2F'X
    DATA = '1B3440'X.
  :ATRCMD
    CTLCHAR = '30'X
    DATA = '1B3440'X.
  :ATRCMD
    CTLCHAR = '31'X
    DATA = '1B3441'X.
  :ATRCMD
    CTLCHAR = '32'X
    DATA = '1B3448'X.
  :ATRCMD
    CTLCHAR = '33'X
    DATA = '1B3449'X.
  :ATRCMD
    CTLCHAR = '34'X
    DATA = '1B3442'X.
  :ATRCMD
    CTLCHAR = '35'X
    DATA = '1B3443'X.
  :ATRCMD
    CTLCHAR = '36'X
    DATA = '1B344A'X.
  :ATRCMD
    CTLCHAR = '37'X
    DATA = '1B3440'X.
  :ATRCMD
    CTLCHAR = '38'X
    DATA = '1B3444'X.
  :ATRCMD
    CTLCHAR = '39'X
    DATA = '1B3445'X.
  :ATRCMD
    CTLCHAR = '3A'X
    DATA = '1B344C'X.
  :ATRCMD
    CTLCHAR = '3B'X
    DATA = '1B344D'X.
  :ATRCMD
    CTLCHAR = '3C'X
    DATA = '1B3446'X.
  :ATRCMD
    CTLCHAR = '3D'X
    DATA = '1B3447'X.
  :ATRCMD
    CTLCHAR = '3E'X
    DATA = '1B344E'X.
  :ATRCMD
    CTLCHAR = '3F'X
    DATA = '1B3440'X.
  :SETUP
    DATA = '1B283A1B293A'X.
  :GCS
    DATA = '1B3E41'X.
  :ACS
    DATA = '1B3C40'X.
  :NLCS
    DATA = '1B3E42'X.
  :CLRSCN
    DATA = '1B214C'X.
  :CSRADR
    ROWTENS =   3
    ROWONES = 0
    COLHUNDS =   4
    COLTENS = 0
    COLONES = 0
    DATA = '1B582020'X.
  :INSCSR
    DATA = '1B5A'X.
  :ALARM
    DATA = '07'X.
  :XCSRADR
    ROW =  4
    COLHIGH =  5
    COLLOW =  6
    DATA = '1B7820202040'X.
  :SCNSIZE
    SIZE = 1920
    DATA = '1B2072212120382250'X.
  :SCNSIZE
    SIZE = 2000
    DATA = '1B20722121203C2250'X.
  :SCNSIZE
    SIZE = 3564
    DATA = '1B20722121203C2444'X.
  :STRBYP
    DATA = '1012'X.
  :ENDBYP
    DATA = '1014'X.

:DKBDTBL.
  :ATN
    DATA = '01'X.
  :BSP
    DATA = '08'X.
  :CSRUP
    DATA = '1B41'X.
  :CSRDOWN
    DATA = '1B42'X.
  :CSRLEFT
    DATA = '1B44'X.
  :CSRRIGHT
    DATA = '1B43'X.
  :DLT
    DATA = '1B51'X.
  :DUP
    DATA = '04'X.
  :ENTER
    DATA = '1B3803'X.
  :ERSINP
    DATA = '1B4B'X.
  :ERRRESET
    DATA = '1B217A03'X.
  :ERRRESET
    DATA = '12'X.
  :ERRRESET
    DATA = '1B72'X.
  :ERRRESET
    DATA = '1B52'X.
  :FLDPLUS
    DATA = '1B2B'X.
  :FLDMINUS
    DATA = '1B4D'X.
  :FLDMINUS
    DATA = '1B6D'X.
  :FLDMINUS
    DATA = '1B2D'X.
  :FLDEXIT
    DATA = '0D'X.
  :HELP
```

```
  DATA = '1B3F'X.
:HOME
  DATA = '1B48'X.
:INSERT
  DATA = '1B502008'X.
:NEWLINE
  DATA = '0A'X.
:PRINT
  DATA = '1B5703'X.
:PRINT
  DATA = '10'X.
:PAGUP
  DATA = '1B04'X.
:PAGDOWN
  DATA = '1B15'X.
:SYSREQ
  DATA = '1B53'X.
:SYSREQ
  DATA = '1B73'X.
:FLDADV
  DATA = '09'X.
:FLDBSP
  DATA = '1B32'X.
:CLEAR
  DATA = '1B4C03'X.
:TSTREQ
  DATA = '14'X.
:TSTREQ
  DATA = '1B54'X.
:TSTREQ
  DATA = '1B74'X.
:HEX
  DATA = '1B60'X.
:FKEY
  KEY = F1
  DATA = '1B6103'X.
:FKEY
  KEY = F2
  DATA = '1B6203'X.
:FKEY
  KEY = F3
  DATA = '1B6303'X.
:FKEY
  KEY = F4
  DATA = '1B6403'X.
:FKEY
  KEY = F5
  DATA = '1B6503'X.
:FKEY
  KEY = F6
  DATA = '1B6603'X.
:FKEY
  KEY = F7
  DATA = '1B6703'X.
:FKEY
  KEY = F8
  DATA = '1B6803'X.
:FKEY
  KEY = F9
  DATA = '1B6903'X.
:FKEY
  KEY = F10
  DATA = '1B6A03'X.
:FKEY
  KEY = F11
  DATA = '1B6B03'X.
:FKEY
  KEY = F12
  DATA = '1B6C03'X.
:FKEY
  KEY = F13
  DATA = '1B216103'X.
:FKEY
  KEY = F14
  DATA = '1B216203'X.
:FKEY
  KEY = F15
  DATA = '1B216303'X.
:FKEY
  KEY = F16
  DATA = '1B216403'X.
:FKEY
  KEY = F17
  DATA = '1B216503'X.
:FKEY
  KEY = F18
  DATA = '1B216603'X.
:FKEY
  KEY = F19
  DATA = '1B216703'X.
:FKEY
  KEY = F20
  DATA = '1B216803'X.
:FKEY
  KEY = F21
  DATA = '1B216903'X.
:FKEY
  KEY = F22
  DATA = '1B216A03'X.
:FKEY
  KEY = F23
  DATA = '1B216B03'X.
:FKEY
  KEY = F24
  DATA = '1B216C03'X.
:CSRSEL
  DATA = '03'X.
:ERSEOF
  DATA = '05'X.
:ERSEOF
  DATA = '1B49'X.
:FLDMRK
  DATA = '06'X.
:PA1
  DATA = '1B216D03'X.
:PA1
  DATA = '1B5031'X.
:PA1
  DATA = '1B7031'X.
:PA2
  DATA = '1B216E03'X.
:PA2
  DATA = '1B5032'X.
:PA2
  DATA = '1B7032'X.
:PA3
  DATA = '1B216F03'X.
:PA3
  DATA = '1B5033'X.
:PA3
  DATA = '1B7033'X.
:FCSRLEFT
  DATA = '1B3C'X.
:FCSRRIGHT
  DATA = '1B3E'X.
:RQDTAB
  DATA = '1B1B09'X.
:WORDUS
  DATA = '1B1B57'X.
:WORDUS
  DATA = '1B1B77'X.
:HLFIDXUP
  DATA = '1B1B59'X.
:HLFIDXUP
  DATA = '1B1B79'X.
:STRUS
  DATA = '1B1B55'X.
:STRUS
  DATA = '1B1B75'X.
:PAGEND
  DATA = '1B1B50'X.
:PAGEND
  DATA = '1B1B70'X.
:STOPCODE
  DATA = '1B1B53'X.
:STOPCODE
  DATA = '1B1B73'X.
:HLFIDXDN
  DATA = '1B1B48'X.
:HLFIDXDN
  DATA = '1B1B68'X.
:END
  DATA = '1B1B4A'X.
:END
  DATA = '1B1B6A'X.
:CARRTN
```

```
      DATA = '1B1B0D'X.
    :FWDTAB
      DATA = '1B1B1B09'X.
    :CENTER
      DATA = '1B1B43'X.
    :CENTER
      DATA = '1B1B63'X.
    :BOLD
      DATA = '1B1B42'X.
    :BOLD
      DATA = '1B1B62'X.
    :NEXTSTOP
      DATA = '1B1B4E'X.
file:NEXTSTOP
      DATA = '1B1B6E'X.
    :DSPSYM
      DATA = '1B1B1B48'X.
    :TOPPAG
      DATA = '1B1B1B41'X.
    :RQDSPC
      DATA = '1B1B20'X.
    :BOTPAG
      DATA = '1B1B1B42'X.
    :STRLINE
      DATA = '1B1B1B44'X.
    :ENDLINE
      DATA = '1B1B1B43'X.
    :SCNREFRESH
      DATA = '1B01'X.
    :TOGIND
      DATA = '1B17'X.
    :DISC
      DATA = '1B12'X.
    :READSTS
      DATA = '1B36'X.
:EWSCST.
```

Source Code for 4019 ASCII Printer That Uses the Host Print Transform Function

The source for this example is created when you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command specifying the following parameters:

Device type

DEVTYPE(TRANSFORM)

Manufacturer, model, type

IBM4019

```
:WSCST DEVCLASS=TRANSFORM.

    :TRNSFRMTBL.
    :PRTDTASTRM
      DATASTREAM=IBMPPDS2.
    :NOPRTBDR
      OPTION=TOP
      ORIENT=PORTRAIT
      DATA =  240.
    :NOPRTBDR
      OPTION=LEFT
      ORIENT=PORTRAIT
      DATA =  360.
    :NOPRTBDR
      OPTION=RIGHT
      ORIENT=PORTRAIT
      DATA =  360.
    :NOPRTBDR
      OPTION=BOTTOM
      ORIENT=PORTRAIT
      DATA =  240.
    :NOPRTBDR
      OPTION=TOP
      ORIENT=LANDSCAPE
      DATA =  360.
    :NOPRTBDR
      OPTION=LEFT
      ORIENT=LANDSCAPE
      DATA =  240.
    :NOPRTBDR
      OPTION=RIGHT
      ORIENT=LANDSCAPE
      DATA =  240.
    :NOPRTBDR
      OPTION=BOTTOM
      ORIENT=LANDSCAPE
      DATA =  360.
    :INITPRT
      DATA ='1B5B4B050006310100A111141B461B481B4F1B541
B57001B35001B2D001B5801FF1B5B5C0400900090001B36'X.
    :RESETPRT
      DATA ='1B4F1B541B57001B2D001B35001B5
B5C020048001B5801FF1B5B460300030101'X.

    :SPACE
      DATA ='20'X.
    :BSP
      DATA ='08'X.
    :CARRTN
      DATA ='0D'X.
    :FORMFEED
      DATA ='0C'X.
    :LINEFEED
      DATA ='0A'X.
    :RVSLINEFEED
      DATA ='1B5D'X.
    :HORRMOV
      DIRECTION=FWD
      VAROFFSET=    2
      VARLEN=    2
      VARTYPE=LOWHIGH
      CNVNUM=    1
      CNVDEN=  120
      DATA ='1B640000'X.
    :HORRMOV
      DIRECTION=BCK
      VAROFFSET=    2
      VARLEN=    2
      VARTYPE=LOWHIGH
      CNVNUM=    1
      CNVDEN=  120
      DATA ='1B650000'X.
    :STRBOLD
      DATA ='1B451B47'X.
    :ENDBOLD
      DATA ='1B461B48'X.
    :STRSUBS
      DATA ='1B5301'X.
    :ENDSUBS
      DATA ='1B54'X.
    :STRSUPS
      DATA ='1B5300'X.
    :ENDSUPS
      DATA ='1B54'X.
    :STRUS
      DATA ='1B2D01'X.
    :ENDUS
      DATA ='1B2D00'X.
    :VARLSPC
      VAROFFSET=   11
      VARLEN=    1
      VARTYPE=LOWHIGH
      CNVNUM=    1
      CNVDEN=  144
      DATA ='1B5B5C0400900090001B41001B32'X.
    :CPICOR
      CPI=10
      ASCIIFNT=  254
      FNTWTH=   84
      FNTATR=    1
      DATA =''X.
    :CPICOR
      CPI=12
      ASCIIFNT=  254
      FNTWTH=   84
      FNTATR=    1
      DATA =''X.
    :CPICOR
      CPI=15
      ASCIIFNT=  254
      FNTWTH=   84
```

```
      FNTATR=    1
      DATA =''X.
    :PRTORIENT
      ORIENT=PORTRAIT
      DATA ='1B6B'X.
    :PRTORIENT
      ORIENT=LANDSCAPE
      DATA ='1B6C'X.
    :PRTORIENT
      ORIENT=RTT180
      DATA ='1B6B'X.
    :PRTORIENT
      ORIENT=RTT270
      DATA ='1B6C'X.
    :DWRSLT
      DRAWER=PAPER
      DATA ='1B5B460300010100'X.
    :DWRSLT
      DRAWER=ENVELOPE
      DATA ='1B5B460300030200'X.
    :DWRSLT
      DRAWER=DRAWER1
      DATA ='1B5B460300030101'X.
    :DWRSLT
      DRAWER=DRAWER2
      DATA ='1B5B460300030102'X.
    :PAGLENI
      VAROFFSET=    3
      VARLEN=    1
      VARTYPE=LOWHIGH
      CNVNUM=    1
      CNVDEN=    1
      DATA ='1B430000'X.
    :PAGLENL
      VAROFFSET=    2
      VARLEN=    1
      VARTYPE=LOWHIGH
      DATA ='1B4300'X.
    :PAGSIZXFM.
    :PAGSIZE
      PAGWTH= 8352
      PAGLEN=11952
      DATA ='1B5B4605000000000600'X.
    :PAGSIZE
      PAGWTH=10368
      PAGLEN=14544
      DATA ='1B5B4605000000000300'X.
    :PAGSIZE
      PAGWTH=10440
      PAGLEN=15120
      DATA ='1B5B4605000000000500'X.
    :PAGSIZE
      PAGWTH=12240
      PAGLEN=15840
      DATA ='1B5B4605000000000100'X.
    :PAGSIZE
      PAGWTH=11952
      PAGLEN=16848
      DATA ='1B5B4605000000000400'X.
    :PAGSIZE
      PAGWTH=12240
      PAGLEN=20160
      DATA ='1B5B4605000000000200'X.
    :EPAGSIZXFM.
    :ENVSIZXFM.
    :ENVSIZE
      ENVWTH=10800
      ENVLEN= 5580
      DATA ='1B5B4605000000000100'X.
    :ENVSIZE
      ENVWTH=12780
      ENVLEN= 5580
      DATA ='1B5B4605000000000200'X.
    :ENVSIZE
      ENVWTH=13680
      ENVLEN= 5540
      DATA ='1B5B4605000000000300'X.
    :ENVSIZE
      ENVWTH=12528
      ENVLEN= 6192
      DATA ='1B5B4605000000000400'X.
    :ENVSIZE
      ENVWTH=12960
      ENVLEN= 9216
      DATA ='1B5B4605000000000500'X.
    :ENVSIZE
      ENVWTH=14112
      ENVLEN= 9936
      DATA ='1B5B4605000000000600'X.
    :EENVSIZXFM.
    :FNTGRP.
    :FNTGRPE
      MINFID=  260
      MAXFID=  273
      FNTSTR='120F1B5701'X
      FNTEND='1B5700'X
      FNTWTH=''X.
    :FNTGRPE
      MINFID=  240
      MAXFID=  246
      FNTSTR='14121B5701'X
      FNTEND='1B5700'X
      FNTWTH=''X.
    :EFNTGRP.
    :ASCCPINFO.
    :CODEPAGE
      CODEPAGE=  437
      DATA ='1B5B540400000001B5'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  850
      DATA ='1B5B540400000000352'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  851
      DATA ='1B5B540400000000353'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  852
      DATA ='1B5B540400000000354'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  856
      DATA ='1B5B540400000000358'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  857
      DATA ='1B5B540400000000359'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  860
      DATA ='1B5B5404000000035C'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
```

