4777 Magnetic Stripe Unit and
4778 PIN-Pad Magnetic Stripe Reader

IBM

DOS Programming Guide

SA34-2206-00

4777 Magnetic Stripe Unit and
4778 PIN-Pad Magnetic Stripe Reader

IBM

DOS Programming Guide

SA34-2206-00

Note!

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

First Edition (September 1994)

Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult your IBM representative to be sure you have the latest edition and any Technical Newsletters.

IBM does not stock publications at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader’s comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Department 78C, 1001 W. T. Harris Boulevard West, Charlotte, NC 28262-8563, U.S.A. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

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 IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

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 Licensing, IBM Corporation, 208 Harbor Drive, Stamford, Connecticut 06904-2501, U.S.A.

License

You may use the files which make up Feature Code 3921 (Programs) with the IBM 4777 or 4778 only in accordance with the IBM System Programs License Agreement that accompanies the Programs.

Trademarks

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:

IBM
Operating System/2
OS/2
Personal System/2
PS/2
PS/ValuePoint
TopView

About This Book

This book tells you how to control the IBM* 4777 Magnetic Stripe Unit and the IBM 4778 PIN-Pad Magnetic Stripe Reader in a Disk Operating System (DOS) environment. It explains how to customize, load, and use the 4777 and 4778 device drivers. This guide also helps you write application programs that access the device drivers.

Who Should Read This Book

The information in this book is intended for people who write, or maintain, system and application programs that work with the 4777 Magnetic Stripe Unit and the 4778 PIN-Pad Magnetic Stripe Reader.

How This Book Is Organized

This book contains the following sections:

Chapter 1, “Introducing the 4777 Magnetic Stripe Unit and the 4778 PIN-Pad Magnetic Stripe Reader,” describes the 4777 Magnetic Stripe Unit device and the 4778 PIN-Pad Magnetic Stripe Reader.

Chapter 2, “Using the 4777 and 4778 Device Drivers,” describes the operations and function calls available under the device drivers for the 4777 Magnetic Stripe Unit and 4778 PIN-Pad Magnetic Stripe Reader in a DOS environment.

Chapter 3, “Loading the 4777 and the 4778 Device Drivers,” tells you how to load and initialize the 4777 and the 4778 device drivers for DOS on your system.

Chapter 4, “Programming for the 4777 MSRE and 4778 MSR,” tells you how to code an application program that can read and write using a MSR or MSRE.

Chapter 5, “Programming for the 4778 PIN-Pad,” tells you how to code an application program that uses the 4778 PIN-Pad.

Chapter 6, “Problem Determination Procedures,” tells you how to do problem determination in an operational environment.

Related Publications

You might need additional information from one or more of the following publications:

The *DOS Technical Reference*, for your DOS operating system

4777 Magnetic Stripe Unit Installation Guide

4778 PIN-Pad Magnetic Stripe Reader Installation Guide

* Trademark of IBM

4777 Magnetic Stripe Unit and 4778 PIN-Pad Magnetic Stripe Reader OS/2 Programming Guide, SA34-2205

4700 Finance Communication System: System Summary, GC31-2016

4700 Financial I/O Planning Guide, GC31-3762

Chapter 1. Introducing the 4777 Magnetic Stripe Unit and the 4778 PIN-Pad Magnetic Stripe Reader

This chapter describes the IBM 4777 Magnetic Stripe Unit and the IBM 4778 PIN-Pad Magnetic Stripe Reader.

The 4777 and 4778 are compatible with the IBM 4700 Finance Communication System family of programs and products. For more information about the 4700 Finance Communication System, see the *4700 Finance Communication System: System Summary*.

The 4777 Magnetic Stripe Unit

The 4777 Magnetic Stripe Unit, available in four models, is a countertop magnetic-stripe unit (see Figure 1-1). The 4777 reads and encodes magnetic-stripe documents that are manually passed through the device.

The 4777 is based on the IBM 4717 Magnetic Stripe Unit. The 4777 attaches to a 4704, or to the serial port or mouse port (auxiliary port) of an IBM Personal System/2* (PS/2*) workstation or a PS/ValuePoint* workstation. You can also attach both the 4777 and the 4778 PIN-Pad Magnetic Stripe Reader to your workstation with a special connector. For more information about installing both devices on the same workstation, see the *4777 Magnetic Stripe Unit Installation Guide*.

Notes:

1. The 4777 cannot be installed on the same system with a 4717 device. Although they are similar in operation, the device drivers for the 4717 devices are not compatible with the 4777 device.
2. Two 4777 devices cannot be attached to a common system.

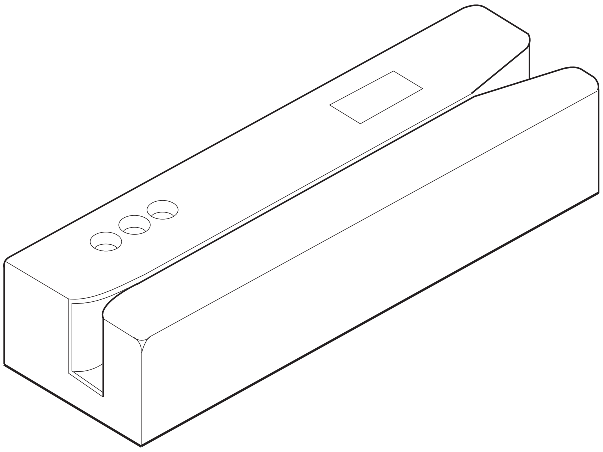

Figure 1-1. 4777 Magnetic Stripe Unit

* Trademark of IBM

Model 001 The 4777 Model 001 is a magnetic-stripe reader that reads, on a single pass, tracks 1 and 2 on credit cards and identification cards. This model reads track 1 and track 2 at 75 or 210 bits per inch (bpi), in accordance with American National Standards Institute (ANSI) standards X4.16-1983 and the International Standards Organization (ISO) standards 7810 and 7811/2-5.

Model 002 The 4777 Model 002 reads tracks 2 and 3 on credit cards and identification cards. It also reads and encodes passbooks. This model reads tracks 2 and 3 of credit cards and identification cards at 75 or 210 bpi. It encodes passbooks using the 4700 specifications or according to the ISO standard 8484. It reads passbooks that are encoded by the IBM 3604 Keyboard Display, by the IBM 4704 Keyboard Display, or in accordance with the ISO standard 8484.

Model 003 The 4777 Model 003 reads and encodes tracks 1 and 2 on credit cards and identification cards in accordance with the ISO and the ANSI specifications. This model reads tracks 1 and 2 at 75 or 210 bpi, and encodes track 1 at 210 bpi and track 2 at 75 bpi. The 4777 Model 003 is useful in an administrative work area of a financial institution that creates the identification cards for personal banking machines when a customer opens an account.

Model 004 The 4777 Model 004 reads tracks 2 and 3 on credit cards and identification cards and reads passbooks. This model reads tracks 2 and 3 at 75 or 210 bpi. It reads passbooks that are encoded by the IBM 3604 Keyboard Display, by the IBM 4704 Keyboard Display, or in accordance with the ISO Standard 8484.

Note: This model is available only in Europe, the Middle East, and Asia.

The 4778 PIN-Pad Magnetic Stripe Reader

The 4778 PIN-Pad Magnetic Stripe Reader, available in three models, is a countertop keypad with or without a magnetic-stripe reader (MSR) (see Figure 1-2 on page 1-3). The three models of the 4778 are described below:

Model 001 Reads tracks 1 and 2 on credit and ID cards on a single pass. Supports application programs requiring a 12-key PIN pad.

Model 002 Supports application programs requiring a 12-key PIN pad.

Model 003 Reads tracks 1, 2, and 3 on credit and ID cards. Supports application programs requiring a 12-key PIN pad.

The keypad is used to enter a personal identification number (PIN) for validating financial transactions. The 4778 keypad accepts and encrypts PINs with enhanced security. The keypad has 10 numeric keys, 2 special keys, and a 16-character display. The magnetic-stripe reader lets your applications read data from magnetic stripes on credit cards or ID cards.

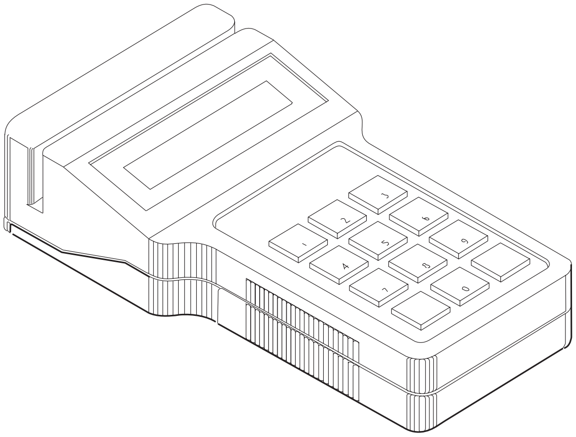

Figure 1-2. 4778 PIN-Pad Magnetic Stripe Reader

You can attach both the 4778 PIN-Pad Magnetic Stripe Reader and the 4777 Magnetic Stripe Unit to your workstation with a special connector. For more information about installing both devices on the same workstation, see the *4778 PIN-Pad Magnetic Stripe Reader Installation (Quick Reference Card)*. The magnetic-stripe reader in the 4778 Model 1 provides the same function as the IBM 4777 Model 001 Magnetic Stripe Unit.

The 4778 PIN keypad is based on the IBM 4718 PIN Keypad. The 4718 attached to the auxillary port of a PS/2. The 4778 can attach to either the serial port or auxillary port of an IBM Personal System/2* (PS/2*) workstation or a PS/ValuePoint* workstation.

Notes:

1. The 4778 PIN-Pad MSR cannot be installed on the same system with a 4718 device. Although they are similar in operation, the device drivers for the devices are not compatible.
2. Two 4778 devices cannot be attached to a common system.

Operations available with the 4778 PIN-Pad Magnetic Stripe Reader:

- Reads data encoded in formats of either the American National Standards Institute (ANSI) X4.16-1983 or the International Organization for Standardization (ISO) 7810 and 7811/2-5.
- Operates in clear or encrypted mode.
- Uses the Data Encryption Standard (DES) algorithm.
- Uses a master key that can be a single-length key or a double-length key.
- Encrypts PINs using the American National Standards Institute (ANSI) X9.8, IBM 3624 Keyboard Display, or IBM 4704 Keyboard Display formats within the keypad.
- Lets you download master keys or enter them using the keypad.
- Generates or verifies a message authentication code (MAC).
- Verifies a PIN or creates PIN offset data using the IBM 3624 Consumer Transaction Facility algorithm.

Chapter 2. Using the 4777 and 4778 Device Drivers

This chapter describes the operations and function calls available under the device drivers for the 4777 Magnetic Stripe Unit and 4778 PIN-Pad Magnetic Stripe Reader in a DOS environment. If you are using the 4777 Magnetic Stripe Unit or 4778 PIN-Pad Magnetic Stripe Reader on a system with the Operating System/2* (OS/2*) operating system, see the information in the *4777 Magnetic Stripe Unit and 4778 PIN-Pad Magnetic Stripe Reader OS/2 Programming Guide*.

4777 Device Driver Operations

The application program interface to the 4777 Magnetic Stripe Unit is supplied by the following device drivers:.

For auxillary port, MSRE2DD.SYS
For RS-232, IBM4777.SYS

These device drivers support the attachment of all four models of the magnetic-stripe unit.

Warning: The 4777 Magnetic Stripe Unit does not differentiate one medium from another. If you try to encode a previously encoded medium, the previously encoded data can be destroyed. Because of this and other operations that require encoding expertise, only authorized banking personnel should do encode operations.

If you load the device driver, your application program can use it to pass data to and from the 4777 Magnetic Stripe Unit by device-driver function calls. The device driver enables you to:

Select which tracks the 4777 reads or writes
Specify track-encoding parameters for reading or writing data
Read or encode magnetic-stripe data
Check the validity of data that is read or encoded

4778 Device Driver Operations

The application program interface to the 4778 PIN Pad is supplied by the following device drivers:.

for auxillary port, PIN2DD.SYS
for RS-232, IBM4778P.SYS

The application program interface to the 4778 magnetic stripe reader is supplied by the following device drivers:.

for auxillary port, MSR2DD.SYS
for RS-232, IBM4778M.SYS

These device drivers support the attachment of all models of the 4778.

* Trademark of IBM

Your application program passes data to and from the device drivers by standard function calls such as Open, Close, Read, and IOCTL. The device drivers enable the keypad and the magnetic stripe reader to do the following:

Read magnetic stripe data
Read nonencrypted data
Read encrypted data
Read and set device information
Send control data to the 4778
Receive data from the 4778

Device Driver Modes

The device drivers operate in either synchronous or asynchronous mode.

Synchronous Mode
Synchronous mode is the default operating mode for the 4777 and the 4778. When you start a device driver operation, all other processing on your workstation is suspended. It stays suspended until the completion of the I/O operation. If the device driver detects a Cancel key, it automatically disables the operation. For more information about the Cancel key, see “Loading the Device Driver” on page 3-2.

Asynchronous Mode
You can switch the operating mode of the 4777 or 4778 and associated device driver to asynchronous mode. In this mode, the driver does not wait for the completion of the I/O operation; control is returned to the application through DOS after the function call is made. In asynchronous mode, the application program must periodically request the status of the I/O operation to determine when it is completed. Also, the application program must detect Cancel key input and disable the device.

If the asynchronous function involves passing data to and from the driver, the application must preserve the integrity of the buffer until the driver indicates that the requested operation is completed.

Device Driver Function Calls

To request services from the 4777 or 4778 device drivers, place a function call into register AX and issue an INT 21H function call. You can issue the following DOS function call codes to the driver:

Code	Function
3DH	Open
3EH	Close
3FH	Read
40H	Write (4777 only)
44H	I/O Control (IOCTL)

Open

Before you make any I/O requests, you must open the driver. After you open the driver you should issue a set binary mode function call, see “01H: Set Binary Mode” on page 4-2 and “01H: Set Binary Mode” on page 5-2. To open the driver, issue the DOS function call INT 21H with the register values shown below.

Registers on entry

(AX)=3D00H	Open for reading
(AX)=3D01H	Open for writing (4777 only)
(AX)=3D02H	Open for reading and writing (4777 only)
(DS:DX)	For 4777, the address (Segment:Offset) of the ASCII string MAGDEV, ended by a byte of binary zeros. For 4778, the address (Segment:Offset) of the ASCII string PINMSR$ or PINDEV, ended by a byte of binary zeros.

Registers on return

(AX) The device-driver file handle if carry flag is not set; DOS Status if the carry flag is set (see “DOS Status” on page 4-7)

This function call gives the *handle* for subsequent requests. The handle is a 16-bit binary value that is returned after the completion of an Open call. It must be saved by the application and returned in register BX on all subsequent calls. The ASCII device name to open the 4777 magnetic stripe reader encoder (MSRE) is MAGDEV. The ASCII device name to open the 4778 keypad driver is PINDEV. The ASCII device name to open the 4778 magnetic stripe reader (MSR) driver is PINMSR$.

Close

To close the 4777 MSRE, 4778 PIN Pad, or 4778 MSR, issue the DOS functioncall INT 21H and set the following register values:

Registers on entry

(AX)=3E00H	Close
(BX)	File handle

The file handle must be the same as the one received in the AX register during the Open function call.

Registers on return

(AX)	The device-driver file handle if carry flag is not set; DOS Status if the carry flag is set (see “DOS Status” on page 4-7)

Read

To read data from the 4777 or 4778, issue the DOS function call INT 21H with the register values shown below. This function enables the unit for data input. For more information about magnetic-stripe-unit data, see “Using the API to Read and Write Data with an MSRE” on page 4-9.

