

PYTHON™

**DDS-DC DATA COMPRESSION
DAT TAPE DRIVES**

**Model: 4322 (3.50-Inch, Internal Tape Drive)
4542 (5.25-Inch, Internal Tape Drive)
4352 (External Subsystem)**

PRODUCT DESCRIPTION MANUAL

January 1992

ARCHIVE®
CORPORATION

PYTHON™

DDS-DC DATA COMPRESSION DAT DRIVES

4322: 3.5-Inch, Internal Tape Drive

4542: 5.25-Inch, Internal Tape Drive

4352: External Subsystem

PRODUCT DESCRIPTION MANUAL

January 1992

*CORNER
TAPE*

**ARCHIVE
CORPORATION
1650 Sunflower Avenue
Costa Mesa, CA 92626
(714)641-1230/FAX: 714/641-2590**

REVISION RECORD

DATE	REV	REVISED PAGES
01/92	002	i through viii, 1-1 through 1-3, 1-6, 1-8 through 1-11, 2-2, 2-4 through 2-6, 2-8, 2-9 and 2-10 deleted, 3-6, 3-10 through 3-12, 4-1 through 4-2, 4-10 through 4-11, 5-1, 5-4 through 5-8, 6-1 through 6-16, 7-3 7-8, 8-2 through 8-14, 9-1, A1-A3, C1, C-4 through C-7, C-10, C-12

Manual Part Number 27296-002

Copyright 1991 and 1992 Archive Corporation. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior written permission from Archive Corporation.

Archive, Python, and QICstream are trademarks of Archive Corporation.

Product features and specifications described in this manual are subject to change without notice. The manufacturer is not liable for any damages or for the loss of information resulting from the performance or use of the information contained herein.

CONTENTS

Preface	viii
Chapter 1. Introduction	1-1
1.1 Overview.	1-1
1.2 Python Models	1-3
1.2.1 Python Model Numbers	1-3
1.2.2 Python DAT Drives	1-4
1.3 Features	1-6
1.4 Typical System Configurations	1-7
1.5 DAT Technology Overview	1-8
1.5.1 Helical Scan Recording	1-8
1.5.2 Tape Formats	1-8
1.5.3 Data Compression---General	1-9
1.5.4 Flash EEPROM.	1-10
1.6 Software	1-10
1.7 About This Manual.	1-11
Chapter 2. Specifications.	2-1
2.1 Overview.	2-1
2.2 Physical Specifications	2-1
2.3 Power Specifications	2-4
2.4 Drive Performance Specifications.	2-6
2.5 Environmental Requirements.	2-7
2.6 DAT Cassette Specifications	2-7
2.7 Regulatory Compliance.	2-8
Chapter 3. Installation	3-1
3.1 Introduction	3-1
3.2 Guidelines and Cautions (4322 and 4542 Internal Models)	3-1
3.3 Unpacking and Inspection	3-2
3.4 Cabling and Connectors	3-2
3.4.1 Cabling Considerations	3-2
3.4.2 Electrical Characteristics	3-2
3.4.3 Single-Ended SCSI Connector---Internal Models 4322 and 4542.	3-3
3.4.4 Single-Ended SCSI Connector---Model 4352	3-4
3.5 Installing the Python 4322 and 4542 Internal Drives.	3-5
3.5.1 Setting Operational Switches	3-6
3.5.2 Mounting the Drive Unit	3-12
3.5.3 Completing Connections	3-15
3.6 Installing the Python 4352 External Drive.	3-16
3.6.1 Selecting the SCSI Address	3-17
3.6.2 Completing the Interface Connection	3-17
3.6.3 Connecting the Power Cord	3-18
Chapter 4. Drive Operation	4-1
4.1 Introduction	4-1
4.2 Data Compression Operation	4-1

4.3	Front Panel LED Operation	4-2
4.4	Loading/Unloading the Cassette	4-5
4.4.1	Loading/Unloading a Cassette (Normal Operation)	4-5
4.4.2	Unloading a Cassette (Manual Operation).	4-6
4.5	Using a Blank Cassette	4-9
4.6	Using a Cassette Containing Data	4-10
4.7	Using a Prerecorded Audio Cassette	4-10
4.8	Loading Revised Firmware via Archive Firmware Cassette.	4-11
Chapter 5.	SCSI Interface	5-1
5.1	Introduction	5-1
5.2	SCSI-1 Interface.	5-1
5.3	ANSI X3.131 Conformance Statement (SCSI-1).	5-4
5.4	SCSI-2 Interface.	5-5
5.5	ANSI X3T9.2 Conformance Statement (SCSI-2) (Working Draft, Rev 10)	5-7
Chapter 6.	Tape Formats	6-1
6.1	Introduction	6-1
6.1.1	DDS Format	6-1
6.1.2	DDS-DC Format	6-2
6.1.3	Commonalities Between the Formats	6-2
6.2	Entities (DDS-DC Format Only).	6-3
6.3	Tape Layout.	6-4
6.4	Single Data Space Layout	6-4
6.4.1	Device Area	6-5
6.4.2	Reference and System Area.	6-5
6.4.3	Data Area	6-7
6.4.4	End-of-Data (EOD) Area	6-7
6.5	Two-Partition Layout	6-7
6.6	Data Storage Organization.	6-8
6.7	Individual Track Layout	6-9
6.7.1	Main Area	6-9
6.7.2	Subareas	6-10
6.8	Group Structure	6-11
6.8.1	Data Frames	6-12
6.8.2	Index.	6-12
6.8.3	ECC Frame.	6-12
6.8.4	Amble Frames	6-13
6.9	Indexing.	6-13
6.9.1	Group Information Table (GIT).	6-13
6.9.2	Block Access Table (BAT).	6-14
6.10	Vendor Group.	6-15
6.11	Read-After-Write	6-16
Chapter 7.	Data Compression.	7-1
7.1	Introduction	7-1
7.1.1	Data Compression Considerations	7-1
7.1.2	DCLZ Algorithm.	7-2
7.1.3	Hardware Compression.	7-3
7.1.4	Data Integrity	7-3

7.2 DCLZ Algorithm.	7-4
7.2.1 Simplified Compression Operation	7-4
7.2.2 Dictionary.	7-5
7.2.3 Simplified Decompression Operation	7-6
7.3 AHA3101 Data Compression Chip	7-7
Chapter 8. Theory of Operation	8-1
8.1 Overview.	8-1
8.2 Block Diagrams	8-2
8.3 Python DDS-DC DAT Drives	8-2
8.3.1 Helical Scan Recording -- Four-Head Design	8-7
8.3.2 Motors and Control Circuits.	8-8
8.3.3 Flash EEPROM.	8-10
8.3.4 Sensors	8-10
8.3.5 Audio Output Jack	8-10
8.4 DAT Cassette	8-11
Chapter 9. Maintenance and Reliability	9-1
9.1 Maintenance	9-1
9.1.1 Head Cleaning	9-1
9.1.2 Automatic Drive Spin-Down and Write	9-2
9.1.3 Guidelines for High Temperature or Humidity Conditions	9-2
9.2 Reliability	9-3
9.2.1 Mean-Time-Between-Failure	9-3
9.2.2 Mean-Time-To-Repair	9-3
Appendix A. Glossary of Terms	A-1
Appendix B. Acronyms and Measurements	B-1
B.1 Abbreviations and Acronyms.	B-1
B.2 Measurements	B-2
Appendix C. Data Compression---SCSI Information	C-1
C.1 Introduction	C-1
C.2 MODE SELECT Page	C-1
C.2.1 Compression Control---Byte 02.	C-2
C.2.2 Decompression Control---Byte 03.	C-2
C.3 MODE SENSE Page.	C-4
C.3.1 Compression Control---Byte 02.	C-5
C.3.2 Decompression Control---Byte 03.	C-5
C.3.3 Compression Algorithm---Byte 07	C-6
C.3.4 Decompression Algorithm---Byte 11	C-6
C.4 Request Sense Data	C-6
C.5 Implementation Notes	C-7
C.5.1 Retrieval of Compressed Data from Noncompressing Modes	C-7
C.5.2 SCSI Command Operation for Noncompressing Modes	C-9
C.6 WRITE DATA BUFFER (3Bh)	C-10
C.6.1 WRITE DATA BUFFER Command Descriptor Block	C-10
C.6.2 Command Descriptor Block Field Descriptions	C-11
C.6.3 WRITE DATA BUFFER Data Header	C-11
C.6.4 Completion Status	C-12

FIGURES

Figure 1-1. Python 4322 DAT Drive	1-4
Figure 1-2. Python 4542 DAT Drive	1-5
Figure 1-3. Python 4352 DAT Drive	1-5
Figure 1-4. SCSI System Sample Configurations	1-7
Figure 2-1. Python 4322 DAT Drive General Dimensions.	2-2
Figure 2-2. Python 4542 DAT Drive General Dimensions.	2-3
Figure 2-3. Python 4352 External Subsystem General Dimensions	2-3
Figure 3-1. Python 4322 Switchbank Access	3-6
Figure 3-2. Python 4542 Switchbank Access	3-7
Figure 3-3. DIP Switch Default Settings	3-8
Figure 3-4. SCSI Device Address Selection (4322 and 4542)	3-9
Figure 3-5. JP6 External SCSI Address Port	3-12
Figure 3-6. Mounting Hole Locations (4322)	3-13
Figure 3-7. Mounting Hole Locations (4542)	3-14
Figure 3-8. Power and SCSI Connectors---4322 DAT Drive	3-15
Figure 3-9. Power and SCSI Connectors---4542 DAT Drive	3-16
Figure 3-10. Python 4352 Rear Panel	3-17
Figure 3-11. Daisy chain Diagram.	3-18
Figure 4-1. Python 4322 Front Panel	4-3
Figure 4-2. Python 4542 Front Panel	4-3
Figure 4-3. Python 4352 Front Panel	4-4
Figure 4-4. Cassette Loading (4322 DAT Drive)	4-5
Figure 4-5. Manually Removing a Cassette---Python 4322 Drive	4-7
Figure 4-6. Manually Removing a Cassette---Python 4542 Drive	4-8
Figure 4-7. Manually Removing a Cassette---Python 4352 Drive	4-9
Figure 6-1. Single Data space Tape Layout	6-5
Figure 6-2. Reference and System Area Layout.	6-6
Figure 6-3. Two-Partition Tape Layout.	6-8
Figure 6-4. Individual Track Layout	6-9
Figure 6-5. Group Structure	6-11
Figure 6-6. Group Structure---DDS and DDS-DC Formats	6-12
Figure 8-1. Simplified Block Diagram -- Python DDS-DC Drives.	8-3
Figure 8-2. Block Diagram -- SCSI Controller Python Models 4322NT and 4542NT DDS-DC Drives	8-4
Figure 8-3. Block Diagram -- SCSI Controller Python Models 4322NP, 4542NP, and 4352XP DDS-DC Drives	8-5
Figure 8-4. Python State Diagram	8-6
Figure 8-5. Four-Head Design	8-7
Figure 8-6. Alternate Azimuth Angles	8-8
Figure 8-7. DAT Cassette	8-11
Figure 8-8. Cassette Design Features	8-12

Figure 8-9. Write-Protect Tab on DAT Cassette8-13

Figure C-1. LBAs of Tape in Example. C-8

TABLES

Table 1-1. Python DDS-DC SCSI DAT Models---Cross Reference 1-3

Table 1-2. Chapter Descriptions.1-11

Table 2-1. Physical Specifications for Python DAT Drives. 2-2

Table 2-2. Power Specifications for the 4322 and 4542 Drives. 2-4

Table 2-3. Power Connector Pin Assignments (Models 4322 and 4542) 2-5

Table 2-4. Power Specifications for the Python 4352 Drives. 2-5

Table 2-5. Performance Specifications: Python DDS-DC DAT Drive (Data
Compression Enabled) 2-6

Table 2-6. Environmental Specifications: Python DDS-DC Drives 2-7

Table 2-7. Regulatory Compliances 2-8

Table 3-1. Single-Ended SCSI Connector Pin Assignments: Models 4322 and 4542. 3-4

Table 3-2. Single-Ended SCSI Connector Pin Assignments: Model 4352 3-5

Table 4-1. LED Summary. 4-4

Table 5-1. Status Codes---SCSI-1 5-1

Table 5-2. Message Codes---SCSI-1. 5-2

Table 5-3. Commands---SCSI-1. 5-3

Table 5-4. Status Codes---SCSI-2 5-5

Table 5-5. Message Codes---SCSI-2. 5-5

Table 5-6. Commands---SCSI-2. 5-6

Table 6-1. Bit configurations of Flag Byte (DDS-DC Format Only)6-15

Table 7-1. Simplified Compression 7-4

Table 7-2. Simplified Decompression 7-6

Table 7-3. Dictionary 7-7

Table 8-1. Tape Lengths and Capacities8-12

Table 9-1. Reliability Specifications. 9-3

Table C-1. Data Compression MODE SELECT Page C-1

Table C-2. DDE and RED Bit Configurations C-3

Table C-3. Algorithms---MODE SELECT Page. C-4

Table C-4. Data Compression MODE SENSE Page C-4

Table C-5. Algorithms---MODE SENSE Page. C-6

Table C-6. Sense Data Values and Descriptions C-7

Table C-7. Command Descriptor Block: WRITE DATA BUFFER.C-10

Table C-8. Field Descriptions: WRITE DATA BUFFER Command Descriptor Block. . . C-11

Table C-9. Data Header: WRITE DATA BUFFERC-11

Table C-10. Completion Status: WRITE DATA BUFFERC-12

PREFACE

The Archive Python™ DDS-DC Digital Audio Tape (DAT) products---Models 4322NT, 4322NP, 4542NT, 4542NP, and 4352XP---are computer DAT drives with data compression. These drives also contain an embedded Small Computer System Interface (SCSI) controller that provides a single-ended SCSI-1 or SCSI-2 interface.

The 4322 models are 3.5-inch, internal drives. The 4542 models designate 5.25-inch versions of these 3.5-inch internal drives, factory-built with a 5.25-inch bezel and mounting rails. The 4352 model is a self-contained, external subsystem with power supply.

The 4-digit, 2-letter model numbers used for Python DAT drives are explained in Section 1.2.1. In this manual, the short, 4-digit model designation (4322, for example) refers to all of the 4-digit, 2-letter models included in that model designation. For example, 4322 refers to the 4322NT and 4322NP models.

This document gives you an in-depth look at the Python 4322, 4542, and 4352 DDS-DC drives. It is intended for those familiar with basic tape drive technology or who are using or evaluating Python DAT data compression drives.

This manual contains all the technical information pertinent to the internal and external Python DDS-DC DAT drives except for details of the SCSI-1 and SCSI-2 interfaces. Detailed technical information about the Python SCSI interfaces (including a complete command summary) is contained in the *Python Technical Reference Manual* (P/N 25356-002; DDS SCSI commands only) or the *Python SCSI Manual* (P/N 27298-001) available in 1992. However, Appendix C of this manual provides specific, important SCSI information pertaining to data compression.

4320NT - NO COMPRESSION
DDS STANDARD

CHAPTER 1

INTRODUCTION

1.1 Overview

Archive Python Digital Audio Tape (DAT) drives are designed for computer environments requiring high performance, high capacity data storage. Based on a proven 3.5-inch mechanism, the Python drives described here---Models 4322NT and 4322NP (3.5-inch), 4542NT and 4542NP (5.25-inch), and 4352XP (external)---provide hardware data compression which supports the industry standard Digital Data Storage Data Compression (DDS-DC) format.

All Python data compression drives provide an embedded, single-ended Small Computer Systems Interface (SCSI) controller. The drives are equipped with either a 512 kilobyte (KB) on-drive buffer or a 1 megabyte (MB) buffer to facilitate efficient operation. In addition, the 4322NP, 4542NP, and 4352XP models offer leading-edge electronically erasable, programmable, read-only memory (flash EEPROM), which enables qualified Archive OEMs to download revised firmware to the drive via three methods: using the Python drive's serial port, the SCSI bus, or a specialized Archive firmware cassette.

The Python DDS-DC drives comply with the American National Standards Institute (ANSI) and European Computer Manufacturers Association (ECMA) DDS format, which ensures interchange compatibility of digital data stored on a small removable DAT cassette (approximately 2 inches x 3 inches x 0.4 inch) using 4 mm tape. The Python data compression drives also comply with the ANSI/ECMA Digital Data Storage Data Compression (DDS-DC) format, which is the industry standard format for DAT data compression and is a superset of the DDS format. These Python drives use an advanced DCLZ (data compression LempelZiv) algorithm to compress data by up to four or more times. More importantly, however, DDS-DC data compression is transparent to the host software and SCSI driver, enabling rapid integration and preserving existing software investments.

Because the DDS-DC format is a superset of the DDS format, the Python data compression drives are fully compatible with the DDS format for reading and writing standard, uncompressed data. To switch operation from compressed to uncompressed mode, the host computer can issue a SCSI command. On internal Python DDS-DC models, a hardware switch is also available to enable or disable the DDS pass-through (uncompressed) mode as an initial power-on default.