```
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  861
      DATA ='1B5B5404000000035D'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  862
      DATA ='1B5B5404000000035E'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  863
      DATA ='1B5B5404000000035F'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  864
      DATA ='1B5B54040000000360'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  865
      DATA ='1B5B54040000000361'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  869
      DATA ='1B5B54040000000365'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  874
      DATA ='1B5B5404000000036A'X.
    :ASCIICTL
      ASCII='14'X
      DATA ='1B5E14'X.
    :ASCIICTL
      ASCII='15'X
      DATA ='1B5E15'X.
    :CODEPAGE
      CODEPAGE=  899
      DATA ='1B5B54040000000383'X.
    :CODEPAGE
      CODEPAGE= 1051
      DATA ='1B5B5404000000041B'X.
    :EASCCPINFO.
:EWSCST.
```

Source Code for 4019 ASCII Printer That Uses the Emulator on the Controller

The source for this example is created when you use the Retrieve Work Station Customizing Object Source (RTVWSCST) command specifying the following parameters:

Device type

DEVTYPE(4019)

Keyboard language type

KBDTYPE(USI)

```
:WSCST DEVCLASS=ASCIIPRT.

  :PDFTMAPTBL.
    :PDFTEBCTBL
      DATA =
     /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
       '2020838485A0C68687A45B2E3C282B21'X      /* 4- */
       '268288898AA18C8B8DE15D242A293B5E'X      /* 5- */
       '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
       '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
       '9D6162636465666676869AEAFD0ECE7F1'X     /* 8- */
       'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
       'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
       'BD9CBE169FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
       '7B41424344454647484 92D939495A2E4'X     /* C- */
       '7D4A4B4C4D4E4F505152FB968197A398'X      /* D- */
       '5C20535455565758595AFDE299E3E0E5'X      /* E- */
       '30313233343536373839FCEA9AEBE900'X.     /* F- */
    :PFNTWTH
      DATA =
     /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
       '0A0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
       '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
       '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
       '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
       '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
       '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
       '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
       '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
       '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
       '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
       '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
       '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */
  :EPDFTMAPTBL.

    :PFCNTBL.
    :DFTFNTID
      CPI=10
      DEFAULT=   11.
    :DFTFNTID
      CPI=12
      DEFAULT=   85.
    :DFTFNTID
      CPI=171
      DEFAULT=  254.
    :DFTFNTID
      CPI=PROP
      DEFAULT=  159.
    :DFTFNTID
      CPI=5
      DEFAULT=  244.
    :DFTFNTID
      CPI=855
      DEFAULT=  265.
    :DFTFNTID
      CPI=133
      DEFAULT=  204.
    :DFTFNTID
      CPI=15
      DEFAULT=  223.
    :DFTFNTID
      CPI=20
      DEFAULT=  281.
    :DFTFNTID
      CPI=27
      DEFAULT=  290.
```

```
:VARLSPC
  VAROFFSET=   11
  VARLEN=    1
  VARTYPE=LOWHIGH
  VARMAX=  255
  ADJUST= 0
  CNVNUM=   10
  CNVDEN=    1
  DATA ='1B5B5C0400900090001B41101B32'X.
:PAGLENI
  VAROFFSET=    3
  VARLEN=    1
  VARTYPE=LOWHIGH
  VARMAX=  255
  ADJUST= 0
  CNVNUM= 1440
  CNVDEN=    1
  DATA ='1B43000B'X.
:PAGLENL
  VAROFFSET=    2
  VARLEN=    1
  VARTYPE=LOWHIGH
  VARMAX=  255
  ADJUST= 0
  DATA ='1B4342'X.
:CODPAGVAR
  VAROFFSET=    7
  VARLEN=    2
  VARTYPE=HIGHLOW
  DATA ='1B5B5405000000035200'X.
:FNTGPFT
  DATA ='1B5B4905000000000000'X.
:FWDRMOV
  VAROFFSET=    2
  VARLEN=    2
  VARTYPE=LOWHIGH
  VARMAX=32767
  ADJUST= 0
  CNVNUM=   12
  CNVDEN=    1
  DATA ='1B640F00'X.
:BCKRMOV
  VAROFFSET=    2
  VARLEN=    2
  VARTYPE=LOWHIGH
  VARMAX=32767
  ADJUST= 0
  CNVNUM=   12
  CNVDEN=    1
  DATA ='1B650F00'X.
:CPI
  CPI=5
  DATA ='00121B5701'X.
:CPI
  CPI=855
  DATA ='00120F1B5701'X.
:CPI
  CPI=10
  DATA ='1B570012'X.
:STRPROP
  DATA ='1B57001B5001'X.
:CPI
  CPI=12
  DATA ='1B5700121B3A'X.
:CPI
  CPI=15
  DATA ='1B5700120F'X.
:CPI
  CPI=171
  DATA ='1B5700120F'X.
:FONTQLTY
  FONTCPI=10
  QLTYTYPE=DRAFT
  DATA ='1B490112'X.
:FONTQLTY
  FONTCPI=12
  QLTYTYPE=DRAFT
  DATA ='1B49011B3A'X.
:FONTQLTY
  FONTCPI=171
  QLTYTYPE=DRAFT
  DATA ='1B49010F'X.
:FONTQLTY
  FONTCPI=PROP
  QLTYTYPE=DRAFT
  DATA ='1B49011B3A'X.
:FONTQLTY
  FONTCPI=10
  QLTYTYPE=LETTER
  DATA ='1B490312'X.
:FONTQLTY
  FONTCPI=12
  QLTYTYPE=LETTER
  DATA ='1B49031B3A'X.
:FONTQLTY
  FONTCPI=171
  QLTYTYPE=LETTER
  DATA ='1B49030F'X.
:FONTQLTY
  FONTCPI=PROP
  QLTYTYPE=LETTER
  DATA ='1B49031B3A'X.
:FONTQLTY
  FONTCPI=10
  QLTYTYPE=TEXT
  DATA ='1B490212'X.
:FONTQLTY
  FONTCPI=12
  QLTYTYPE=TEXT
  DATA ='1B49021B3A'X.
:FONTQLTY
  FONTCPI=171
  QLTYTYPE=TEXT
  DATA ='1B49020F'X.
:FONTQLTY
  FONTCPI=PROP
  QLTYTYPE=TEXT
  DATA ='1B49021B3A'X.
:STRBOLD
  DATA ='1B45'X.
:ENDBOLD
  DATA ='1B46'X.
:PRTFEED
  FEEDTYPE=MANUAL
  DATA ='1B5B460300010100'X.
:DWRSLT
  DRAWER=DRAWER1
  DATA ='1B5B460300030101'X.
:DWRSLT
  DRAWER=DRAWER2
  DATA ='1B5B460300030102'X.
:DWRSLT
  DRAWER=ENVELOPE
  DATA ='1B5B460300030200'X.
:PRTORIENT
  ORIENT=PORTRAIT
  DATA ='1B6B'X.
:PRTORIENT
  ORIENT=LANDSCAPE
  DATA ='1B6C'X.
:PRTORIENT
  ORIENT=RTT180
  DATA ='1B6B'X.
:PRTORIENT
  ORIENT=RTT270
  DATA ='1B6C'X.
:STRUS
  DATA ='1B2D01'X.
:ENDUS
  DATA ='1B2D00'X.
:STRSUPS
  DATA ='1B5300'X.
:ENDSUPS
  DATA ='1B54'X.
:STRSUBS
  DATA ='1B5301'X.
:ENDSUBS
  DATA ='1B54'X.
:RVSIDX
  DATA ='1B5D'X.
:INITVON
  DATA ='00001B5B4B070006310184002 4B8
        1B5B5C0400900090001B5B540500
        00000352001B6B'X.
:INITPRT
  DATA ='1B5B4B03000031011B5B5C020090
```

```
          001B5B5405000000035200'X.
  :SPACE
    DATA ='20'X.
  :BELL
    DATA ='07'X.
  :CARRTN
    DATA ='0D'X.
  :LINEFEED
    DATA ='0A'X.
  :FORMFEED
    DATA ='0C'X.
  :PRTCTL
    DATA ='00'X.
  :BSP
    DATA ='08'X.
  :COLLATE
    DATA =   12.
  :PAGSIZPFT
    PAGWTH=12240
    PAGLEN=20160
    PAPER=MANUAL.
  :PAGSIZPFT
    PAGWTH=12240
    PAGLEN=15840
    PAPER=DRAWER1.
  :PAGSIZPFT
    PAGWTH=12240
    PAGLEN=15840
    PAPER=DRAWER2.
  :PAGSIZPFT
    PAGWTH=12240
    PAGLEN= 5760
    PAPER=ENVELOPE.
  :MARGIN
    OPTION=TOP
    ORIENT=PORTRAIT
    DATA =  240.
  :MARGIN
    OPTION=LEFT
    ORIENT=PORTRAIT
    DATA =  360.
  :MARGIN
    OPTION=RIGHT
    ORIENT=PORTRAIT
    DATA =  360.
  :MARGIN
    OPTION=BOTTOM
    ORIENT=PORTRAIT
    DATA =  240.
  :MARGIN
    OPTION=TOP
    ORIENT=LANDSCAPE
    DATA =  360.
  :MARGIN
    OPTION=LEFT
    ORIENT=LANDSCAPE
    DATA =  240.
  :MARGIN
    OPTION=RIGHT
    ORIENT=LANDSCAPE
    DATA =  240.
  :MARGIN
    OPTION=BOTTOM
    ORIENT=LANDSCAPE
    DATA =  360.
  :ASCIICTL
    ASCII='16'X
    DATA ='1B5B5405000002O1B5009E1B5B54050
          00000035200'X.
  :ASCIICTL
    ASCII='14'X
    DATA ='1B5E14'X.
  :ASCIICTL
    ASCII='15'X
    DATA ='1B5E15'X.
  :ASCIICTL
    ASCII='18'X
    DATA ='1B5E18'X.

:PMLGMAPTBL.

  :PMLGEBCTBL
    EBCDICCP=   29
    ASCIICP=  850
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '2020838485A0C68687A4E82E3C282B21'X      /* 4- */
     '268288898AA18C8B8DE192242A293B99'X      /* 5- */
     '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
     '9B90D2D3D4D6D7D8DED03A23D1273D22'X      /* 7- */
     '9D6162636465666676869AEAF60EC7BF1'X     /* 8- */
     'F86A6B6C6D6E6F707172A6A77DF75DCF'X      /* 9- */
     'E6947374757677787797AADA840ED5BA9'X     /* A- */
     'BD9CBEFAB8F5F4ACABF3AA7CEEF95C9E'X      /* B- */
     'E7414243444546474849F0937E95A2E4'X      /* C- */
     '914A4B4C4D4E4F505152FB968197A398'X      /* D- */
     'EFF65354555657585595AFDE25EE3E0E5'X     /* E- */
     '30313233343536373839FCEA9AEBE900'X.     /* F- */
  :PFNTWTH
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '0A0A0A0A0A0A0A0A0C0E0A0A0A0A0A'X        /* 4- */
     '0C0A0A0A0A060606060C0E0A0A0A0A0E'X      /* 5- */
     '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
     '0A0C0C0C0C080808080C0A0A0E060A0A'X      /* 7- */
     '0E0A0C0A0C0A080C0C060E0E0A0C0A0A'X      /* 8- */
     '0A060C060E0C0A0C0C0A0A0A0A0A0A0A'X      /* 9- */
     '0C0A0A080C0C0E0C0C0A0A0A0A0E0A0A'X      /* A- */
     '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
     '0C0E0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X    /* C- */
     '0E0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
     '0A0A0C0E0E0E0E0E0E0C0A0E0A0E0E0E'X      /* E- */
     '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

  :PMLGEBCTBL
    EBCDICCP=   30
    ASCIICP=  853
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '2020838485A000917BA4802E3C282B21'X      /* 4- */
     '268288898AA18C8B8DE1A6982A293B5E'X      /* 5- */
     '2D2FB686B7B500925BA5AD2C255F3E3F'X      /* 6- */
     '0090D2D3D4D6D7D8DED53A99B8273D9A'X      /* 7- */
     'F461626364656676869E886C7ED007C'X       /* 8- */
     'F86A6B6C6D6E6F707172A99B9FF700CF'X      /* 9- */
     'E6947374757677787797AE78FC6EC0040'X     /* A- */
     'FA9CBE7DBDF55D00AB24A89DACF9EF9E'X      /* B- */
     '8741424344454647484 9F0937E95A2E5'X     /* C- */
     'A74A4B4C4D4E4F50515260965C81A300'X      /* D- */
     '81F65354555657585595AFDE223E3E0E4'X     /* E- */
     '30313233343536373839FCEA22EBE900'X.     /* F- */
  :PFNTWTH
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '0A0A0A0A0A0A0A0A0C0E0A0A0A0A0A'X        /* 4- */
     '0C0A0A0A0A060606060C0E080A0A0A0A'X      /* 5- */
     '0A0A0E0E0E0E0E0E0A0E0A0A0A0A0A0A'X      /* 6- */
     '0A0C0C0C0C08080808060A0E0C060A0E'X      /* 7- */
     '0A0A0C0A0C0A080C0C060C0A0A0C0C0A'X      /* 8- */
     '0A060C060E0C0A0C0C0A0C0C060A0E0A'X      /* 9- */
     '0C0A0A080C0C0E0C0C0A0E0E0C0E0E0A'X      /* A- */
     '0A0A0A0A0C0A0A0A0A0A0E0E0C0E0A0A'X      /* B- */
     '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0C'X      /* C- */
     '0C0A0E0C0E0E0E0C0E0E0A0C0A0C0C0C'X      /* D- */
     '0C0A0C0E0E0E0E0E0E0C0A0E0A0E0E0E'X      /* E- */
     '0A0A0A0A0A0A0A0A0A0A0A0E0A0E0E0A'X.     /* F- */