Registers on entry

(AX)=3F00H	Read
(BX)	File Handle

(CX) Number of bytes to be read (must be large enough to accept all data entered)

(DS:DX) Address (Segment:Offset) of the input buffer

Registers on return

(AX) Number of bytes read if carry flag is not set; DOS Status if carry flag is set (see “DOS Status” on page 4-7)

Registers on INT 24H Entry

(DI) Error code value

- **01H** Function unavailable (4777 only)
- **02H** Device not ready (4777 or 4778 not attached)
- **03H** Unknown command (unsupported function call)
- **0BH** Read fault (operator canceled Read or buffer too small)
- **0CH** General failure (device failure)

Write (4777 only)

To write data to the magnetic-stripe unit, issue DOS function call INT 21H and set the following register values:

Registers on entry

(AX)=4000H Write

(BX) File handle

(CX) Number of bytes to be written

(DS:DX) Address (Segment:Offset) of the data buffer that contains the data to be written

Registers on return

(AX) Number of bytes written if carry flag is not set; DOS Status if carry flag is set (see “DOS Status” on page 4-7)

Registers on INT 24H Entry

(DI) Error code value

- **01H** Function unavailable
- **02H** Device not ready (magnetic-stripe unit not attached)
- **03H** Unknown command (unsupported function call)
- **0AH** Write fault (operator canceled Write or buffer too small)
- **0CH** General failure (device failure)

Chapter 3. Loading the 4777 and the 4778 Device Drivers

This chapter tells you how to load and initialize the 4777 and the 4778 device drivers for DOS on your system.

The application program interface (API) to the 4777 Magnetic Stripe Unit is supplied by the IBM4777.SYS and MSRE2DD.SYS device drivers. These device drivers support the attachment of all models of the magnetic-stripe unit. IBM4777.SYS supports a 4777 that is attached to a serial port. MSRE2DD.SYS supports a 4777 that is attached to a mouse port.

After you load the 4777 device driver, your application programs can use it to pass data to and from the 4777 Magnetic Stripe Unit by device-driver function calls. The 4777 device driver can operate in either synchronous or asynchronous mode. See Chapter 2, “Using the 4777 and 4778 Device Drivers,” for more information about using the API to control the 4777.

The API to the 4778 is supplied by the IBM4778P.SYS, IBM4778M.SYS, MSR2DD.SYS, and PIN2DD.SYS device drivers. The device drivers give you a DOS-level interface to the 4778 functions. IBM4778P.SYS and IBM4778M.SYS support a 4778 that is attached to a serial port. PIN2DD.SYS and MSR2DD.SYS support a 4778 that is attached to a mouse port. The DOS function calls let your application programs read clear or encrypted data, use the PIN security functions, and perform magnetic-stripe operations.

Prerequisites

The 4777, 4778, and software for the device drivers can be used on the following systems:

Personal System/2

PS/ValuePoint

DOS Version 3.3 (or higher)

Loading the Device Driver

To load and initialize the 4777 or 4778 device drivers, include the following device statement in your CONFIG.SYS file:

```
DEVICE=[d:][path]Name.SYS[Options]
```

Note: The brackets [ ] indicate optional parameters.

The device statement parameters are:

d:path
The identifier for the disk or diskette drive (d:) and the directory-search sequence (path) to locate the IBM4777.SYS file.

Name.SYS
The name of the device driver.

	Auxillary Port	**RS-232**
4777	MSRE2DD.SYS	IBM4777.SYS
4778 PINPad	PIN2DD.SYS	IBM4778P.SYS
4778 MSR	MSR2DD.SYS	IBM4778M.SYS

Notes:

1. For 4777, do not confuse the device driver name with MAGDEV, which is used to open the device. (See “Open” on page 2-3.)
2. For 4778, do not confuse the device driver name with PINDEV, which is used to open the 4778 PIN pad. (See “Open” on page 2-3.)
3. For 4778, do not confuse the device driver name with PINMSR$, which is used to open the 4778 MSR. (See “Open” on page 2-3.)

Options
The optional parameters that suppress automatic error messages, prevent the setting and return of error codes through DOS, and let you reassign the designated Cancel key. At least one blank must precede the options list. There cannot be any blanks between the options.

You can specify the optional parameters in any combination and sequence:

```
/X/Y/Cx/M/K:val
```

or

```
/X/Y/Cx/M/K: ;val
```

/X This option suppresses automatic error messages. When certain problems are detected by the power-on test or during subsequent operations, the device driver automatically displays these error messages on your screen. If you use this option, your application program must display the messages.

/Y This option prevents the setting of the error bit and error codes returned to DOS. When an error is detected in a function call or the operation is canceled, an error bit and an error code are returned to DOS and processed by the INT 24H critical-error handler. If you do not have an error handler, the DOS `Abort, Retry, Ignore` message appears on the screen.

Note: The /Y option suppresses the return of all error statuses from the device driver to DOS, and the carry flag is not set. Your application program must issue a Read Status request to check the results of all operations.

/Cx (serial attach only)
This option designates the communication port to which the 4777 Magnetic Stripe Unit device driver is attached. The default communication port is COM1.

/M (4778 only)
This option lets you attach a 4778 Magnetic Stripe Reader and use it as if it were a 4777 Model 1 Magnetic Stripe Unit. This option allows you to replace a 4717 or 4777 Model 1 with a 4778 and not require a re-write of the application program that controls the magnetic stripe reader.

/K:[0;]*val*
This option designates one of the computer keyboard keys as Cancel, where *val* is a 1-to-3-digit decimal number for the standard or extended American National Standard Code for Information Interchange (ASCII) key-code you want to use. If you omit the /K option, the default Cancel key is the Esc key on your keyboard (*val*=27).

If the key you use as Cancel returns a standard ASCII code when you press it, use the form `/K:val`.

If the key you use as Cancel returns an extended ASCII code, use the form `/K: ;val`.

Note: This option applies only to synchronous operation. When the driver is in synchronous mode, all keystrokes other than the designated Cancel key are ignored.

Chapter 4. Programming for the 4777 MSRE and 4778 MSR

This chapter tells you how to code an application program that can read and write using a magnetic stripe reader (MSR) or magnetic stripe reader encoder (MSRE).

Note: The 4778 MSR is controlled and has the same function as a 4777 Model 001.

IOCTL Function Calls

You issue the DOS IOCTL function calls to pass I/O control strings to the device driver to:

Set binary mode
Read parameter, status, configuration, and error data from the device driver
Write control messages to the device driver
Request the status of the device

The following registers and their I/O control functions are described below:

Registers on entry

(AH)=**44H** The function number.

(AL) The function value:

- **00H** Retrieve device information (information returned in register DX).
- **01H** Set binary mode.
- **02H** Read (parameter, status, configuration, and error data).
- **03H** Write is used for track selection, encoding parameters, mode select, disarming, and TopView data reads and writes. (See “TopView Support” on page 4-16 for more information about TopView functions.)

(BX) File handle.

(CX) Number of bytes to be read (the input data buffer must be large enough to accept all data entered).

(DS:DX) Data buffer address for IOCTL Read (AX=4402H).
Data buffer address for IOCTL Write (AX=4403H).

(DL) Bit 5 is equal to 1 in DL for IOCTL Set Binary (AX=4401H) call.

Registers on return

(AX) The number of bytes transferred for IOCTL Read or Write if the carry flag is not set; DOS Status if the carry flag is set (see “DOS Status” on page 4-7)

Note: IOCTL never starts INT 24H.

00H: Get Device Information

You use this IOCTL function to get information about the magnetic-stripe unit. The device information is returned in register DX. For information about bit definitions of the returned information, see the *DOS Technical Reference*. You must use this function before you set any device information. For information about the status of the device driver, see “Status Reporting” on page 4-7.

01H: Set Binary Mode

You use this IOCTL function to set device information (specifically, the device channel operating mode). You should set binary mode after issuing a device open, see “Open” on page 2-3. The device data is set in register DL.

When the channel is in ASCII mode, DOS checks for the → (1AH) (end-of-file) character. When the channel is in binary mode, no special checks are performed by DOS. This means that all bit patterns are sent across the channel. Set binary mode immediately after you open the device driver and remain in binary mode for as long as the device driver is open. Before your application program sets binary mode, it should read the device information. This saves the current device settings.

Use the following assembly language procedure to set binary mode:

```
MOV AH, 44H         ;DOS IOCTL function call
MOV AL,            ;IOCTL function  , get device information
MOV BX,dev_handle   ;Magnetic device handle
INT 21H            ;Issue DOS IOCTL function call
OR DL,  1     B     ;Set bit 5 on; binary mode
MOV DH,            ;DH must be   to set information
MOV AH, 44H         ;DOS IOCTL function call
MOV AL,1           ;IOCTL function 1 set device information
MOV BX,dev_handle   ;Magnetic device handle
INT 21H            ;Issue DOS IOCTL function call
```

02H: Read

You use this IOCTL function to read status information. The information returned depends on the retrieval state that is in effect at the time this function call is issued. You set the retrieval state by using the IOCTL Write Function Call (see “03H: Write” on page 4-4). The returned status can be any one of the following:

Device-driver status
Magnetic-stripe-unit parameters in effect
Error counter statistics
Power-on test status or configuration

The first byte in the inbound data reflects the existing retrieval status of the driver. The following is a list of the retrieval states:

00	Driver status
01	Magnetic-stripe-unit parameters
02	Error statistics
03	Power-on test status and configuration

Driver Status: For information about the status of the device driver, see “Status Reporting” on page 4-7.

Magnetic-Stripe-Unit Parameters: When the device driver is in retrieval state 01, an IOCTL Read, (AX)=4402H, returns a message comprised of 1 and 1 to 25 bytes of the magnetic-stripe-unit parameters. If you have defined these parameters, they are now in effect. For more information about the parameters, see “Loading Magnetic-Stripe-Unit Parameters” on page 4-4.

Error Statistics: When you set the device driver to retrieval state 02, an IOCTL Read, (AX)=4402H, causes the driver to return an 8-byte or 11-byte message that consists of 02H and 7 or 10 bytes.

The device driver accumulates error statistics starting at power-on. These error statistics are in the form of 8-bit binary values. Each counter can count up to FFH. If another count is recorded, the counter goes to 80H and remains at that value until power-off. Reading the counters does not affect their value.

Counter	**Statistic**
Byte 1	Track-1 read errors
Byte 2	Track-2 read errors
Byte 3	Track-3 read errors
Byte 4	Encode read-back errors (for single-track encode only)
Byte 5	Read operator cancels
Byte 6	Encode operator cancels
Byte 7	All other magnetic-stripe-unit errors
Byte 8	Track-1 encode errors (for multitrack encode only)
Byte 9	Track-2 encode errors (for multitrack encode only)
Byte 10	Track-3 encode errors (for multitrack encode only)

Note: Bytes 8, 9, and 10 are sent only when your application has set parameters for a multitrack encode operation. For multitrack encodes, byte 4 contains the accumulated total of encode errors for all tracks; and bytes 8, 9, and 10 contain errors detected on a particular track. If your application is doing a single-track encode operation, only bytes 1 through 7 are received. Byte 4 contains the encode error count.

Read Power-On Test Status and Configuration Data: A 4-byte message is returned to the application in response to IOCTL Read, (AX)=4402H, when in retrieval state 03.

Byte 0 03H

Byte 1 Power-On Test Status

00H	No errors
30H	System unit model is not supported
31H	Transmission error
33H	Device-driver installation error
34H	Magnetic-stripe device attached to the wrong connector
36H	Break in device cable, or no device attached
37H	No response from magnetic-stripe unit
38H	Device self-test failure
39H	Magnetics CONFIG.SYS option specification error

Byte 2 Read Capability

Bits 7–4	Reserved
Bit 3	Asynchronous mode
Bit 2	Track 1
Bit 1	Track 2
Bit 0	Track 3

Byte 3 Encode Capability

Bits 7–4	Reserved
Bit 3	Reserved
Bit 2	Track 1
Bit 1	Track 2
Bit 0	Track 3

03H: Write

You use an IOCTL Write to send information to the device driver or to read and write data in a TopView environment. The first byte of the outbound data string defines the message type.

Byte 0	
00H	Load magnetic-stripe-unit parameters
01H	Set synchronous/asynchronous mode
02H	No operation
03H	Disable magnetic-stripe unit
04H	Set 4704 read compatibility mode
05H	Set non-4704 read compatibility mode
06H	TopView read data
07H	TopView write data
08H	Reserved
FFH	Set retrieval state

Loading Magnetic-Stripe-Unit Parameters: This message is sent to the device driver by an IOCTL command with:

Byte 0 00H (load magnetic-stripe-unit parameters).

Byte 1 Track selection bit definitions:

7	Reserved
6	Read track 1
5	Read track 2
4	Read track 3
3	Reserved
2	Encode track 1
1	Encode track 2
0	Encode track 3.

Byte 2 Track-1 data and longitudinal redundancy check (LRC) parity:

Value	**Meaning**
00H	Odd data, odd LRC parity
01H	Even data, even LRC parity
02H	Odd data, even LRC parity
03H	Even data, odd LRC parity.

Byte 3 Track-1 bits per character (including the parity bit):

Value	**Meaning**
05H	5 bits per character
06H	6 bits per character
07H	7 bits per character.

Note: If you specify values outside these ranges, the driver uses the default value.

Byte 4 Track 1 - start-of-message character.

Byte 5 Track 1 - alternate start-of-message character.

Byte 6 Track 1 - end-of-message character.

Byte 7 Track 1 - alternate end-of-message character.

Byte 8 Track 1 - format control (encoding only).

If bit 7 of byte 8 is equal to 0, the byte is a single record.
If bit 7 is equal to 1, the byte is a double record, and
bits 6 through 0 are the number of interrecord zero characters.

Byte 9 Track 1 - number of leading zero characters.

Bytes 10-17 Same as bytes 2 through 9, except for track 2.

Bytes 18-25 Same as bytes 2 through 9, except for track 3.

If the request in the Track Selection Code (byte 1) is outside the range for the attached magnetic device, error status 8101H is returned in Status Word 1 as the result of an IOCTL Read Device-Driver Status command. (See “02H: Read” on page 4-2.) In addition, the device driver reduces the code for the selected track so that it is within the capacity of the selected device.

For example, if the attached magnetic device supports reading and encoding only on track 2, but the application program requests (in byte 1) read tracks 2 and 3, the device driver changes the track selection code in byte 1 to select track 2. Error status 8101H is returned in Status Word 1 as the result of an IOCTL Read Device-Driver Status command. (See “02H: Read” on page 4-2.)

In the Encoding Format Control (bytes 8, 16, and 24), if bit 7=1 (indicating double recording) and if bits 0 through 6 are set to 0, the device driver resets bits 0 through 6 to a value of five interrecord characters, and error status 8101H is set in Status Word 1.

If bit 7=0 (single recording), and if bits 0 through 6 are not set to 0, the device driver sets bits 0 through 6 to 0 and error status 8101H is set in Status Word 1.

For the remaining parameters, if the requested track (set in byte 1) for a read or encode request is within the capacity of the attached device, but the parameters in bytes 2 and 3 are unacceptable, the device driver sets parameters to their defaults, and error status 8101H is set in Status Word 1.

Set Mode: When byte 0 in an outbound control message is 01H, byte 1 determines the mode to which the driver is set. The outbound data message is:

0100H (set asynchronous mode)
0101H (set synchronous mode)

Disable Magnetic-Stripe Unit: When byte 0 in the outbound control message is 03H, the device driver disables the magnetic-stripe unit. All the indicators on the device are switched off.

Set 4704 Read Compatibility Mode: When byte 0 in the outbound control message is 04H, the device driver is set to 4704 read compatibility mode. When this mode is in effect for multitrack read operations, the device driver returns data to the application input buffer if data from at least one track is valid. If errors are detected on *all* tracks, no data is returned. The format of the data returned to the application is described in “4704 Read Compatibility Mode” on page 4-11. Any parameters set by the IOCTL Write 00H command are not affected by this command.

Set Non-4704 Read Compatibility Mode: When byte 0 in the outbound control message is 05H, the device driver is set to non-4704 read compatibility mode. This is the default read mode. This command is necessary only when the IOCTL Write 04H command has been processed and you want the application program to return to the default read mode. Any parameters set by the IOCTL Write 00H command are not affected by this command.

When the non-4704 read compatibility mode is in effect for multitrack read operations, the device driver returns no data to the application program input buffer if an error is detected in the data read from any track. Data is returned to the application input buffer only if data from *all* tracks is *valid*. The format of the data returned to the application is described in “Non-4704 Read Compatibility Mode” on page 4-11.

TopView Read Data: For information about TopView Read, see “Read Data” on page 4-17.

TopView Write Data: For information about TopView Write, see “Write Data” on page 4-18.

Set Retrieval Status: When byte 0 in the outbound control message is FFH, byte 1 sets up certain conditions in the device driver.