Tape capacity and sustained data transfer rate using DDS-DC are dependent upon the characteristics of the files being compressed, along with other parameters, including the speed of the host system, and the operating system and application software used. Nevertheless, Archive Python DDS-DC drives typically provide a doubling of storage capacity and transfer rate---and a maximum quadrupling of storage capacity and transfer rate---when compared with computer DAT drives without data compression. That is, Python data compression drives provide a typical 2.6 gigabyte (GB) storage capacity on a 60-meter DDS data cassette and 4.0 GB on a 90-meter DDS data cassette. The drive sustained transfer rate at this typical 2:1 compression ratio is 366 kilobytes/second (KB/sec), or 22 megabytes/minute (MB/minute).

With highly redundant files, such as many database files, Python DDS-DC drives can achieve a 4:1 compression ratio which provides a nominal maximum storage capacity of 5.2 GB on a 60-meter DDS data

cassette and 8.0 GB on a 90-meter DDS data cassette. The transfer rate at this 4:1 compression ratio is 732 KB/sec (44 MB/min). Although this 4:1 compression ratio (and corresponding 8.0 GB capacity and 44 MB/minute transfer rate) is termed a "maximum", these values are a *nominal maximum* only for the more conventional file types. With more highly redundant data, Python DDS-DC drives are capable of much higher compression ratios, capacities, and transfer rates.

Python DDS-DC drives combine inherent DAT technology and DDS-DC data compression capabilities with Archive's proven computer grade design to provide unmatched reliability and performance characteristics among DDS products. These drives are ideal for PC, workstation, network, and minicomputer applications such as:

- Backup of high capacity fixed disks.
- Data interchange between systems.
- Software distribution.
- Online data collection.
- Direct-access secondary storage for text, graphics or multimedia information of all types.
- Archival storage.

Built using long-wearing materials and custom Large Scale Integration (LSI) components, the Python 3.5-inch SCSI DAT drive used in all these DDS-DC models was engineered for heavy-duty computer applications. Providing carefully controlled tape handling and rapid, smooth operation, the Python design promotes long life for key components such as the rotating DAT drum, drive heads, and the media itself. One major benefit of this new, computer grade engineering is low power consumption -- typically below 7 Watts.

All Python DDS-DC drives contain an embedded SCSI controller that supports SCSI-1 (*ANSI X3.131, 1986*) and SCSI-2 (*ANSI X3T9.2 Working Draft, Revision 10*). The 3.5-inch and 5.25-inch internal drive form factors are tailored for easy installation in today's computers, and the full-featured embedded SCSI controller facilitates easy integration into a variety of systems. The internal models (4322 and 4542) are hardware switch-selectable for SCSI-1 or SCSI-2, and all models are software selectable for SCSI-1 or SCSI-2. The Python DDS-DC drives also provide synchronous or asynchronous SCSI and a high speed burst rate of 5 MB/second.

Python DDS-DC products provide unmatched reliability through three levels of error correction code (ECC) and the four-head design, which allows read-after-write (RAW) error detection and correction. Python internal data compression models also contain an onboard serial port that provides the capability for extensive testing of the Python drives.

All Archive Python drives comply with the DDS tape format standard---*American National Standard Helical-Scan Digital Computer Tape Cartridge 3.81 mm (0.150 in) for Information Interchange*. The Python drives are designed to use data-grade DDS DAT media, not audio DAT media, and provide maximum data integrity and reliability with computer DAT cassettes officially qualified by Archive. The Archive-qualified Model M31300 DDS data cassette (60 m), Model M32000 DDS data cassette (90 m), and M7301 DDS head-cleaning cassette (or equivalents supplied by Archive subsidiary companies) are recommended.

1.2 Python Data Compression Models

The Python SCSI drives with data compression are as follows:

- 3.5-inch, half-high DDS-DC DAT drives that mount internally (Models 4322NT and 4322NP).
- 5.25-inch, half-high DDS-DC DAT drives that consist of a 3.5-inch drive with 5.25-inch mounting rails and bezel that mount internally in a 5.25-inch half-high space (Models 4542NT and 4542NP).
- A complete external subsystem containing the 3.5-inch DDS-DC DAT drive and built-in worldwide power supply (Model 4352XP).

Table 1-1 lists a cross reference of data-compression models and features.

Table 1-1. Python DDS-DC SCSI DAT Models---Cross Reference

FEATURE	4322NT	4322NP	4542NT	4542NP	4352XP
Buffer Size					
512 KB	P/N 25977-0xx	P/N 27264-0xx	P/N 27402-0xx	P/N 27373-0xx	P/N 27265-0xx
1024 KB	P/N 25977-5xx	P/N 27264-5xx	P/N 27402-5xx	P/N 27373-5xx	P/N 27265-5xx
Form Factor	3.50" H.H.	3.50" H.H.	5.25" H.H. ⁽¹⁾	5.25" H.H. ⁽¹⁾	External Subsystem
Mounting	Internal	Internal	Internal	Internal	External
SCSI	Single-Ended	Single-Ended	Single-Ended	Single-Ended	Single-Ended
Data					
Compression	Yes	Yes	Yes	Yes	Yes
Conventional					
EEPROM	Yes	No	Yes	No	No
Flash					
EEPROM	No	Yes	No	Yes	Yes

⁽¹⁾ 3.5" H.H. drive in 5.25" bezel and mounting rails.

NOTES: Certain part numbers listed may be available only as a special order.

Part numbers listed are for generic Archive Python DDS-DC OEM products only. Contact your distributor or dealer for the correct distribution part numbers for ordering information.

1.2.1 Python Model Numbers

The six-position model numbers assigned by Archive to Python drives denote specific information as follows:

1	2	3	4	5	6
—	—	—	—	—	—

Position 1 shows the tape drive type. This character is always 4 for Archive DAT products, which refers to the 4-millimeter tape size.

Position 2 shows the form factor. This character is either a 5, which refers to the 5.25-inch internal form factor, or a 3, which refers to either the 3.5-inch internal form factor or an external Python subsystem, which (like all Archive Python drives) uses a 3.5-inch internal mechanism.

Position 3 is a marketing option.

Position 4 denotes the following: 0 for single-ended SCSI interface models without data compression; 1 for differential SCSI interface models without data compression; and 2 for single-ended SCSI interface models with data compression.

Position 5 also shows whether the drive is an internal or external unit. This character is an N for internal or an X for external.

Position 6 shows the interface and PROM type. For the SCSI-1 or SCSI-2 compatible interface (default setting SCSI-2) with the standard PROM, this character is a T. For the SCSI-1 or SCSI-2 compatible interface (default setting SCSI-2) with flash EEPROM, this character is a P.

1.2.2 Python DAT Drives

Figures 1-1, 1-2, and 1-3 illustrate the Python 4322, 4542, and 4352 drives, respectively.

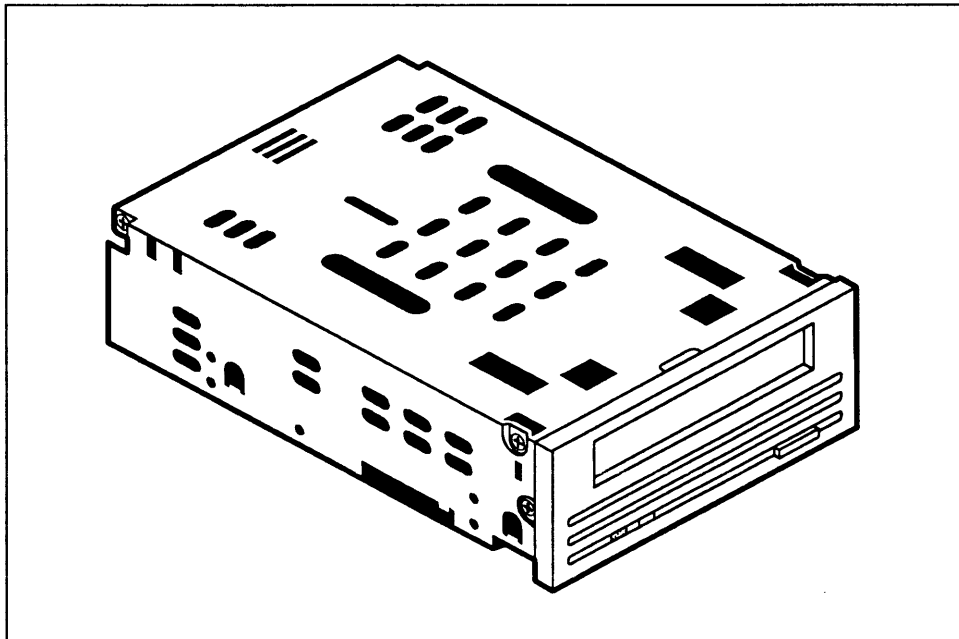


Figure 1-1
Python 4322 DAT Drive

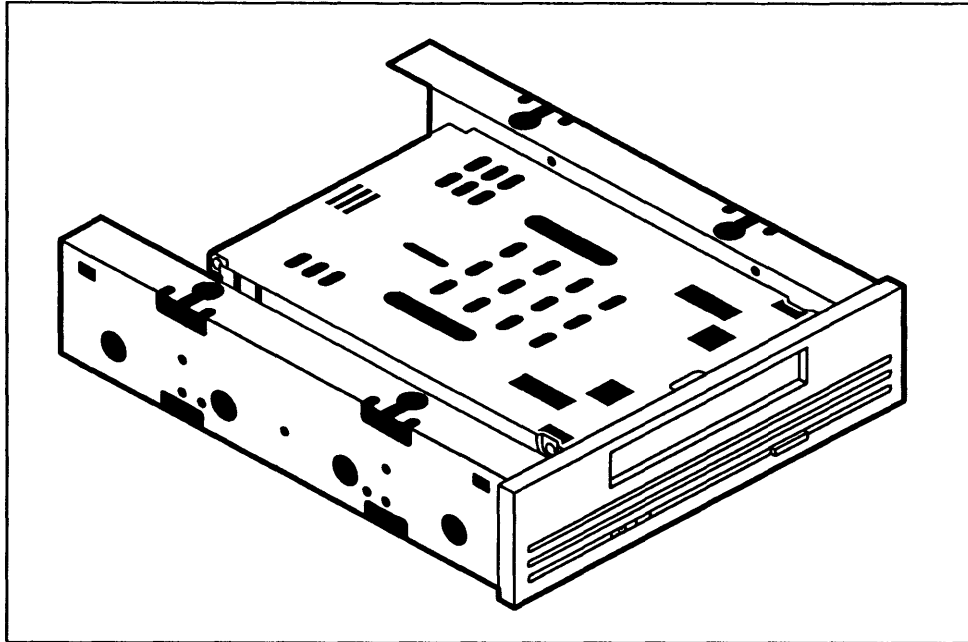


Figure 1-2
Python 4542 DAT Drive

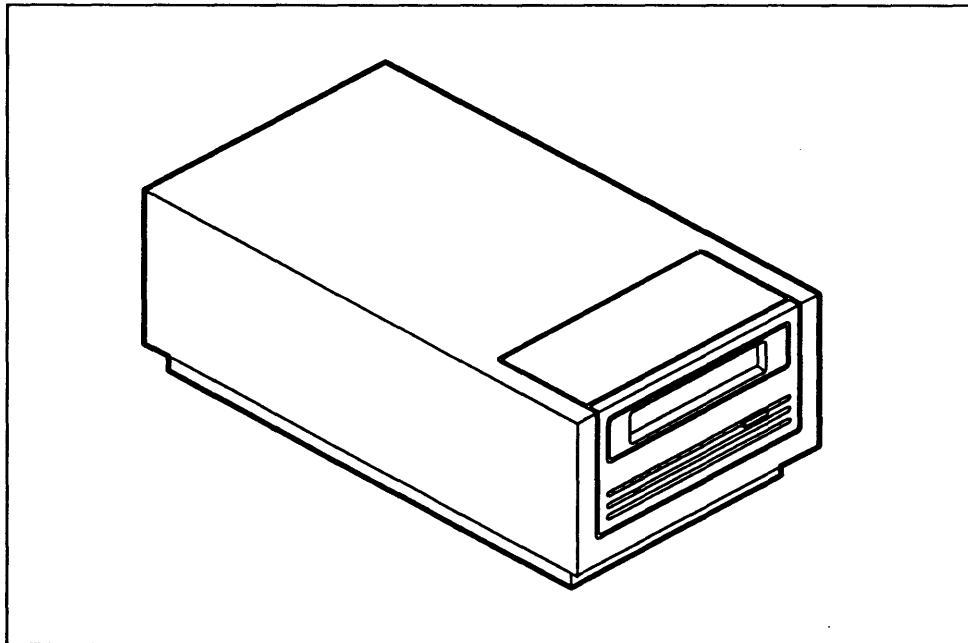


Figure 1-3
Python 4352 DAT Drive

1.3 Features

The Python data compression drives embody Archive's commitment to engineering reliable and durable tape drive products which implement leading-edge technology. In summary form, key features of the Python DDS-DC drives are:

- All Python DDS-DC drives based on proven 3.5-inch Python DAT drive, with the identical drive mechanism used in standard (non-compressed) Python products
- 3.5-inch internal form factor for installation in a 3.5-inch half-high space (Model 4322NT and 4322NP)
- 3.5-inch DAT drive with factory-installed 5.25-inch mounting rails and bezel for installation in a 5.25-inch half-high space (Models 4542NT and 4542NP)
- External subsystem with built-in, auto-sensing, worldwide power supply (Model 4352)
- Advanced onboard DDS-DC hardware using DCLZ (Data Compression LempelZiv) data compression algorithm
- ANSI/ECMA DDS and DDS-DC tape format compliance for compatibility and interchange
- Reads and writes both DDS uncompressed and DDS-DC compressed data and data cassettes
- Typical capacity of 4 GB, nominal maximum capacity of 8 GB on a 90-meter DDS cassette
- High speed random access of 20 seconds (average) to any file on a 60 m tape; 30 seconds for a 90 m tape
- High speed transfer rates for fast backups:
 - 183 KB/sec (11 MB/min)---uncompressed data
 - 366 KB/sec (22 MB/min) typical---compressed data
 - 732 KB/sec (44 MB/min) nominal maximum---compressed data
- High performance SCSI burst transfer rate of 5 MB/sec with up to 1 MB on-drive data buffer to facilitate the most efficient use of the host computer
- Four-head design with RAW error detection and rewrites
- Three levels of ECC to ensure data integrity
- Uncorrectable error rate of less than 1 in 10^{15} bits
- Flash EEPROM ("NP" and "XP") models enable electrically upgradeable drive firmware
- Custom Archive designed LSI circuitry to reduce component count and boost drive reliability
- Advanced, single-chip, DAT formatter LSI
- Low power consumption---less than 7 Watts (typical) for internal drives
- Single-ended SCSI connection with these features:
 - Embedded full LSI, high speed SCSI controller
 - Switch selectable SCSI-1 or SCSI-2 interface on internal models (as default setting at power up) for flexibility in system integration
 - Software selectable SCSI-1 or SCSI-2 interface on all models
 - Software selectable synchronous or asynchronous SCSI data transfer

- Onboard output jack on internal models for configuring the SCSI address if repackaged in an external box
- Compatibility with the Archive SCSI Viper quarter-inch cartridge drives
- Onboard serial port for internal diagnostics (internal models only---4322 and 4542)
- Automatic power-on self tests (switch option on internal models)
- Manual emergency cassette ejection procedure

1.4 Typical System Configurations

The SCSI standard supports up to eight SCSI addresses or IDs. These IDs refer to host adapters or peripheral devices such as printers, magnetic disks, or tape drives. The eight devices or hosts are daisy chained together. Figure 1-4 shows sample configurations of SCSI systems.