  :PMLGEBCTBL
    EBCDICCP=   37
    ASCIICP=  850
    DATA =
   /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
     '2020838485A0C68687A4BD2E3C282B7C'X      /* 4- */
     '26828889BAA18C8B8DE121242A293BAA'X      /* 5- */
     '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
     '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
     '9D6162636465666676869AEAFD0ECE7F1'X     /* 8- */
     'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
     'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
     '5E9CBEFA9FF5F4ACABF35B5DEEF9EFF2'X      /* B- */
     '7B4142434445464748 49F0939495A2E4'X     /* C- */
     '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
     '5C2053545556575859 5AFDE299E3E0E5'X     /* E- */
     '30313233343536373839FCEA9AEBE900'X.     /* F- */
  :PFNTWTH
    DATA =
```

```
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '0A0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
   '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
   '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
   '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
   '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
   '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
   '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
   '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
   '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
   '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
   '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
   '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=   38
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '2020838485A0C68687A45B2E3C282B21'X      /* 4- */
   '268288898AA18C8B8DE15D242A293B5E'X      /* 5- */
   '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
   '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
   '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
   'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
   'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
   'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
   '7B41424344454647484 9F0939495A2E4'X     /* C- */
   '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
   '5C205354555657585 95AFDE299E3E0E5'X     /* E- */
   '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '0A0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
   '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
   '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
   '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
   '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
   '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
   '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
   '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
   '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
   '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
   '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
   '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=   45
  ASCIICP=  880
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '2020838461A0C79F87865B2E3C282B21'X      /* 4- */
   '2682A98985A18C9692E15D242A293B5E'X      /* 5- */
   '2D2FB68E41B5C6AC808FD82C255F3E3F'X      /* 6- */
   'F390A8D3DED6D79591603A2340273D22'X      /* 7- */
   'F4616263646566676869 98E5D0ECFDAD'X     /* 8- */
   'F86A6B6C6D6E6F70717288E4E7F7F2CF'X      /* 9- */
   'A5F1737475767778797AB8D5D1EDFCB8'X      /* A- */
   'FA4CBEDDBDF5A7ABA68D9DE3E6F9EFEE'X      /* B- */
   'D4414243444546474849F09394EAA28B'X      /* C- */
   'D24A4B4C4D4E4F505152B7FB819CA379'X      /* D- */
   'A420535455565758595A32E299E8E08A'X      /* E- */
   '3031323334353637383933EB9A9BE900'X.     /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* 4- */
   '0C0A0A0A0C060606060C0A0A0A0A0A0A'X      /* 5- */
   '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
   '0A0C0C0C0E080808080A0A0A0A060A0A'X      /* 7- */
   '0A0A0C0A0C0A080C0C060A0C0C0C0A0A'X      /* 8- */
   '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
   '0A0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
   '0A0C0A0E0C0A0A0A0C0C0C0E0C0A0A08'X      /* B- */
   '0C0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
   '0E0A0E0C0C0E0E0E0C0E0E0C0C0C080C0C'X    /* D- */
   '0E0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
   '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  256
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '2020838485A0C68687A45B2E3C282B21'X      /* 4- */
   '268288898AA18C8B8DE15D242A293B5E'X      /* 5- */
   '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
   '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
   '9D6162636465666768 69AEAFD0ECE7F1'X     /* 8- */
   'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
   'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
   'BD9CBEF169FF5F4ACABF3AA7CEEF9EFF2'X     /* B- */
   '7B41424344454647484 9F0939495A2E4'X     /* C- */
   '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
   '5C205354555657585 95AFDE299E3E0E5'X     /* E- */
   '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '0A0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
   '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
   '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
   '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
   '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
   '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
   '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
   '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
   '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
   '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
   '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
   '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  257
  ASCIICP=  855
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '202080828486888A8C8E5B2E3C282B21'X      /* 4- */
   '2690929496989A9EEF815D242A293B5E'X      /* 5- */
   '2D2F838587898B8D8F917C2C255F3E3F'X      /* 6- */
   '9395972D999B9CA0A2603A2340273D22'X      /* 7- */
   '9A6162636465666768 69A6A8AAACB5B7'X     /* 8- */
   'BD6A6B6C6D6E6F707172C6D0D2D4D6D8'X      /* 9- */
   'DE7E737475767778797AE1E3E5E7E9EB'X      /* A- */
   'EDF1F3F5F7F9FB9E9DA1A3A5A7A9ABAD'X      /* B- */
   '7B41424344454647484 9B6B8BEC7D1D3'X     /* C- */
   '7D4A4B4C4D4E4F505152D5D7DDE0E2E4'X      /* D- */
   '5CCF535455565758595AE6E8EAECEEF2'X      /* E- */
   '3031323334353637383 9F4F6F8FAFC00'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '0A0A0C0A0A0A0A0606060A0A0A0A0A0A'X      /* 4- */
   '0C0E0E0C0C0C0C0E0E0E0A0A0A0A0A0A'X      /* 5- */
   '0A0A0E0E0E0C08080A0E0A0A0A0A0A0A'X      /* 6- */
   '0E0E0E0A0E0E0E0A0A0A0A0A060A0A'X        /* 7- */
   '0C0A0C0A0C0A080C0C060C0A0C0A0A0C'X      /* 8- */
   '0C060C060E0C0A0C0C0A0C0C0E0C0A0A'X      /* 9- */
   '0C0A0A080C0C0E0C0C0A0C0A0A0C0E0C'X      /* A- */
   '0C0E0A0E0A0E0C0C0E0E0E0E0E0C0E0E'X      /* B- */
   '0A0E0E0E0E0C0C0E0E080E0E0E0E0E0E'X      /* C- */
   '0A0A0E0C0E0E0E0C0E0E0E0E0E0C0E'X        /* D- */
   '0A0A0C0E0E0E0E0E0E0C0E0E0E0E0E'X        /* E- */
   '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.       /* F- */

:PMLGEBCTBL
  EBCDICCP=  258
  ASCIICP=  852
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '20208384EEA0C79F87865B2E3C282B21'X      /* 4- */
   '2682A98985A18C9692E15D242A293B5E'X      /* 5- */
   '2D2FB68EF1B5C6AC808F7C2C255F3E3F'X      /* 6- */
   'F390A8D3DED6D79591603A2340273D22'X      /* 7- */
   'F4616263646566676869 98E5D0ECFDAD'X     /* 8- */
   'F86A6B6C6D6E6F70717288E4E7F7F2CF'X      /* 9- */
   'A57E737475767778797AB8D5D1EDFCB8'X      /* A- */
   'FAA4BEDDBDF5A7ABA68D9DE3E6F9EF9E'X      /* B- */
   '7B41424344454647484 9F09394EAA28B'X     /* C- */
   '7D4A4B4C4D4E4F505152B7FB819CA3DB'X      /* D- */
   '5CF653545556758595AD4E299E8E08A'X       /* E- */
   '3031323334353637383 9D2EB9A9BE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
```

```
    '0A0A0A0A080A0A0A0A0A0A0A0A0A0A'X      /* 4- */
    '0C0A0A0A0C060606060C0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0A0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0E080808080A0A0A0A060A0A'X      /* 7- */
    '0A0A0C0A0C0A080C0C060A0C0C0C0A0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A060C0A0A0A0A'X      /* 9- */
    '0A0A0A080C0C0E0C0C0A0C0E0E0E0E0C'X      /* A- */
    '0A0E0A0E0C0A0A0A0C0C0C0E0C0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E0C0C0C080C0A'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0C0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  259
  ASCIICP=  899
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838485A0B08687A45B2E3C282B21'X      /* 4- */
    '268288898AA18C8B8DB15D242A293B5E'X      /* 5- */
    '2D2FB28EB3B4B58F80A57C2C255F3E3F'X      /* 6- */
    'B690B7B8B9BA00BCBD603A2340273D22'X      /* 7- */
    'BE6162636465666676869AEAFBFC0C1C2'X     /* 8- */
    'C36A6B6C6D6E6F707172A6A791C492C5'X      /* 9- */
    'C67E737475767778797AADA8C7C8C9CA'X      /* A- */
    '9B9C9D9E9FE0CBACABCCAACDCECFD0D1'X      /* B- */
    '7B41424344454647484 9D2939495A2D3'X     /* C- */
    '7D4A4B4C4D4E4F505152D4968197A398'X      /* D- */
    '5C20535455565758595AD5D699D7D8D9'X      /* E- */
    '3031323334353637383 9DADB9ADCDD00'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0A0A0A0A060A0A'X      /* 6- */
    '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  260
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838485A0C68687A45B2E3C282B21'X      /* 4- */
    '268288898AA18C8B8DE15D242A293B5E'X      /* 5- */
    '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
    '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
    '9D6162636465666676869AEAFD0ECE7F1'X     /* 8- */
    'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
    'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
    'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
    '7B41424344454647484 9F0939495A2E4'X     /* C- */
    '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
    '5C20535455565758595AFDE299E3E0E5'X      /* E- */
    '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.       /* F- */

:PMLGEBCTBL
  EBCDICCP=  273
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020837B85A0C68687A48E2E3C282B21'X      /* 4- */
    '26828889BAA18C8B8D7E9A242A293B5E'X      /* 5- */
    '2D2FB65BB7B5C78F80A5942C255F3E3F'X      /* 6- */
    '9B90D2D3D4D6D7D8DE603A23F5273D22'X      /* 7- */
    '9D6162636465666676869AEAFD0ECE7F1'X     /* 8- */
    'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
    'E6E1737475767778797AADA8D1EDE8A9'X      /* A- */
    'BD9CBEFA9F40F4ACABF3AA7CEEF9EFF2'X      /* B- */
    '8441424344454647484 9F093D095A2E4'X     /* C- */
    '814A4B4C4D4E4F505152D5967D97A398'X      /* D- */
    '9920535455565758595AFDE25CE3E0E5'X      /* E- */
    '3031323334353637383 9FCEA5DEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0E0A0A0A0A0A'X      /* 4- */
    '0C0A0A0A0A060606060C0E0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0C0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0C0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0E0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  274
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838440A0C6865CA45B2E3C282B21'X      /* 4- */
    '267B88897DA18C8B8DE15D242A293B5E'X      /* 5- */
    '2D2FB68EB7B5C78F80A5972C255F3E3F'X      /* 6- */
    '9B90D2D3D4D6D7D8DE603A2385273D22'X      /* 7- */
    '9D6162636465666676869AEAFD0ECE7F1'X     /* 8- */
    'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
    'E6F9737475767778797AADA8D1EDE8A9'X      /* A- */
    'BD9CBEFA9FF5F4ACABF3AA7CEE7EEFF2'X      /* B- */
    '8241424344454647484 9F0939495A2E4'X     /* C- */
    '8A4A4B4C4D4E4F505152D5968197A398'X      /* D- */
    '8720535455565758595AFDE299E3E0E5'X      /* E- */
    '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0C0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.       /* F- */

:PMLGEBCTBL
  EBCDICCP=  275
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838485A06086D0A4902E3C282B21'X      /* 4- */
    '267B8889BAA18C8B8DE124802A293B5E'X      /* 5- */
    '2D2FB68EB7B5408F5DA5872C255F3E3F'X      /* 6- */
    '9B5BD2D3D4D6D7D8DEC63AE5C7273D22'X      /* 7- */
    '9D6162636465666676869AEAFD0ECE7F1'X     /* 8- */
    'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
    'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
    'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
    'E4414243444546474849F0939495A27B'X      /* C- */
    '824A4B4C4D4E4F505152D5968197A398'X      /* D- */
    '5C20535455565758595AFDE299E3E023'X      /* E- */
    '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0C0A0A0A0A0A'X      /* 4- */
```

```
    '0C0A0A0A0A060606060C0A0E0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0C080808080A0A0E0E060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  276
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '202083845BA0C68687A4852E3C282B21'X      /* 4- */
    '267B88897DA18C8B8DE1EF242A293B5E'X      /* 5- */
    '2D2FB68EB7B5C78F80A5972C255F3E3F'X      /* 6- */
    '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
    '9D6162636465666676869AEAFD0ECE7F1'X     /* 8- */
    'F86A6B6C6D6E6F707172A6A7915C92CF'X      /* 9- */
    'E6F97374757677787 97AADA8D1EDE8A9'X     /* A- */
    'BD9CBEFA9FF5F4ACABF3AA7CEE7E5DF2'X      /* B- */
    '82414243444546474849F0939495A2E4'X      /* C- */
    '8A4A4B4C4D4E4F505152D5968197A398'X      /* D- */
    'F720535455565758595AFDE299E3E0E5'X      /* E- */
    '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X        /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0C0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  277
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838485A0C67B87A4232E3C282B21'X      /* 4- */
    '26828889BAA18C8B8DE1CF8F2A293B5E'X      /* 5- */
    '2D2FB68EB7B5C72480A59B2C255F3E3F'X      /* 6- */
    'D090D2D3D4D6D7D8DE603A929D273D22'X      /* 7- */
    '4061626364656676869AEAFD0ECE7F1'X       /* 8- */
    'F86A6B6C6D6E6F707172A6A77BF75B5D'X      /* 9- */
    'E6817374757677787 97AADA8D1EDE8A9'X     /* A- */
    'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
    '91414243444546474849F0939495A2E4'X      /* C- */
    '864A4B4C4D4E4F505152D5967E97A398'X      /* D- */
    '5C20535455565758595AFDE299E3E0E5'X      /* E- */
    '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X        /* 4- */
    '0C0A0A0A0A060606060C0A0E0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0A0A0A0A0A0A'X        /* 6- */
    '0A0C0C0C0C080808080A0A0E0E060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0C0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0C0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.       /* F- */