Your application can retrieve several kinds of information concerning the 4777 Magnetic Stripe Unit. To differentiate which kind of information is returned on the IOCTL Read, the application issues an IOCTL Write, with (AX)=4403H. This sets the desired retrieval status for the IOCTL Read (AX)=4402H. The IOCTL Write sends a control message to the driver to set the retrieval statuses that are shown below. The values of these bytes are:

FF00	Set Retrieve Driver Status (default)
FF01	Set Retrieve Magnetic-Stripe-Unit Parameters
FF02	Set Retrieve Error Statistics
FF03	Set Retrieve Power-On Test Status and Configuration Data

If byte 1 is a value other than 00H through 03H, then 8103H is set in Word 1 of the device-driver status.

The device driver retains various data, parameters, and the status indicated above. It returns this information with the retrieval state condition in effect at the time of the IOCTL Read. (See “02H: Read” on page 4-2.)

After power-on and the completion of all IOCTL Reads, the retrieval status is set to 00H (retrieve device-driver status). Therefore, it is unnecessary to set a retrieval status to obtain driver status after the completion of function calls.

Status Reporting

The 4777 device driver indicates status to the application with two different mechanisms:

DOS status A value returned to the application in register AX after completion of a function call if the carry flag is set

Device-driver status A value returned to the application program as the result of an IOCTL Read Status

This is done for both synchronous and asynchronous operations.

DOS Status

The device driver returns a status word to DOS at the completion of each INT 21H function call. DOS status is returned to the application program in register AX under the following conditions:

If you use the /Y option, no error codes are returned to DOS; therefore, the carry flag is reset. Register AX is equal to the byte transfer count. Function call status must be obtained by the IOCTL Read Status command. (See “02H: Read” on page 4-2.)

If the /Y option is not used and no error occurred, the carry flag is reset and register AX is equal to the byte transfer count.

If the /Y option is not used and an error did occur, the carry flag is set and register AX is equal to the DOS error code.

When control is returned to the calling program, a carry flag indicates whether the function call was successful. If the carry flag is not set, the call is successful. If the carry flag is set, the call failed and an error code is returned in register AX.

Note: When you use the /Y parameter to load the device driver, the error codes are stored in the driver’s status and your application program must read and test the status to check for errors. For more information about DOS error codes, see the *DOS Technical Reference*.

All error conditions cause the INT 24H critical-error handler to be started. The application program can supply a critical-error handler or use the handler supplied with DOS for error recovery. However, IOCTL errors do not call the critical-error handler. Because DOS does not initiate an INT 24H on an IOCTL error, the only error indication the application program receives is that the carry flag is set and register AX contains a DOS error code. For a list and explanation of the DOS error codes, see the *DOS Technical Reference*.

Device-Driver Status

The device driver maintains three status words on the progress of an I/O operation. The contents of these words are appended to a 00H byte and passed back to the application program as a 7-byte message in response to an IOCTL Read Status call. The three status words of the magnetic-stripe unit are:

Word 1: Driver Status
Word 2: I/O Status
Word 3: Data length of a completed I/O operation

Your application program must allocate an adequate buffer to store the status message (7 bytes). This status-message buffer should not be the same as the data buffer. If the buffer is too short, 0208H is set into Status Word 2.

The format of each inbound status message is:

Byte 0: 00H
Byte 1: Word 1 low byte
Byte 2: Word 1 high byte
Byte 3: Word 2 low byte
Byte 4: Word 2 high byte
Byte 5: Word 3 low byte
Byte 6: Word 3 high byte

Status Word 1

When you issue a Read, IOCTL Write, IOCTL Read Parameters, or IOCTL Read Error Counters request, the bits shown below are set in Word 1. If the /Y option is not in effect, the bits are also set in the Device-Request-Header Status Word. The application program determines the status of the call by an IOCTL Read Status if you need to analyze the Request-Header Status Word.

All code points in the range 8xxx indicate that the call was not completed successfully and cause DOS to set the carry flag and an error code in register AX (if the /Y option is not in effect). The code points of the 4777 Magnetic Stripe Unit are:

0000H Status previously reported and cleared

0100H Call done without error

8101H Requested function unavailable, device parameters out of range when loading magnetic-stripe-unit parameters

8102H Device not ready or bad command sequence

8103H Unknown command

810AH Write Fault - Synchronous Mode Cancel, or allocated buffer is less than 2 bytes

810BH Read Fault - Synchronous Mode Cancel, or allocated buffer is less than 4 bytes

810CH General Failure - cannot communicate with the device

Word 1 is cleared to 0000H after the status message is sent to the application program.

Status Word 2

If an I/O operation is in progress, Word 2 of the driver status contains FFFFH. The only function calls that the driver accepts in this status are IOCTL Read Status and a cancel request. If any other call is sent to the driver, it is rejected as a Bad Sequence (Status Word 1 = 8102H).

After the status message is sent to the application program, Status Word 2 is cleared to 0000H, except when an I/O operation is in progress (I/O busy).
The I/O busy status is cleared (set to 0000H) on completion of the function call.

Note: When the application program operates in synchronous mode, the application program never sees the FFFFH status.

The following status codes are set in Status Word 2 to indicate the existing I/O status of the 4777 Magnetic Stripe Unit:

0000H	No I/O operation in progress
0030H	System unit model not supported
0031H	Transmission error
0032H	Adapter hardware error
0033H	Device-driver installation error
0034H	Device attached to wrong connector
0036H	Device cable break
0037H	No response from the 4777 Magnetic Stripe Unit
0038H	Device self-test failure
0039H	Magnetics CONFIG.SYS option specification error
0208H	Allocated buffer too short
FFFFH	I/O operation in progress (busy)

Status Word 3

Status Word 3 conveys data length information. It is always 0000H, except when an I/O operation is completed. This status represents the length of the data read or written. If the I/O operation completes without an error (Word 2 = 0000H), the data length in Word 3 should equal:

(CX) for Write
(AX) for Read

Word 3 is cleared to 0000H after an IOCTL Read Status is performed.

Using the API to Read and Write Data with an MSRE

This section describes the data formats and operations that are used to read and write magnetic-stripe data.

Reading Data

The following steps must be taken for a 4777 read operation:

1. Open the device driver with (AX)=3D00H for read or (AX)=3D02H for read and write. You have to issue Open only once each session.
2. Issue IOCTL call (AX)=4401H to set binary mode on. You have to issue Set Binary Mode only once each session.
3. Issue Read calls (AX)=3F00H as required to enable the 4777 data input operation.

When a Read call is made by your application program, the driver enables the device and the green light on the unit comes on. You can now pass media though the slot.

The 4777 device driver permits the reading of multiple magnetic media tracks on a single pass of the medium.

In synchronous mode, control does not return to DOS until the Read is completed (successful card passes and data checks) or until you press the Cancel key (error completion).

In asynchronous mode, control returns to DOS as soon as the device driver accepts the Read call. The application then must periodically perform an IOCTL Read Status to validate completion of the Read. If you cancel, your application must intercept the key it defines as Cancel and send an IOCTL Write Disable call ("03H: Write" on page 4-4) to the driver.

Default parameters and the non-4704 read compatibility mode are set automatically in the device driver. This enables the device to read any of the supported tracks, as described in "Non-4704 Read Compatibility Mode" on page 4-11. Track selection is set to read track 2; however, the driver lets the application load user-specified parameters into the device driver.

Shown below are the device-driver Read Defaults associated with each track.

Track-1 Read

SOM = 05H
EOM = 1FH
Bits per character = 07H
Odd data parity
Even LRC parity

Track-2 Read

SOM = 0BH or 0DH
EOM = 0FH or 0CH
Bits per character = 05H
Odd data parity
Even LRC parity

Track-3 Read

SOM = 0BH or 0DH
EOM = 0FH or 0CH
Bits per character = 05H
Odd data parity
Even LRC parity

User-Specified Operation

If the default parameters described earlier in this document do not suit your application, your application program can set new and unique parameters in the device driver. This is done by the IOCTL Write command to load magnetic-stripe-unit parameters ("03H: Write" on page 4-4). This permits special definition of the operational characteristics of the device and the data expected to and from each track.

Input Data Formats

Depending on the type of compatibility mode in effect, you can use two input data formats:

Non-4704 read compatibility mode
4704 read compatibility mode

Non-4704 Read Compatibility Mode:

The format of the data passed to the application for tracks 1, 2, and 3 is as follows:

```
1 Track

 Lx     Tx data


2 Tracks

 Lx     Tx data      Ly     Ty data


3 Tracks

 Lx     Tx data      Ly     Ty data      Lz      Tz data
```

The **L** fields are 1-byte length fields that define the length of the **T** fields. The L byte and the 00H byte are not included in the length value.

The **T** fields are the data read from the respective track and include the start of message (SOM) and end of message (EOM), but exclude the LRC.

The 00H byte defines the end of the inbound 4777 data message.

The lower-numbered track is always the first data segment in the message. The next-lower track is the next data segment. The third and last data segment is the highest-numbered track and it contains the existing track-3 data.

If two SOMs are defined in the parameter table (SOM 1 and SOM 2), the driver accepts either SOM as a valid start of message. The same is true for dual EOMs.

The device driver assists error recovery. If the data on any of the selected tracks has an error that pertains to format, parity, or LRC checks, the red light comes on and no data is placed in the application input buffer. The device stays enabled and waits for a subsequent pass of the medium. You can pass the medium until a good read occurs, or you can cancel the operation. If you cancel the Read, no data is returned to the application.

The device driver signals a cancel to DOS by placing 0BH (Read Fault) status into the status byte of the Request Header and register DI on return to DOS. Error statistics on the number of error passes and the cancel action are kept in the device driver. In asynchronous operation, the application program disables the device in the event of a cancel. The canceled operation is counted in the error statistics the same as in synchronous operation.

4704 Read Compatibility Mode: The format of the data passed to the application for tracks 1, 2, and 3 is as follows:

```
L  Header       Flag  SOM    Data...        EOM

                          Length
                        One Record
```

A record is the input data from one track, as shown above. If multiple tracks are used, the application input buffer contains multiple records.

The L field indicates the length of the data, as shown above.

The Header field is a constant: 0EH.

The Flag byte indicates which track produced the data and whether the data is the first record, an intermediate record, or the last record in a sequence. The bit definitions for this byte are as follows:

Bits 7–4

Reserved

Bits 3,2

00 Intermediate record
01 Last record
10 First record
11 Only record

Bits 1,0

00 Track-1 data
01 Track-2 data
10 Track-3 data
11 Reserved

A 00H byte is appended to the last record but is not included in any length field.

The device driver affects error recovery and the data placed in the application input buffer. Data read from any track without error is always placed in the application input buffer. If no SOM is detected on a track, the track is considered blank and the following record is placed in the buffer:

```
 4    E   Flag   SOM   EOM
```

If an SOM is detected but the data contains a parity error or an LRC error, the track is considered to be *invalid* and no record is placed in the buffer. If any data is placed in the application input buffer, the Read operation is considered to be *valid* and the red light remains off. If errors are detected in the data read from *all* tracks, then the Read operation fails and the red light comes on.

The device driver signals a cancel to DOS by placing 0BH (Read Fault) status into the status byte of the Request Header and register DI on return to DOS. The device driver stores error statistics on the number of error passes and the cancel action. In asynchronous operation, the application program disables the device in the event of a cancel. The canceled operation is counted in the error statistics the same as in synchronous operation.

Data Output

You must open the driver before you attempt to write data. Binary mode must also be set on. A Write call enables the encoder for writing. If operation is in synchronous mode, control is returned to the application program after the I/O operation is successfully completed or you end the operation by the Cancel key.

In asynchronous mode, control returns to the application program after the Write call is accepted by the driver.

After the Write call is accepted, the operation is similar to Read.

Writing Data

Below are the device-driver Write defaults associated with each track. Certain defaults vary, depending on what model of the 4777 is attached during device-driver initialization. To alter the default settings, use the IOCTL Write function call.

Track-1 Write

Records written = single
Leading zero bits = 70 (10 characters)
Bits per character = 07
Odd data parity
Even LRC parity

Track-2 Write

Records written = double for Model 2
Records written = single for Model 3
Leading zero bits = 300 (60 characters) for Model 2
Leading zero bits = 25 (5 characters) for Model 3
Interrecord zero bits = 25 (5 characters)
Bits per character = 05
Odd data parity
Even LRC parity

Track-3 Write

Records written = double
Leading zero bits = 300 (60 characters)
Interrecord zero bits = 25 (5 characters)
Bits per character = 05
Odd data parity
Even LRC parity

If you are using a 4777 Model 002 or Model 003, track selection is set to encode track 2. You must supply, in the output data format shown below, the output data that you want encoded. This includes the SOM and EOM.

For default operation, the data must be in the form `B 1 2 3 4 5...... F`. Each byte contains the character encoded in the high and low halfbytes. Depending on your current parameter setting for bits-per-character, you might ignore certain bits in the high halfbytes if they fall out of the range of the setting that you have specified.

User-Specified Operation

If the default parameters described earlier in this document are not suitable for a given application, your application program can set new and unique parameters in the driver. You do this by using the IOCTL Write command to load magnetic-stripe-unit parameters. This permits special definition of the operational attributes of the device and the data expected to and from each track.

Output Data Format for Single-Track Encode

A single-track encode operation is identified to the device driver when the application has set the Track Selection Code to a value that indicates that only one of the three tracks is used for encoding. (See "Loading Magnetic-Stripe-Unit Parameters" on page 4-4.)

You must open the driver (set register AX=3D01H or 3D02H) before you attempt to enable the device for writing. When the application program issues a Write function call, the device driver enables the encoder, and the flashing green light on the unit comes on. This indicates that the unit is ready to encode.

The format of the data to the device driver is as follows:

```
  SOM      Data         EOM
```

The driver takes the number of low-order bits of each output byte (as specified by the bits-per-character parameter) and calculates the correct parity based on these bits. These bits, plus the parity bit, form the character to be encoded.

For example, if the application program passes the byte 35H, and 5 bits per character is in effect, the driver takes the low 4 bits (the halfbyte 5) as the character. It calculates the correct parity bit (5th bit) and passes it to the unit for encoding. If you specify 6 bits per character, 15H + parity is encoded. At 7 bits per character, 35H + parity is encoded. This procedure applies to the SOM and EOM and also the data bytes.

The device driver calculates the LRC and adds it to the encoded record.

The encoded data is read back by a read head that trails the encode head in the device. The read-back data is checked and compared to the original encode data. If the check matches the data, the encode successfully ends. If a read-back error occurs, the red light on the unit comes on along with the flashing green light. You can make encode attempts until the read-back check verifies a match. At that time, the operation successfully ends. The device driver accumulates the number of encode failures and reports them to your application.

You can cancel the encode by the same procedures as defined for Read.

Output Data Format for Multitrack Encode

A multiple-track encode operation is identified to the device driver when the application sets the Track Selection Code (see "Loading Magnetic-Stripe-Unit Parameters" on page 4-4) to a value that indicates that two or more of the tracks are used for encoding.

You must open the driver (set register AX=3D01H or 3D02H) before you make any attempt to enable the device for writing. When the application program issues a

Write function call, the device driver enables the encoder and the flashing green light comes on. This indicates that media can be passed through the slot for encoding.

The format of the data to the driver is shown below. Note that multitrack data must be contiguous in the application buffer when passed to the driver.

```
2 Tracks

  Lx  SOMx DATAx EOMx Ly SOMy DATAy EOMy

           Lx                  Ly

 Valid combinations:      DATAx  DATAy

                       trk 1  trk 2
                       trk 2  trk 3

3 Tracks

  Lx   SOMx DATAx EOMx Ly SOMy DATAy EOMy Lz SOMz DATAz EOMz

            Lx                  Ly                Lz

Valid combinations:     DATAx   DATAy   DATAz

                     trk1    trk 2   trk 3

L (byte) = 1-byte length of encode data (includes SOM, DATA, EOM)
```

Note: The device driver determines which track gets the data segments. This is based on the values set in the Track Selection byte when the parameters are loaded. The lower-numbered track must always be the first data segment in the message. The next-lower track is the next data segment. The third and last data segment is always track 3 if present.

The driver takes the number of the low-order bit of each output byte (as specified by the bits-per-character parameter) and calculates the correct parity based on these bits. These bits, plus the parity bit, form the character to be encoded.