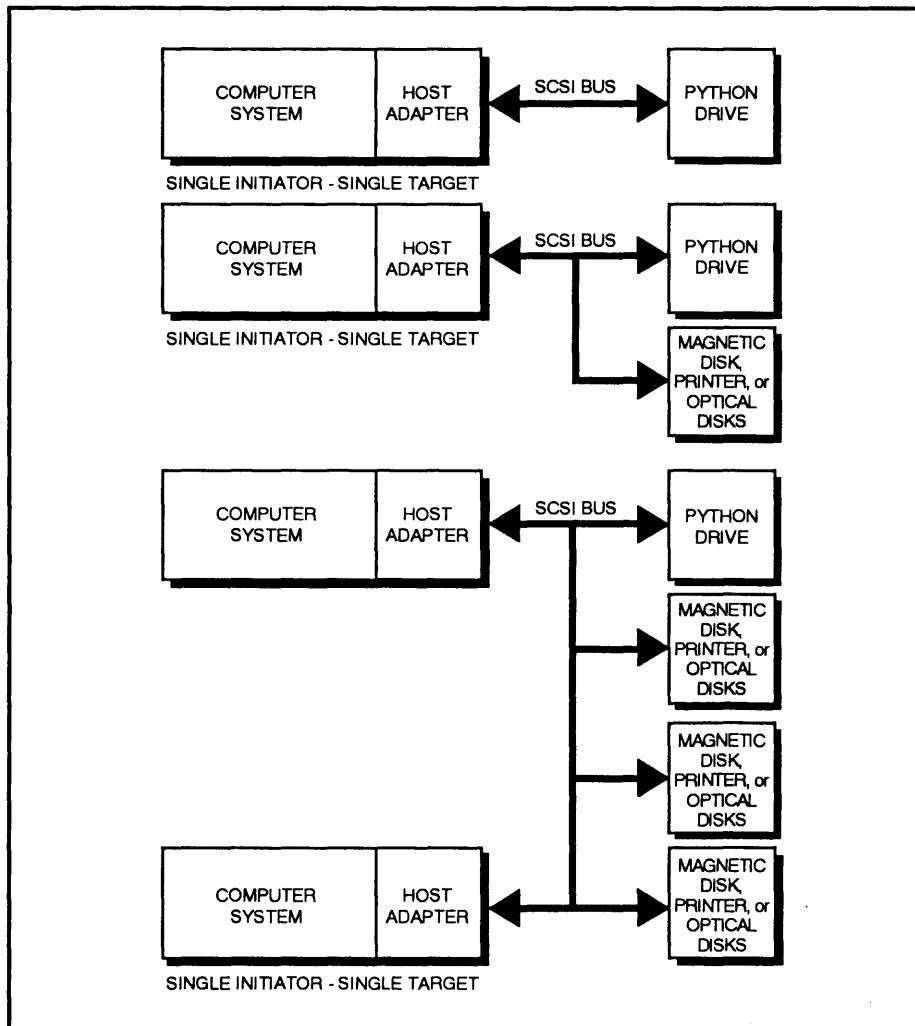


Figure 1-4
SCSI System Sample Configurations

1.5 DAT Technology Overview

First developed for the audio electronics market, DAT technology was first applied in computer peripherals in the late 1980s. Unlike traditional magnetic tape audio cassette products, DAT technology proves inherently reliable through the *helical scan recording method*, which provides a high recording density with a very low error rate. All DAT products, including computer implementations, use the helical scan recording method. This recording method has been used in professional video tape recorders (VTRs) since 1956 and in home video cassette recorders (VCRs) since 1974. In 1986, DAT products using helical scan technology were first developed for audio applications. DAT consumer products are specifically designed for digital audio recording and playback and compete with such products as analog audio cassette decks and compact disk (CD) players.

1.5.1 Helical Scan Recording

Helical scan recording was originally developed as method of efficiently recording high-quality television signals on a relatively slow moving tape. It requires that both the tape and the recording head move simultaneously. This recording method results in an extremely high recording density, far higher than can be achieved with stationary-head devices such as 1/2-inch open-reel or 1/4-inch cassette tapes. (See Chapter 8, "Helical Scan Recording -- Four-Head Design" for additional information.)

In helical scan recording, both the read and write heads are located on a rapidly rotating cylinder or drum. The cylinder is tilted at an angle in relation to the vertical axis of the tape. As the tape moves horizontally, it wraps around the part of the circumference of the cylinder (90°) so that the head enters at one edge of the tape and exits at the other edge before the tape unwraps.

The horizontal movement of the tape in combination with the angular movement of the cylinder causes the track to be recorded diagonally across the tape rather than straight down its length. The resulting recorded track, nearly one inch, is approximately eight times longer than the width of the tape.

1.5.2 Tape Formats

Archive Python DDS-DC drives are designed to use the industry standard DDS and DDS-DC tape formats. These two formats are summarized in the following text.

DDS Tape Format

This standard format was codeveloped by DDS manufacturers to support DAT devices as computer peripherals. The objectives of DDS are to maximize storage capacity and performance; to facilitate data interchange; to provide compatibility with existing tape storage command sets; and to provide extremely fast random access. The DDS format also takes advantage of the helical scan recording method and the inherent error correction capability of the DAT technology to augment error detection and correction.

The format consists of a finite sequence of data groups with each data group being a fixed-length recording area. A data group is made up of 22 data frames and 1 ECC frame; each frame is made up of two helical scan tracks. The advantages of the fixed-length data group is that ECC is easily generated, and buffering requirements are simplified. (See Chapter 6, "Tape Formats", for additional information.)

Although data groups are fixed-length and always contain 22 data frames, the DDS format is designed such that variable-length computer records can be stored in the fixed-length data groups.

The native transfer rate of 11 million bytes/minute (183 KB/sec) is a characteristic of the DDS format for computer DAT technology. At this transfer rate, a full 1.3 GB of information can be written in 120 minutes, and a full 2.0 GB of information can be written in 180 minutes. Another important speed consideration is that once the information is written, specific files or data sets can be located at up to 200 times the speed at which they were originally written. The resulting average time to randomly access any file on a 60-meter DAT cassette is 20 seconds.

DDS-DC Tape Format

A superset of the basic DDS DAT format, DDS-DC drives can write compressed and uncompressed data to the same cassette. Thus, because DDS-DC is based on the DDS format, backward compatibility is maintained.

Introduced by the DDS Manufacturers Group and approved by ANSI and ECMA, DDS-DC is a *record compression* industry-standard format that provides support for *lossless* compression algorithms based on substitution---such as those of the Lempel-Ziv family.

This format supports compressed and uncompressed records. A recorded DAT cassette may contain compressed records, uncompressed records, filemarks, and setmarks. Compressed records exist within recorded objects called *entities*. Entities and uncompressed records are collected into groups.

Many aspects of the DDS-DC format are identical to those of the DDS format:

- The series of transformations (randomizing, interleaving, generation and inclusion of two Reed-Solomon error-correcting codes, and etc.) applied to a group before recording
- The tape layout
- The third group-based level of Reed-Solomon error-correcting codes (C3)

The only differences between the DDS and DDS-DC formats are in the contents of the groups.

The combination of DAT technology and the DDS-DC format provides a solid core around which computer DAT drives with exceptional performance and reliability can be designed, such as the Python data compression products.

1.5.3 Data Compression---General

Data compression is based on reducing the redundancy that occurs naturally in data streams of text, graphics, code, and other data. Reducing or eliminating such redundancy prior to recording the information to tape significantly increases the amount of data that can be recorded on a given amount of tape.

Data compression causes repeated strings of data to be recognized and replaced by symbols or codewords that encode the strings or point back to the original occurrence of the string. In this way, data compression uses fewer characters to represent the original data.

The Lempel-Ziv (LZ) algorithms are a family of string-based compression algorithms developed in the late 1970s by Messrs Lempel and Ziv. The Python DDS-DC DAT drives use the DCLZ algorithm, which is based on a Lempel-Ziv algorithm that was enhanced by Welch. DCLZ builds a symbol dictionary that represents data strings identified as repeated events in incoming data. The algorithm then writes the symbols to the tape. This compressed data is then decompressed with the original data strings resubstituted for the symbols from the dictionary when the data is read back. (See Chapter 7, "Data Compression", for detailed information about the algorithm.)

1.5.4 Flash EEPROM

Another technological advancement incorporated into the Python 4322NP, 4542NP, and 4352XP drives is flash EEPROM, which is useful should the drive's SCSI firmware need to be upgraded at some point. With the permanently installed, electrically upgradeable, flash EEPROM memory, revised SCSI firmware for the Python drive can be loaded into the drive via any one of three methods:

- Archive OEM firmware cassette (See Section 4.8.)
- Host SCSI bus
- Python drive serial port (internal drives) (See Chapter 8.)

This feature enables qualified Archive OEMs needing to revise DDS-DC drive SCSI firmware to do so rapidly and at a reduced cost. Flash EEPROM should also prolong the life cycle of a drive because many new techniques---such as increasing the capacity of the drive through support for longer tapes---may require only a firmware upgrade.

1.6 Software

One of the most cost-effective uses of DAT drives is to back up fixed disks. The software required to perform a disk backup runs under the control of the host computer's operating system. Compatibility with a wide range of software is also an important consideration in system integration. The Python DDS-DC computer DAT drives are designed to take advantage of the host computer's standard magnetic tape backup software or, optionally, to use backup software provided by Archive and other suppliers.

These Python drives comply with the QIC-104 standard ensuring compatibility with the Archive Viper and other quarter-inch cartridge software. (Python drives maintain full SCSI-interface compatibility with the Archive Viper SCSI 150-MB quarter-inch cartridge product.) Also, use of variable-length records and the ability to overwrite previously recorded data allow the Python drives to run software originally written for 1/2-inch reel-to-reel tape backup.

Standard Python DDS drives have been proven compatible with an exceptionally wide range of SCSI host adapters and interfaces, and network, operating system, and application software.

A major advantage of DDS-DC data compression is that it is software transparent. Software transparency means that, generally, no modification is required to network, operating system, application, and device driver software already proven compatible with Python DDS drives to run with Python DDS-DC products.

1.7 About This Manual

The remaining chapters and the appendices in this manual are briefly described in Table 1-2. A glossary of terms is also included.

Table 1-2. Chapter Descriptions

NUMBER	TITLE	DESCRIPTION
2	Specifications	Contains physical, performance, environmental, power, drive tape handling, and DAT cassette specification tables. Also, contains the SCSI-1 and SCSI-2 conformance statements.
3	Installation	Provides cautions, unpacking tips, inspection information, and installation/connection steps including cabling requirements and connector pinouts.
4	Drive Operation	Explains the simple operation of the DDS-DC drives.
5	SCSI Interface	Lists general information about the SCSI-1 and SCSI-2 interfaces. (See NOTE below.)
6	Tape Formats	Explains the DDS and the DDS-DC tape formats.
7	Data Compression	Describes the data compression algorithm and explains pertinent information for effective use of data compression.
8	Theory of Operation	Details the functional operation of various assemblies of the Python DDS-DC drives.
9	Maintenance and Reliability	Presents maintenance procedures and reliability information.
App A	Glossary	Defines key terms.
App B	Acronyms and Measurements	Lists the acronyms and measurements used in the manual.
App C	Data Compression ---SCSI Information	Describes the specific SCSI information that pertains to data compression

NOTE: *This manual provides all technical information about the Python 4322, 4542, and 4352 drives for hardware and software development and integration except complete SCSI information on the SCSI-1 and SCSI-2 commands themselves. Refer to the Python Technical Reference Manual (P/N 25356-002, DDS-only) or the Python SCSI Manual (P/N 27298-001) for detailed information about the SCSI interface, including a complete command summary. Appendix C, however, does provide important, specific SCSI information pertaining to data compression.*

NOTES

CHAPTER 2

SPECIFICATIONS

2.1 Overview

Archive Python DDS-DC DAT drives provide exceptional reliability in storing large amounts of computer data. This chapter includes technical specifications for the internal (4322 and 4542) and external (4352) SCSI Python drives. This information covers the following specifications and requirements:

- Physical specifications
- Power requirements
- Drive performance specifications
- Environmental requirements
- DAT cassette specifications
- Regulatory compliance
- ANSI Conformance Statements (SCSI-1 and SCSI-2)

2.2 Physical Specifications

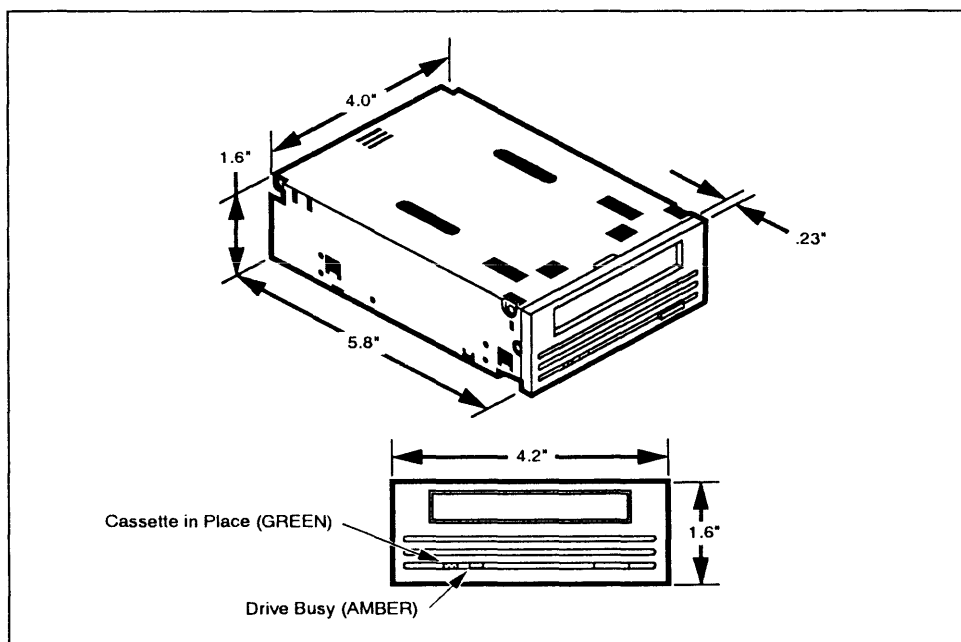
The physical specifications of Python drives are listed in Table 2-1.

NOTE: *Installation, operation, and most technical information for the 4322NT and 4322NP drives is the same. In this guide, the model designation 4322 is used to refer to both drives. Likewise, installation, operation, and most technical information for the 4542NT and 4542NP drives is the same. In this guide, the model designation 4542 is used to refer to both drives. Also, the 4352XP drive is referred to as model 4352. Information specific to only one of the drive models is identified with the complete 4-digit, 2 letter model number.*

Table 2-1. Physical Specifications for Python DAT Drives

SPECIFICATION	4322	4542	4352
Height	1.6 in/ 41 mm	1.6 in/ 41 mm	3.4 in/ 85 mm
Width	4.0 in/101 mm	5.7 in/146 mm	5.3 in/135 mm
Length	6.0 in/152 mm	7.1 in/180 mm	9.3 in/235 mm
Weight	2.0 lbs/0.9 kg	2.4 lbs/1.1 kg	5.2 lbs/2.4 kg

Figure 2-1 illustrates the Python 4322 drive showing its general dimensions.



**Figure 2-1
Python 4322 Computer DAT Drive General Dimensions**

Figure 2-2 illustrates the Python 4542 drive showing its general dimensions.

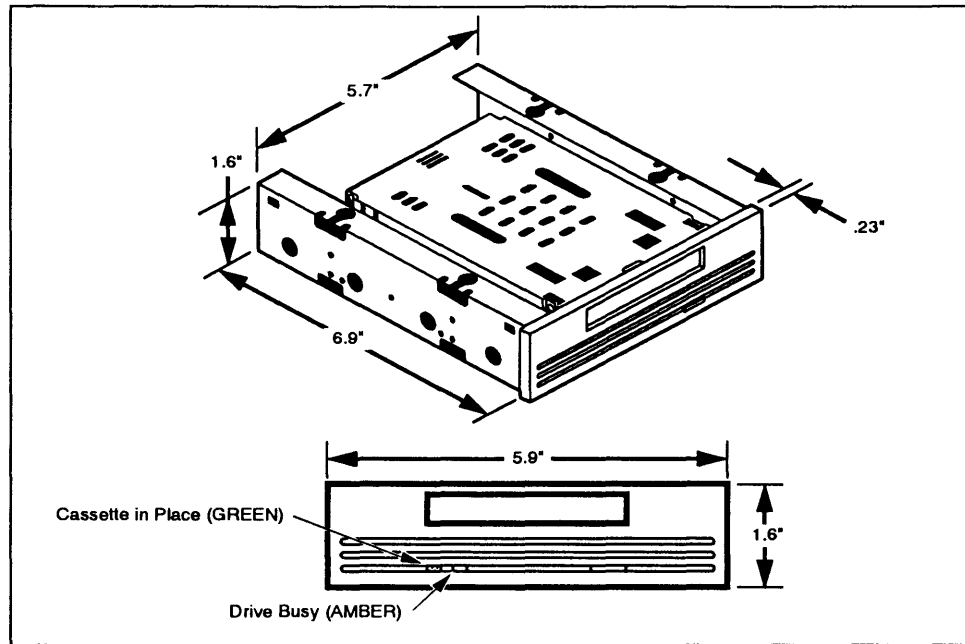


Figure 2-2
Python 4542 Computer DAT Drive General Dimensions

Figure 2-3 illustrates the Python 4352 subsystem showing its general dimensions.