:PMLGEBCTBL
  EBCDICCP=  278
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020837B85A0C67D87A4F52E3C282B21'X      /* 4- */
    '26608889BAA18C8B8DE1CF8F2A293B5E'X      /* 5- */
    '2D2FB623B7B5C72480A5942C255F3E3F'X      /* 6- */
    '9B5CD2D3D4D6D7D8DE823A8E99273D22'X      /* 7- */
    '9D6162636465666676869AEAFD0ECE7F1'X     /* 8- */
    'F86A6B6C6D6E6F707172A6A791F7925D'X      /* 9- */
    'E6817374757677787 97AADA8D1EDE8A9'X     /* A- */
    'BD9CBEFA9F5BF4ACABF3AA7CEEF9EFF2'X      /* B- */
    '84414243444546474849F093D095A2E4'X      /* C- */
    '864A4B4C4D4E4F505152D5967E97A398'X      /* D- */
    '9020535455565758595AFDE240E3E0E5'X      /* E- */
    '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X        /* 4- */
    '0C0A0A0A0A060606060C0A0E0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0A0A0A0A0A0A'X        /* 6- */
    '0A0C0C0C0C080808080A0A0E0E060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0C0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0C0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.       /* F- */

:PMLGEBCTBL
  EBCDICCP=  279
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838440A0C6865CA4F82E3C282B21'X      /* 4- */
    '267B88897DA18C8B8DE1F5242A293B5E'X      /* 5- */
    '2D2FB68EB7B5C78F80A5972C255F3E3F'X      /* 6- */
    '9B90D2D3D4D6D7D8DEE63A9C85273D22'X      /* 7- */
    '9D6162636465666676869AEAFD0ECE7F1'X     /* 8- */
    '5B6A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
    '60F97374757677787 97AADA8D1EDE8A9'X     /* A- */
    'BD23BEFA9F5DF4ACABF3AA7CEE7EEFF2'X      /* B- */
    '82414243444546474849F0939495A2E4'X      /* C- */
    '8A4A4B4C4D4E4F505152D59681D0A398'X      /* D- */
    '8720535455565758595AFDE299E3E0E5'X      /* E- */
    '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X        /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0C0A0A0A0A0A'X        /* 6- */
    '0A0C0C0C0C080808080C0A0A0A060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0E0E0E0A'X.         /* F- */

:PMLGEBCTBL
  EBCDICCP=  280
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '202083847BA0C6865CA4F82E3C282B21'X      /* 4- */
    '265D88897DA18C8B7EE182242A293B5E'X      /* 5- */
    '2D2FB68EB7B5C78F80A5952C255F3E3F'X      /* 6- */
    '9B90D2D3D4D6D7D8DE973A9CF5273D22'X      /* 7- */
    '9D6162636465666676869AEAFD0ECE7F1'X     /* 8- */
    '5B6A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
    'E68D737475767778797AADA8D1EDE8A9'X      /* A- */
    'BD23BEFA9F40F4ACABF3AA7CEEF9EFF2'X      /* B- */
    '85414243444546474849F09394D0A2E4'X      /* C- */
    '8A4A4B4C4D4E4F505152D5968160A398'X      /* D- */
    '8720535455565758595AFDE299E3E0E5'X      /* E- */
    '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X        /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
```

```
    '0A0A0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0C080808080C0A0A0A060A0A'X    /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X    /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X    /* 9- */
    '0C060A080C0C0E0C0C0A0A0A0E0E0E0A'X    /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X    /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X    /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X    /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X    /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.   /* F- */

:PMLGEBCTBL
  EBCDICCP=  281
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '2020838485A0C68687A49C2E3C282B7C'X     /* 4- */
   '26828889BAA18C8B8DE121BE2A293BAA'X     /* 5- */
   '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X     /* 6- */
   '9B90D2D3D4D6D7D8DE603A2340273D22'X     /* 7- */
   '9D6162636465666768 69AEAFD0ECE7F1'X    /* 8- */
   'F86A6B6C6D6E6F707172A6A791F792CF'X     /* 9- */
   'E6EE737475767778797AADA8D1EDE8A9'X     /* A- */
   'BD5B5CFA9FF5F4ACABF35E5D7EF9EFF2'X     /* B- */
   '7B41424344454647484 9F0939495A2E4'X    /* C- */
   '7D4A4B4C4D4E4F505152D5968197A398'X     /* D- */
   '2420535455565758595AFDE299E3E0E5'X     /* E- */
   '3031323334353637383 9FCEA9AEBE900'X.   /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
   '0C0A0A0A0A060606060C0A0A0A0A0A0A'X     /* 5- */
   '0A0A0E0E0E0E0E0E0E0A0A0A0A0A0A'X       /* 6- */
   '0A0C0C0C0C080808080A0A0E0A060A0A'X     /* 7- */
   '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X     /* 8- */
   '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X     /* 9- */
   '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X     /* A- */
   '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X     /* B- */
   '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X     /* C- */
   '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X     /* D- */
   '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X     /* E- */
   '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.      /* F- */

:PMLGEBCTBL
  EBCDICCP=  282
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '2020838485A07B867EA45B2E3C282B21'X     /* 4- */
   '26828889BAA18C8B8DE15D242A293B5E'X     /* 5- */
   '2D2FB68EB7B5238F5CA5E42C255F3E3F'X     /* 6- */
   '9B90D2D3D4D6D7D8DE603AC7E5273D22'X     /* 7- */
   '9D6162636465666768 69AEAFD0ECE7F1'X    /* 8- */
   'F86A6B6C6D6E6F707172A6A791F792CF'X     /* 9- */
   'E6877374757677787 97AADA8D1EDE8A9'X    /* A- */
   'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X     /* B- */
   'C6414243444546474849F0939495A2D0'X     /* C- */
   'EF4A4B4C4D4E4F505152D5968197A398'X     /* D- */
   '8020535455565758595AFDE299E3E040'X     /* E- */
   '3031323334353637383 9FCEA9AEBE900'X.   /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
   '0C0A0A0A0A060606060C0A0A0A0A0A0A'X     /* 5- */
   '0A0A0E0E0E0E0E0E0E0A0A0A0A0A0A'X       /* 6- */
   '0A0C0C0C0C080808080A0A0E0A060A0A'X     /* 7- */
   '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X     /* 8- */
   '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X     /* 9- */
   '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X     /* A- */
   '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X     /* B- */
   '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X     /* C- */
   '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X     /* D- */
   '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X     /* E- */
   '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.      /* F- */

:PMLGEBCTBL
  EBCDICCP=  283
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '2020838485A0C68687D05B2E3C282B7C'X     /* 4- */
   '26828889BAA18C8B8DE15D502A293BAA'X     /* 5- */
   '2D2FB68EB7B5C78F8023A42C255F3E3F'X     /* 6- */
   '9B90D2D3D4D6D7D8DE603AA540273D22'X     /* 7- */
   '9D6162636465666768 69AEAFD0ECE7F1'X    /* 8- */
   'F86A6B6C6D6E6F707172A6A791F792CF'X     /* 9- */
   'E6F9737475767778797AADA8D1EDE8A9'X     /* A- */
   'BD9CBEFA9FF5F4ACABF35E21EE7EEFF2'X     /* B- */
   '7B41424344454647484 9F0939495A2E4'X    /* C- */
   '7D4A4B4C4D4E4F505152D5968197A398'X     /* D- */
   '5C2053545556575859 5AFDE299E3E0E5'X    /* E- */
   '3031323334353637383 9FCEA9AEBE900'X.   /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
   '0C0A0A0A0A060606060C0A0E0A0A0A0A'X     /* 5- */
   '0A0A0E0E0E0E0E0E0E0C0A0A0A0A0A'X       /* 6- */
   '0A0C0C0C0C080808080A0A0E0A060A0A'X     /* 7- */
   '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X     /* 8- */
   '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X     /* 9- */
   '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X     /* A- */
   '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X     /* B- */
   '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X     /* C- */
   '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X     /* D- */
   '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X     /* E- */
   '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.      /* F- */

:PMLGEBCTBL
  EBCDICCP=  284
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '2020838485A0C68687D05B2E3C282B7C'X     /* 4- */
   '26828889BAA18C8B8DE15D242A293BAA'X     /* 5- */
   '2D2FB68EB7B5C78F8023A42C255F3E3F'X     /* 6- */
   '9B90D2D3D4D6D7D8DE603AA540273D22'X     /* 7- */
   '9D6162636465666768 69AEAFD0ECE7F1'X    /* 8- */
   'F86A6B6C6D6E6F707172A6A791F792CF'X     /* 9- */
   'E6F9737475767778797AADA8D1EDE8A9'X     /* A- */
   'BD9CBEFA9FF5F4ACABF35E21EE7EEFF2'X     /* B- */
   '7B41424344454647484 9F0939495A2E4'X    /* C- */
   '7D4A4B4C4D4E4F505152D5968197A398'X     /* D- */
   '5C2053545556575859 5AFDE299E3E0E5'X    /* E- */
   '3031323334353637383 9FCEA9AEBE900'X.   /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
   '0C0A0A0A0A060606060C0A0E0A0A0A0A'X     /* 5- */
   '0A0A0E0E0E0E0E0E0E0C0A0A0A0A0A'X       /* 6- */
   '0A0C0C0C0C080808080A0A0E0A060A0A'X     /* 7- */
   '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X     /* 8- */
   '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X     /* 9- */
   '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X     /* A- */
   '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X     /* B- */
   '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X     /* C- */
   '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X     /* D- */
   '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X     /* E- */
   '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.      /* F- */

:PMLGEBCTBL
  EBCDICCP=  285
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '2020838485A0C68687A4242E3C282B7C'X     /* 4- */
   '26828889BAA18C8B8DE1219C2A293BAA'X     /* 5- */
   '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X     /* 6- */
   '9B90D2D3D4D6D7D8DE603A2340273D22'X     /* 7- */
   '9D6162636465666768 69AEAFD0ECE7F1'X    /* 8- */
   'F86A6B6C6D6E6F707172A6A791F792CF'X     /* 9- */
   'E6EE737475767778797AADA8D1EDE8A9'X     /* A- */
   'BD5BBEFA9FF5F4ACABF35E5D7EF9EFF2'X     /* B- */
   '7B41424344454647484 9F0939495A2E4'X    /* C- */
   '7D4A4B4C4D4E4F505152D5968197A398'X     /* D- */
   '5C2053545556575859 5AFDE299E3E0E5'X    /* E- */
   '3031323334353637383 9FCEA9AEBE900'X.   /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
   '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
   '0C0A0A0A0A060606060C0A0E0A0A0A0A'X     /* 5- */
   '0A0A0E0E0E0E0E0E0E0A0A0A0A0A0A'X       /* 6- */
```

```
    '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  290
  ASCIICP=  911
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '20A1A2A3A4A5A6A7A8A9E12E3C282B7C'X      /* 4- */
    '26AAABACADAEAF00B000215C2A293BE3'X       /* 5- */
    '2D2F000000000000000000002C255F3E3F'X     /* 6- */
    '00000000000000000000003A2340273D22'X     /* 7- */
    '00B1B2B3B4B5B6B7B8B9BA00BBBCBDBE'X       /* 8- */
    'BFC0C1C2C3C4C5C6C7C8C90000CACBCC'X       /* 9- */
    '007ECDCECFD0D1D2D3D4D500D6D7D8D9'X       /* A- */
    '000000000000000000000DADBDCDDDEDF'X      /* B- */
    '7B41424344454647484900000000000000'X     /* C- */
    '004A4B4C4D4E4F505152000000000000'X       /* D- */
    '242053545556578595A000000000000'X        /* E- */
    '3031323334353637383900000000000000'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X       /* 5- */
    '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X       /* 6- */
    '0A0C0C0C0C080808080C0A0A0A060A0A'X       /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X       /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X       /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X       /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X       /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X       /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X       /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X       /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.      /* F- */

:PMLGEBCTBL
  EBCDICCP=  293
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838485A0C68687A45B2E3C282B21'X       /* 4- */
    '268288898AA18C8B8DE15D242A293B5E'X       /* 5- */
    '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X       /* 6- */
    '9B90D2D3D4D6D7D8DE603A2340273D22'X       /* 7- */
    '9D6162636465666768 69AEAFD0ECE7F1'X      /* 8- */
    'F86A6B6C6D6E6F707172A6A791F792CF'X       /* 9- */
    'E67E737475767778797AADA8D1EDE8A9'X       /* A- */
    'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X       /* B- */
    '7B41424344454647484 9F0939495A2E4'X      /* C- */
    '7D4A4B4C4D4E4F505152D5968197A398'X       /* D- */
    '5C2053545556575859 5AFDE299E3E0E5'X      /* E- */
    '303132333435363738 39FCEA9AEBE900'X.     /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X         /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X       /* 5- */
    '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X       /* 6- */
    '0A0C0C0C0C080808080A0A0A060A0A'X         /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X       /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X       /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X       /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X       /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X       /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X       /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X       /* E- */
    '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.        /* F- */

:PMLGEBCTBL
  EBCDICCP=  297
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838440A0C6865CA4F82E3C282B21'X       /* 4- */
    '267B88897DA18C8B8DE1F5242A293B5E'X       /* 5- */
    '2D2FB68EB7B5C78F80A5972C255F3E3F'X       /* 6- */
    '9B90D2D3D4D6D7D8DEE63A9C85273D22'X       /* 7- */
    '9D6162636465666768 69AEAFD0ECE7F1'X      /* 8- */
    '5B6A6B6C6D6E6F707172A6A791F792CF'X       /* 9- */
    '60F97374757677787 97AADA8D1EDE8A9'X      /* A- */
    'BD23BEFA9F5DF4ACABF3AA7CEE7EEFF2'X       /* B- */
    '824142434445464748 49F0939495A2E4'X      /* C- */
    '8A4A4B4C4D4E4F505152D59681D0A398'X       /* D- */
    '872053545556575859 5AFDE299E3E0E5'X      /* E- */
    '303132333435363738 39FCEA9AEBE900'X.     /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X         /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X       /* 5- */
    '0A0A0E0E0E0E0E0E0E0E0C0A0A0A0A0A'X       /* 6- */
    '0A0C0C0C0C080808080C0A0A0A060A0A'X       /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X       /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X       /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X       /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X       /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X       /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X       /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X       /* E- */
    '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.        /* F- */