The device driver calculates the LRC and adds it to the encoded record.

The encoded data is read back by a read head that trails the encode head in the device. (All tracks must be read back correctly.) The read-back data is checked and compared to the original encode data. If the check matches the data, the encode successfully ends. If a read-back error occurs, the red light on the unit comes on along with the flashing green light.

You can make encode attempts until the read-back check verifies a match. At that time, the operation successfully ends. The driver accumulates the number of encode failures and reports them to your application.

You can cancel the encode using the same procedures as defined for Read.

Device Status Lights

The 4777 Magnetic Stripe Unit has lights that inform you of various conditions.

Read Status

Green Light steady on - Ready to read

This light comes on as a result of a Read to the driver. You should now pass a medium through the slot. After the medium has passed through the slot, the green light goes off and the yellow light comes on.

Yellow Light - In process, wait

This light comes on after you pass the media through the reader. It stays on until the validity checks on the data are completed.

Red Light - Error detected, try again

The red light comes on if either of these conditions occurs:

If *any* of the data read from the active tracks fails the validity checks (in the non-4704 read compatibility mode)

If *all* of the data read from the active tracks fails the validity checks (in the 4704 read compatibility mode *or* the non-4704 read compatibility mode)

The red light remains on until:

- A subsequent good read occurs.
- The operation is canceled.
- The 4777 is disabled.

Write Status

Flashing Green Light - Ready to encode

When the driver issues a Write call, the green light flashes two or three times per second. You can now pass media through the encoder.

Yellow Light - In process, wait

This light comes on immediately after the medium is passed through the encode slot. It stays on until after the read-back check is complete.

Red Light - Error detected, try again

The steady On of the red light shows a bad read-back check. It stays on until:

- A subsequent good encode occurs.
- The operation is canceled.
- The magnetic-stripe unit is disabled.

TopView Support

This section is only for the applications that you use when you are running the 4777 in a TopView environment.

In a TopView environment, your applications cannot use the standard Read and Write function calls to enable the 4777 for reading and encoding. You must use an IOCTL (function 44H) in place of the Read and Write. This TopView IOCTL function performs the same functions as the Read and Write functions perform in a non-TopView environment.

To take advantage of TopView's multitasking features, run the device driver in asynchronous mode. If the driver is in synchronous mode and is enabled for read or write, no other TopView application can get control of the device until you pass a magnetic stripe and end the operation. See "4777 Device Driver Operations" on page 2-1 for additional information about synchronous- and asynchronous-mode operation.

Critical-Error Handler (INT 24H)

Because the TopView Read and Write data operations use IOCTL function 44H instead of Read (3FH) and Write (40H), any errors detected by the device driver do *not* call the INT 24H critical-error handler. Therefore, it is necessary for your application to always perform an IOCTL Read status operation to determine whether the Read and Write functions successfully completed.

IOCTL Write

You can use the IOCTL Write function call to read and write device data in the TopView environment.

You can use the Read and Write functions 3FH and 40H for data transmission, but they operate correctly only in the non-TopView environment.

The application informs the driver if the request is for a Read or Write by the value of the first byte in the data buffer passed to the device driver. The definition of this first byte ID is:

Byte 0 = 06H (Read magnetic-device data)
Byte 0 = 07H (Write magnetic-device data)

Before you issue these IOCTL Read and Write functions, the application must perform any functions that are required by the existing Read and Write operations (such as Open Driver and Set Binary Mode).

Read Data

To read magnetic data in a TopView environment, send an INT 21H with the following register values:

(AX)=4403H IOCTL Write.

(BX) File handle.

(CX) The size (in bytes) of the input data buffer. This size must be large enough for any possible input and at least 5 bytes (including the first byte containing the read option ID).

(DS:DX) The address (Segment:Offset) of the input data buffer. The first byte in the data buffer must be a 06H to identify an IOCTL Read magnetic-device data operation. Input data from the magnetic device will follow the 06H ID byte after a successful read.

The following value is returned from the TopView read request.

(AX) Number of bytes transferred if carry flag is not set. These bytes do not include the 06H ID byte.

DOS status if the carry flag is set. (See “DOS Status” on page 4-7.)

Write Data

To encode magnetic data in a TopView environment, send an INT 21H with the following register values:

(AX)=4403H IOCTL Write.

(BX) File Handle.

(CX) The number of bytes to write (must be at least 3 bytes: ID, SOM, and EOM).

Note: CX includes the 07H ID byte that precedes the write data.

(DS:DX) Address (Segment:Offset) of the input data buffer. The first byte in the data buffer must be a 07H to identify a Write magnetic-device data operation. This is followed by the write data stream.

Values are returned from the TopView read request.

(AX) Number of bytes transferred if carry flag is not set. These bytes do not include the 07H ID byte.

DOS status if carry flag is set. (See “DOS Status” on page 4-7.)

Before sending the TopView Write, the application must perform any functions that are required by the standard (40H) Write function (such as Open Device Driver or Set Binary Mode).

Chapter 5. Programming for the 4778 PIN-Pad

This chapter describes the IOCTL function calls, the status reporting, TopView support, data encryption functions, cryptographic key management, and the download utility for the 4778 PIN-pad device driver for the serial port, (IBM4778P.SYS) and for the auxiliary port, (PIN2DD.SYS).

IOCTL Function Calls

You send the IOCTL DOS function calls to pass I/O control strings to the device driver to:

Set or read device information
Send control strings to the device
Receive control data from the device
Read data in a TopView environment
Use the enhanced security functions

Note: The IOCTL function never calls INT 24H.

The following registers and their IOCTL functions are described below:

Registers on entry

(AH)=44H The function number.

(AL) The function value:

- **00H** Get device information (information returned in DX).
- **01H** Set device information (data is in DX).
- **02H** IOCTL Read is used to read status and configuration data.
- **03H** IOCTL Write is used to write control data to the device driver.

(BX) File Handle.

(CX) The length of the data buffer for the IOCTL Read and Write.

(DS:DX) Address (Segment:Offset) of the data buffer for an IOCTL Read and Write.

(DL) Data for Get or Set Information.

Registers on return

(AX) The number of bytes transferred if the carry flag is not set; DOS status if the carry flag is set (see “DOS Status” on page 5-4).

00H: Get Device Information

You use this IOCTL function to get information about the PIN keypad. The data is returned in register (DX). For information about bit definitions of the returned information, see the *DOS Technical Reference* manual. You must use this function before you set any device information. For information on checking device status see, “Status Reporting” on page 5-4.

01H: Set Binary Mode

You use this IOCTL function to set device information (specifically, the device-channel operating mode). You should set binary mode after issuing a a device open, see “Open” on page 2-3. To activate this function, set the register AX to 4401H. The device data is set in register DL; DH must be set to 0.

When the channel is in ASCII mode, DOS checks for 1AH (→) end-of-file characters. When the channel is in binary mode, DOS does not perform any special checks. This means that all bit patterns are sent across the channel. Use bit 5 of register DL to set binary mode immediately after you get device information. Remain in this mode for as long as the PIN keypad remains open. Before your application program sets binary mode, it should read the device information. This saves the existing device settings. Use the following assembly language procedure to set binary mode:

```
MOV AH, 44H       ;DOS IOCTL function call
MOV AL,          ;IOCTL function  , get device information
MOV BX,dev_handle  ;PIN keypad handle
INT 21H          ;Issue DOS IOCTL function call
OR DL,  1     B   ;Set binary mode
MOV DH,          ;DH must be   to set information
MOV AH,44H        ;DOS IOCTL function call
MOV AL,1         ;IOCTL function 1 set device information
MOV BX,dev_handle  ;PIN keypad handle
INT 21H          ;Issue DOS IOCTL function call.
```

02H: Read Keypad Data

You use this IOCTL function to read status conforming to the retrieval state that is in effect when this function call is used. You activate the retrieval state by using the IOCTL Write Function Call. (See “03H: Write Keypad Data” on page 5-3 for information about how to do this.) The returned status can be either of the following:

Device-driver status
POR (Power-On Reset) status and configuration

The first byte in the inbound data stream indicates the retrieval state that is in effect; 00H for device-driver status and 03H for POR status and configuration. To activate this function, set register AX to 4402H. The status data is returned in the buffer pointed to by DS:DX.

Driver Status: An IOCTL Read command returns the status of the device driver when the driver is in the Retrieve Device Driver Status (retrieval state 0) state. For more information about the status see “Status Reporting” on page 5-4. In the remainder of this chapter, this operation is referred to as IOCTL Read Status.

Read POR Status and Configuration: A 6-byte message is returned to the application in response to IOCTL Read, (AX)=4402H, when in the 03 Retrieval State.

Byte 0 = 03H
Byte 1 = 00H (not used)
Byte 2 = 00H (not used)
Byte 3 = 00H (not used)

Byte 4 = 00H (not used)
Byte 5 = configuration status:

- 00000000B no PIN keypad attached
- xxxxxx01B clear PIN keypad mode
- xxxxxx10B encrypted PIN keypad mode
- xxxxx1xxB CONFIG.SYS option specification error
- xxxx0xxxB synchronous mode
- xxxx1xxxB asynchronous mode
- xxx0xxxxB non-4700 compatibility mode
- xxx1xxxxB 4700 compatibility mode
- xx1xxxxxx POR diagnostic test failure

Bytes 1, 2, 3, and 4 are always zero. These bytes are returned to maintain compatibility with existing 4700 Finance Communication System applications that expect to see 6 bytes. In this chapter, this operation is referred to as *IOCTL Read POR Status and Configuration*.

03H: Write Keypad Data

You use this IOCTL function to set device-driver states, disable the PIN keypad, and perform security functions. To activate this function (with an outbound data buffer pointed to by DS:DX), set register AX to 4403H. Bytes 0 and 1 of the outbound data string define the action that is taken:

0100H Set Asynchronous Mode (default): To set asynchronous mode, send an IOCTL Write command with data bytes 0 and 1 of the data buffer equal to 0100H.

0101H Set Synchronous Mode (default): The device driver defaults to synchronous mode, after POR. To return the device driver from asynchronous mode to synchronous mode, send an IOCTL Write command with data bytes 0 and 1 of the data buffer equal to 0101H.

0102H Set 4700 PIN Compatibility Mode (default): To set the device driver to 4700 compatibility mode, use the IOCTL Write command. Set the data bytes 0 and 1 of the data buffer equal to 0102H. This mode ensures that PIN data passed to the application conforms to the 4700 PIN-keypad format.

0103H Set Non-4700 PIN Compatibility Mode: To set the device driver to non-4700 compatibility mode, use the IOCTL Write command. Set the data bytes 0 and 1 of the data buffer equal to 0103H.

02H No operation.

03H Disable PIN Keypad: To disable the PIN keypad, use the IOCTL Write command. Set data byte 0 of the data buffer equal to 03H. This causes the indicator on the LCD to be cleared. Word 2 of the device-driver status message goes from FFFFH to 0000H (see “Status Reporting” on page 5-4).

05H Security functions (see “Security Functions” on page 5-17).

06H TopView Read (see “Read PIN-Keypad Data” on page 5-7).

FF03H Set Retrieve POR Status and Configuration Data: To set the POR status and configuration message so that they return on the succeeding IOCTL Read function call, use the IOCTL Write command. Set data bytes 0 and 1 of the data buffer equal to

FF03H. To receive POR status or a configuration message, the IOCTL Read must immediately follow the IOCTL Write function call. When the function call is complete, the device driver automatically returns to Retrieve Driver Status state.

FF00H Set Retrieve Driver Status (default): To set the device driver in a state where the driver status is returned on the succeeding IOCTL Read function call, use the IOCTL Write command. Set data bytes 0 and 1 of the data buffer equal to FF00H.

Status Reporting

Two different status mechanisms are available to the application program:

DOS Status A value returned to the application in register AX after completion of a function call if the carry flag is set

Device-Driver Status A value returned to the application as the result of an IOCTL Read Status

This is done for both synchronous and asynchronous operations.

DOS Status

The device driver returns a status word to DOS at the completion of each INT 21H function call. DOS status is returned to the application program in register AX under the following conditions:

If you used the /Y option, no error codes are returned to DOS, and the carry flag is reset. Register AX is equal to the byte transfer count. Function call status must be obtained by the IOCTL Read Status command (see “02H: Read Keypad Data” on page 5-2).

If the /Y option is not used, and no error occurred, the carry flag is reset and register AX is equal to the byte transfer count.

If the /Y option is not in effect and an error did occur, the carry flag is set and register AX is equal to the DOS error code.

When control is returned to the calling program, a carry flag indicates whether the function call was successful. If the carry flag is not set, the call is successful. If the carry flag is set, the call failed and an error code is returned in AX.

Note: Error codes are not returned to DOS if the driver was loaded with the /Y parameter. The error codes are stored in the driver’s status and the application program must read and test the status to check for errors. For more information about DOS error codes, see the *DOS Technical Reference* manual.

All error conditions cause the INT 24H critical-error handler to be called. The application program can supply a critical-error handler or use the handler supplied with DOS for error recovery. However, IOCTL errors do not call the critical-error handler. Because DOS does not initiate an INT 24H on an IOCTL error, the only error indication the application program receives is that the carry flag is set and register AX contains a DOS error code. For a list and explanation of the DOS error codes, see the *DOS Technical Reference* manual.

Device-Driver Status

The device driver maintains three status words on the progress of an I/O operation. The contents of these words are appended to a 00H type byte and passed back to the application program as a 7-byte message in response to an IOCTL Read Status. The definition of the three status words of the 4778 PIN keypad are:

Word 1: Driver status (status of the previous function call)
Word 2: I/O status
Word 3: Data length of a completed I/O operation

Your application program must allocate an adequate buffer to store the status message (7 bytes). This buffer should be different than the data buffer.

The format of the inbound status message is as follows:

Byte 0: 00H
Byte 1: Word 1 low byte
Byte 2: Word 1 high byte
Byte 3: Word 2 low byte
Byte 4: Word 2 high byte
Byte 5: Word 3 low byte
Byte 6: Word 3 high byte

Status Word 1

When you issue a Read, IOCTL Write, or Read POR Status and Configuration request, the bits shown below are set into WORD 1. If the /Y option is not in effect, these bits are also set in the Device Request Header Status Word. The application program determines the status of the call by an IOCTL Read Status if you need to analyze the Request Header Status Word.

All code points in the range 8xxx indicate that the call was not completed successfully and result in DOS setting the carry flag and an error code in AX (if the /Y option is not in effect). The status word 1 values of the 4778 PIN keypad are defined as follows:

0000H Status previously reported and cleared

0100H Call done without error

8102H Device not ready or bad command sequence

8103H Unknown command

810AH Write Fault

Buffer length error on an IOCTL write function call
Security function write error

810BH Read Fault

Buffer length error on Read or IOCTL Read POR Status and Configuration function calls
Security function read error
Synchronous Cancel of the Read operation

810CH General Failure - cannot communicate with the device

Word 1 is cleared to 0000 after the status message is read by the application program.

Status Word 2

If an I/O operation is in progress, Word 2 of the device-driver status contains FFFFH. The only function calls the device driver accepts in this state are IOCTL Read Status and a disable request. If any other call is sent to the device driver, it is rejected as a Bad Sequence (Status Word 1 = 8102H). After the status message is sent to the application program, Status Word 2 is cleared to 0000H, except when an I/O operation is in progress (I/O busy). The I/O busy status is cleared (set to 0000H) when the call is completed.

Note: When the application program operates in synchronous mode, the application program never receives an FFFFH status.

The following status codes are set in Status Word 2 to indicate the current I/O status of the 4778 PIN keypad:

0000H	No I/O operation in progress
0208H	Buffer length error
FFFFH	I/O operation in progress (busy)

Status Word 3

Status Word 3 conveys data length information. It is always 0000H, except when an I/O operation is completed. This status represents the length of the data read or written. If the I/O operation completes without an error, the data length in Word 3 should equal:

CX for IOCTL Write
AX for Read

Word 3 is cleared to 0000H after an IOCTL Read Status is performed.

TopView Support

Use this information only when you are running the 4778 PIN keypad in a TopView environment.

When you use a TopView environment, your applications cannot use the standard Read function to prepare the 4778 for reading.