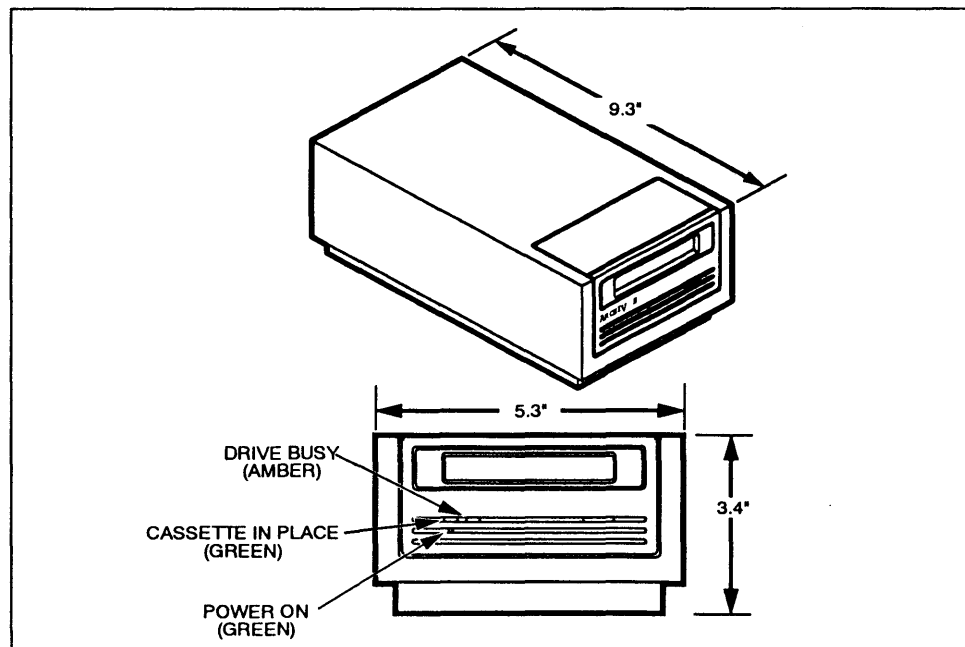


Figure 2-3
Python 4352 External Subsystem General Dimensions

2.3 Power Specifications

Table 2-2 lists the power specifications for the Python 4322 and 4542 drives. (Power specifications are measured at the tape drive power connector and are nominal values.)

Table 2-2. Power Specifications for the Python 4322 and 4542 Drives

SPECIFICATION	MEASUREMENT	
DC Voltage	+12 VDC	+5 VDC
Voltage Tolerance	+ or - 10%	+ or - 7%
Operational Current	210 milliamps	900 milliamps
Standby Current	210 milliamps	700 milliamps
Peak	1.40 amps	1.30 amps
Power Sequence	None	None
Ripple (peak to peak)	100 mV	100 mV
Power Dissipation (Operating)	< 5.0 Watts (excluding surge)	< 4.0 Watts (excluding surge)

Table 2-3 lists pin assignments for the power connector for the internal models (4322 and 4542).

Table 2-3. Power Connector Pin Assignments (Models 4322 and 4542)

PIN	ASSIGNMENT
1	+12 VDC
2	+12 return
3	+5 return
4	+5 VDC

The Python 4352 external drives have a built-in power supply that senses the incoming voltage and automatically adapts to voltages within the range of 100 to 240 volts, 50 to 60 Hz. Table 2-4 lists its power specifications.

Table 2-4. Power Specifications for the Python 4352 Drives

SPECIFICATION	AC INPUT VOLTAGE		
	100 (Japan)	120 (US)	240 (European)
AC Input Current	230 milliamps	200 milliamps	125 milliamps
AC Input Power	12.0 Watts	12.0 Watts	12.5 Watts

2.4 Drive Performance Specifications

Table 2-5 lists performance specifications for Python DDS-DC DAT drives with data compression.

Table 2-5. Performance Specifications: Python DDS-DC DAT Drives (Data Compression Enabled)

FEATURE	SPECIFICATION
Capacity	2.6 gigabytes on a 60 meter DAT cassette (typical) 5.2 gigabytes on a 60 meter DAT cassette (maximum ¹) 4.0 gigabytes on a 90 meter DAT cassette (typical) 8.0 gigabytes on a 90 meter DAT cassette (maximum ¹)
Recording Density	61,000 bits/inch
Flux Density	76,250 flux transfers/inch
Track Packing Density	1,869 tracks/inch
Areal Density	114 Megabits/square inch
Error Recovery	Read-after-write Reed Solomon ECC (C3---3 levels)
Unrecoverable Errors	Less than 1 in 10 ¹⁵ data bits
Tape Drive Type	Computer grade 4 Direct Drive (4DD) mechanism
Head Configuration	2 Read heads, 2 Write heads
Recording Format (compression)	DDS-DC (DCLZ)
Recording Media	4-mm DAT tape (qualified media recommended)
Recording Method	Helical scan (R-DAT)
Cassette	2.9 in. x 2.1 in. x 0.4 in.
Tape length:	60 m (197 ft) (2-hour cassette) 90 m (295 ft) (3-hour cassette)
Transfer Rate (Sustained)	366 KByte/second typical 732 KByte/second maximum ¹
Synchronous Transfer Rate (Burst)	5 MByte/second maximum
Asynchronous Transfer Rate (Burst)	5 MByte/second maximum
Search/Rewind Speed	200 X normal speed
Average Access Time	< 20 sec (60 meter tape) < 30 sec (90 meter tape)
Drum Rotation Speed	2000 revolutions/minute (RPM)
Tape Speed	0.32 inch/second
Head-to-Tape Speed	123 inch/second

¹Nominal maximum only; can be exceeded for highly compressible data.

2.5 Environmental Requirements

Table 2-6 lists the environmental specifications for Python drives. The Python internal DAT drives may be mounted either vertically (drive left side up or right side up) or horizontally.

Table 2-6. Environmental Specifications: Python DDS-DC Drives

SPECIFICATION	OPERATIONAL	NONOPERATIONAL
Temperature	+41° to +113°F ⁽¹⁾ (+ 5° to + 45°C)	- 40° to +149°F ⁽²⁾ (- 40° to + 65°C)
Thermal Gradient	2° C/minute (no condensation)	Below condensation
Relative Humidity	20% to 80% noncondensing ⁽¹⁾	0% to 90% noncondensing ⁽²⁾
Maximum Wet Bulb Temperature	78.8°F (26°C)	No condensation
Altitude	-100 to +4,575 meters	-300 to +15,250 meters
Vibration, peak to peak displacement	0.9 mm (1-17 Hz)	
Vibration peak acceleration	0.73 g (17 to 500 Hz) (Sweep rate less than 1 octave/minute)	1.5 g (5 to 500 Hz)
Acoustic Level Idling	49 dBA maximum	
Acoustic Level Operational	53 dBA maximum (measured in suitable enclosure at 3-ft distance and operator height)	
Shock (1/2 sine wave)	10 g's peak, 11 msec	50 g's peak, 11 msec

⁽¹⁾Mechanism and media

⁽²⁾Mechanism

2.6 DAT Cassette Specifications

Python DDS-DC DAT drives provide maximum data integrity and reliability when Archive-qualified DAT cassettes are used as the recording medium. Archive maintains an ongoing program to qualify manufacturers of DAT cassettes. The following cassettes are recommended:

- DDS data cassette: Archive Model M31300, 60-meter tape (Archive P/N: 25408-10x; order multiples of five)
- DDS data cassette: Archive Model M32000, 90-meter tape (Archive P/N: 25744-10x; order multiples of five)

- DDS cleaning cassette: Archive Model M7301 (Archive P/N: 25495-10x; order multiples of five)

These cassettes fully meet the specifications in the document *Proposed ANSI Helical-Scan Digital Computer Tape Cassette 3.81 mm (0.150 in) for Information Interchange, (ASC X3 Project No. 668)*. Only data grade DAT cassettes with the official DDS logo are recommended for use in Python drives.

Contact your Archive sales representative for information on qualified DDS data and cleaning cassette manufacturers and models.

2.7 Regulatory Compliance

All Python DAT drives comply with the regulations listed in the Table 2-7.

Table 2-7. Regulatory Compliances

AGENCY	REGULATION
CSA	C22.2, No. 220-M 1986
TUV-RHEINLAND	EN 60 950
UL	1950
FCC	Class B ⁽¹⁾
VDE	0871, Class B

⁽¹⁾ Required compliance for external model; verification on file for internal models.

The Python DDS-DC drives also conform to the following safety standards:

- UL 1950, "Information Technology Equipment, Including Electrical Business Equipment" (First Edition)
- CSA C22.2, No. 950-M89, "Safety of Information Technology Equipment, Including Electrical Business Equipment" (First Edition) (All Python internal drives, external drives pending)
- EN 60 950, "Safety of Information Technology Equipment, Including Electrical Business Equipment" (First Edition)

Use the Python DDS-DC drives only in equipment where the combination has been determined to be suitable by an appropriate certification organization (for example, Underwriters Laboratories Inc. or the Canadian Standards Association in North America). You should also consider the following safety points.

- Install the Python drive in an enclosure that limits the user's access to live parts, gives adequate system stability, and provides the necessary grounding for the drive.
- Provide the correct voltages (+5 VDC and +12 VDC) based on the regulation applied---Extra Low Voltage (SEC) for UL and CSA and Safety Extra Low Voltage for BSI and VDE.

CHAPTER 3

INSTALLATION

3.1 Introduction

This chapter tells you how to install the Python DDS-DC DAT drives. Some of the information relates to all Python models drives; other information is specifically based on the drive model. The following paragraphs briefly outline the organization of this chapter.

- Section 3.2 lists guidelines for handling the 4322 and 4542 drives.
- Section 3.3 contains general information that you should read before you begin the installation.
- Section 3.4 gives specific cabling requirements and connector pinouts for the drives.
- Section 3.5 describes installing Model 4322, the 3.5-inch internal drive, and Model 4542, the 3.5-inch drive with 5.25-inch mounting rails and bezel.
- Section 3.6 describes installing Model 4352, the external subsystem.

3.2 Guidelines and Cautions (4322 and 4542 Internal Models)

The following guidelines and cautions apply to handling and installing Python 4322 and 4542 internal drives. Keep them in mind as you install the drive.

- Because Python internal drives contain components that are sensitive to static electricity, the drives are shipped in a protective anti-static bag. DO NOT remove the drive from the anti-static bag until you are ready to install it.
- Before you remove the drive from the anti-static bag, touch a metal or grounded surface to discharge any static electricity buildup from your body.

CAUTION: *If you touch static-sensitive parts of the drive, such as a printed circuit board, and discharge static electricity, the components may be damaged.*

- Hold the drive by its edges only and avoid direct contact with any printed circuit board exposed.
- Lay the drive only on top of the anti-static bag or return it to the bag when you need to lay it down.

3.3 Unpacking and Inspection

Although Python drives are inspected and carefully packaged at the factory, damage may occur during shipping. Follow these steps for unpacking the drive.

1. Visually inspect the shipping containers; notify your carrier immediately of any damage.
2. Place shipping containers on a flat, clean, stable surface; then, carefully remove and verify the contents against the packing list.

If parts are missing or the equipment is damaged, notify your Archive representative.

3. Save the containers and packing materials for any future reshipment.

3.4 Cabling and Connectors

The Python DDS-DC drives provide standard single-ended SCSI transmission. ANSI SCSI standards specify the technical requirements for correctly cabling and connecting single-ended devices. This section provides information about SCSI cabling and connectors for the 4322, 4542, and 4352 drives.

Refer to Sections 3.5 and 3.6 for actual installation instructions.

3.4.1 Cabling Considerations

Either 50-pin flat cable or 25-signal twisted-pair cable with a maximum length of 6 meters (19 feet) may be used to connect the Python drives to the SCSI host adapter output. If twisted-pair cabling is used, connect the twisted pairs to physically opposing contacts on the connector.

A stub length no greater than 0.1 meter should be used off the mainline connection within any connected equipment.

The cable characteristic impedance should not be less than 90 ohms nor greater than 140 ohms. A cable characteristic impedance of greater than 100 ohms is recommended.

To minimize noise and ensure even distribution of terminator power, the minimum recommended conductor size is 28 AWG (0.08042 mm²).

3.4.2 Electrical Characteristics

This section lists measurements of various electrical signals in relation to the single-ended SCSI connection. For these measurements, SCSI bus termination is assumed to be external to the SCSI device.

All signals except GROUND and TEMPWR must be terminated at both ends of the cable. Each signal termination consists of 220 ohms (+ or -5%) to TEMPWR and 330 ohms (+ or -5%) to GROUND and must meet the following specifications:

- Terminators must supply a characteristic impedance of 100 to 132 ohms.
- External terminators must be powered by the TEMPWR line, and units providing terminator power to the cable must have:
 - $V_{\text{TERM}} = 4.25 \text{ to } 5.25 \text{ VDC}$
 - 900 milliamps minimum source drive capability
- The external drive normally supplies terminator power to the SCSI bus.
- When TEMPWR matches the above values, the voltage of released signal lines must be at least 2.5 VDC.
- When a driver asserts a line and pulls it to 0.5 VDC, the current available to the signal line driver may not exceed 48 milliamps. The first two terminators may only supply 44.8 milliamps of this current.
- When at least one device supplies TEMPWR, these conditions may be met by any valid configuration of targets and initiators.

All signals use open-collector drivers. The output characteristics (measured at the connector of the drive) of signals driven by the Python drive are:

- Signal assertion (low-level output voltage):* 0.0 to 0.5 VDC at 48 milliamps sinking
- Signal negation (high-level output voltage):* 2.5 to 5.25 VDC

Signals received by the Python drive have the following characteristics.

- Signal assertion (low-level input voltage):* 0.0 to 0.8 VDC
- Signal negation (high-level input voltage):* 2.0 to 5.25 VDC
- Maximum input load (low-level input current):* - 0.4 at 0.5 VDC
- Minimum input hysteresis:* 0.2 VDC

3.4.3 Single-Ended SCSI Connector – Internal Models 4322 and 4542

The Python 4322 and 4542 internal drives provide a 50-pin, right-angle, dual-row connector on the main PCB at the rear of the drive. The pin assignments for this single-ended connector are listed in Table 3-1.

NOTE: All odd pins except pin 25 are connected to signal ground at the drive. Pin 25 is left open. A signal name or abbreviation preceded by a - (dash) indicates that the signal is active-low.

Table 3-1. Single-Ended SCSI Connector Pin Assignments: Models 4322 and 4542

PIN	ASSIGNMENT
2	-DB(0)
4	-DB(1)
6	-DB(2)
8	-DB(3)
10	-DB(4)
12	-DB(5)
14	-DB(6)
16	-DB(7)
18	-DB(P)
20	GROUND
22	GROUND
24	GROUND
26	TERMINATOR POWER ¹
28	GROUND
30	GROUND
32	-ATN
34	GROUND
36	-BSY
38	-ACK
40	-RST ²
42	-MSG
44	-SEL
46	-C/D
48	-REQ
50	-I/O

¹ The +5V drive supply is available on the SCSI connector as a terminator power option. This pin is connected to the +5V through a diode. The option is a hardware jumper on the rear PCB of internal Python drives with terminator power disabled as a factory default. On external Python drives, terminator power is enabled.

² ANSI defines -RST as a bidirectional pin. On the Python drive, -RST is input only.

3.4.4 Single-Ended SCSI Connector -- Model 4352

The Python 4352 external drive provides two 50-pin, shielded connectors (ANSI Alternative 2) on the rear panel of the drive. These connectors consist of two rows of ribbon contacts spaced 2.16 mm (0.085 in) apart.

These two connectors facilitate adding the drive to a daisy-chain configuration. Either connector is a "SCSI IN" connection; the other is a "SCSI OUT" connection. When the Python drive is the last device in the chain, a terminator is plugged in the "SCSI OUT" connector.

The pin assignments for these single-ended connectors are listed in Table 3-2.

NOTE: *Pins 1-12 and 14-25 are connected to ground. Pin 13 is left open. A signal name or abbreviation preceded by a - (dash) indicates that the signal is active-low.*

Table 3-2. Single-Ended SCSI Connector Pin Assignments: Model 4352

PIN	ASSIGNMENT
26	-DB(0)
27	-DB(1)
28	-DB(2)
29	-DB(3)
30	-DB(4)
31	-DB(5)
32	-DB(6)
33	-DB(7)
34	-DB(P)
35	GROUND
36	GROUND
37	GROUND
38	TERMPWR
39	GROUND
40	GROUND
41	-ATN
42	GROUND
43	-BSY
44	-ACK
45	-RST
46	-MSG
47	-SEL
48	-C/D
49	-REQ
50	-I/O

3.5 Installing the Python 4322 and 4542 Internal Drives

The Python 4322 is a 3.5-inch drive that mounts internal to the computer in a 3.5-inch half high space. The Python 4542 is a 3.5-inch drive with mounting rails and bezel for internal installation in a 5.25-inch half high space.

Installing these two drives consists of a few easy steps:

1. Set operational switches (Section 3.5.1).
2. Mount the drive unit (Section 3.5.2).
3. Complete the power and interface connections (Section 3.5.3).

The installation procedure is the same for both models except physically mounting the unit in the computer. The following text explains the installation steps for both models.

3.5.1 Setting Operational Switches

Before you install the 4322 or 4542 drive in the computer, set the eight-position switch bank mounted on the PCB at the rear of the drive.

Figure 3-1 illustrates the switchbank location for the 4322 drive.