:PMLGEBCTBL
  EBCDICCP=  310
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838485A0C68687A45B2E3C282B21'X       /* 4- */
    '268288898AA18C8B8DE15D242A293B5E'X       /* 5- */
    '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X       /* 6- */
    '9B90D2D3D4D6D7D8DE603A2340273D22'X       /* 7- */
    '9D6162636465666768 69AEAFD0ECE7F1'X      /* 8- */
    'F86A6B6C6D6E6F707172A6A791F792CF'X       /* 9- */
    'E67E737475767778797AADA8D1EDE8A9'X       /* A- */
    'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X       /* B- */
    '7B41424344454647484 9F0939495A2E4'X      /* C- */
    '7D4A4B4C4D4E4F505152D5968197A398'X       /* D- */
    '5C2053545556575859 5AFDE299E3E0E5'X      /* E- */
    '303132333435363738 39FCEA9AEBE900'X.     /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X         /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X       /* 5- */
    '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X       /* 6- */
    '0A0C0C0C0C080808080A0A0A060A0A'X         /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X       /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X       /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X       /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X       /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X       /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X       /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X       /* E- */
    '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.        /* F- */

:PMLGEBCTBL
  EBCDICCP=  330
  ASCIICP=  880
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838461A0C79F87865B2E3C282B21'X       /* 4- */
    '2682A98985A18C9692E15D242A293B5E'X       /* 5- */
    '2D2FB68E41B5C6AC808FD82C255F3E3F'X       /* 6- */
    'F390A8D3DED6D79591603A2340273D22'X       /* 7- */
    'F461626364656667686998E5D0ECFDAD'X       /* 8- */
    'F86A6B6C6D6E6F70717288E4E7F7F2CF'X       /* 9- */
    'A5F1737475767778797AB8D5D1EDFCB8'X       /* A- */
    'FA4CBEDDBDF5A7ABA68D9DE3E6F9EFEE'X       /* B- */
    'D4414243444546474849F09394EAA28B'X       /* C- */
    'D24A4B4C4D4E4F505152B7FB819CA379'X       /* D- */
    'A42053545556575859 5A32E299E8E08A'X      /* E- */
    '303132333435363738 3933EB9A9BE900'X.     /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X       /* 4- */
    '0C0A0A0A0C060606060C0A0A0A0A0A0A'X       /* 5- */
    '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X       /* 6- */
    '0A0C0C0C0E080808080A0A0A0A060A0A'X       /* 7- */
```

```
    '0A0A0C0A0C0A080C0C060A0C0C0C0A0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A060C0A0A0A0A'X      /* 9- */
    '0A0A0A080C0C0E0C0C0A0C0E0E0E0E0C'X      /* A- */
    '0A0C0A0E0C0A0A0A0C0C0C0E0C0A0A08'X      /* B- */
    '0C0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0E0A0E0C0E0E0E0C0E0E0C0C0C080C0C'X      /* D- */
    '0E0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  340
  ASCIICP=  876
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838485A0C68687A45B2E3C282B21'X      /* 4- */
    '268288898AA18C8B8DE15D242A293B5E'X      /* 5- */
    '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
    '9B90D2D3D4D6D7C4DE603A2340273D22'X      /* 7- */
    'A7616263646566676869AEAFD0ECE7F1'X      /* 8- */
    'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
    'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
    'BD9C9DFAB8F5F4ACABF3AA7CEEF9EFDB'X      /* B- */
    '7B41424344454647484924D609495A2E4'X      /* C- */
    '7D4A4B4C4D4E4F505152FB7E8197A35E'X      /* D- */
    '5CF6535455565758595AFD5F99E3E0E5'X      /* E- */
    '3031323334353637383FCEA9AEBE900'X.      /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  420
  ASCIICP=  864
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020F1F0E09FC1C2A2C3C02E3C282B7C'X      /* 4- */
    '26A5C40000C6C7A8A9C821242A293BDC'X      /* 5- */
    '2D2FC9AACAABCBADCCAE7C2C255F3E3F'X      /* 6- */
    'CDAFCECFD0D1D2BCD3AC3A2340273D22'X      /* 7- */
    'BD616263646566676869D4BED5EBD6D7'X      /* 8- */
    'D86A6B6C6D6E6F707172DFC5D9ECEEED'X      /* 9- */
    'DADD737475767778797AF7BAE1F8E2FC'X      /* A- */
    'E3FBF9FA999A00009D9EE4EFE5F2E6F3'X      /* B- */
    'BB41424344454647484 9A1E700F400E8'X      /* C- */
    'BF4A4B4C4D4E4F505152E9F5FDF6EAB0'X      /* D- */
    'DEF6535455565758595AB1B200B3B4B5'X      /* E- */
    '3031323334353637383900B6B7B8B900'X.     /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* 4- */
    '0C0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* 5- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* 6- */
    '0A0A0A0A0A0A0A0A0A0A0A0A060A0A0A'X      /* 7- */
    '0A0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0A0A0A0A'X      /* 9- */
    '0A0A0A080C0C0E0C0C0A0A0A0A0A0A0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E0A0A0A0A0A0A'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0A0A0A0A0A'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  423
  ASCIICP=  851
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '20A4A5A6A7A8A9AAACAD5B2E3C282B21'X      /* 4- */
    '26B5B6B7B8BDBEC6C7CF5D242A293B5E'X      /* 5- */
    '2D2FD0D1D2D3D4D500007C2C255F3E3F'X      /* 6- */
    '00868D8F0090929598603A9CF5273D22'X      /* 7- */
    '8E6162636465666768 69D6D7D8DDDEE0'X      /* 8- */
    '996A6B6C6D6E6F707172E1E2E3E4E5E6'X      /* 9- */
    '9AF9737475767778797AE7E8E9EAEBEC'X      /* A- */
    '009B9D9EA09FA2A3FBFDEDEEF2F3F4F6'X      /* B- */
    'F741424344454647484 9F0FA83858488'X      /* C- */
    'EF4A4B4C4D4E4F505152F1828A898C8B'X      /* D- */
    'F8205354555657585 95AAB9493969781'X      /* E- */
    '3031323334353637383798780000000'X.      /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0E0E0E0C0C0C0E0E080A0A0A0A0A0A'X      /* 4- */
    '0C0E0E0E0E0E0E0E0C0E0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0A0A0A0A0A0A0A'X      /* 6- */
    '0A0E0E0E0C0A0E0E0E0A0A0A0A060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060C0C0A0A0A0A'X      /* 8- */
    '0E060C060E0C0A0C0C0A0A0A0A0C0A0A'X      /* 9- */
    '0E0A0A080C0C0E0C0C0A0A0A0A0A0A0A'X      /* A- */
    '0A0A0A0A06060A0C0C0C0A0A0A0A0A0C'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0C0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E0A0A0A0A0606'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0A0A0C0C0C'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0C0A0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  424
  ASCIICP=  862
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2080818283848586878 89B2E3C282B7C'X      /* 4- */
    '26898A8B8C8D8E8F909121242A293BAA'X      /* 5- */
    '2D2F92939495969798997C2C255F3E3F'X      /* 6- */
    '009A0000200000005F603A2340273D22'X      /* 7- */
    '00616263646566676869AEAF000000F1'X      /* 8- */
    'F86A6B6C6D6E6F707172000000000000'X      /* 9- */
    'E67E737475767778797A0000000000052'X      /* A- */
    '5E9C9DF900F5F4ACAB005B5D2D000078'X      /* B- */
    '7B41424344454647484 92D0000000000'X      /* C- */
    '7D4A4B4C4D4E4F505152000000000000'X      /* D- */
    '5CF6535455565758595AFD0000000000'X      /* E- */
    '3031323334353637383900000000000'X.      /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0A0808080A0A0A0A0A060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0E'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0A0A0C0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  500
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838485A0C68687A45B2E3C282B21'X      /* 4- */
    '268288898AA18C8B8DE15D242A293B5E'X      /* 5- */
    '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
    '9B90D2D3D4D6D7D8DE603A2340273D22'X      /* 7- */
    '9D616263646566676869AEAFD0ECE7F1'X      /* 8- */
    'F86A6B6C6D6E6F707172A6A791F792CF'X      /* 9- */
    'E67E737475767778797AADA8D1EDE8A9'X      /* A- */
    'BD9CBEFA9FF5F4ACABF3AA7CEEF9EFF2'X      /* B- */
    '7B41424344454647849F0939495A2E4'X      /* C- */
    '7D4A4B4C4D4E4F505152D5968197A398'X      /* D- */
    '5C2053455565758595AFDE299E3E0E5'X      /* E- */
    '3031323334353637383 9FCEA9AEBE900'X.     /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
```

```
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  870
  ASCIICP=  852
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '20208384EEA0C79F87865B2E3C282B21'X      /* 4- */
    '2682A98985A18C9692E15D242A293B5E'X      /* 5- */
    '2D2FB68EF1B5C6AC808F7C2C255F3E3F'X      /* 6- */
    'F390A8D3DED6D79591603A2340273D22'X      /* 7- */
    'F461626364656667686998E5D0ECFDAD'X      /* 8- */
    'F86A6B6C6D6E6F70717288E4E7F7F2CF'X      /* 9- */
    'A57E737475767778797AB8D5D1EDFCB8'X      /* A- */
    'FAA4BEDDBDF5A7ABA68D9DE3E6F9EF9E'X      /* B- */
    '7B41424344454647484 9F09394EAA28B'X     /* C- */
    '7D4A4B4C4D4E4F505152B7FB819CA3DB'X      /* D- */
    '5CF6535455565758595AD4E299E8E08A'X      /* E- */
    '3031323334353637383 9D2EB9A9BE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A080A0A0A0A0A0A0A0A0A0A0A'X      /* 4- */
    '0C0A0A0A0C060606060C0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0A0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0E080808080A0A0A0A060A0A'X      /* 7- */
    '0A0A0C0A0C0A080C0C060A0C0C0C0A0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A060C0A0A0A0A'X      /* 9- */
    '0A0A0A080C0C0E0C0C0A0C0E0E0E0E0C'X      /* A- */
    '0A0E0A0E0C0A0A0A0C0C0C0E0C0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E0C0C080C0A'X        /* D- */
    '0A0A0C0E0E0E0E0E0E0C0C0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  871
  ASCIICP=  850
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020838485A0C68687A4E82E3C282B21'X      /* 4- */
    '268288898AA18C8B8DE192242A293B99'X      /* 5- */
    '2D2FB68EB7B5C78F80A5DD2C255F3E3F'X      /* 6- */
    '9B90D2D3D4D6D7D8DED03A23D1273D22'X      /* 7- */
    '9D616263646566676869AEAF60EC7BF1'X      /* 8- */
    'F86A6B6C6D6E6F707172A6A77DF75DCF'X      /* 9- */
    'E6947374757677787 97AADA840ED5BA9'X     /* A- */
    'BD9CBEFAB8F5F4ACABF3AA7CEEF95C9E'X      /* B- */
    'E7414243444546474849F0937E95A2E4'X      /* C- */
    '914A4B4C4D4E4F505152FB968197A398'X      /* D- */
    'EFF6535455565758595AFDE25EE3E0E5'X      /* E- */
    '3031323334353637383 9FCEA9AEBE900'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0A0C0E0A0A0A0A0A'X      /* 4- */
    '0C0A0A0A0A0A060606060C0E0A0A0A0A0E'X    /* 5- */
    '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0C080808080C0A0A0E060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0A0C0A0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0A0A0A0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0A0E0A0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0C0E0E0E0E0E0C0C0E0E080A0A0A0E0A0A'X    /* C- */
    '0E0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0A0E0A0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  880
  ASCIICP=  855
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020808284868888A8C8E5B2E3C282B21'X     /* 4- */
    '2690929496989A9EEF815D242A293B5E'X      /* 5- */
    '2D2F838587898B8D8F917C2C255F3E3F'X      /* 6- */
    '9395972D999B9CA0A2603A2340273D22'X      /* 7- */
    '9A616263646566676869A6A8AAACB5B7'X      /* 8- */
    'BD6A6B6C6D6E6F707172C6D0D2D4D6D8'X      /* 9- */
    'DE7E737475767778797AE1E3E5E7E9EB'X      /* A- */
    'EDF1F3F5F7F9FB9E9DA1A3A5A7A9ABAD'X      /* B- */
    '7B41424344454647484 9B6B8BEC7D1D3'X     /* C- */
    '7D4A4B4C4D4E4F505152D5D7DDE0E2E4'X      /* D- */
    '5CCF535455565758595AE6E8EAECEEF2'X      /* E- */
    '3031323334353637383 9F4F6F8FAFC00'X.    /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0C0A0A0A0A0606060A0A0A0A0A0A'X      /* 4- */
    '0C0E0E0C0C0C0C0E0E0E0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0C08080A0E0A0A0A0A0A0A'X      /* 6- */
    '0E0E0E0A0E0E0E0A0A0A0A0A0A060A0A'X      /* 7- */
    '0C0A0C0A0C0A080C0C060C0A0C0A0A0C'X      /* 8- */
    '0C060C060E0C0A0C0C0A0C0C0E0C0A0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0C0A0A0C0E0C'X      /* A- */
    '0C0E0A0E0A0E0C0C0E0E0E0E0E0C0E0E'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080E0E0E0E0E0E'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E0E0E0E0C0E0E'X      /* D- */
    '0A0A0C0E0E0E0E0E0E0C0E0E0E0E0E0E'X      /* E- */
    '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

:PMLGEBCTBL
  EBCDICCP=  892
  ASCIICP=  876
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020000000000000000005B2E3C282B21'X     /* 4- */
    '2600000000000000000005D242A293B00'X     /* 5- */
    '2D2F008E0000008F00A5002C25003E3F'X      /* 6- */
    '000000000000C400003A2340273D22'X        /* 7- */
    'A7616263646566676869000000000000'X      /* 8- */
    '006A6B6C6D6E6F70717200000009200'X       /* 9- */
    '0000737475767778797A000000000000'X      /* A- */
    '009C9D000000000000000007C000000DB'X     /* B- */
    '7B41424344454647484900600000000'X       /* C- */
    '7D4A4B4C4D4E4F505152007E0000005E'X      /* D- */
    '5C2053545556575859 5A005F99000000'X     /* E- */
    '3031323334353637383900009A000000'X.     /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X        /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0C08080808 0A0A0A0A060A0A'X     /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
    '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
    '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
    '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
    '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
    '0A0A0C0E0E0E0E0E0C0A0E0E0E0E0E'X        /* E- */
    '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.       /* F- */