You must use the IOCTL (function 44H) in place of the Read function (3FH). This TopView IOCTL function performs the same function that the Read function (3FH) performs in a non-TopView environment.

To take advantage of the TopView environment multitasking features, run the device driver in asynchronous mode. If the driver is in synchronous mode, no other TopView application can get control until a PIN is entered.

Note: TopView functions are not supported in an Operating System/2* (OS/2*) MVDM environment.

* Trademark of IBM

INT 24H Critical-Error Handler

Because the TopView Read PIN keypad data operations use IOCTL function 44H instead of Read (3FH), any errors detected by the device driver do *not* start the INT 24H critical-error handler. Therefore, it is necessary for your application to always perform an IOCTL Read status operation to determine if the Read function completed successfully.

IOCTL Read

You can use the IOCTL Read function call to read device data in either the TopView or non-TopView environment. You can use the Read function (3FH) for data transmission but it operates correctly only in the non-TopView environment.

The application informs the driver if the request is for a Read by the value of the first byte in the data buffer passed to the device driver. The definition of this first byte ID is as follows:

Byte 0 (first byte) = 06H (Read data)

Before you use this IOCTL Read function, the application must perform any functions that were required by the existing Read operations (such as Open Driver).

Read PIN-Keypad Data

Send an INT 21H to request TopView Read. TopView Read requires the following register values:

(AH)=44H Function Number.

(AL)=03H Function Value.

(BX) File Handle.

(CX) Size of the input data buffer (in bytes). The size must be large enough for any possible input and at least 5 bytes (including the first byte containing the read option ID).

(DS:DX) Address (Segment:Offset) of the input data buffer. The first byte in the data buffer must be 06H to identify an IOCTL Read PIN Keypad device data operation. Input data from the PIN keypad will follow the 06H ID byte after a successful read. The data buffer must also be large enough to hold the PIN data returned (see “PIN Keypad Data” on page 5-8).

The following value is returned from the TopView Read request.

(AX) Number of bytes read if carry flag is not set; DOS status if carry flag is set (see “DOS Status” on page 5-4)

PIN Keypad Data

The format of the PIN data passed to the application is:

```
 7EH             † PIN data          ††† | 7FH |
 7FH          (32 bytes max)
```

The leading 7EH indicates encrypted PIN data, and a leading 7FH indicates nonencrypted PIN data.

Nonencrypted Keypad Data

The maximum data length that the PIN keypad can buffer and return is 32 digits. The device driver stores ASCII data, along with the 7FH header and trailer bytes, into the application data buffer.

The device driver translates digits 0 through 9 into ASCII codes as shown below. It should be noted that the clear data stream is identical in the 4700 compatible and noncompatible modes.

PIN Keypad Key	Output to Application
0	30H
1	31H
2	32H
3	33H
4	34H
5	35H
6	36H
7	37H
8	38H
9	39H

4704 Format

The PIN keypad always returns 8 bytes of data to the driver. The encrypted PIN data is then translated into 4704 compatible data format. This translation process starts by converting the 8-byte encrypted data stream into 3-3-2 format (resulting in a 24-byte data stream). The 3-3-2 data stream is then translated into 4704 PIN-compatible format by the table below. The 3-3-2 conversion is accomplished by retaining bits 5 through 7 as a byte value, bits 2 through 4 as a byte value, and bits 0 and 1 as a byte value.

For example, assume that the following 8-byte encrypted data stream is returned to the device driver from the 4778:

```
6B 22 34 EF C6 B7 63 94
```

The device driver converts the 8-byte data stream into 3-3-2 format. This results in the following bytes:

```
 3  2  3  1     2  1  5     7  3  3  6  1  2  5  5  3  3     3  4  5
```

The 3-3-2 data is then translated into a 4704 compatible format (using the table shown below). This results in the following bytes:

```
 A  C  A  E 1  C  E  6 1  2  A  A  4  E  C  6  6  A  A 1  A  8  6 1
```

The table used to convert these bytes is:

3-3-2 Data	**Translate To**
0	10H
1	0EH
2	0CH
3	0AH
4	08H
5	06H
6	04H
7	02H

For encrypted PIN data, a 7EH is placed into the data stream as the first byte, identifying the data as encrypted data. The data stream is ended with a 7FH (same as the clear data stream). The translated data stream, along with the header and trailer bytes, always returns 26 bytes of data to the application program.

Non-4704 Encrypted Format

All PIN block requests must be issued through the enhanced security functions of the device driver when in noncompatibility mode. The data is returned to the application program in the 32-byte control block. For more information about the enhanced security functions, see “Security Functions.”

Download Utility

The 4778 download utility (DNLD4778.COM) lets you download or enter required data into the 4778.

The data types requiring download support are:

Data Encryption Standard (DES) Keys
The two types of DES keys are:

Master Key The master key is used for PIN 4704 block encryption and also as a transport key that downloads the remaining keys and the initial chaining value (ICV) after the master key has been downloaded.

Encryption Keys The remaining encryption keys are 8-byte keys that are used for various encryption processes such as PIN encryption and message authentication code (MAC) generation.

Initial Chaining Value
The initial chaining value is an 8-byte data block used in the generation and verification of MACs.

PIN-Verification Parameters
The PIN-verification parameters are used to create PIN offset data in PINs and also when verifying PINs at the 4778.

To use the download utility, do the following:

1. Type the name of the program (DNLD4778) at the DOS prompt.
2. Press **Enter**. You are then presented with six options:

Load DES keys
Load initial chaining value
Load PIN verification parameters
Set mode
Read serial number
Read device driver information

Note: The download utility does not correctly display screen menus with an IBM 4707 display in 480-mode.

Load DES Keys Option

When you select the Load DES keys option on the first menu, a second menu is displayed. This menu lets you select the type of DES key (master key or encryption key) you want to place in the 4778.

Master Key Options

The master key is a 16-byte key that can be loaded into either the 4778 from your workstation or the 4778 PIN keypad. The first time you place a master key into the PIN keypad, you must enter it with the 4778 or load it into the 4778 in nonencrypted form from the workstation.

Load Master Key: This option lets you load the master key into the 4778 using your workstation keyboard. You can choose any of the following four options of the load master key function:

Load 8-byte master key (nonencrypted)
Load 8-byte master key (encrypted)
Load 16-byte master key (nonencrypted)
Load 16-byte master key (encrypted)

If you load an 8-byte master key into the 4778, it is duplicated as the second 8 bytes of the double-length master key in the 4778. If you load the master key in encrypted form, it is decrypted by the 4778 using the resident master key.

Enter the master key in response to the prompt. Use your workstation keyboard to enter the master key in hexadecimal format. If the new master key has correct parity, the old key is replaced and the triple encryption of the serial number with the new master key is returned and displayed. If the new master key has incorrect parity, an error code is returned and displayed. The information returned to you following the download of a master key is displayed on the screen until you press one of the workstation keyboard keys.

To leave the load master key options, enter a null response to the Enter key prompt. (You form a null response by pressing the **Enter** key on the keyboard without entering data.)

Enter Master Key: This option lets you manually enter the master key into the 4778, using the 4778 PIN keypad. You can choose any of the following four options of the enter master key function:

Enter 8-byte master key - single entry
Enter 8-byte master key - dual entry
Enter 16-byte master key - single entry
Enter 16-byte master key - dual entry

The PIN keypad arrow indicator on the display comes on to indicate the start of master key entry and remains on until the operation is complete. If you press a key that is not valid or enter a byte with bad parity, the operation ends at that point; otherwise, it concludes automatically when the last digit of the master key has been entered.

If an 8-byte master key is entered into the 4778, it is duplicated as the second 8 bytes of the double-length master key in the 4778. If the dual-entry option is selected, the two entries are exclusive-ORed together to produce the master key.

You must enter the master key in 3-3-2 format (see "Converting a Master Key to Keypad-Entry Format" on page 5-36) at the 4778 keypad. You enter the master key in response to the informational message displayed on the screen when the 4778 arrow indicator is on. If the new master key has correct parity, the old key is replaced, and the triple encryption of the serial number with the new master key is returned and displayed. If the new master key has incorrect parity, an error code is returned and displayed, and the 4778 arrow indicator flashes. The information returned to you after you enter a master key is displayed on the screen until you press a workstation keyboard key.

The following list displays the number of keystrokes needed for each type of master key entry:

8-byte single-entry master key: 24 keypad strokes at the 4778
8-byte dual-entry master key: 48 keypad strokes (24 by each individual)
16-byte single-entry master key: 48 keypad strokes
16-byte dual-entry master key: 96 keypad strokes (48 by each individual)

Dual-Entry of the Master Key: Dual entry of the master key is a security measure. In theory, two people would each know one-half of the key but not the entire key. The two components of the key would be logically combined within the 4778 to form the entire master key.

Generating the Two Components of a Dual-Entry Master Key: Each component of the dual-entry master key has the form of a valid key. Each key is either 8 or 16 bytes long and has valid parity on every byte. The PIN keypad combines the two components with an exclusive-OR operation, then adjusts the resulting key to have correct parity before storing it.

Entering the Dual-Entry Master Key on the PIN Keypad: For an 8-byte dual-entry master key, each person enters 24 digits on the keypad. For a 16-byte key, each person enters 48 digits. In either case, the first person enters all keystrokes for the first component; then the second person enters all keystrokes for the second component.

Leaving the Enter Master Key Option: To leave the enter master key option after the PIN keypad is enabled, press the **Cancel** key on the workstation keyboard. (The **Cancel** key is customized at power-on time by the DEVICE statement in the CONFIG.SYS file.)

Encryption Key Option

This option lets you load encryption keys into the 4778 using your workstation keyboard. All encryption keys that you download must be triple-encrypted under a variant of the master key.

Warning: If you fail to download the keys triple-encrypted under the master key, unexpected keys result in the 4778 because the keys are decrypted at the 4778 using the master key.

You are prompted for the variant used to decrypt the key at the 4778. If you are not using variants, enter zero (**0** or **00**) in the variant offset prompt. You will also be prompted for the key offset. This information is used to determine the exact location where the key is stored in the 4778. All responses to the prompt must be in hexadecimal format. (If the key is to be stored in location 10, the response to the key offset prompt should be **A** or **0A**.)

If the encryption key has correct parity, the old key is replaced, and the encryption of the serial number with the new key is returned. If the encryption key has incorrect parity, an error code is returned and displayed. The information returned to you following the download of an encryption key is displayed on the screen until you press a workstation keyboard key.

To leave the load-encryption-key option, enter a null response to any of the enter data prompts. (Form a null response by pressing the **Enter** key on the workstation keyboard with no data entered.) The data prompts are for variant offset, key offset, and the actual key to be downloaded.

Exit Option

You exit the Load DES Key Menu and return to the main menu by pressing **F3**.

Load Initial Chaining Value Option

This option lets you load an initial chaining value (ICV) into the 4778 using the workstation interface. The ICV must be triple-encrypted under a variant of the master key.

Warning: Failure to download the ICV that is triple-encrypted under the master key results in an unexpected ICV in the 4778. An unexpected ICV occurs because the ICV is decrypted at the 4778 using the master key.

You are prompted for the variant to be used when you are decrypting the ICV. If you are not using variants, respond to the variant offset prompt by entering zero (**0** or **00**). All responses to the prompt must be in hexadecimal format.

After the 4778 receives the ICV, it sends a return code to the workstation. All error messages are displayed until you press a workstation keyboard key.

You leave the load ICV option by entering a null response to any of the enter data prompts. (A null response is formed by pressing the **Enter** key on the workstation keyboard with no data entered.) The data prompts are for variant offset and the encrypted form of the ICV that is to be downloaded.

Load PIN Verification Parameters Option

This option lets you load parameters for PIN verification and create PIN offset data. You are prompted for the length of the PIN to be checked; acceptable lengths range from 0H (1 digit checked) to FH (16 digits checked). The length of the PIN to be checked is a zero offset. For example, if you enter **0** in response to the length prompt, the 4778 will check 1 digit of the PIN. If you enter **F** in response to the length prompt, the 4778 will check 16 digits of the PIN. You are then prompted for the 16-byte decimalization table. All responses to the prompt must be in hexadecimal format.

After the 4778 receives the PIN verification parameters, it sends a return code to the workstation. All error messages remain displayed until you press a workstation keyboard key.

To leave this option, enter a null response to any of the enter data prompts. (A null response is formed by pressing the **Enter** key on the workstation keyboard with no data entered.) The data prompts are for the PIN length and the decimalization table to be downloaded.

Set Mode Option

This option lets you set the 4778 into either clear or encrypted mode.

Set Clear Mode Option

You use this option to set the 4778 into clear (nonencrypted) data mode. This mode lets the application program request unencrypted data from the 4778. Message authentication codes can be generated by the 4778 when in clear mode; however, a create PIN block request results in an error.

Set Encrypted Mode Option

You use this option to set the 4778 into encrypted data mode. This mode lets the application program request PIN blocks from the 4778. You can also generate message authentication codes using the 4778 in encrypted mode; however, requests for clear data result in an error.

Warning: Placing the 4778 into encrypted mode destroys all DES keys. You must reload all DES keys into the 4778, following the Set Encrypted Mode request.

Exit Option

You leave the Set Mode Menu and return to the main menu by pressing **F3**.

Read Serial Number Option

You use this option to read the serial number stored in the 4778. This option is supplied as a diagnostic because encryption of the serial number is returned from the 4778 on key management options.

Read Device-Driver Version Option

You use this option to read the device-driver version information from the 4778 PIN keypad driver. This information is an ASCII string in the following format:
`VERSION x.xx`

Exit Option

You leave the download utility and return to DOS by pressing **F3**.

Messages

The following messages can be generated by the download utility. These messages are in addition to the messages from the device driver.

Information Messages

Following successful entry or loading of a DES key, the encrypted result of the serial number is displayed on the screen. You should decrypt the information to verify that the key was loaded correctly. The encrypted serial number result will be displayed until you press a key on the workstation keyboard.

Warning: Placing the 4778 into encrypted mode destroys all DES keys. All DES keys must be reloaded into the 4778 following the Set Encrypted Mode request.

Error Messages

The possible error messages from the download utility are listed below. The user action item is intended for the download utility only. (Other applications handle error conditions differently.)

Invalid selection

Explanation: You selected an option that is not valid on one of the displayed menus.

User Response: Select a valid option from the displayed menu.

Invalid data

Explanation: A non-hexadecimal representation was entered in response to a data prompt.

User Response: Enter only hexadecimal representations (valid hexadecimal digits are 0 through 9 and A through F).

Insufficient data

Explanation: You did not enter enough data in response to a data prompt.

User Response: Supply the required data for the task.

Invalid command

Explanation: The device driver sent to the 4778 a command that was not recognized.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table. Also verify that correct versions of the device driver and download utility are running.

Check-sum error

Explanation: An unrecoverable communication error occurred in the transmitted message.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table.

Data not available

Explanation: The device driver requested data from the 4778 but no data was available at the PIN keypad.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table.

More data required

Explanation: You did not enter enough data in response to a requested function.

User Response: Enter data that meets the minimum requirement.

Too much data

Explanation: You entered too much data in response to a requested function.

User Response: Enter data that meets the maximum requirement.

Invalid keypad key pressed

Explanation: You pressed a key that was incorrect on the 4778 for the required function.

User Response: Press only keys 0 through 7 on the 4778 when you are entering a master key through the keypad.

DES key has incorrect parity

Explanation: The Data Encryption Standard (DES) key that was downloaded or entered into the 4778 had incorrect parity.

User Response: Correct the key and use correct parity.

PIN keypad write error

Explanation: The 4778 detected an internal error.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table.

No function pending

Explanation: The device driver sent an ABORT command to the 4778 but no function was active at the 4778.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table.

Stored DES key has incorrect parity

Explanation: The DES key stored at the 4778 has incorrect parity.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table.

Invalid in clear mode

Explanation: The requested function is not valid in clear mode.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table.

Invalid in encrypted mode

Explanation: The requested function is not valid in encrypted mode.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table.

Too many data frames received

Explanation: The 4778 received more data than it expected.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table.

Too few data frames received

Explanation: The 4778 expected more data than it received.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table.

Incorrect variant specified

Explanation: The variant requested is not valid for the requested function.

User Response: Specify a variant that is valid for the requested function.

Variant has incorrect parity

Explanation: The variant requested is stored in the 4778 with incorrect parity.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table.