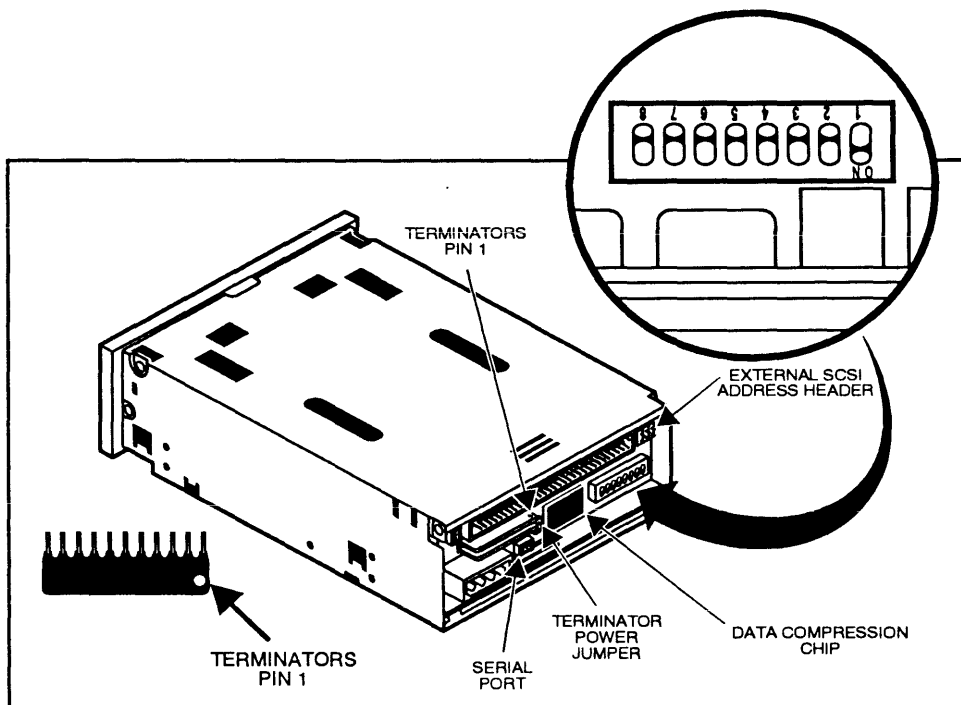


Figure 3-1
Python 4322 Switchbank Access

Figure 3-2 illustrates the switchbank location for the 4542 drive.

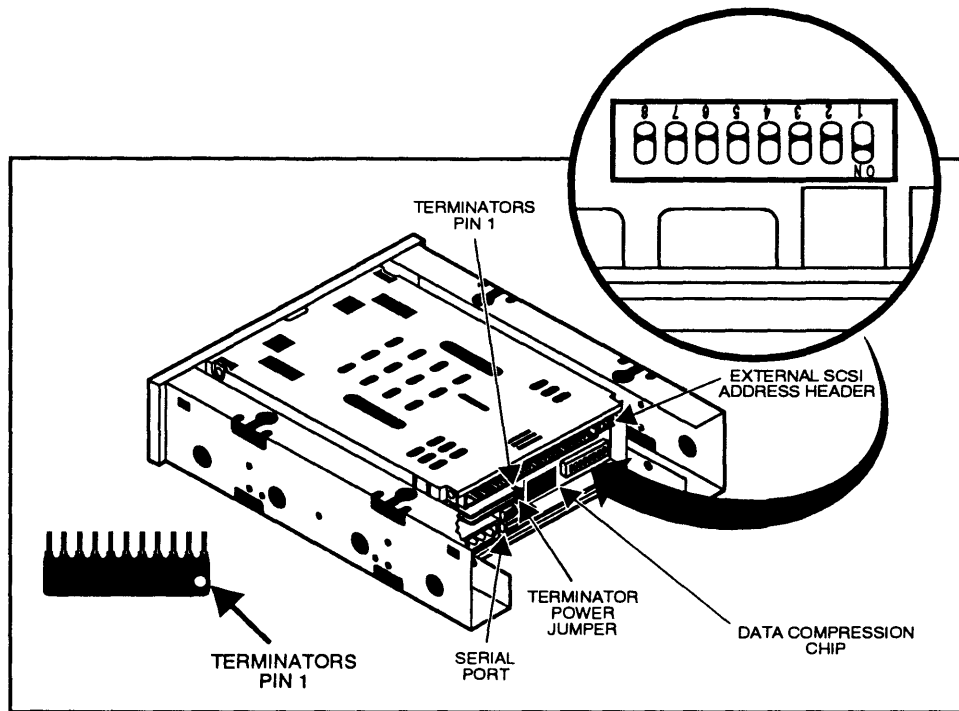


Figure 3-2
Python 4542 Switchbank Access

These switches allow you to set up the following configuration options:

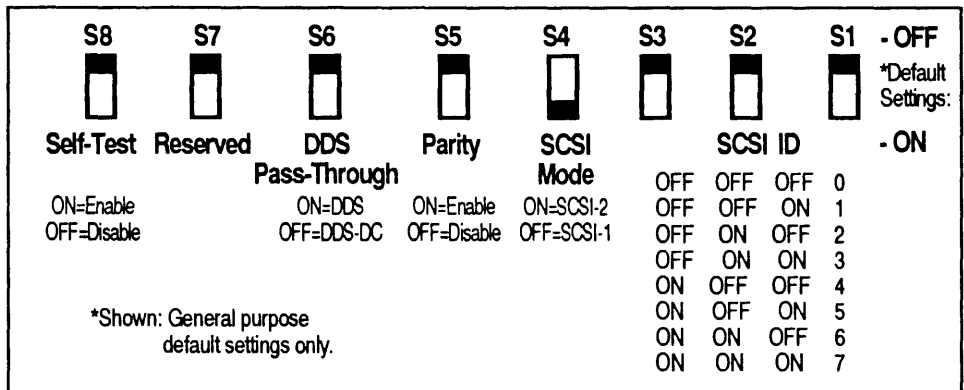
- SCSI device address (S1 through S3): Default = SCSI ID 0 (S1 through S3 = OFF)
- SCSI mode at power-up (S4): Default = SCSI-2 (S4 = ON)
- Parity check enable/disable (S5): Default = Parity disabled (S5 = OFF)
- DDS pass-through mode enable/disable (S6): Default = Pass-through mode disabled (S6=OFF)

NOTE: When S6 is OFF, DDS-DC data compression is enabled. When S6 is ON, DDS-DC data compression is disabled.

- Power-on self-test enable/disable (S8): Default = Power-on self-test disabled (S8 = OFF)

NOTE: Switch S7 is reserved and must be in the OFF position.

Figure 3-3 shows the default settings for the 4322 and 4542 drives (view looking from the rear of the drive).



**Figure 3-3
Dip Switch Default Settings**

NOTE: *The drive must be turned OFF; then, ON in order for switch settings to take effect, or a SCSI Bus Reset must be received.*

If the default settings are correct for your system, go to Section 3.5.2. DDS-DC data compression enabled (S6 = OFF) is also set as a factory default.

If you need to change any default settings, refer to the appropriate following section; make the changes and then go to Section 3.5.2.

SCSI Device Address (S1 through S3)

The three switches S1 through S3 correspond to the SCSI device address identification bits 0 (LSB) through 2 (MSB), respectively. Figure 3-4 shows the switch settings for the eight possible SCSI device addresses. The default setting is SCSI device address 0 (S1 through S3 = OFF). Be sure that no other device on the SCSI bus has the same SCSI address.

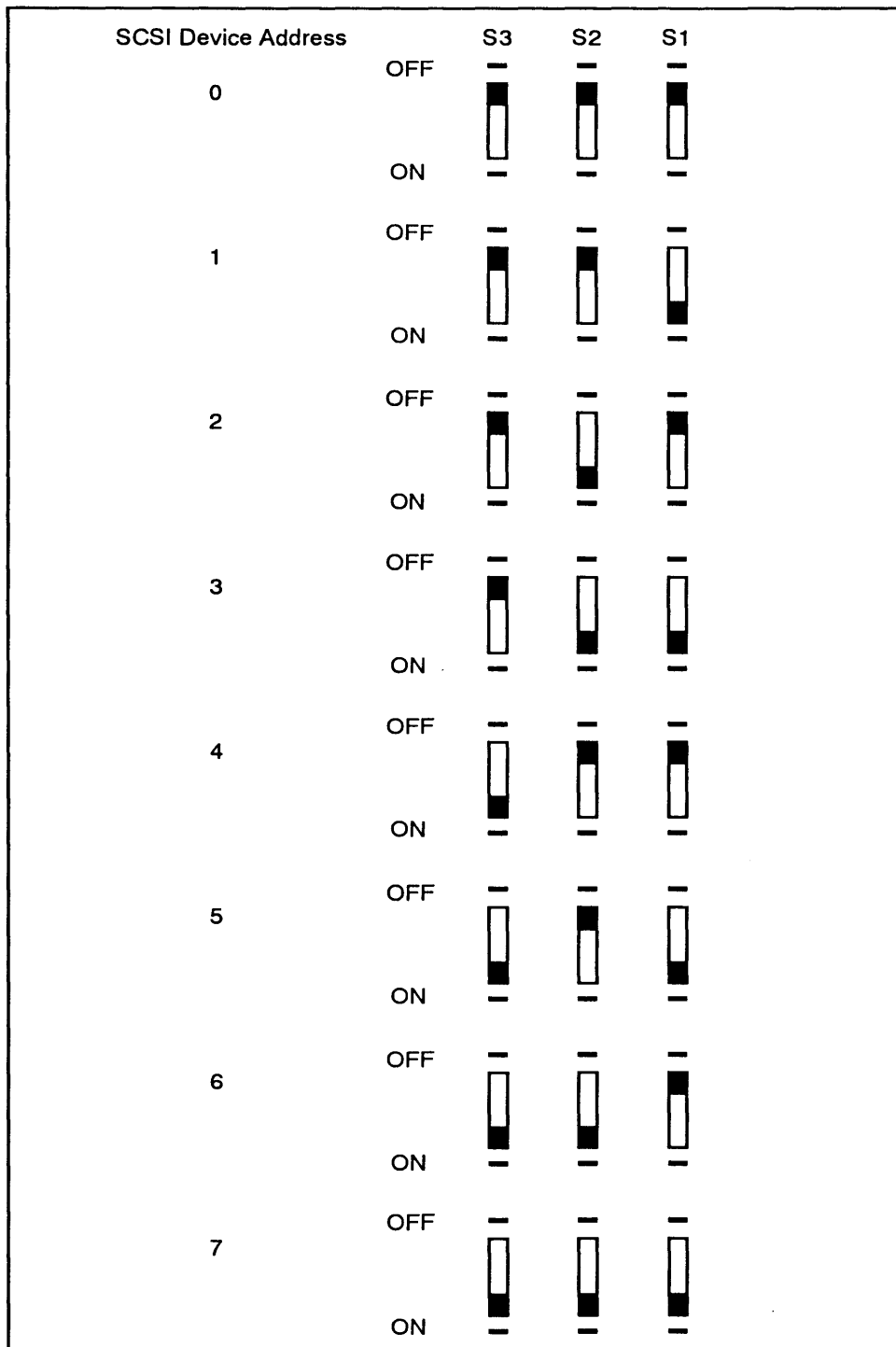


Figure 3-4. SCSI Device Address Selection (4322 and 4542)

SCSI Mode at Power-Up (S4)

The S4 switch selects either SCSI-1 or SCSI-2 as the default operational mode for the Python SCSI bus at power-up. The default is SCSI-2 (S4 = ON), a superset of SCSI-1. Chapter 5 provides more information about the SCSI-1 and SCSI-2 interfaces.

S4 = ON selects SCSI-2
S4 = OFF selects SCSI-1

Parity Check Enable/Disable (S5)

The S5 switch enables or disables parity checking for the SCSI bus. The default is parity disabled (S5 = OFF).

S5 = ON enables parity checking
S5 = OFF disables parity checking

DDS Pass-Through Mode Enable/Disable (S6)

The S6 switch enables or disables DDS pass-through mode. The default is DDS pass-through mode disabled (S6=OFF).

S6 = ON enables DDS pass-through mode
S6 = OFF disables DDS pass-through mode

NOTES: *When S6 is OFF, DDS-DC data compression is enabled during writing. When S6 is ON, DDS-DC data compression is disabled. When reading, DDS-DC compressed data is always decompressed, regardless of the position of S6.*

The function of the S6 switch can be over-ridden by the proper SCSI MODE SELECT command issued from the host computer. Regardless of the position of S6, the MODE SELECT command can enable or disable data compression.

Reserved Switch (S7)

The S7 switch is reserved and should be left in its factory default setting.

Power-on Self-Test Mode Enable/Disable (S8)

The S8 switch enables or disables execution of power-on self-test diagnostics when the power comes ON. The default is power-on self-test mode disabled (S8 = OFF). If ON, the drive responds to SCSI commands after successful completion of the test (about 5 seconds).

S8 = ON enables power-on self-test mode
S8 = OFF disables power-on self-test mode

SCSI-1/SCSI-2 Interface

The Python 4322 and 4542 drives allow you to choose either SCSI-1 or SCSI-2 as the default interface with the computer system. However, the factory default for the drives is the SCSI-2. (Chapter 5 provides summary information about the SCSI-1 and SCSI-2 interfaces.)

The SCSI-1 interface conforms to the ANSI X3.131-1986 standard. SCSI-2, which conforms to ANSI X3T9.2, Revision 10, is a standard that augments the SCSI-1 command set and generally includes everything in SCSI-1.

If the hardware default setting is SCSI-2 for power up, issuing a CHANGE DEFINITION command allows SCSI-1 to be selected until the next power up. To turn data compression OFF, the host computer issues a SCSI MODE SELECT command.

Terminator Power

At the rear of the Python 4322 and 4542 drives, a 2-pin header allows you to enable +5-volt terminator power if needed for terminators or other SCSI devices. The factory default for internal Python DDS-DC drives is with terminator power disabled (jumper shunt over one pin). To enable terminator power, place the jumper shunt over the two pins. (See Figure 3-1 for the 4322 drive and Figure 3-2 for the 4542 drive.) Be sure it is firmly in place.

CAUTION: *If the jumper is installed, be careful not to short the TERMPWR signal to ground.*

The terminator power fuse is located beside the terminator power jumper to prevent damage to drive components in case the terminator power is shorted. If terminator power is enabled and the SCSI cable is connected upside down, this fuse may blow to prevent damage to the drive itself. In that case, return the drive to an authorized repair facility.

Serial Port

A unique feature of the Python internal 4322 and 4542 drives is the 3-pin serial port available at the rear of the drive unit. This serial port allows access to internal drive diagnostics stored in firmware. These internal diagnostics allow standalone testing of the drive through connection to an external computer.

Should this capability be needed, contact your Archive sales representative about receiving the Archive Python Serial Port Diagnostics Kit, which includes a TTL logic converter, manual, cables, and host PC software.

In addition to the serial port connector, another larger connector on the PCB at the rear of the drives allow access to the drive for manufacturing testing. (See Figures 3-8 and 3-9.)

External SCSI Address Port

A 6-pin header on a PCB at the rear of the drives allow remote SCSI address selection. For example, if an OEM chooses to enclose the Python 4322 or 4542 in an external housing, the 6-pin header allows the SCSI address selection to be brought out the rear panel. Figure 3-5 illustrates this 6-pin header (JP6).

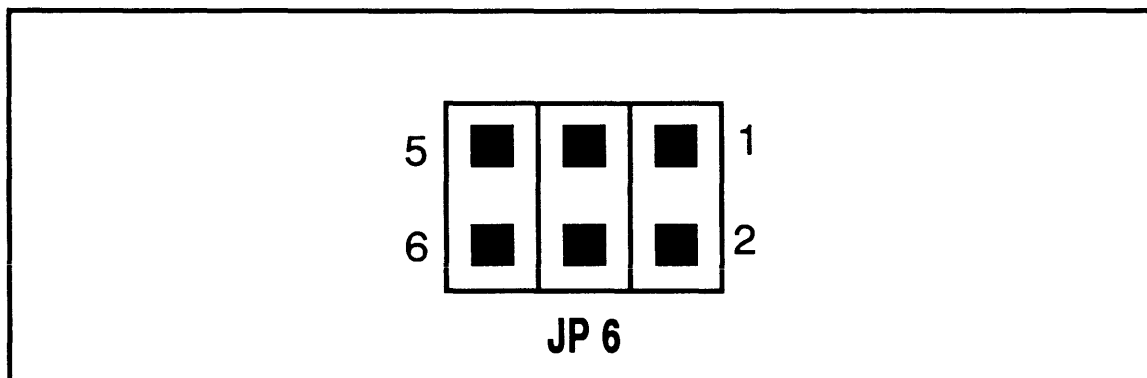


Figure 3-5
JP6 External SCSI Address Port

NOTE: Follow the switch settings for selecting the SCSI address as illustrated in Figure 3-4.