:PMLGEBCTBL
  EBCDICCP=  893
  ASCIICP=  877
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '2020008400000086000005B2E3C282B21'X     /* 4- */
    '260000000000000000E15D242A293B5E'X      /* 5- */
    '2D2F008E0000008F00A5002C255F3E3F'X      /* 6- */
    'A60000000000A9C400603A2340273D22'X      /* 7- */
    'A7616263646566676869000000009B'X        /* 8- */
    '006A6B6C6D6E6F707172000091F792CF'X      /* 9- */
    '007E737475767778797A000000000000'X      /* A- */
    '009C9D0000F500000000007C00F9EFDB'X      /* B- */
    '7B41424344454647484900009400000'X       /* C- */
    '7D4A4B4C4D4E4F50515200008100005E'X      /* D- */
    '5C2053545556575859 5A000099000000'X     /* E- */
    '3031323334353637383900009A000000'X.     /* F- */
:PFNTWTH
  DATA =
 /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
    '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X        /* 4- */
    '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
    '0A0A0E0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
    '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
    '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
    '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
```

```
      '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
      '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
      '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
      '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
      '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
      '0A0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=  905
      ASCIICP=  853
      DATA =
     /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
       '2020838485A000917BA4802E3C282B21'X      /* 4- */
       '268288898AA18C8B8DE1A6982A293B5E'X      /* 5- */
       '2D2FB686B7B500925BA5AD2C255F3E3F'X      /* 6- */
       '0090D2D3D4D6D7D8DED53A99B8273D9A'X      /* 7- */
       'F461626364656676869E886C7ED007C'X       /* 8- */
       'F86A6B6C6D6E6F707172A99B9FF700CF'X      /* 9- */
       'E6947374757677787 97AE78FC6EC0040'X     /* A- */
       'FA9CBE7DBDF55D00AB24A89DACF9EF9E'X      /* B- */
       '87414243444546474849F0937E95A2E5'X      /* C- */
       'A74A4B4C4D4E4F50515260965C81A300'X      /* D- */
       '81F6535455565758595AFDE223E3E0E4'X      /* E- */
       '303132333435363737 3839FCEA22EBE900'X.  /* F- */
    :PFNTWTH
      DATA =
     /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
       '0A0A0A0A0A0A0A0A0C0E0A0A0A0A0A'X      /* 4- */
       '0C0A0A0A0A060606060C0E080A0A0A0A'X      /* 5- */
       '0A0A0E0E0E0E0E0E0A0E0A0A0A0A0A'X      /* 6- */
       '0A0C0C0C0C080808080A0A0A0A060A0E'X      /* 7- */
       '0A0A0C0A0C0A080C0C060C0A0A0C0C0A'X      /* 8- */
       '0A060C060E0C0A0C0C0A0C0C060A0E0A'X      /* 9- */
       '0C0A0A080C0C0E0C0C0A0E0E0C0E0E0A'X      /* A- */
       '0A0A0A0A0C0A0A0A0A0A0E0E0C0E0A0A'X      /* B- */
       '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0C'X      /* C- */
       '0C0A0E0C0E0E0E0C0E0E0A0C0A0C0C0C'X      /* D- */
       '0C0A0C0E0E0E0E0E0E0C0A0E0A0E0E0E'X      /* E- */
       '0A0A0A0A0A0A0A0A0A0A0A0E0E0A0E0A'X.     /* F- */