Invalid command field

Explanation: The 4778 received an incorrect value.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table.

Invalid data length

Explanation: The download utility used a data length that was not valid.

User Response: Run the workstation and 4778 diagnostics, and follow the recommended symptom-action table.

Security Functions

The 4778 PIN keypad supports a set of enhanced security functions that let you use the PIN keypad to do cryptographic functions at your workstation. These functions include enhanced data encryption functions and cryptographic key management functions. For information about how to use the cryptography of the PIN keypad, see “Cryptographic Function Details” on page 5-33.

Security Function Calls

You call all the enhanced security functions by an IOCTL Write function call. You make the function call with a fixed data buffer of 32 bytes (CX = 0020H). The first byte of the Data Buffer must contain 05H, which indicates a security function request. The second data byte contains the number of the security function requested. The rest of the data buffer contains additional data required to perform the request as defined below. On return, the data buffer contains the response to the function call.

Note: In the buffer examples, bytes are in sequence from left to right and top to bottom.

All functions requiring keypad input are performed asynchronously if the device driver is in asynchronous mode. All other functions are performed synchronously, regardless of mode. The supported functions are as follows:

Command (Second Buffer Byte)	**Description**
00H	Read Status
01H	Read Serial Number
02H	Set PIN Keypad Mode
03H	Enter Master Key
04H	Load Master Key
05H	Load Key
06H	Load Initial Chaining Vector
07H	Reserved
08H	Load PIN Verification Parameters
09H	Create PIN Block
0AH	Verify PIN Block
0BH	Create PIN Offset Data
0CH	Generate Message Authentication Code
0DH	Verify Message Authentication Code
0EH	Reserved
0FH	LCD
10H–FFH	Reserved

Read Status

The Read Status function gets hardware status, Basic Assurance Test (BAT) data, or diagnostic information from the PIN keypad. The data buffer is configured as follows:

5H	H	xxH	xxH	aH	xxH	xxH	xxH
xxH	xxH	xxH	xxH	xxH	xxH	xxH	xxH
xxH	xxH	xxH	xxH	xxH	xxH	xxH	xxH
xxH	xxH	xxH	xxH	xxH	xxH	xxH	xxH

Note: Any box that contains xxH is irrelevant to the current operation.

Where **a** is the option for the operation requested, the values for byte 5 can be:

00H Return hardware test status.

01H Rerun BAT tests and return hardware test status.

02H Run keyboard test. The device will scan the keypad until you press each key from left to right in ascending sequence. The device then pads the data to 16 digits, encrypts the results using a fixed cryptographic key (1 1 1 1 1 1 1 1H), and returns the 8-byte result.

03H Return PIN microcode version information.

On return, the data buffer has the following configuration:

```
 5H    H  RCL  RCH   aH  xxH  xxH  xxH

xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

RD   RD1  RD2  RD3  RD4  RD5  RD6  RD7

RD8  RD9  RDA  RDB  RDC  RDD  xxH  xxH
```

Where:

RCL,RCH Security function return code

RD0–RDD From 1 to 14 response bytes

RD0 Option 1 or 2, One byte with bit significance:

bit 0 1 if one or more keys are closed
bit 1 1 if EEPROM check sum verification failed
bit 2 1 if random access memory (RAM) test failed
bits 3–6 Reserved for future use
bit 7 0 for clear mode; 1 for encrypted mode

RD0 Option 2, 8 bytes (`FEB7B9253F35EB D` if correct)

RD0–RDD Option 3, 14-byte ASCII string `"v.vv, mm/dd/yy"`, where `v.vv`=version, `mm`=month, `dd`=day, and `yy`=year

Read Serial Number

This function reads the serial number stored in the PIN keypad. The data buffer is configured as follows:

```
 5H   1H  xxH  xxH  xxH  xxH  xxH  xxH

xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH
```

On return, the data buffer has the following configuration:

```
 5H   1H  RCL  RCH  xxH  xxH  xxH  xxH

xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

RD   RD1  RD2  RD3  RD4  RD5  RD6  RD7
```

Where:

RCL,RCH Security function return code, refer to “Security Function Return Codes” on page 5-33

RD0–RD7 16 BCD digits representing the serial number (`477841 sssssssFF`):

4778 PIN keypad machine type
41 Plant of control

0sssssss Serial number
FF Serial number set flag

Set PIN Keypad Mode

This function sets the PIN keypad mode for either encrypted or nonencrypted operation.

Note: When you use this function to set the encrypted mode, all loaded encryption keys become invalid. Consequently, ensure that you reload all the encryption keys whenever you request the encryption mode.

When called, the data buffer is configured as follows:

```
  5H   2H  xxH  xxH  xmH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH
```

Where:

m Mode to be set:

- **0** Set nonencrypted mode
- **1** Set encrypted mode

On return, the data buffer has the following configuration:

```
  5H   2H  RCL  RCH   mH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH
```

Where:

RCL,RCH Security function return code
m Mode to be set:

- **0** Set nonencrypted mode
- **1** Set encrypted mode

Enter Master Key

This function lets you enter a master key into the 4778 by the using the 4778 keypad. The data is entered into the 4778 keypad in 3-3-2 format. To convert the keystrokes, use the “Hexadecimal-to-Keystroke Conversion” on page 5-36.

When called, the data buffer is configured as follows:

```
  5H   3H  xxH  xxH  tmH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH
```

Where:

t Type of key:

0000 If 16 bytes (48 keystrokes)
0001 If 8 bytes (24 keystrokes)

m Method of entry:

0000 If single entry
0001 If dual entry (two pieces to be exclusive ORed); see “Dual-Entry of the Master Key” on page 5-11

Note: If an 8-byte key is entered, it will be duplicated as the second 8 bytes of the double-length master key in the PIN keypad. For more information, see “Key Variant Description” on page 5-34.

On return, the data buffer has the following configuration:

```
  5H   3H  RCL  RCH  tmH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

 RD   RD1  RD2  RD3  RD4  RD5  RD6  RD7
```

Where:

RCL,RCH Security function return code

RD0–RD7 8 data bytes representing the triple encryption of the device serial number with the entered key

t Same as the calling value

m Same as the calling value

Load Master Key

Using the 4778 device driver, the Load Master Key function loads a new master key into the PIN keypad. When called, the data buffer is configured as follows:

```
  5H   4H  xxH  xxH  tmH  xxH  xxH  xxH

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 KD8  KD9  KDA  KDB  KDC  KDD  KDE  KDF

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH
```

Where:

t Type of key:

0000 If 16 bytes
0001 If 8 bytes

m Method of entry:

0000 If clear
0001 If triple-encrypted under resident key

KD0–KDF Key data bytes (8 or 16 bytes)

Note: If an 8-byte key is entered, it is duplicated as the second 8 bytes of the double-length master key in the PIN keypad. For more information, see "Key Variant Description" on page 5-34.

On return, the data buffer has the following configuration:

```
  5H   4H  RCL  RCH  tmH  xxH  xxH  xxH

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 KD8  KD9  KDA  KDB  KDC  KDD  KDE  KDF

 RD   RD1  RD2  RD3  RD4  RD5  RD6  RD7
```

Where:

RCL,RCH Security function return code

RD0–RD7 8 data bytes representing the triple encryption of the device serial number with the loaded key

t Same as calling value

m Same as calling value

Load Key

The Load Key function loads keys into the PIN keypad Security Processor nonvolatile storage. The keys are passed from the application to the PIN keypad in triple-encrypted form, using a suitable variant of the master key. The keys are decrypted to check for valid parity but are stored in encrypted form for later use. When called, the data buffer must be configured as follows:

```
  5H   5H  xxH  xxH  kkH  vvH  xxH  xxH

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH
```

Where:

kk Key identifier (index number = 00 to FF)

vv Variant table descriptor:

- **00** For variants not used
- **03–06** If you are using a variant (depending on the key function)

KD0–KD7 Key data bytes

On return, the data buffer has the following configuration:

```
  5H   5H  RCL  RCH  kkH  vvH  xxH  xxH

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

 RD   RD1  RD2  RD3  RD4  RD5  RD6  RD7
```

Where:

RCL,RCH Security function return code

RD0–RD7 8 data bytes that represent the encryption of the device serial number with the loaded key

kk Same as calling value

vv Same as calling value

Load ICV

The Load ICV function stores an initial chaining vector (ICV) that is internal to the PIN keypad Security Processor. The ICV, which can be used in the Generate MAC or Verify MAC function, is an 8-byte quantity. It is passed from the computer, in triple-encrypted form, using a variant of the master key. When called, the data buffer must be configured as follows:

```
  5H   6H  xxH  xxH  xxH  vvH  xxH  xxH

 ID   ID1  ID2  ID3  ID4  ID5  ID6  ID7

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH
```

Where:

vv Variant table descriptor:

00 For variants not used
02 If you are using a variant

ID0–ID7 8-byte initial chaining vector, triple encrypted under the indicated variant of the master key

On return, the data buffer has the following configuration:

```
  5H   6H  RCL  RCH  xxH  vvH  xxH  xxH

 ID   ID1  ID2  ID3  ID4  ID5  ID6  ID7

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH
```

Where:

RCL,RCH Security function return code

vv Same as calling value

Load PIN Verification Parameters

This function loads the parameters needed by the PIN keypad to verify PINs or create PIN offset data. After the parameters are loaded, they are stored in nonvolatile storage and used until the command is reissued. When called, the data buffer must be configured as follows:

```
  5H   8H  xxH  xxH   pH  xxH  xxH  xxH

 DD   DD1  DD2  DD3  DD4  DD5  DD6  DD7

 DD8  DD9  DDA  DDB  DDC  DDD  DDE  DDF

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH
```

Where:

p Length of PIN to be checked: 0H–FH (1–16)

DD0–DDF Decimalization table

Note: The PIN Check Length (p) is a zero-based parameter. For example, if you specify that p=0H, one digit of the PIN will be verified. If you specify that p=FH, 16 digits will be verified.

On return, the data buffer has the following configuration:

```
  5H   8H  RCL  RCH   pH  xxH  xxH  xxH

 DD   DD1  DD2  DD3  DD4  DD5  DD6  DD7

 DD8  DD9  DDA  DDB  DDC  DDD  DDE  DDF

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH
```

Where:

RCL,RCH Security function return code

p Same as calling value

Create PIN Block

This function is used to encrypt a personal identification number (PIN) entered in the keypad using the key and format indicated by the calling data. You can use this function only while the PIN keypad is in encrypted mode (see “Set PIN Keypad Mode” on page 5-20). When called, the data buffer is configured as follows:

```
  5H   9H  xxH  xxH   kH  vvH  faH  xxH

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 PD   PD1  PD2  PD3  PD4  PD5  xxH  xxH

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH
```

Where:

k Key format:

0000 Master key (first 8 bytes)
0001 1-byte internal key, short pointer
0011 8-byte key data field

vv Variant table descriptor:

00 Variants not used
03 Use fixed variant 03

KD0–KD7 1-byte key pointer (KD0) or 8-byte key encrypted under the suitable variant of the master key (KD0–KD7)

f PIN block format:

0000 IBM 4704
0001 ANSI X9.8
0010 IBM 3624

a Pad character (0–F)

PD0–PD5 12-digit personal account number (PAN) in BCD format

On return, the data buffer has the following configuration:

```
  5H   9H  RCL  RCH   kH  vvH  faH  xxH

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 PD   PD1  PD2  PD3  PD4  PD5  xxH  xxH

 RD   RD1  RD2  RD3  RD4  RD5  RD6  RD7
```

Where:

RCL,RCH Security function return code

RD0–RD7 Encrypted PIN block

k Same as calling value

vv Same as calling value

KD0–KD7 Same as calling value

f Same as calling value

a Same as calling value

PD0–PD5 Same as calling value

Verify PIN

This function uses PIN offset and validation data that is read from a magnetic-stripe card to verify PIN entries. The PIN keypad encrypts the data and compares it to the data passed with the command according to the IBM 3624 algorithm. (See "Verifying 3624 PINs" on page 5-41.) When called, the data buffer is configured as follows:

```
  5H   AH  xxH  xxH  akH  vvH  xxH  xxH

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 VD   VD1  VD2  VD3  VD4  VD5  VD6  VD7

 OD   OD1  OD2  OD3  OD4  OD5  OD6  OD7
```

Where:

a 0 = offset data not present; 1 = offset data present

k Key format:

0000 Master key (first 8 bytes)
0001 1-byte internal key, short pointer
0011 8-byte key data field

vv Variant table descriptor:

00 Variants not used
04 Use fixed variant 04

KD0–KD7 1-byte key pointer (KD0) or 8-byte key encrypted under the suitable variant of the master key (KD0–KD7)

VD0–VD7 Verification data read from the magnetic card

OD0–OD7 Offset data read from the magnetic card (if required)

On return, the data buffer has the following configuration:

```
  5H   AH  RCL  RCH  akH  vvH  xxH  RD

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 VD   VD1  VD2  VD3  VD4  VD5  VD6  VD7

 OD   OD1  OD2  OD3  OD4  OD5  OD6  OD7
```

Where:

RCL,RCH Security function return code

RD0 Response data byte:

00 PIN checks
FF PIN not valid

a Same as calling value

k Same as calling value

vv Same as calling value

KD0–KD7 Same as calling value

VD0–VD7 Same as calling value

OD0–O07 Same as calling value

Create PIN Offset Data

This function is used to generate a PIN Offset parameter that is used to verify PINs using the IBM 3624 algorithm. (See “Verifying 3624 PINs” on page 5-41.) The PIN keypad computes the offset by combining validation data with the PIN (which has been entered through the keypad). When called, the data buffer is configured as follows:

```
  5H   BH  xxH  xxH   kH  vvH  xxH  xxH

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 VD   VD1  VD2  VD3  VD4  VD5  VD6  VD7

 xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH
```

Where:

k Key format:

0000 Master key (first 8 bytes)
0001 1-byte internal key, short pointer
0011 8-byte key data field

vv Variant table descriptor:

00 00 for variants not used
04 04 use fixed variant 04

KD0–KD7 1-byte key pointer (KD0) or 8-byte key encrypted under the suitable variant of the master key (KD0–KD7)

VD0–VD7 Validation data to be used in creating the offset

On return, the data buffer has the following configuration:

```
  5H   BH  RCL  RCH   kH  vvH  xxH  xxH

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 VD   VD1  VD2  VD3  VD4  VD5  VD6  VD7

 OD   OD1  OD2  OD3  OD4  OD5  OD6  OD7
```

Where:

RCL,RCH Security function return code

OD0–OD7 Offset data up to 16 binary coded decimal (BCD) digits, padded with Fs for less than 16 digits checked

k Same as calling value

vv Same as calling value

KD0–KD7 Same as calling value

VD0–VD7 Same as calling value

Generate Message Authentication Code

This function generates a message authentication code on a data string of up to 65 528 (64KB less 8) bytes long. When called, the data buffer is configured as follows:

```
  5H   CH  xxH  xxH  IkH  VaH  VbH  xxH

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 ID   ID1  ID2  ID3  ID4  ID5  ID6  ID7

 OFL  OFH  SGL  SGH  LNL  LNH  xxH  xxH
```

Note: Enter this data as a multiple of eight.

Where:

I ICV format:

0100 Internal ICV
0110 ICV in data

k Key format:

0000 Master key (1st 8 bytes)
0001 1-byte internal key, short pointer
0011 8-byte key data field

va Variant table descriptor:

00 Variants not used
06 Use fixed variant 06 to decrypt data key

vb ICV variant table descriptor:

00 Variants not used
02 Use fixed variant 02 to decrypt ICV data key

KD0–KD7 1-byte key pointer (KD0) or 8-byte key encrypted under the suitable variant of the master key (KD0–KD7)

ID0–ID7 ICV encrypted under the appropriate variant of the master key (if required)

OFL,OFH Offset of data address

SGL,SGH Segment of data address

LNL,LNH Length of data (maximum of FFF8H)

On return, the data buffer has the following configuration:

```
  5H   CH  RCL  RCH  IkH  VaH  VbH  xxH

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 ID   ID1  ID2  ID3  ID4  ID5  ID6  ID7

 RD   RD1  RD2  RD3  RD4  RD5  RD6  RD7
```

Where:

RCL,RCH Security function return code

RD0–RD7 8-byte message authentication code (for systems requiring only 4 bytes, the application selects the leftmost 4 bytes to be retained)

I Same as calling value

k Same as calling value

Va, Vb Same as calling value

KD0–KD7 Same as calling value

ID0–ID7 Same as calling value

Verify Message Authentication Code

This function verifies a message authentication code (MAC). The MAC to be verified is the last 4 or 8 bytes of the data passed. If only 4 bytes are passed, they are compared with the 4 most significant bytes of the MAC, which is internally computed on the preceding data bytes. The maximum number of bytes (including the MAC) that can be passed is 65 532 (64KB less 4) bytes. When called, the data buffer is configured as follows:

```
  5H   DH  xxH  xxH  IkH  VaH  VbH  xxH

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 ID   ID1  ID2  ID3  ID4  ID5  ID6  ID7

 OFL  OFH  SGL  SGH  LNL  LNH  xxH  xxH
```

Note: Enter this data as a multiple of four.