To turn S1, S2, or S3 ON, place a jumper shunt over the two vertical pins for each switch required.

To turn S1, S2, or S3 OFF, remove the jumper shunt from the two vertical pins for each switch required.

Set the DIP switches S1, S2, and S3, all to OFF to use the external SCSI address port.

3.5.2 Mounting the Drive Unit

The Python 4322 and 4542 internal drives can be installed in three different orientations: one horizontally (eject button right) and two vertically (eject button up or eject button down).

The 4322 drive chassis contains threaded mounting holes for M3.0 metric screws. Four are located on the bottom and five are on each side of the frame. See Figure 3-6.

The 4542 drive chassis contains threaded mounting holes for M3.0 metric screws. Four are located on the bottom, and eight are on each side of the frame. See Figure 3-7.

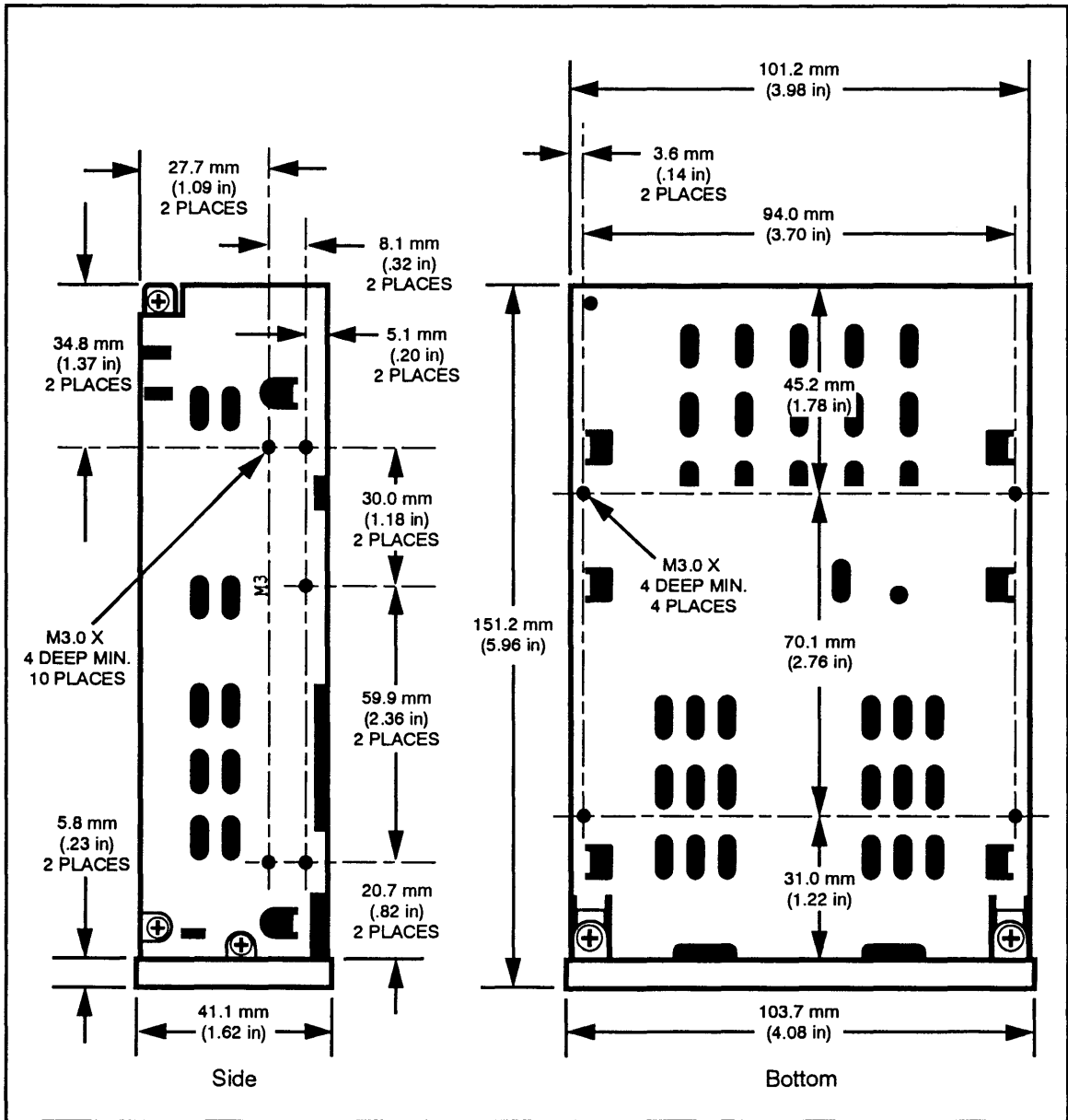


Figure 3-6
Mounting Hole Locations (4322)

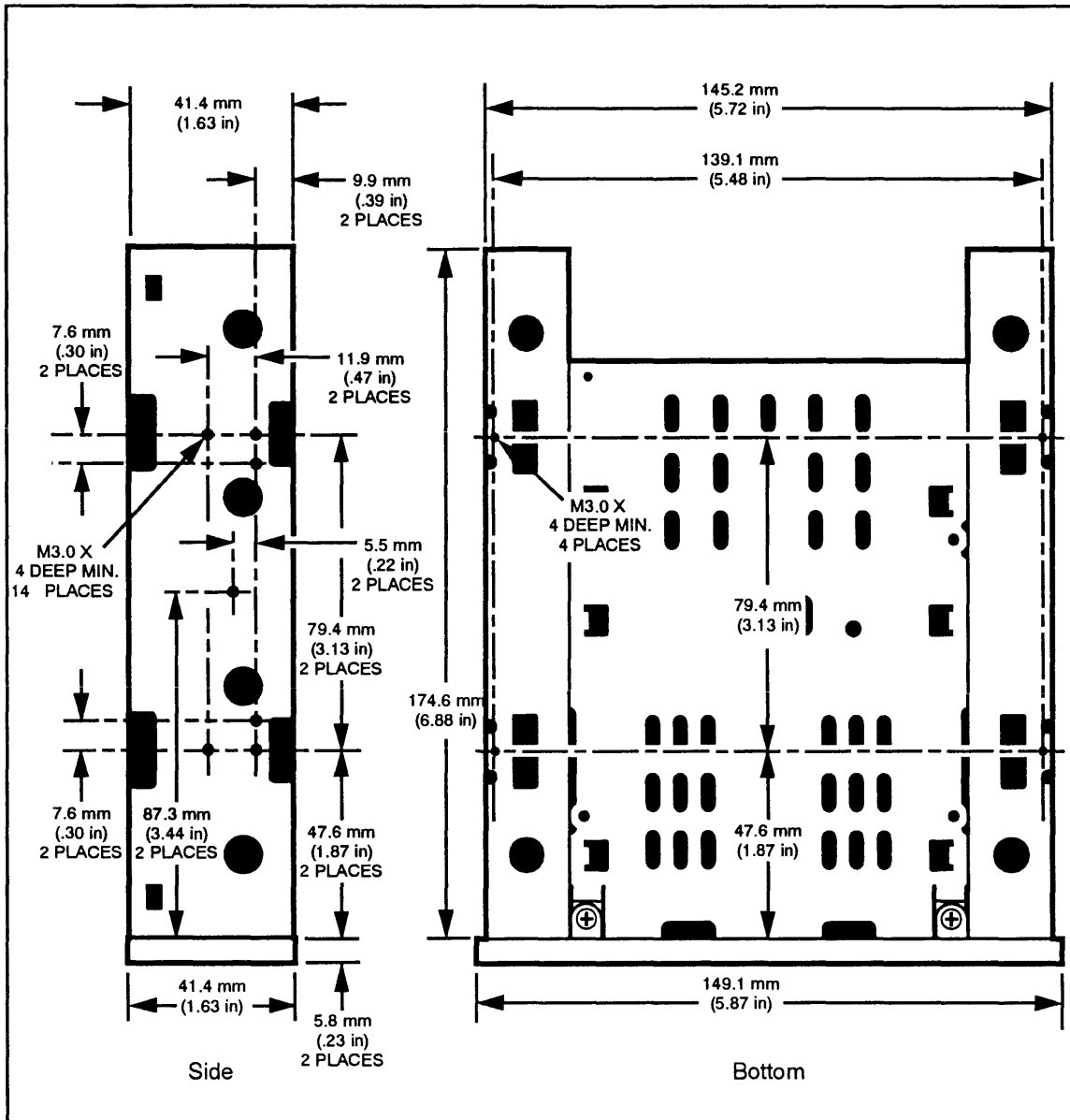


Figure 3-7
Mounting Hole Locations (4542)

3.5.3 Completing Connections

The power and interface connectors for the Python 4322 and 4542 internal drives are located at the back of the drive unit. Figure 3-8 illustrates these connections for the 4322 drive, and Figure 3-9 illustrates these connections for the 4542 drive.

NOTE: Turn off all power before inserting connectors. Pin 1 on the SCSI connector is to your right as you look at the back of the drive. (See Figures 3-8 and 3-9.) Your SCSI cable should have Pin 1 highlighted by a color strip. Be sure to mate Pin 1 on the cable to Pin 1 on the drive. Failure to do so could make the Python drive inoperative.

- The recommended *power mating connector* requires an AMP 1-48024-0 housing with AMP 60617-1 pins or equivalent.
- The mating interface connector for 4322 and 4542 drives is a single-ended connector as described in Section 3.4 with pin assignments as listed in Table 3-1. When you make the connection, be sure pin 1 of the connector aligns with pin 1 on the Python SCSI connector. (See Figures 3-8 and 3-9.)
- The layout for the mating interface connector for the 6-pin external SCSI address header is shown in Figure 3-5.

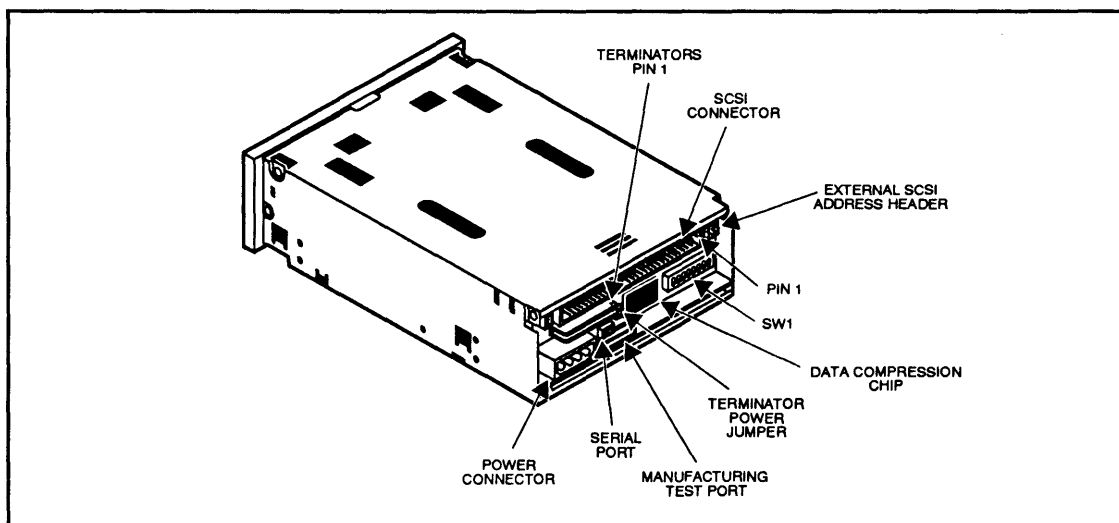


Figure 3-8
Power and SCSI Connectors -- 4322 DAT Drive

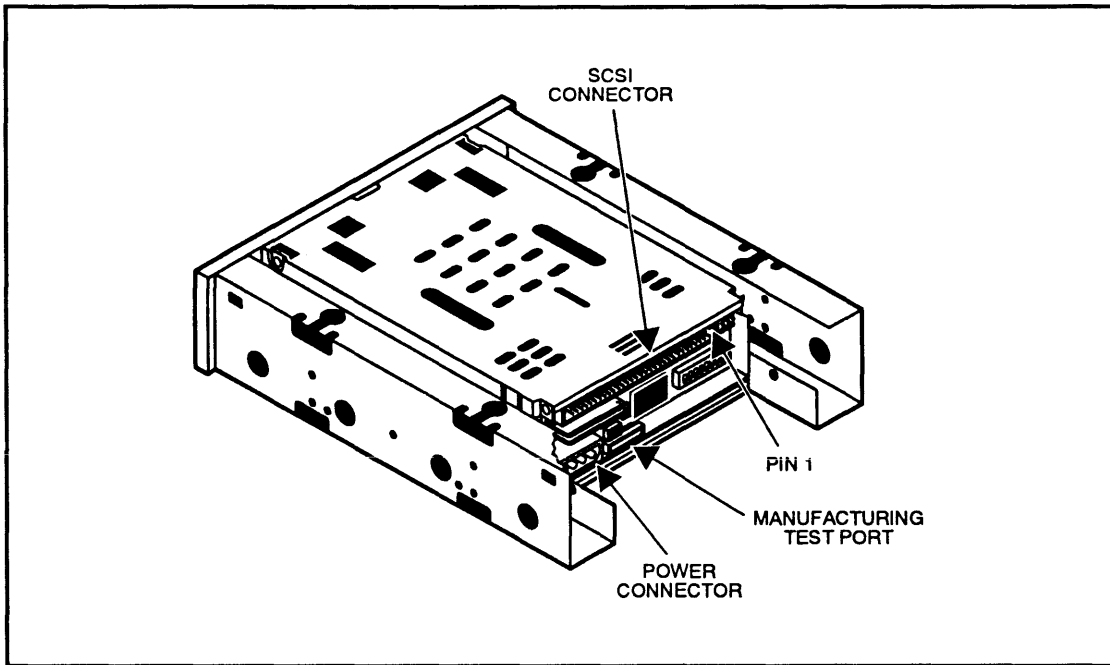


Figure 3-9
Power and SCSI Connectors -- 4542 DAT Drive

3.6 Installing the Python 4352 External Drive

The Python Model 4352 is a compact external drive that connects as a turnkey subsystem to the computer. The drive may be operated vertically or horizontally. Generic Python 4352 external drives are set by default at the factory for:

- SCSI-2 as the power-on interface
- Parity disabled
- Power-on self-test disabled
- Terminator power is supplied to the SCSI bus
- DDS pass-through mode disabled (data compression ON)

NOTE: *General purpose default settings shown previously in Figure 3-3 also apply to the Model 4352.*

Installing these drives consists of a few easy steps:

1. Select SCSI address (Section 3.6.1).
2. Complete the interface connection (Section 3.6.2).
3. Complete the power cord connection (Section 3.6.3).

3.6.1 Selecting the SCSI Address

The rear panel of the external Python drive unit contains the SCSI address selection push switch, the two interface connectors, the ON/OFF switch, digital audio output jack (some models), and the power cord connection. Figure 3-10 illustrates the rear panel.

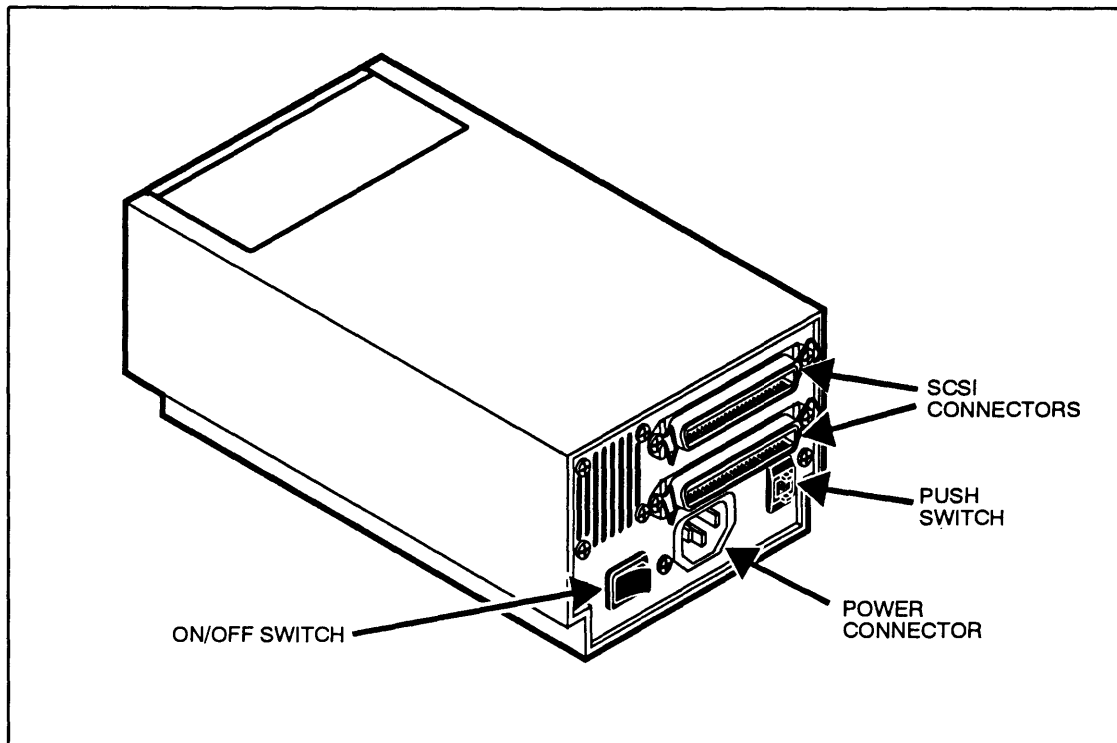


Figure 3-10
Python 4352 Rear Panel

Locate the SCSI address push switch. Select the SCSI address for the Python drive by pressing the + or - button until the desired address (0 through 7) appears in the window. Turn the unit power OFF; then, ON.