    :PMLGEBCTBL
      EBCDICCP=  259
      ASCIICP=  850
      DATA =
     /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
       '20200000000000001A2F5B003C28B800'X      /* 4- */
       '00EE3C3E00000000000005DFA0029F95E'X     /* 5- */
       '2D000000BA003C3E0000EF0C0B5F3E00'X      /* 6- */
       '00000000001C00000060F6F1F82722A9'X      /* 7- */
       '0000E100000000000000DAC3C0B30000'X      /* 8- */
       '00000000E60000000000C2C5C1000000'X      /* 9- */
       '007E0000009E00000000BFB4D9C40000'X      /* A- */
       '00000000000000000000000000079CCFBE'X    /* B- */
       '7B000000001B0000F418F00400007C00'X      /* C- */
       '7D00F50000001900000000FE00EE0000'X      /* D- */
       '5C20001A000000000000000AA101E0000'X     /* E- */
       '00FBFDFC0000000000009D0000000000'X.     /* F- */
    :PFNTWTH
      DATA =
     /* -0-1-2-3-4-5-6-7-8-9-A-B-C-D-E-F */
       '0A0A0A0A0A0A0A0A0C0A0A0A0A0A0A'X      /* 4- */
       '0C0A0A0A0A060606060C0A0A0A0A0A0A'X      /* 5- */
       '0A0A0E0E0E0E0E0E0E0A0A0A0A0A0A'X      /* 6- */
       '0A0C0C0C0C080808080A0A0A0A060A0A'X      /* 7- */
       '0E0A0C0A0C0A080C0C060E0E0C0C0C0A'X      /* 8- */
       '0A060C060E0C0A0C0C0A0A0A0E0A0E0A'X      /* 9- */
       '0C0A0A080C0C0E0C0C0A0A0A0E0E0E0A'X      /* A- */
       '0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A'X      /* B- */
       '0A0E0E0E0E0C0C0E0E080A0A0A0A0A0A'X      /* C- */
       '0A0A0E0C0E0E0E0C0E0E060C0C0C0C0C'X      /* D- */
       '0A0A0C0E0E0E0E0E0E0C0A0E0E0E0E0E'X      /* E- */
       '0A0A0A0A0A0A0A0A0A0A0E0E0E0E0A'X.       /* F- */
  :EPMLGMAPTBL.
:EWSCST.
```

Appendix C. Character to Hexadecimal Value Tables

The following tables are provided for your convenience; however, because workstation customizing is dependent on the type of device you are customizing, the values in these tables may not correspond to those that your device supports. Use these tables as a basis to help you find the correct hexadecimal values for your device. To use these tables, use the numbers running along the top of the table as the first hexadecimal digit. For example, the DLE key in Table C-1 is '10'X.

ASCII Character Code to Hexadecimal Value Chart

Table C-1. ASCII Character Code to Hexadecimal Value Conversion Table

Second Hex Digit	First Hex Digit							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENG	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	–	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	—	o	DEL

EBCDIC Character Code to Hexadecimal Value Chart

The characters and the hexadecimal values for this table vary depending on the code page and character set you select when you retrieve the workstation customizing source. For the EBCDIC character code information for all of the supported code pages, see the *National Language Support Planning Guide*.

Table C-2. EBCDIC Character Code to Hexadecimal Value Conversion Table

Second Hex Digit	First Hex Digit															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	DS		Sp	&	-				minus	0			\	0
1	SOH	DC1	SOS				/		a	j	°	1	A	J		1
2	STX	DC2	FS	SYN					b	k	s	2	B	K	S	2
3	ETX	TM							c	l	t	3	C	L	T	3
4	PF	RES	BYP	PN					d	m	u	4	D	M	U	4
5	HT	NL	LF	RS					e	n	v	5	E	N	V	5
6	LC	BS	ETB	UC					f	o	w	6	F	O	W	6
7	DEL	IL	ESC	EOT					g	p	x	7	G	P	X	7
8		CAN							h	q	y	8	H	Q	Y	8
9		EM							i	r	z	9	I	R	Z	9
A	SMM	CC	SM		¢	!	¦	:								
B	VT	CU1	CU2	CU3	.	$	,	#	{	}	└	┘				
C	FF	IFS		DC4	<	*	%	@	≤	¤	┌	┐				
D	CR	IGS	ENQ	NAK	(	)	—	'	(	)	[	]				
E	SO	IRS	ACK		+	;	>	XXX	+	±	≥	≠				
F	SI	IUS	BEL	SUB	|	¬	?	"	+	■	•	–				

Appendix D. Setting Up to Customize a Display

The following procedures are provided to assist you in setting up a display for the workstation customizing procedures. The first procedure helps you get a Sign On screen to be displayed correctly on an ASCII display before you customize it. The second procedure helps you set up a 3477 display and verify it is a Model H when you want to attach an ASCII printer.

Setting Up to Customize an ASCII Display

When the Sign On screen is not displayed correctly, you need to do some customizing before planning for the actual customization. This is because you cannot experiment with the device unless you can at least sign on to the AS/400 system.

The following procedure is provided to help you display a Sign On screen correctly, so that you can determine the remaining the device characteristics you want to customize.

Note: If your ASCII workstation is totally incompatible with the device type you specified in your device description, this procedure may not help you. If you still do not have a Sign On screen displayed on your workstation after following this procedure, try changing the device type and model you specified in the device description for the workstation and for the Retrieve Work Station Customizing Object Source (RTVWSCST) command.

1. Use the RTVWSCST command to retrieve the default source for the device type you selected for your device description. You need the following information for the command parameters:

   Device type
   This should be the same as the device type you specified when you created the device description.

   Keyboard language type
   This should match either the language type supported for your AS/400 system or the language that you want to use for this display.

   Source member name
   This name should be unique so that you can recognize and keep track of your workstation customizing source.

   Source file name / library
   This is QTXTSRC unless you have created a source file specifically for your workstation customizing source members. The library you specify here can contain both the source file and member and the compiled customizing object.

   Text description
   This is any text that you may want to use to help you identify this source member.

2. Use the Start SEU (STRSEU) command to edit the source file member you just created. You will need your device reference manual to map at least the following functions correctly. For more complete information about the tags described here, see Chapter 8, "Customizing ASCII Displays."

   CLRSCN (Clear Screen) Tag
   Check the device reference manual to find the hexadecimal value that your display supports to clear the screen. If the value in your workstation customizing source is not the same as the value listed in your device reference manual, change the hexadecimal value in the source accordingly. (See "Update Screen Tags" on page 8-20 for more information about this tag.)

   Note: If your display does not support the Clear Screen command, try mapping it to one or more other commands supported by your display that will do the equivalent of clearing the screen. For example, on a VT-52** display, sequences for two separate VT-52 commands, Set Home and Clear to End of Screen, can be specified in the DATA keyword for the CLRSCN tag to perform the clear screen function.

   CSRADR (Set Cursor Address) Tag
   This affects the positions of the characters shown on the ASCII display. Use the device reference manual to determine the values for the keywords associated with this tag. Change the values on the CSRADR tag keywords to match those given in your device reference manual. If these values do not apply to your device type, set these keywords to 0. (See "Set Cursor Address (CSRADR) Tag" on page 8-21 for more information about this tag.)

   DSCNTBL (Update Screen Table) Tag
   You do not need to change all of the keyword values for this tag, only the ADDRMOD (address mode) and CHARATR (character) keywords to begin testing the device. Find the values your device uses for these keywords in the device reference manual and specify the correct addressing mode supported by your display. This is CHAR for numeric addressing, and BINARY if your display uses graphic characters. The CHARATR keyword is normally set to FIELD depending on the display type you selected when you retrieved the workstation customizing source.

3. Change the following keyboard function tags so that you can complete the testing for planning workstation customizing. (For more information about these and other keyboard function tags, see "Keyboard Function Tags" on page 8-29.)

ENTER (Enter)
This value is usually '0D'X (carrier return)

TSTREQ (Test request)
This value is usually '1B74'X (ESC + t)

4. Use the Create Work Station Customizing Object (CRTWSCST) command to compile the workstation customizing object. You need the following information for the command parameters:

   WSCST name/Library
   The name you want to call the customizing object. This is the name you will specify in your device description for the ASCII display. The library may be the same as the one you specified for the source file or you may create a library specifically for the compiled customization objects.

   Source member
   This is the same source member name you used to retrieve the source in step 1.

   Text description
   This is a text description you can use to help you further identify the customizing object.

5. Use the Change Device Description (Display) (CHGDEVDSP) command and press F10 (Additional parameters) to show the *Workstation customizing object* (WSCST) field. Type the name of the workstation customizing object you created in the previous step and specify the library where the object is stored.
6. Vary on the device and see if the AS/400 Sign On display is shown. (If the device is already varied on, you must vary it off and then vary it on again for the customizing object to be used with the device.)

Your ASCII display should have at least minimal function and a Sign On screen displayed. If you cannot get the display to this point, you should try changing the device type and model in the device description for the display, and vary the device off and then on again. To determine the correct device type to select, you need to compare the characteristics of your unsupported ASCII display with the characteristics of the supported IBM displays. To help you with this comparison, see the *ASCII Work Station Reference and Example*. When you find a supported display that has at least some matching characteristics to your ASCII display, you can then use the device type the AS/400 system selected to retrieve the workstation customizing source and try this procedure again to get to an AS/400 Sign On display.

Setting Up a 3477 Twinaxial Display

The following procedure helps you set up the 3477 display. It also provides the extra steps to verify that the display is a Model H and can support an attached ASCII printer. If you only want to set up the display, then you can leave out the settings for the printer. These are in step 6 of the procedure. To set up the display (and verify that the 3477 is a Model H), do the following:

1. Turn on the power for the display while pressing and holding the spacebar. The Offline Setup menu appears.
2. Move the cursor to the *Customize workstation* option and press the Enter key. The first Customization display appears.
3. Move the cursor to specify the *Display-printer for the terminal mode* option and press the spacebar.
4. Move the cursor to the appropriate address for the display and press the spacebar to select the address.
5. Move the cursor to the appropriate address for the printer and press the spacebar to select the address.
6. Use the cursor and the spacebar to make appropriate selections for the following fields. (You will need to page down to get to all these fields.)
   - Character set
   - Printer character set (usually one of the multilingual sets)
   - Printer emulation (if laser printer select 5219; otherwise select 4214)
   - Keyboard type (standard)
   - ASCII Printer type (user-defined)
   - ASCII Printer ID (select any ID number except 00)
7. Press F3 (Save and exit) to save these settings and return to the Offline Setup display.
8. Press F3 (Save and exit) to return to communication mode.

If you are working with a 3477 display and the final Customization Setup menu did not show the ASCII printer type or ASCII printer ID fields, then your 3477 display is not a Model H and you cannot customize an attached ASCII printer using the OS/400 workstation customizing functions.

Appendix E. Workstation Customizing Planning Work Sheets

You may copy the following maps and tables and use them for work sheets to help you plan for and work through the customizing process for displays and printers.

Matching Command Parameters

Use the following commands to set up and use the OS/400 workstation customizing functions:

- Retrieve Work Station Customizing Object Source (RTVWSCST) command
- Create Work Station Customizing Object (CRTWSCST) command
- Create Device Description (Display) (CRTDEVDSP) command
- Create Device Description (Printer) (CRTDEVPRT) command

| Devices that do not use the host print transform function
| have matching keywords and parameters for these com-
| mands, as shown in Table E-1. Printers that use the host
| print transform function have matching keywords and param-
| eters for these commands, as shown in Table E-2.

An X in these tables indicates that the parameter should have the same value for the device description commands as it does in the workstation customizing commands. For example, if you are going to customize an ASCII display, the device type (DEVTYPE) parameter for the RTVWSCST command should be the same value as the device type (TYPE) parameter you specified in the CRTDEVDSP command.

Note: You should create the device descriptions for the workstations you want to customize before you retrieve a source file member or create the customizing object.

Table E-1. Matching Command Parameters--Devices That Do Not Use Host Print Transform Function

Parameter (KEYWORD)	CRTDEVDSP	CRTDEVPRT	RTVWSCST	CRTWSCST
Device type (TYPE, DEVTYPE)	X	X	X	
Keyboard language type (KBDTYPE, LNGTYPE)	X	X	X	
Workstation customizing object (WSCST)	X	X		X
Library (LIB)	X	X	X	X
Source file (SRCFILE)			X	X
Source member (SRCMBR)			X	X

| *Table E-2. Matching Command Parameters--Printers That Use Host Print Transform Function*

Parameter (KEYWORD)	CRTDEVPRT	RTVWSCST	CRTWSCST
Manufacturer, type, and model (MFRTYPMDL)	X (except when value is *WSCST)	X	
Workstation customizing object (WSCST)	X		X
Library (of workstation customizing object)	X		X
Source member (SRCMBR)		X	X
Source file (SRCFILE)		X	X
Library (of source file)		X	X

Work Sheets for Planning to Customize a Display Workstation

Following are work sheets you can copy and use in planning to customize a display workstation.

Command Parameters for ASCII and Twinaxial Displays		
Command	Parameter	Value
CRTDEVDSP	TYPE	
	KBDTYPE	
	WSCST	
	LIB	
RTVWSCST	DEVTYPE	
	KBDTYPE	
	SRCMBR	
	SRCFILE	
	LIB	
	TEXT	
CRTWSCST	WSCST	
	LIB	
	SRCMBR	
	SRCFILE	
	LIB	

Twinaxial Displays

Twinaxial Keyboard Translation Table Entries (Mode= Shift State=)									
Scan Code	Key or Function		Hexadecimal Data		Scan Code	Key or Function		Hexadecimal Data	
	Old	New	Old	New		Old	New	Old	New

ASCII Displays

ASCII to Keyboard Function Mapping Table		
Tag	**Key Sequence**	**Hexadecimal Data**
ATN		
BASE		
BOTPAG		
BSP		
CLEAR		
CLOSE		
CSRUP		
CSRDOWN		
CSRLEFT		
CSRRIGHT		
CSRSEL		
DISC		
DLT		
DUP		
ENTER		
ERSINP		
ERSEOF		
ERRRESET		
FCSRLEFT		
FCSRIGHT		
FLDADV		
FLDBSP		
FLDEXIT		
FLDPLUS		
FLDMINUS		
FLDMRK		
HELP		
HEX		
HOME		
INSERT		
LATINON		
NEWLINE		
NTLON		
PRINT		
PAGDOWN		
PAGUP		
PA1		
PA2		
PA3		
READSTS		
RVS		

ASCII to Keyboard Function Mapping Table		
Tag	**Key Sequence**	**Hexadecimal Data**
SHIFTIN		
SHIFTOUT		
SCNREFRESH		
SYSREQ		
TOGIND		
TOPPAG		
TSTREQ		
Note: You may have more than one entry for each tag.		

ASCII to Keyboard Function Mapping Table (Text Functions)		
Tag	**Key Sequence**	**Hexadecimal Data**
PAGEND		
RQDTAB		
FWDTAB		
WORDUS		
STRUS		
HLFIDXUP		
HLFIDXDN		
STOPCODE		
END		
CARRTN		
CENTER		
BOLD		
NEXTSTOP		
RQDSPC		
DSPSYM		
STRLINE		
ENDLINE		

ASCII to Keyboard Function Mapping Table (Function Keys)		
Tag	**Key Sequence**	**Hexadecimal Data**
F1		
F2		
F3		
F4		
F5		
F6		
F7		
F8		
F9		
F10		
F11		

ASCII to Keyboard Function Mapping Table (Function Keys)		
Tag	**Key Sequence**	**Hexadecimal Data**
F12		
F13		
F14		
F15		
F16		
F17		
F18		
F19		
F20		
F21		
F22		
F23		
F24		
Note: You may have more than one entry for each function key.		

Work Sheets for Customizing ASCII Printers That Use the Host Print Transform Function

The following tables are for you to copy and use as work sheets for customizing an ASCII printer that uses host print transform function.

Table E-3. Command Parameters for ASCII Printers That Use the Host Print Transform Function

Command	Parameter	Value
CRTDEVPRT	TRANSFORM	*YES
	MFRTYPMDL	
	WSCST	
	LIB	
RTVWSCST	DEVTYPE	*TRANSFORM
	MFRTYPMDL	
	SRCMBR	
	SRCFILE	
	LIB	
	TEXT	
CRTWSCST	WSCST	
	LIB	
	SRCMBR	
	SRCFILE	
	LIB	

Work Sheets for ASCII Printer Functions

Table E-4. Customizing Printer Controls

Tag	Description	Data
BELL	Bell	DATA=
CARRTN	Carrier return	DATA=
INITPRT	Initialize printer	DATA=
RESETPRT	Reset printer	DATA=
JOGOUTTRAY	Jog output tray	DATA=
DUPXPRT	Set duplex printing	DATA=
NXTDUPXPRT	Select next side printing in duplex	DATA=
SMPXPRT	Set simplex printing	DATA=
TUMDUPXPRT	Set tumble duplex printing	DATA=

Table E-5. Customizing Printer Data Streams

Tag	Description	Data
PRTDTASTRM	Printer data stream	DATASTREAM=

Table E-6. Customizing Print Media Size

Tag	Description	Data
ENVSIZXFM	Set envelope size for trans-form	None.