Where:

I ICV format:

0100 Internal ICV
0110 ICV in data

k Key format:

0000 Master key (first 8 bytes)
0001 1-byte internal key, short pointer
0011 8-byte key data field

va Variant table descriptor:

00 Variants not used
06 Use fixed variant 06 to decrypt data key

vb ICV Variant table descriptor:

00 Variants not used
02 Use fixed variant 02 to decrypt ICV data key

KD0–KD7 1-byte key pointer (KD0) or 8-byte key encrypted under the suitable variant of the master key (KD0–KD7)

ID0–ID7 ICV encrypted under the appropriate variant of the master key (if required)

OFL,OFH Offset of data address

SGL,SGH Segment of data address

LNL,LNH Length of data (maximum of FFF8H)

On return, the data buffer has the following configuration:

```
  5H   DH  RCL  RCH  IkH  VaH  VbH  RD

 KD   KD1  KD2  KD3  KD4  KD5  KD6  KD7

 ID   ID1  ID2  ID3  ID4  ID5  ID6  ID7

 OFL  OFH  SGL  SGH  LNL  LNH  xxH  xxH
```

Where:

RCL,RCH Security function return code

RD0 Response data byte:

00 MAC checks
FF MAC not valid

OFL,OFH Same as calling value

SGL,SGH Same as calling value

LNL,LNH Same as calling value

Va, Vb Same as calling value

Writing to the Display

This function lets your application write up to 16 characters to the 4778 display. When called, the data buffer is configured as follows:

```
 5H   FH  xxH  xxH  xxH  xxH  xxH  xxH

xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

OFL  OFH  SGL  SGH  LNL  LNH  xxH  xxH
```

Where:

OFL,OFH Offset of data to be displayed

SGL,SGH Segment of data to be displayed

LNL,LNH Length of data to be displayed

On return, the data buffer has the following configuration:

```
 5H   FH  RCL  RCH  xxH  xxH  xxH  xxH

xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

xxH  xxH  xxH  xxH  xxH  xxH  xxH  xxH

OFL  OFH  SGL  SGH  LNL  LNH  xxH  xxH
```

Where:

OFL,OFH Same as calling value

SGL,SGH Same as calling value

LNL,LNH Same as calling value

RCL,RCH Security function return code

Security Function Return Codes

The following hexadecimal return codes can be issued by the 4778 Security function:

Code	**Description**
00	Complete (no error)
01	Command not valid
02	Check sum mismatch
03	Data not available
04	More data required to process function
05	Too much data passed
06	Key pressed is not valid
07	Encryption key has bad parity
08	PIN keypad write error
09	Tried to abort with no function pending
0A	Bad parity on EEPROM stored key
0B	Command not valid in clear mode
0C	Command not valid in encrypted mode
0D	Too many data frames received
0E	Too few data frames received
0F	Incorrect variant for command
10	Variant parity is not valid
11	Field not valid for the command
12	Data length of MAC functions not valid
13	Invalid display data length
14	Truncated display data

Cryptographic Function Details

This information describes in detail the cryptographic functions supported by the 4778 PIN keypad and how you use them.

To use cryptography with a 4700 Finance Communication System, you should be familiar with *4700 Finance Communication System, Controller Programming Library, Volume 5: Cryptographic Programming*.

Because the Data Encryption Algorithm (DEA) is in the public domain, the security of the functions of the 4778 PIN keypad that use the DEA depends on the security of the key that is used in processing the algorithm. Therefore, after you load or enter cryptographic keys into the keypad, the keys cannot be read and are stored in nonvolatile EEPROM memory. This memory resides in a tamper-resistant security processor. Provisions for loading keys into the PIN keypad have been made so that you can design a secure method for handling your keys when you are isolated from the PIN keypad. You should randomly generate your keys and store and distribute them in a secure, controlled, and auditable manner.

Loading Keys

The first key you load into the keypad is the master key. You must load (or enter) the master key into the keypad before any cryptographic operations can become valid. This is the only key loaded into the keypad in Clear (nonencrypted) form. For protection purposes, this key should be 128 bits (16 bytes) long. However, for compatibility with the existing 4704 encrypting PIN keypad feature, provisions have been made so that you can load a 64-bit (8-byte) master key. When you load an 8-byte master key, the 8 bytes are duplicated. This ensures that a full 16 bytes are available for key-management functions. These functions can then use your master key.

Triple-Encrypted Keys: After you have loaded the master key, you can load additional 8-byte keys into the keypad (if you desire). You load these keys *triple-encrypted* under the master key or variant of the master key (for an explanation of variants, see “Key Variant Description”). Triple encryption is a cryptographic process in which you first encrypt the 8 bytes of data with the first 8 bytes of a double-length key. You then decrypt the result with the second 8 bytes of the double-length key. After that, encrypt the result again, using the first 8 bytes of the double-length key. If you use the same 8 bytes for the encryption and decryption steps (for an 8-byte master key), the final result will be the same as if a single encryption step were performed with a single-length (8-byte) key.

The 4778 PIN keypad can store 256 keys (this is in addition to the master key). These keys are stored in the EEPROM and triple encrypted under the appropriate master key variant until they are used.

Key Variant Description

A variant of a cryptographic key is a new key that is formed by combining the original key with a nonsecret quantity. In the 4778, the nonsecret quantity is called a Variant Descriptor Byte (VDB). Each byte of the original key is combined in an exclusive-OR operation with the VDB to produce the new key. For example:

```
Original key:       1 23 45 67 89 AB CD EF    (hex)
VDB:              55

Exclusive-OR the VDB with each byte of the original key
to obtain the variant key.

                  1 23 45 67 89 AB CD EF
            XOR  55 55 55 55 55 55 55 55

Variant key:       54 76 1  32 DC FE 98 BA
```

The 4778 contains a fixed table of Variant Definition Bytes. The table, which is shown below, is organized as 16 sets of 4 VDBs. Only 6 variant bytes are defined, corresponding to the six PIN-keypad commands that require them. Whenever you use a variant with a command, you must specify a Variant Descriptor that designates which of the VDB sets to use.

Figure 5-1. Key Variants

Index	Variant a	Variant b	Variant c	Variant d
1	X'12'	X'00'	X'00'	X'00'
2	X'90'	X'00'	X'00'	X'00'
3	X'06'	X'00'	X'00'	X'00'
4	X'2E'	X'00'	X'00'	X'00'
5	X'44'	X'00'	X'00'	X'00'
6	X'82'	X'00'	X'00'	X'00'
7–16	X'00'	X'00'	X'00'	X'00'

Note: X'00' is a special case. It indicates that variants are not to be used at all. This is equivalent to using a variant of X'00'.

Only certain variants can be used with each command:

Command	Variant Use
Load Key	Va3, Va4, Va5, or Va6 are used to decrypt the key being loaded.
Load ICV	Va2 is used to decrypt the ICV being loaded.
Create PIN Block	Va3 is used to decrypt the PIN key.
Verify PIN	Va4 is used to decrypt the PIN verification key.
Generate MAC	Va5 is used to decrypt the MAC key.
Verify MAC	Va6 is used to decrypt the MAC key.

Variants ensure that a key can be used only for its intended function. For example, a security problem could result if a MAC verification key could also be used for the MAC generation function. The manner in which the keys are stored in the PIN keypad prevents this problem. Each key is stored in encrypted form (encrypted under a variant of the master key by using one of the Variant Descriptor Bytes). The VDB is specified with the Load Key command. When used for the intended function, the key is decrypted using the correct variant of the master key (and is successfully recovered). If used for a different function, the wrong variant of the master key is used, resulting in an incorrect key.

Converting a Master Key to Keypad-Entry Format

Master keys are generated as either 8-byte or 16-byte values. If you are going to enter the key into the PIN keypad with the Enter Master Key command, you must first convert the key to a format containing only digits 0 through 9. In this process, convert each byte of the key into three keystrokes. This means that an 8-byte key is converted to 24 keystrokes and a 16-byte key is converted to 48 keystrokes.

To express the key in the form of keystrokes, write the key in hexadecimal, then use the table in Figure 5-2 on page 5-37 to convert each pair of hexadecimal digits to three-keystroke values.

For example:

```
      73 A  11 C3 8  6F CE 22       Key




      Convert each pair,
      using the hexadecimal
      to keystroke tables.



   3 4 3
      5   1
          4
           6   2
             4
                3 3 2
                  6 3 2
                     1   3

Enter this sequence on the
keypad     3 4 3 5   1   4   6   2 4     3 3 2 6 3 2 1   3
```

Hexadecimal-to-Keystroke Conversion

When you generate keys by a random process, any hexadecimal character is possible. Because the keypad has only decimal characters, you must translate the Enter Master Key command bytes (which appear as two hexadecimal characters) into 3-3-2 decimal format. To do this, use the table in Figure 5-2 on page 5-37.

When you read the table, you will notice that multiple hexadecimal bytes result in identical decimal input (00H and 01H both result in 001D). This is because each byte is required to have odd parity (the parity bit is the least-significant bit). This means that if you entered a byte such as 000D, a parity error would occur. If keys are generated by an automatic process, the process corrects parity so identical keys of correct parity reside at all nodes.

Note: Do not use the conversion table for routine data conversions to and from 3-3-2 format. That is because parity is accounted for in the table. Use the table only for encryption and key and keystroke generation.

1H		1	21H	1			41H	2			61H	3		1
2H		2	22H	1		3	42H	2		3	62H	3		2
3H		2	23H	1		3	43H	2		3	63H	3		2
4H	1		24H	1	1	1	44H	2	1	1	64H	3	1	
5H	1		25H	1	1	1	45H	2	1	1	65H	3	1	
6H	1	3	26H	1	1	2	46H	2	1	2	66H	3	1	3
7H	1	3	27H	1	1	2	47H	2	1	2	67H	3	1	3
8H	2		28H	1	2	1	48H	2	2	1	68H	3	2	
9H	2		29H	1	2	1	49H	2	2	1	69H	3	2	
AH	2	3	2AH	1	2	2	4AH	2	2	2	6AH	3	2	3
BH	2	3	2BH	1	2	2	4BH	2	2	2	6BH	3	2	3
CH	3	1	2CH	1	3		4CH	2	3		6CH	3	3	1
DH	3	1	2DH	1	3		4DH	2	3		6DH	3	3	1
EH	3	2	2EH	1	3	3	4EH	2	3	3	6EH	3	3	2
FH	3	2	2FH	1	3	3	4FH	2	3	3	6FH	3	3	2
1 H	4		3 H	1	4	1	5 H	2	4	1	7 H	3	4	
11H	4		31H	1	4	1	51H	2	4	1	71H	3	4	
12H	4	3	32H	1	4	2	52H	2	4	2	72H	3	4	3
13H	4	3	33H	1	4	2	53H	2	4	2	73H	3	4	3
14H	5	1	34H	1	5		54H	2	5		74H	3	5	1
15H	5	1	35H	1	5		55H	2	5		75H	3	5	1
16H	5	2	36H	1	5	3	56H	2	5	3	76H	3	5	2
17H	5	2	37H	1	5	3	57H	2	5	3	77H	3	5	2
18H	6	1	38H	1	6		58H	2	6		78H	3	6	1
19H	6	1	39H	1	6		59H	2	6		79H	3	6	1
1AH	6	2	3AH	1	6	3	5AH	2	6	3	7AH	3	6	2
1BH	6	2	3BH	1	6	3	5BH	2	6	3	7BH	3	6	2
1CH	7		3CH	1	7	1	5CH	2	7	1	7CH	3	7	
1DH	7		3DH	1	7	1	5DH	2	7	1	7DH	3	7	
1EH	7	3	3EH	1	7	2	5EH	2	7	2	7EH	3	7	3
1FH	7	3	3FH	1	7	2	5FH	2	7	2	7FH	3	7	3

Figure 5-2 (Part 1 of 2). Hexadecimal-to-Keystroke Conversion Table

8 H	4			A H	5		1	C H	6		1	E H	7		
81H	4			A1H	5		1	C1H	6		1	E1H	7		
82H	4		3	A2H	5		2	C2H	6		2	E2H	7		3
83H	4		3	A3H	5		2	C3H	6		2	E3H	7		3
84H	4	1	1	A4H	5	1		C4H	6	1		E4H	7	1	1
85H	4	1	1	A5H	5	1		C5H	6	1		E5H	7	1	1
86H	4	1	2	A6H	5	1	3	C6H	6	1	3	E6H	7	1	2
87H	4	1	2	A7H	5	1	3	C7H	6	1	3	E7H	7	1	2
88H	4	2	1	A8H	5	2		C8H	6	2		E8H	7	2	1
89H	4	2	1	A9H	5	2		C9H	6	2		E9H	7	2	1
8AH	4	2	2	AAH	5	2	3	CAH	6	2	3	EAH	7	2	2
8BH	4	2	2	ABH	5	2	3	CBH	6	2	3	EBH	7	2	2
8CH	4	3		ACH	5	3	1	CCH	6	3	1	ECH	7	3	
8DH	4	3		ADH	5	3	1	CDH	6	3	1	EDH	7	3	
8EH	4	3	3	AEH	5	3	2	CEH	6	3	2	EEH	7	3	3
8FH	4	3	3	AFH	5	3	2	CFH	6	3	2	EFH	7	3	3
9 H	4	4	1	B H	5	4		D H	6	4		F H	7	4	1
91H	4	4	1	B1H	5	4		D1H	6	4		F1H	7	4	1
92H	4	4	2	B2H	5	4	3	D2H	6	4	3	F2H	7	4	2
93H	4	4	2	B3H	5	4	3	D3H	6	4	3	F3H	7	4	2
94H	4	5		B4H	5	5	1	D4H	6	5	1	F4H	7	5	
95H	4	5		B5H	5	5	1	D5H	6	5	1	F5H	7	5	
96H	4	5	3	B6H	5	5	2	D6H	6	5	2	F6H	7	5	3
97H	4	5	3	B7H	5	5	2	D7H	6	5	2	F7H	7	5	3
98H	4	6		B8H	5	6	1	D8H	6	6	1	F8H	7	6	
99H	4	6		B9H	5	6	1	D9H	6	6	1	F9H	7	6	
9AH	4	6	3	BAH	5	6	2	DAH	6	6	2	FAH	7	6	3
9BH	4	6	3	BBH	5	6	2	DBH	6	6	2	FBH	7	6	3
9CH	4	7	1	BCH	5	7		DCH	6	7		FCH	7	7	1
9DH	4	7	1	BDH	5	7		DDH	6	7		FDH	7	7	1
9EH	4	7	2	BEH	5	7	3	DEH	6	7	3	FEH	7	7	2
9FH	4	7	2	BFH	5	7	3	DFH	6	7	3	FFH	7	7	2

Figure 5-2 (Part 2 of 2). Hexadecimal-to-Keystroke Conversion Table

PIN Formats

The 4778 supports three different formats of encrypted PINs. These formats are the 4704 EPP format, 3624 PIN format, and ANSI 9.8 PIN format.