3.6.2 Completing the Interface Connection

The Python 4352 drive provides two SCSI connectors to allow daisy chaining. (See Figure 3-10.) Either connector can connect to the host computer or any SCSI device in a daisy chain.

NOTE: *Turn off all power before connecting cables and the terminator.*

- When the Python drive is the last drive in the chain, a single interface cable is attached to one connector, and a terminating plug is installed in the other connector.
- When the Python drive is within the chain, the interface cable from the preceding device is connected in one connector; an interface cable is also connected from the other connector to the following device. In this case, no termination is required.

Figure 3-11 illustrates these daisy-chain connections.

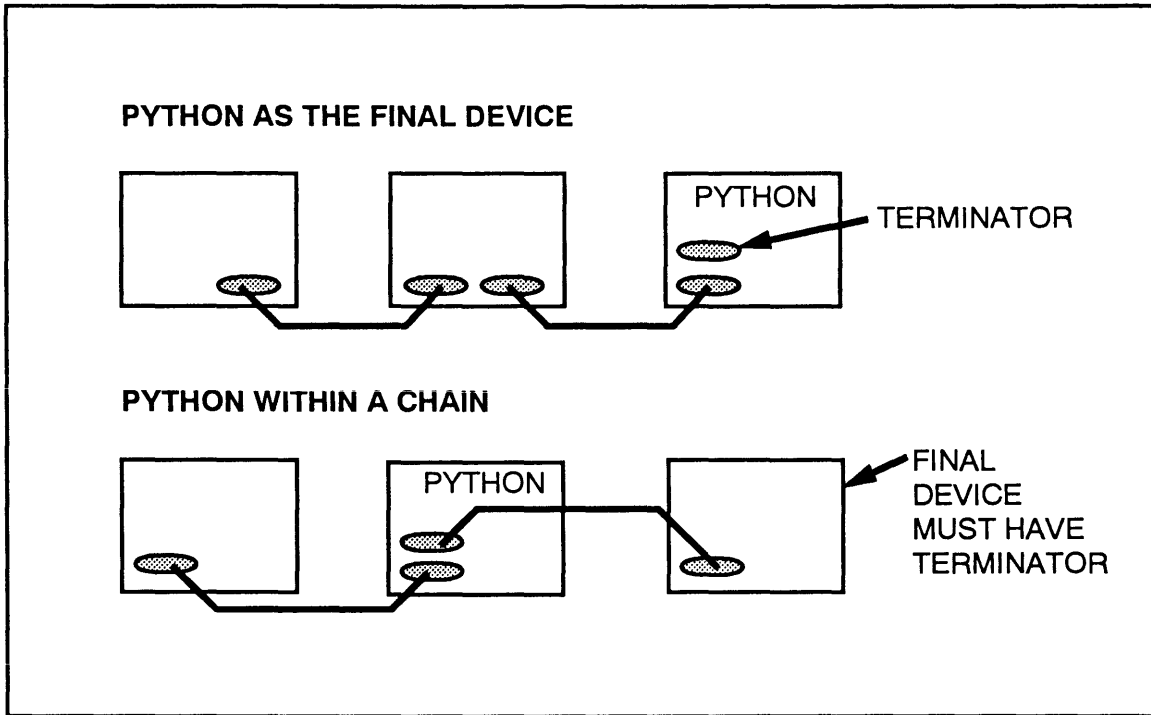


Figure 3-11
Daisy Chain Diagram

The same type of mating connector is used for either of the daisy-chain connections. The mating interface connector for Model 4352 is a single-ended connector as described in Section 3.4 with pin assignments as listed in Table 3-2.

3.6.3 Connecting the Power Cord

See Figure 3-10 for the location of the power cord connector. Insert the power cord mating connector into the connector on the rear panel. Be sure the connection is secure. Plug the other end of the power cord into a wall receptacle.

CHAPTER 4

DRIVE OPERATION

4.1 Introduction

This chapter describes important operational procedures for the Python DDS-DC drives. It covers the following topics:

- Data compression operation
- Operation of the front panel LEDs
- Loading and unloading a cassette
- Using a blank cassette
- Using a cassette with data on it
- Using a prerecorded audio cassette
- Loading revised SCSI firmware (updating flash EEPROM)

NOTE: *Required drive maintenance including using a cleaning cassette is explained in Chapter 9.*

4.2 Data Compression Operation

Default operation for the Python DDS-DC drives is with data compression enabled---the drive automatically compresses all data written to tape and decompresses all compressed data read from tape, in DDS-DC format. Data with high degrees of redundancy, such as structured database files, can be compressed most efficiently, often at a ratio of 4:1 or more. Data with little redundancy, such as executable programs, can be compressed least.

A unique feature of the Python DDS-DC drives is expansion minimization. To provide this feature, the drive continuously monitors the compression ratio. If the "compressed" version becomes larger than the original version, data compression would be automatically turned OFF. Although this situation is very rare, it could occur, for example, if the drive is trying to compress data that has already been compressed.

To switch between compressed mode (writing compressed data, DDS-DC format) and noncompressed mode (writing uncompressed data, DDS format), the host computer executes a SCSI MODE SELECT command.

On internal Python DDS-DC drives, a hardware DIP switch (S6) can also be used to enable or disable data compression.

NOTE: *This switch is designated "DDS pass-through mode" and leaving this switch OFF enables DDS-DC compression. External Python DDS-DC drives are factory set with DDS-DC enabled. See Chapter 3 for more information about this switch.*

The SCSI MODE SELECT command can switch the Python DDS-DC drive into compressed or uncompressed mode for writing data regardless of the position of the S6 DIP switch. When reading, the drive automatically selects compressed or uncompressed mode.

4.3 Front Panel LED Operation

The front panel of the Python DDS-DC drives contains two rectangular LEDs. One rectangular LED indicates the drive status; the other, the cassette status. These two indicators provide operating information for normal as well as error conditions.

The 4352 external subsystem also contains a round, green power-on LED on the front panel.

The amber, rectangular Drive Status LED indicates the following conditions:

- When ON (lit), the drive is reading or writing the tape. (SCSI or DAT activity is present.) During a SCSI Prevent Media Removal command, the LED is always ON.

NOTE: *Do not push the eject button while this LED is ON. If you do, the operation in progress is aborted and the cassette is ejected, possibly causing a loss of data.*

- When flashing rapidly, a hardware fault has occurred or dew (excessively high humidity) was detected. If this situation occurs immediately after power-on (and switch 8 is ON), the power-on self-test may have failed. In that case, the drive will not operate.

NOTE: *Some guidelines for operation in high humidity conditions are listed in Chapter 9.*

The green, rectangular Cassette Status LED indicates the following conditions:

- When ON (lit), a cassette is inserted and does NOT generate excess errors.
- When flashing slowly, a cassette is inserted but generates excessive errors beyond a predefined DDS error threshold. First, clean the drive heads using an approved DDS DAT cleaning cassette (such as the Archive Model M7301).

NOTE: *As routine maintenance, the drive heads should be cleaned after every 25 hours of operation. See Chapter 9 for more information about maintenance.*

If the LED continues flashing or flashes when ejecting the cassette, use a new cassette for future writes. Otherwise, operation is proceeding normally. This signal is a warning only.

- When flashing slowly in conjunction with the amber LED, a prerecorded audio cassette is inserted and is being played automatically.
- When flashing rapidly, the drive could not write the tape correctly (maximum rewrite count exceeded). The WRITE operation failed. First, clean the drive heads using an approved DDS DAT cleaning cassette, such as the Archive Model M7301. If the LED continues flashing, use a new cassette for future writes.

The round, green LED on the 4352 drive illuminates when power is applied to the drive.

Figures 4-1, 4-2, and 4-3 show the front panels of the 4322, 4542, and 4352 drives, respectively.

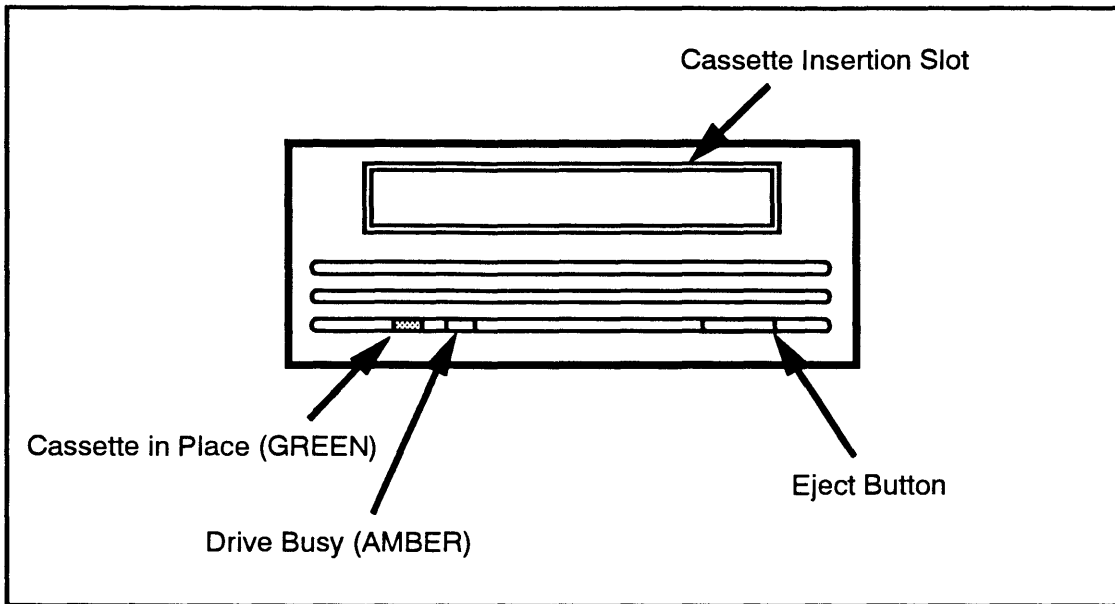


Figure 4-1. Python 4322 Front Panel

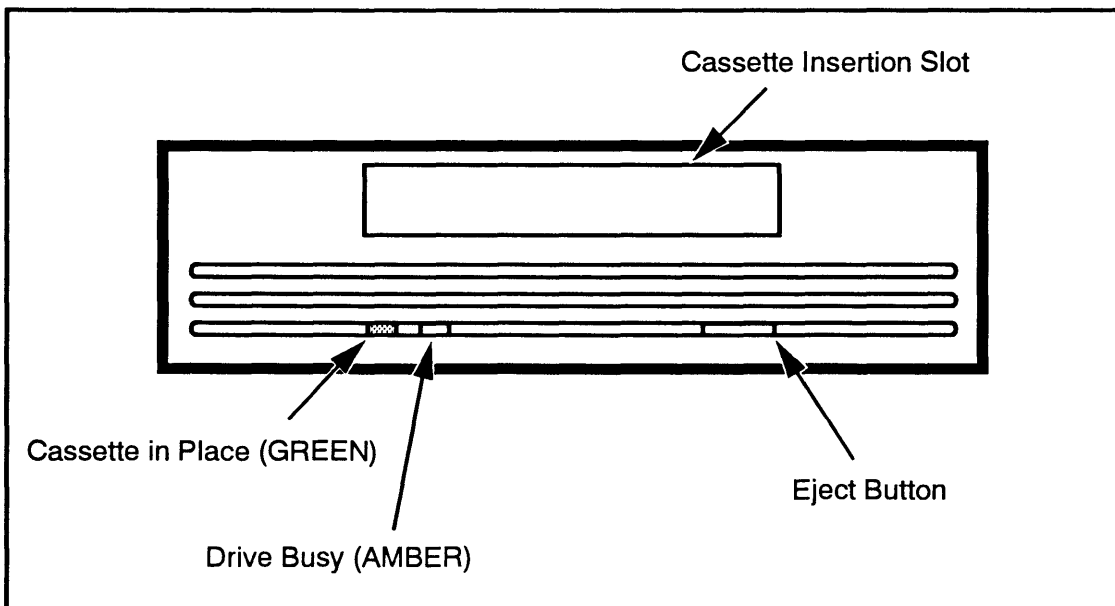


Figure 4-2. Python 4542 Front Panel

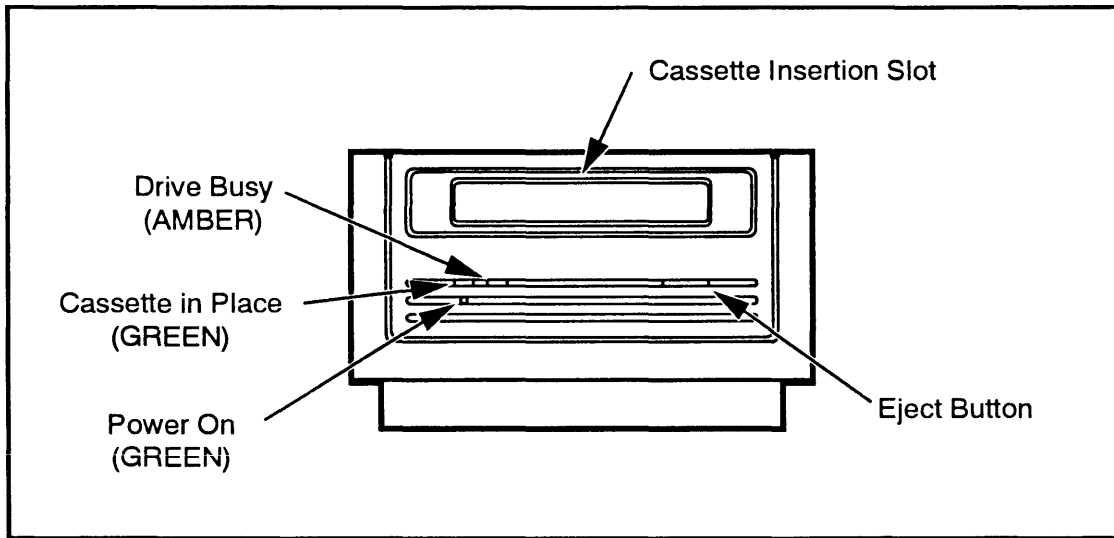


Figure 4-3. Python 4352 Front Panel

Table 4-1 summarizes the operation of the front-panel LEDs. See the previous explanations to remedy fault conditions.

Table 4-1. LED Summary

LED	ACTION	MEANING
Amber	ON (lit)	The drive is reading or writing the tape.
Amber	Flashing Rapidly	A hardware fault occurred, or dew was detected.
Green	ON (lit)	A cassette is inserted and does NOT generate excess errors.
Green	Flashing Slowly	A cassette is inserted but generates excessive errors beyond a predefined error threshold. (Warning only)
Green	Flashing Slowly (with amber LED flashing)	A prerecorded audio cassette is inserted and being played automatically.
Green	Flashing Rapidly	The drive could not write the tape correctly. (Error)
Green, round (4352 ONLY)	ON (lit)	The 4352 drive is powered on.

4.4 Loading/Unloading the Cassette

The cassette insertion slot on the front panel of Python drives provides easy access to the drive. This section explains loading and unloading a cassette under normal operating conditions. It also explains the manual procedure for removing a cassette abnormally lodged in the drive. Under a few exceptional conditions -- such as a power outage, you may need to manually unload a cassette.

4.4.1 Loading/Unloading a Cassette (Normal Operation)

The Python DDS-DC drives have a front-loading cassette insertion mechanism that allows the operator to easily load the cassette by pushing against the middle part of the cassette opening until it is fully recessed into the cassette insertion slot. Insert the cassette with the arrow on the top of the cassette entering the slot first. Figure 4-4 illustrates cassette loading (4322 internal drive shown).