ENVSIZE	Envelope size entry	ENVWTH= ENVLEN= DATA=
ENVSIZE	Envelope size entry	ENVWTH= ENVLEN= DATA=
EENVSIZXFM	End set envelope size for transform	None.
PAGSIZXFM	Set page size for transform	None.
PAGSIZE	Page size entry	PAGWTH= PAGLEN= DATA=
PAGSIZE	Page size entry	PAGWTH= PAGLEN= DATA=
PAGSIZE	Page size entry	PAGWTH= PAGLEN= DATA=
EPAGSIZXFM	End set page size for trans-form	None.

Table E-7. Customizing Highlighting

Tag	Description	Data
STRBOLD	Start bold printing	DATA=
ENDBOLD	End bold printing	DATA=
STRUS	Start under-score	DATA=
ENDUS	End underscore	DATA=

Table E-8. Customizing Horizontal Spacing and Movement

Tag	Description	Data
BSP	Backspace	DATA=
CPI	Set characters per inch	CPI= DATA=
CPI	Set characters per inch	CPI= DATA=
CPI	Set characters per inch	CPI= DATA=
CPICOR	Set characters per inch in COR mode	CPI = ASCIIFNT= FNTWTH= FNTATR= DATA=
CPICOR	Set characters per inch in COR mode	CPI = ASCIIFNT= FNTWTH= FNTATR= DATA=
CPICOR	Set characters per inch in COR mode	CPI = ASCIIFNT= FNTWTH= FNTATR= DATA=
HORRMOV	Horizontal rela-tive movement	DIRECTION= VAROFFSET= VARLEN= VARTYPE= CNVNUM= CNVDEN= DATA=
HORRMOV	Horizontal rela-tive movement	DIRECTION= VAROFFSET= VARLEN= VARTYPE= CNVNUM= CNVDEN= DATA=
STRPROP	Start propor-tional spacing	DATA=
ENDPROP	End propor-tional spacing	DATA=
SPACE	Space	DATA=

Table E-9. Customizing Vertical Spacing and Movement

Tag	Description	Data
FORMFEED	Form feed	DATA=
HLFLINEFEED	Half line feed	DATA=
LINEFEED	Line feed	DATA=
VERRMOV	Vertical relative movement	DIRECTION= VAROFFSET = VARLEN= VARTYPE= CNVNUM= CNVDEN= DATA=
RVSHLFLINEFEED	Reverse half line feed	DATA=
RVSLINEFEED	Reverse line feed	DATA=
LPI	Set lines per inch	LPI= DATA=
LPI	Set lines per inch	LPI= DATA=
LPI	Set lines per inch	LPI= DATA=
LPI	Set lines per inch	LPI= DATA=
VARLSPC	Variable line spacing	VAROFFSET= VARLEN= VARTYPE= CNVNUM= CNVDEN= DATA=

Table E-10. Customizing Indexing

Tag	Description	Data
STRSUBS	Start subscript	DATA=
ENDSUBS	End subscript	DATA=
STRSUPS	Start super-script	DATA=
ENDSUPS	End superscript	DATA=

Table E-11. Customizing Color

Tag	Description	Data
FOREGRND	Foreground color	COLOR= DATA=
FOREGRND	Foreground color	COLOR= DATA=
FOREGRND	Foreground color	COLOR= DATA=
FOREGRND	Foreground color	COLOR= DATA=
FOREGRND	Foreground color	COLOR= DATA=

Table E-12. Customizing the No Print Border

Tag	Description	Data
NOPRTBDR	Set no print border	OPTION= ORIENT= DATA=
NOPRTBDR	Set no print border	OPTION= ORIENT= DATA=
NOPRTBDR	Set no print border	OPTION= ORIENT= DATA=
NOPRTBDR	Set no print border	OPTION= ORIENT= DATA=

Table E-13. Customizing Page Length

Tag	Description	Data
PAGLENI	Set page length in inches	VAROFFSET= VARLEN= VARTYPE= CNVNUM= CNVDEN= DATA=
PAGLENL	Set page length in lines	VAROFFSET= VARLEN= VARTYPE= DATA=

Table E-14. Customizing Paper Drawer Selection

Tag	Description	Data
DWRSLT	Paper drawer selection	DRAWER= DATA=
DWRSLT	Paper drawer selection	DRAWER= DATA=
DWRSLT	Paper drawer selection	DRAWER= DATA=
DWRSLT	Paper drawer selection	DRAWER= DATA=

Table E-15. Customizing Paper Orientation

Tag	Description	Data
PRTORIENT	Paper orientation	ORIENT= DATA=
PRTORIENT	Paper orientation	ORIENT= DATA=
PRTORIENT	Paper orientation	ORIENT= DATA=
PRTORIENT	Paper orientation	ORIENT= DATA=

Table E-16. Customizing Print Quality

Tag	Description	Data
PRTQLTY	Print quality	QLTYTYPE= DATA=
PRTQLTY	Print quality	QLTYTYPE= DATA=
PRTQLTY	Print quality	QLTYTYPE= DATA=

Table E-17. Customizing Fonts

Tag	Description	Data
FNTGRP	Font group	None.
FNTGRPE	Font group entry	MINFID= MAXFID= FNTSTR= FNTEND= FNTWTH=
FNTGRPE	Font group entry	MINFID= MAXFID= FNTSTR= FNTEND= FNTWTH=
EFNTGRP	End font group	None.
INDFNT	Individual font	None.
INDFNTE	Individual font entry	FID= POINTSIZE= FNTSTR= FNTEND= FNTWTH=
INDFNTE	Individual font entry	FID= POINTSIZE= FNTSTR= FNTEND= FNTWTH=
EINDFNT	End Individual font	None.

Table E-18 (Page 1 of 2). Customizing Code Page Support

Tag	Description	Data
ASCCPINFO	ASCII code page information	None.
CODEPAGE	Set code page	CODEPAGE= DATA=
ASCIICTL	ASCII control code mapping	ASCII= DATA=
ASCIICTL	ASCII control code mapping	ASCII= DATA=
ASCIICTL	ASCII control code mapping	ASCII= DATA=
CODEPAGE	Set code page	CODEPAGE= DATA=

Table E-18 (Page 2 of 2). Customizing Code Page Support

Tag	Description	Data
ASCIICTL	ASCII control code mapping	ASCII= DATA=
ASCIICTL	ASCII control code mapping	ASCII= DATA=
ASCIICTL	ASCII control code mapping	ASCII= DATA=
EASCCPINFO	End ASCII code page information	None.
DFTASCCP	Default ASCII code page	ASCIICP=
EBCASCTBL	EBCDIC-to-ASCII mapping table	None.
EBCASCTBLE	EBCDIC-to-ASCII mapping table entry	EBCDICCP= ASCIICP= DATA=
EBCASCTBLE	EBCDIC-to-ASCII mapping table entry	EBCDICCP= ASCIICP= DATA=
EBCASCTBLE	EBCDIC-to-ASCII mapping table entry	EBCDICCP= ASCIICP= DATA=
EEBCASCTBL	End EBCDIC-to-ASCII mapping table	None.

Work Sheet for Customizing ASCII Printers That Use the Emulator on the Display

The following tables are for you to copy and use as work sheets for customizing an ASCII printer that uses the emulator on the display.

Command Parameters for ASCII Printers		
Command	Parameter	Value
CRTDEVPRT	TYPE	
	LNGTYPE	
	WSCST	
	LIB	
RTVWSCST	DEVTYPE	
	KBDTYPE	
	SRCMBR	
	SRCFILE	
	LIB	
	TEXT	
CRTWSCST	WSCST	
	LIB	
	SRCMBR	
	SRCFILE	
	LIB	

Work Sheets ASCII Printers Attached to 3477 Model H, 3486, 3487, and 3488 Displays

Table E-19 (Page 1 of 3). Printer Function Tags for Printers Attached to 3477 Model H Displays

Tag	Description	Data
FWDRMOV	Forward relative movement	VAROFFSET = VARLEN= VARTYPE= VARMAX= ADJUST= CNVNUM= CNVDEN= DATA=
BCKRMOV	Backward relative movement	VAROFFSET = VARLEN= VARTYPE= VARMAX= ADJUST= CNVNUM= CNVDEN= DATA=
PAGLENI	Set page length in inches	VAROFFSET= VARLEN= VARTYPE= VARMAX= ADJUST= CNVNUM= CNVDEN= DATA=
VARLSPC	Variable line spacing	VAROFFSET= VARLEN= VARTYPE= VARMAX= ADJUST= CNVNUM= CNVDEN= DATA=
BSP	Backspace	DATA=
BELL	Bell	DATA=
STRBOLD	Start bold printing	DATA=
ENDBOLD	End bold printing	DATA=
CARRTN	Carriage return	DATA=
CPI	Set characters per inch	CPI= DATA=
CPI	Set characters per inch	CPI= DATA=
CPI	Set characters per inch	CPI= DATA=
CODEPAGE	Set code page	CODEPAGE= DATA=
DBLCHRH	Set double character height	DATA=
STRWIDE	Start double-wide continuous	DATA=

Table E-19 (Page 2 of 3). Printer Function Tags for Printers Attached to 3477 Model H Displays

Tag	Description	Data
ENDWIDE	End double-wide continuous	DATA=
DWRSLT	Drawer selection	DRAWER= DATA=
FNTGPDT	Global fonts for printer definition table	
EFNTGPDT	End global fonts for PDT	
FNTGRNG	Global font range	MINFID= MAXFID= DATA=
FOREGRND	Foreground color	COLOR= DATA=
FORMFEED	Form feed	DATA=
INITPRT	Initialize printer	DATA=
LINEFEED	Line feed	DATA=
LPI	Set lines per inch	LPI= DATA=
LPI	Set lines per inch	LPI= DATA=
PAGLENL	Set page length in lines	VAROFFSET= VARLEN= VARTYPE= VARMAX= ADJUST= DATA=
PRTFEED	Paper feed	FEEDTYPE= DATA=
PRTORIENT	Paper orientation	ORIENT= DATA=
PRTQLTY	Print quality	QLTYTYPE= DATA=
STRPROP	Start proportional space mode	DATA=
ENDPROP	End proportional spacing	DATA=
SETQLTY	Quality font download	QLTYTYPE= DATA=
SPACE	Space	DATA=
STDCHRH	Set standard character height	DATA=
STRSUBS	Start subscript	DATA=
ENDSUBS	End subscript	DATA=
STRSUPS	Start superscript	DATA=
ENDSUPS	End superscript	DATA=
TBLNAME	Table name	DATA=

Table E-19 (Page 3 of 3). Printer Function Tags for Printers Attached to 3477 Model H Displays		
Tag	**Description**	**Data**
TRNEBCDIC	Translation table	
TRNEBCDICE	Translation entry	EBCDIC code= DATA=
ETRNEBCDIC	End translation printer definition table	
STRUS	Start under-score	DATA=
ENDUS	End underscore	DATA=
VERUNT	Set vertical units	DATA=
Note: Printer definition table (PDT) for a 3477 Model H, 3486, 3487, or 3488 Display DEVCLASS = TWINAXPRT		

Additional Work Sheets ASCII Printers Attached to 3486, 3487, and 3488 Displays

Table E-20. Additional Printer Function Tags for Printers Attached to 3486, 3487, and 3488 Displays		
Tag	**Description**	**Data**
ADJHRZORG	Adjust horizontal origin	ADJUST=
ADJVERORG	Adjust vertical origin	ADJUST=
CPI	Set characters per inch	CPI= DATA=
CPI	Set characters per inch	CPI= DATA=
CPI	Set characters per inch	CPI= DATA=
CPICOR	Set characters per inch in COR mode	CPI = ASCIIFNT= FNTWTH= FNTATR= DATA=
DUPXPRT	Set duplex printing	DATA=
HLFLINEFEED	Half line feed	DATA=
JOGOUTTRAY	Jog output tray	
NXTDUPXPRT	Select next side printing in duplex	DATA=
PRTORIENT	Paper orientation	ORIENT= DATA=
RVSHLFLINEFEED	Reverse half line feed	DATA=
RVSLINEFEED	Reverse line feed	DATA=
SMPXPRT	Set simplex printing	DATA=
TUMDUPXPRT	Set tumble duplex printing	DATA=
VERUNTHLF	Set vertical units in half	DATA=
Note: Printer definition table (PDT) for 3486, 3487, or 3488 Displays with DEVCLASS = TWINAXPRT		

Work Sheets for ASCII Printers That Use the Emulator on the Workstation Controller

The following tables are for you to copy and use as work sheets to customize ASCII printers that use the emulator on the workstation controller.

Command Parameters for ASCII Printers		
Command	Parameter	Value
CRTDEVPRT	TYPE	
	LNGTYPE	
	WSCST	
	LIB	
RTVWSCST	DEVTYPE	
	KBDTYPE	
	SRCMBR	
	SRCFILE	
	LIB	
	TEXT	
CRTWSCST	WSCST	
	LIB	
	SRCMBR	
	SRCFILE	
	LIB	

Work Sheet for the Default EBCDIC-to-ASCII Mapping Table																
First Hex Digit	Second Hex Digit															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

Work Sheet for the Font Width Table																
First Hex Digit	Second Hex Digit															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																
Note: There must be one of these tables for each default EBCDIC-to-ASCII mapping table you provide in your source.																

Work Sheet for Printer Function Tags with Only a Data Parameter		
Tag	**Description**	**Hexadecimal Data**
BELL	Bell	DATA=
BSP	Backspace	DATA=
CARRTN	Carrier return	DATA=
ENDBOLD	End bold printing	DATA=
ENDSUBS	End subscript	DATA=
ENDSUPS	End superscript	DATA=
ENDUS	End underscore	DATA=
FNTGPFT	Set font global for PFT	DATA=
FORMFEED	Form feed	DATA=
INITPRT	Initialize printer	DATA=
INITVON	Initialization at vary on	DATA=
LINEFEED	Line feed	DATA=
PRTCTL	Printer control flags Valid values are: '00'X No flags set '01'X Printer response allowed '02'X Page length change in middle of page not allowed '30'X Inhibit computer output reduction (COR) mode '80'X Font widths in 1/1440-inch units	DATA=
RVSIDX	Reverse index	DATA=
SPACE	Space	DATA=
STRBOLD	Start bold printing	DATA=
STRPROP	Start proportional space mode	DATA=
STRSUBS	Start subscript	DATA=
STRSUPS	Start superscript	DATA=
STRUS	Start underscore	DATA=
Note: Printer function table (PFT) DEVCLASS = ASCIIPRT		

Work Sheet for Printer Function Tags with a Variable		
Tag	**Description**	**Data**
CODPAGVAR	Set code page	VAROFFSET= VARLEN= VARTYPE= DATA=
FNTTYPE	Font type	VAROFFSET= VARLEN= VARTYPE= DATA=
PRISPCM	Set primary spacing mode	VAROFFSET= VARLEN= VARTYPE= DATA=

Work Sheet for Printer Function Tags with a Variable		
Tag	**Description**	**Data**
PRISTYLE	Set primary style	VAROFFSET= VARLEN= VARTYPE= DATA=
Note: Printer function table (PFT) DEVCLASS = ASCIIPRT		

Work Sheet for Printer Function Tags with a Variable and Relative Movement		
Tag	**Description**	**Data**
BCKRMOV	Backward relative movement	VAROFFSET= VARLEN= VARTYPE= VARMAX= ADJUST= CNVNUM= CNVDEN= DATA=
FWDRMOV	Forward relative movement	VAROFFSET= VARLEN= VARTYPE= VARMAX= ADJUST= CNVNUM= CNVDEN= DATA=
PAGLENI	Set page length in inches	VAROFFSET= VARLEN= VARTYPE= VARMAX= ADJUST= CNVNUM= CNVDEN= DATA=
PRICHRH	Set primary character height	VAROFFSET= VARLEN= VARTYPE= VARMAX= ADJUST= CNVNUM= CNVDEN= DATA=
VARCPI	Variable characters per inch	VAROFFSET= VARLEN= VARTYPE= VARMAX= ADJUST= CNVNUM= CNVDEN= DATA=
VARLSPC	Variable line spacing	VAROFFSET= VARLEN= VARTYPE= VARMAX= ADJUST= CNVNUM= CNVDEN= DATA=

Work Sheet for Other Printer Function Tags		
Tag	**Description**	**Data**
ASCIICTL	ASCII control code mapping	ASCII= DATA=
COLLATE	Collate width	DATA=
CPI	Set characters per inch	CPI= DATA=
CPI	Set characters per inch	CPI= DATA=
CPI	Set characters per inch	CPI= DATA=
CPI	Set characters per inch	CPI= DATA=
DFTFNTID	Default font ID	CPI= DEFAULT=
DWRSLT	Drawer selection	DRAWER= DATA=
EFNTMAP	End font ID mapping	
FNTMAP	Font ID mapping	CPI= DEFAULT=
FNTMAP	Font ID mapping	CPI= DEFAULT=
FNTMAP	Font ID mapping	CPI= DEFAULT=
FNTMAPE	Font mapping table entry	IBMFNT= ASCIIFNT=
FNTMAPE	Font mapping table entry	IBMFNT= ASCIIFNT=
FNTMAPE	Font mapping table entry	IBMFNT= ASCIIFNT=
FNTMAPE	Font mapping table entry	IBMFNT= ASCIIFNT=
FNTMAPE	Font mapping table entry	IBMFNT= ASCIIFNT=
FONTQLTY	Font quality	FONTCPI= QLTYTYPE= DATA=
FONTQLTY	Font quality	FONTCPI= QLTYTYPE= DATA=
FONTQLTY	Font quality	FONTCPI= QLTYTYPE= DATA=
MARGIN	Set margin	OPTION= ORIENT= DATA=
MARGIN	Set margin	OPTION= ORIENT= DATA=
MARGIN	Set margin	OPTION= ORIENT= DATA=

Work Sheet for Other Printer Function Tags		
Tag	**Description**	**Data**
MARGIN	Set margin	OPTION= ORIENT= DATA=
PAGLENL	Set page length in lines	VAROFFSET= VARLEN= VARTYPE= VARMAX= ADJUST= DATA=
PAGSIZPFT	Set page size for printer function table	PAGWTH= PAGLEN= PAPER=
PRTFEED	Paper feed	FEEDTYPE= DATA=
PRTORIENT	Paper orientation	ORIENT= DATA=
PRTQLTY	Print quality	QLTYTYPE= DATA=

Bibliography

This list of tasks related to workstation customizing followed by the AS/400 publications that describe them is provided to help you find more information about setting up your displays and ASCII printers.

For more information about general AS/400 system management tasks, see:

Advanced Backup and Recovery Guide, SC41-8079

Central Site Distribution Guide, SC41-9993

Basic Security Guide, SC41-0047

System Operator's Guide, SC41-8082

For more information about AS/400 system communications and configuration in general, see:

Communications: Advanced Peer-to-Peer Networking Guide, SC41-8188

Communications: Integrated Services Digital Network Guide, SC41-0003

Communications: Local Area Network Guide, SC41-0004

Communications: Management Guide, SC41-0024

Communications: Operating System/400 Communications Configuration Reference*, SC41-0001

Communications: X.25 Network Guide, SC41-0005

Device Configuration Guide, SC41-8106

For more information about AS/400 national language support, code pages, coded character set identifiers, and character identifiers in general, see:

Character Data Representation Architecture Reference, SC09-1390

Guide to Programming Application and Help Displays, SC41-0011

National Language Support (NLS) Enabling of IBM AS/400 Applications, GG24-3807

National Language Support Planning Guide, GC41-9877

Programming: Work Management Guide, SC41-8078

For more information about setting up and using twinaxial or ASCII workstations with the AS/400 system, see:

ASCII Work Station Reference and Example, SA41-9922

AS/400 9402 Attaching Workstation and Communications Cables*, SA41-0005

Communications: Remote Work Station Guide, SC41-0002

IBM InfoWindow 3477 User's Guide*, GA18-2923

Twinaxial Workstation Controller Port Tester User's Guide, SA41-9944

9404 Attaching Workstation and Communications Cables, SA41-0004

9406 Attaching Workstation and Communications Cables, SA41-9957

For more information about setting up and using ASCII printers with an AS/400 system, see:

Additional Information for IBM InfoWindow 3477, GA18-7084

Application Development Tools: Advanced Printer Function Guide, SC09-1361

Guide to Programming for Printing, SC41-8194

PrintManager Application Programming Interface Reference, S544-3699

Systems Application Architecture Common Programming Interface PrintManager Reference*, S544-3698

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Customer Satisfaction Feedback

Application System/400
Operating System/400
Workstation Customization Function
Programmer's Guide
Version 2

Publication No. SC41-0056-01

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatis-fied	Very Dissatis-fied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

May we contact you to discuss your responses? __ Yes __ No

Phone: (____) _____ Fax: (____) _____

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937-3430**
  Other countries: **(+1)+507+253-5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

Name

Address

Company or Organization

Phone No.

Customer Satisfaction Feedback
SC41-0056-01

IBM®

Cut
Alon

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 245
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-7899

Fold and Tape | **Please do not staple** | Fold and Tape

SC41-0056-01

Cut
Alo

Customer Satisfaction Feedback

Application System/400
Operating System/400
Workstation Customization Function
Programmer's Guide
Version 2

Publication No. SC41-0056-01

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatis-fied	Very Dissatis-fied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

May we contact you to discuss your responses? __ Yes __ No

Phone: (____) _____ Fax: (____) _____

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937-3430**
  Other countries: **(+1)+507+253-5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

Name

Address

Company or Organization

Phone No.

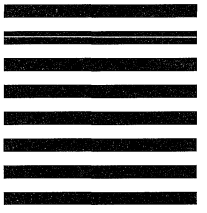
Customer Satisfaction Feedback
SC41-0056-01

IBM®

Cut
Alor

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 245
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-7899

Fold and Tape | **Please do not staple** | Fold and Tape

Cut
Alor

Program Number: 5738-SS1

Printed in Denmark by Bonde's

SC41-0056-01