Figure 5-3 describes the 4704 EPP PIN format.
Figure 5-4 describes the 3624 PIN format.
Figure 5-5 on page 5-40 describes the ANSI 9.8 PIN format.

```
                  8 Bytes

           LEN PIN .......PAD SEQ

           A    123456789 FFF

             encipher using Kp
```

LEN Number of PIN characters entered; a 4-bit value from 1H to DH

PIN From 1 to 13 PIN characters; each is a 4-bit value from 0H to 9H

PAD From 0 to 12 pad characters (13 minus the number of PIN characters); each is a 4-bit value, always FH

SEQ A 1-byte sequence number, from 00H to FFH

Figure 5-3. 4704 EPP Format

```
                  8 bytes

           PIN.................PAD

            1 23 45 6E EE EE EE EE

              encipher using Kp
```

PIN From 1 to 16 PIN characters; each is a 4-bit value from 0H to 9H

PAD From 0 to 15 pad characters (16 minus the number of PIN characters); each is a 4-bit value, always 0H to FH; all must be the same

Figure 5-4. 3624 PIN Format

```
          8 bytes

     LEN  PIN.........PAD   Plaintext PIN


          PAN............   Primary Account Number


     encipher using Kp

Example:
   6 12 34 56 FF FF FF FF   (Customer PIN: 123456)


        22 23 33 44 45 55   (PAN: 111 222 333 444 555)


   6 12 16 75 CC BB BA AA   Formatted PIN (PIN XOR PAN)
```

0 A 4-bit control field; always 0H.

LEN Number of PIN characters entered; a 4-bit value from 4H to CH.

PIN From 4 to 12 PIN characters; each is a 4-bit value from 0H to 9H.

PAD From 2 to 10 keypad characters (14 minus the number of PIN characters). Each 4-bit character must be set to FH.

0000 A 2-byte field; always 0000H.

PAN Twelve 4-bit digits representing the rightmost 12 digits of the primary account number.

XOR An exclusive-OR of the plaintext PIN and the PAN yields the formatted PIN.

Figure 5-5. ANSI 9.8 PIN Format

Verifying 3624 PINs

The PIN verification process compares the PIN entered by a consumer with the encrypted validation data on the consumer's identification card. This determines whether the consumer entered the correct PIN. The verification algorithm used in the 4778 PIN keypad is identical to that used in the IBM 3624 Consumer Transaction Facility and other IBM products.

To verify PINs, the PIN keypad requires the following information:

Validation Data: Data on the consumer's card that is compared to the consumer's PIN.

EPINKEY: Key used to encrypt the validation data.

Offset Data: Optional data required if random or consumer-selected PINs are used.

DECTAB: Decimalization table used to translate hexadecimal numbers to decimal numbers. DECTAB is used to compare PIN digits.

PINMINL: The number of PIN digits to be checked by the EPP.

The method used to verify PINs is shown in Figure 5-6 on page 5-42 and works as follows:

1. The validation data is read from the identification card, padded to 8 bytes (16 digits) if required, and passed to the EPP along with the EPINKEY location (or encrypted key) and PINMINL.
2. The validation data is encrypted with EPINKEY and converted to decimal using the DECTAB.
3. The customer's PIN input is read from the 4778 PIN keypad.
4. The *n* leftmost characters (*n* = length of entered PIN) of the validation data (in decimal format) form the intermediate PIN.
5. The *m* rightmost digits (*m* = PINMINL) of the intermediate PIN are added to the offset data (if used) modulo 10 (without carry) to form the PIN check number.
6. The *m* rightmost digits of the entered PIN are compared with the PIN check number. The results of the comparison are returned to the application.

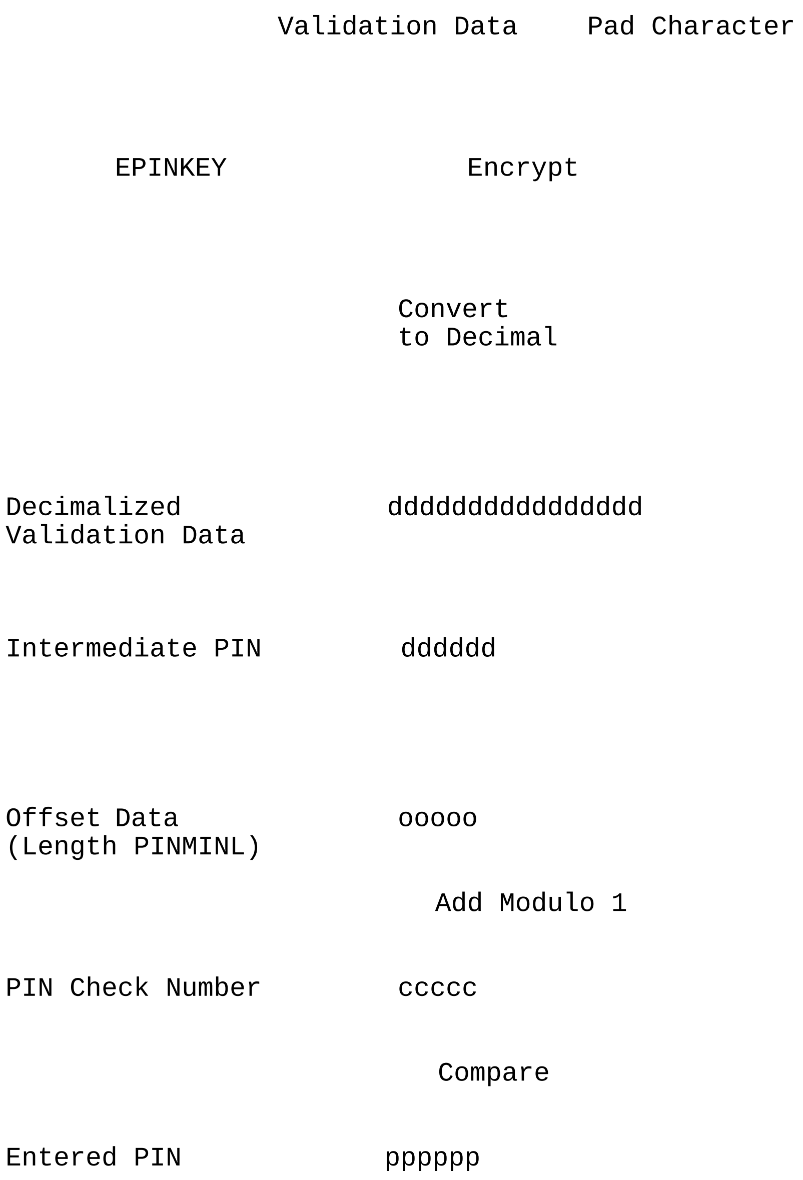

Figure 5-6. Verifying the 3624 PINs

When you create PIN offset data for the magnetic stripe of a new consumer, or when a consumer wants to change a PIN, you use the same algorithm. The only difference is that, instead of adding the offset data modulo 10 to the intermediate PIN to compute the PIN check number, the entered PIN subtracts modulo 10 from the intermediate PIN to compute the offset data to place on the consumer's card.

Using Message Authentication Codes

The PIN keypad produces a message authentication code (MAC) following the conventions defined in ANSI X9.9, Financial Institution Message Authentication. A MAC ensures data integrity when a message is transmitted from one node to another by an unprotected communication link. The MAC is generated at the sending node and is sent along with the message. At the other end, the MAC is verified to ensure that it is the same as the MAC transmitted by the sending node.

If the MAC does not verify, you can assume that some of the data was either intentionally or unintentionally changed. The algorithm is applied, as shown in Figure 5-7, either to the whole body of the message or to specific authentication elements presented to the PIN keypad by the Personal System/2 computer. The data must be 8 bytes (no padding or element extraction is provided by either the PIN keypad or the device driver). The algorithm uses the Cipher Block Chaining mode of the Data Encryption Standard (DES).

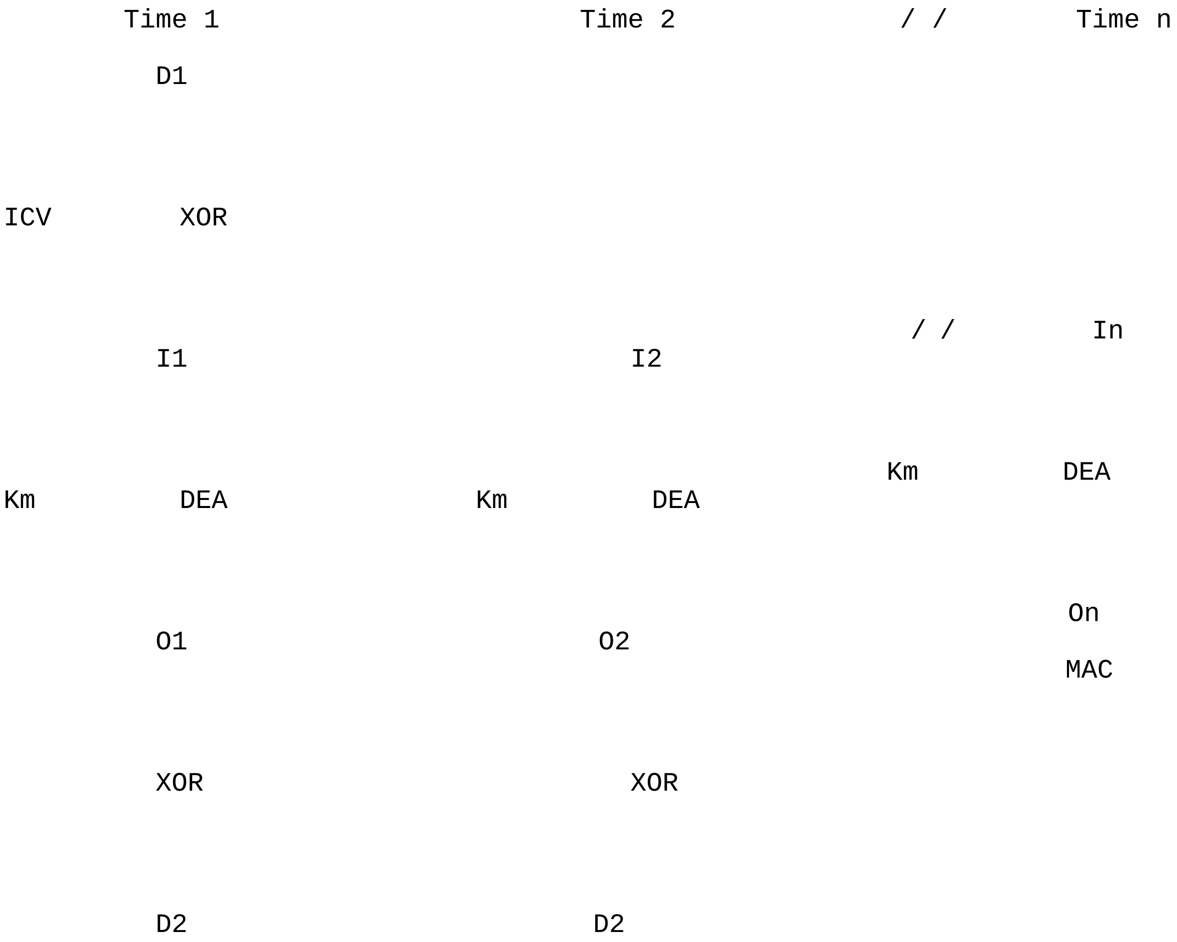

Figure 5-7. Using Message Authentication Codes

Chapter 6. Problem Determination Procedures

This chapter provides information to help you diagnose a problem and determine the source of a 4777 or 4778 problem.

Use the Symptom-Action Table to find the information that most accurately describes the symptom or problem.

Follow the steps in the Action column of the table to solve the problem.

If you cannot find a description of your problem, use the procedure described in “Any other symptoms.”

Symptom-Action Table

Symptom	Action
56xx messages	Record the message number. For additional information about the message, see “Messages” on page 6-2.
4777 Green and yellow lights are on at the same time.	Replace the 4777 Magnetic Stripe Unit.
Magnetic-stripe-unit problems	Ensure that the device is connected correctly. Ensure that the correct device driver is specified and available on your workstation. Ensure that you are using the correct magnetic-stripe document and that the document is not damaged.
4777 red light is on.	A data error occurred. You will be prompted to try again.
The 4778 display green indication is blinking or the 4778 does not beep **during the power on sequence.**	Replace the 4778.
Any other symptoms	Ensure that the 4777 is connected correctly. Ensure that the correct device driver is specified and available on the system disk of your workstation. Ensure that you are using the correct procedures.

Messages

5600

Explanation: The I/O test is complete.

User Response: None

4778 PIN Pad

5611

Explanation: The 4778 PIN-Pad MSR failed.

User Response: Run the test again. If you still get the error, replace the 4778.

5612

Explanation: The requested COMM port is either not installed or is not available for use by the 4778 device driver.

User Response: Verify that the COMM port is installed and that no other device is using that port.

5614

Explanation: The 4778 is not connected to the system unit.

User Response:

1. Ensure that the unit is connected to your workstation.
2. If the error continues, connect the 4778 directly to the serial port on your workstation.
3. If you still get an error, replace the 4778. If you do not get an error, the RS-232 cable, T-connector, or the 4778 is defective.

5615

Explanation: The 4778 driver has an input parameter error.

User Response: Correct the DEVICE statement in your CONFIG.SYS file. For information about the DEVICE statement, see the programming guide for your operating system.

5617

Explanation: The 4778 power-on test failed.

User Response: Replace the 4778.

5618

Explanation: The 4778 has a communication error.

User Response: Replace the 4778.

4777 MSRE

5621

Explanation: The 4777 Magnetic Stripe Unit failed.

User Response: Replace the 4777 Magnetic Stripe Unit.

5622

Explanation: The communication port that you requested is not available, or the 4777 Magnetic Stripe Unit is not attached to the system unit.

User Response: Verify that the unit is attached and that the communication port is not being used.

5623

Explanation: The 4777 Magnetic Stripe Unit is not attached to the system unit.

User Response:

1. Ensure that the unit is connected to your workstation.
2. If the error continues, connect the 4777 Magnetic Stripe Unit directly to the serial port on your workstation (bypassing the special connector).
3. If you still get an error, replace the 4777 Magnetic Stripe Unit. If you do not get an error and you are using a special connector to attach two devices, either the connector or the other device is defective.

5624

Explanation: The 4777 Magnetic Stripe Unit driver has an input parameter error.

User Response: Correct the DEVICE statement on your workstation system disk.

5627

Explanation: The 4777 Magnetic Stripe Unit self-test failed.

User Response: Replace the 4777 Magnetic Stripe Unit.

4778 MSR

5631

Explanation: The magnetic-stripe unit failed.

User Response: Replace the magnetic-stripe unit.

5632

Explanation: The requested COMM port is either not installed or is not available for use by the 4778 device driver.

User Response: Verify that the COMM port is installed and that no other device is using this port.

5633

Explanation: The magnetic-stripe unit is not attached to the system unit.

User Response:

1. Ensure that the unit is connected to your workstation.
2. If the error continues, connect the 4778 directly to the serial port on your workstation (bypassing the special connector).
3. If you still get an error, replace the unit. If you do not get an error and you are using the special connector to attach two devices, the connector or the other device is defective.

5634

Explanation: The magnetic-stripe unit driver has an input parameter error.

User Response: Correct the DEVICE statement in your CONFIG.SYS file, see Chapter 3, "Loading the 4777 and the 4778 Device Drivers" on page 3-1.

5637

Explanation: The magnetic-stripe unit self-test failed.

User Response: Replace the magnetic-stripe unit.

5691

Explanation: A device-driver error occurred.

User Response: Install the device driver again and ensure that you are using the correct parameters on the DEVICE statement in your CONFIG.SYS file.

Index

Communicating Your Comments to IBM

4777 Magnetic Stripe Unit and
4778 PIN-Pad Magnetic Stripe Reader
DOS Programming Guide

Publication No. SA34-2206-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

If you prefer to send comments by mail, use the RCF at the back of this book.

If you prefer to send comments by FAX, use this number:

United States & Canada: 1-800-955-5259

Make sure to include the following in your note:

Title and publication number of this book
Page number or topic to which your comment applies.

Readers' Comments — We'd Like to Hear from You

4777 Magnetic Stripe Unit and
4778 PIN-Pad Magnetic Stripe Reader
DOS Programming Guide

Publication No. SA34-2206-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction					

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate					
Complete					
Easy to find					
Easy to understand					
Well organized					
Applicable to your tasks					

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
SA34-2206-00

IBM

Cut or Fold
Along Line

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

International Business Machines Corporation
RDS Solutions Development
Department 56I
8501 IBM Drive
Charlotte NC 28262-8563

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

IBM

Part Number: 07H5083

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.