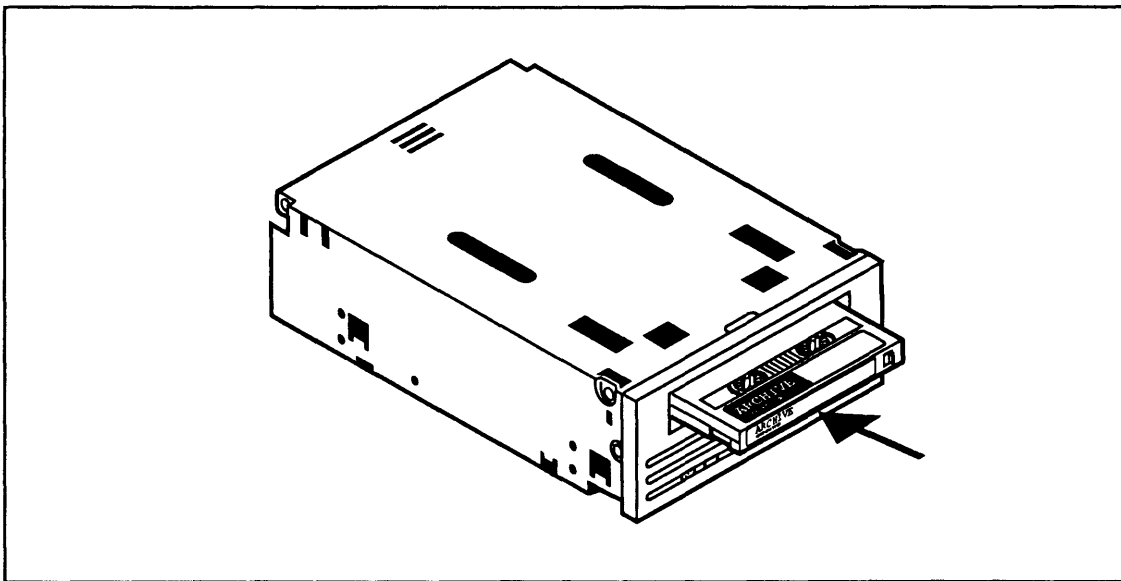


Figure 4-4
Cassette Loading (4322 DAT Drive)

You unload the cassette by pressing the eject (tape unloading) button on the front panel. (See Figures 4-1, 4-2, and 4-3 of the front panels for the location of the eject button for your drive.) Once you press the eject button, the Python drive updates the system log with a running count of drive/tape "soft" errors, rewinds the tape, and then ejects the cassette. It is partially ejected and can then be easily removed from the drive.

NOTE: *The time between pressing the eject button and cassette ejection may be several seconds. Do not power down the Python external drive or the internal drive host computer during this time.*

4.4.2 Unloading a Cassette (Manual Operation)

If a power outage occurs while a cassette is loaded or the automatic removal procedure previously explained fails, you may want to manually unload a cassette from the Python drive. The following steps outline the manual cassette unloading and removal procedure.

NOTE: *This procedure is basically the same for all three Python DDS-DC form factors---only the physical housings are different.*

1. Disconnect power to the computer; then disconnect the unit from the computer. That is, remove the SCSI connections to the host computer.
2. Remove the power connections. For internal models, disconnect the power connection with the host computer. For the 4352, remove the power cord from the drive.
3. For internal drives, remove the drive from the computer.

For the 4352 drive, turn the unit upside down and remove the four screws (two screws on each side) that attach the drive cover to the chassis unit. Remove the drive cover and retain the screws.

Figure 4-5 shows the Python 4322 3.5-inch drive with the cover removed; Figure 4-6 shows the Python 4542 internal drive with the cover removed; Figure 4-7 shows the Python 4352 external drive with the cover removed.

4. Turn the mode motor shaft counterclockwise (from front view) until the posts retract into the cassette, and the mode motor stops.
5. Push the front roller in and turn it clockwise (from top view) until the tape is wound on the supply reel, and the roller stops.
6. Rotate the rear cassette gear counterclockwise (from top view) until the cassette ejects.
7. Reassemble the drive and reconnect it to the computer.

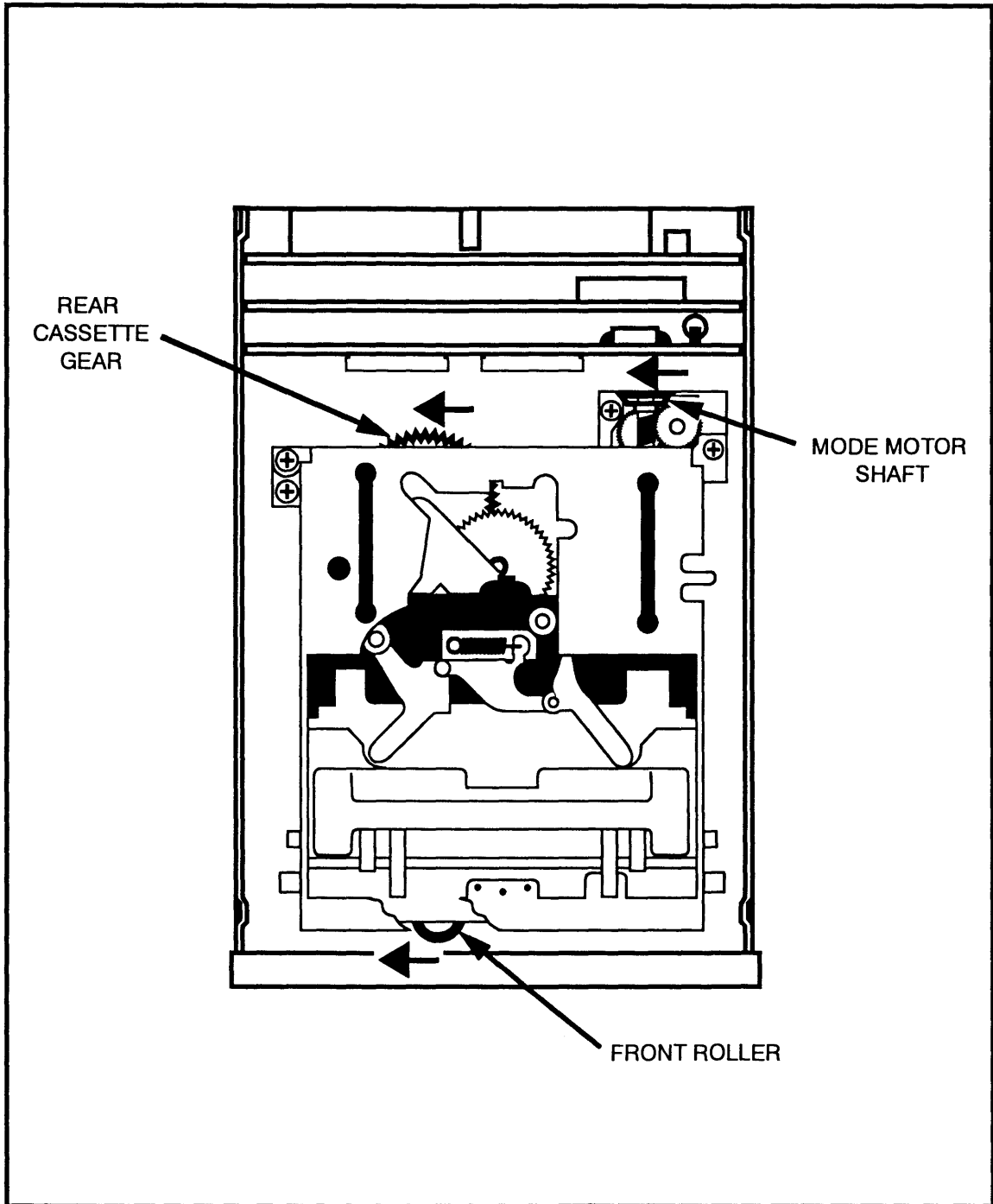


Figure 4-5
Manually Removing a Cassette -- Python 4322 Drive

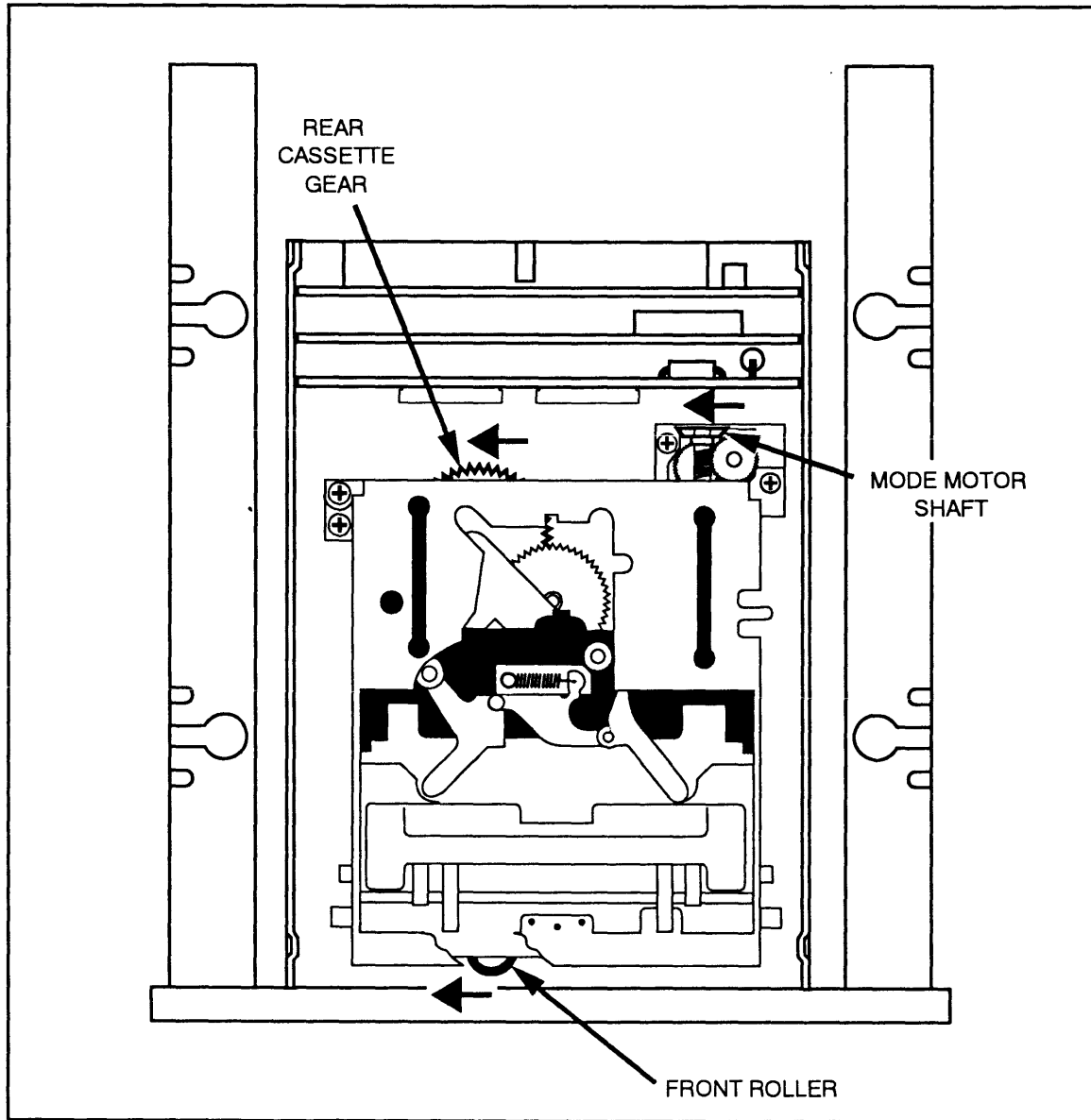


Figure 4-6
Manually Removing a Cassette -- Python 4542 Drive

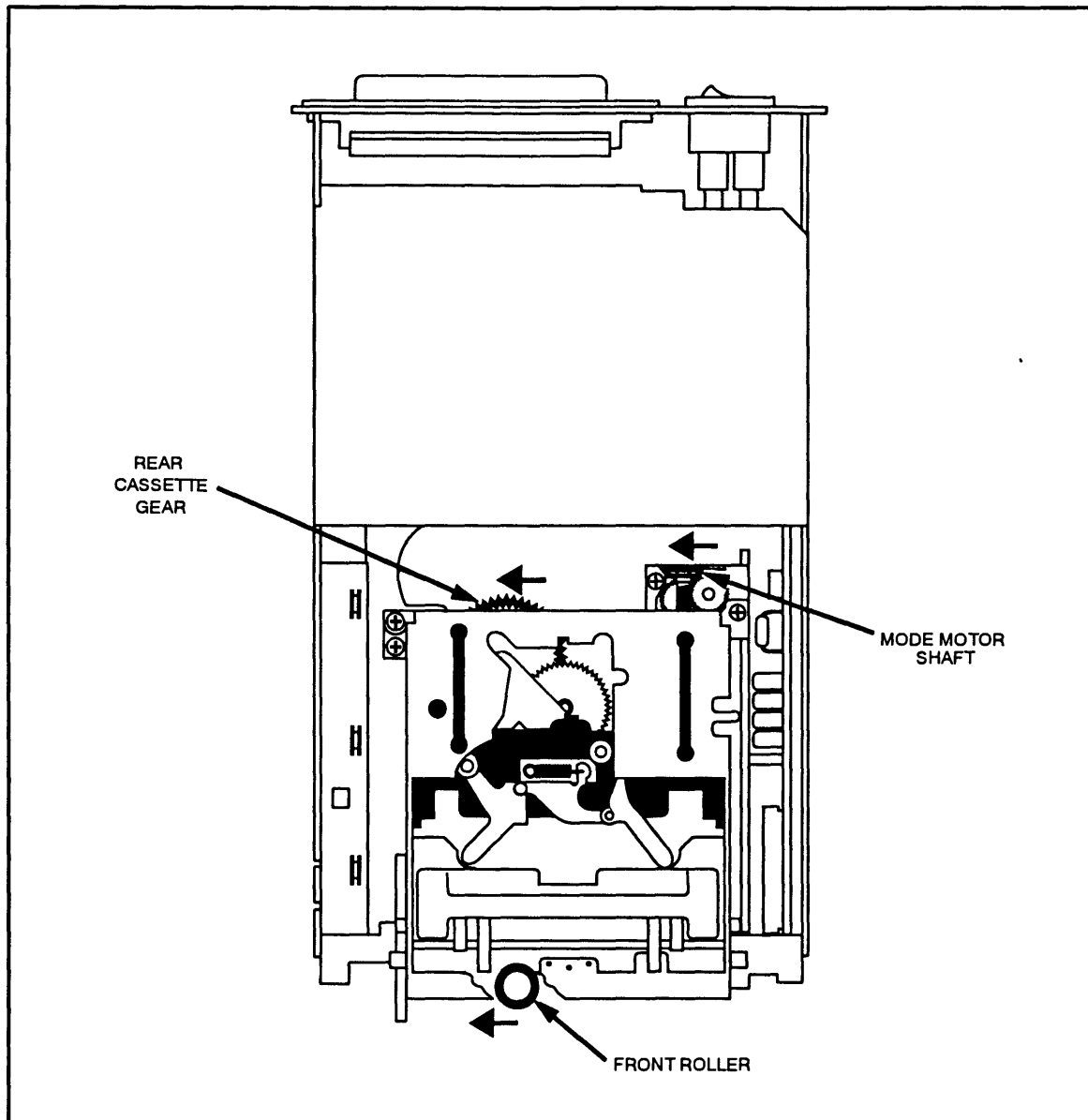


Figure 4-7
Manually Removing a Cassette -- Python 4352 Drive

4.5 Using a Blank Cassette

When you insert a blank DDS cassette into the Python drive for the first time, it is automatically initialized. The Python drive first detects that the tape is blank (about 10-12 seconds). It then initializes the tape when it receives a command that causes writing to the tape.

NOTE: *Initializing the tape takes about 30 seconds. Ejecting the cassette before the initialization is completed causes the procedure to abort. The initialization then restarts from the beginning the next time a WRITE command is received.*

If the first WRITE command stores less than 126 kilobytes of data, the data is placed in the buffer until the next WRITE command exceeds the 126-kilobyte buffer capacity and forces the stored data to actually be written to tape.

The following steps outline a typical sequence for using a blank cassette.

1. Gently push the blank DAT cassette into the cassette insertion slot on the front panel with the arrow on the top side of the cassette entering the opening first. (See Figure 4-4.)

Once the cassette is partially inserted, the drive mechanism automatically completes the cassette insertion and proper positioning.

The amber and green rectangular LEDs on the front panel go ON as the drive checks the cassette to determine its state (blank, write-protected, prerecorded audio, firmware update, and etc.) and positions to the data area, which takes about 10-12 seconds.

2. Start the software application and issue a command.

For example, if you want to back up a file, issue the appropriate command or make the appropriate menu selections from backup application software. The Python drive begins initializing the tape before completing the backup (WRITE) operation. WRITE operations are completed during the initializing operation without delay until all internal buffers are filled.

3. After completing the backup and after the amber rectangular LED on the front panel is OFF, push the eject (tape unload) button on the front panel to remove the cassette. The Python buffer is then emptied to tape, and the tape is rewound.

The Python's data buffer is also emptied to the tape in these three cases:

- A REWIND command is issued.
- The eject button is pushed.
- A delay as defined by MODE SELECT or the one-minute default delay time in which no activity occurs.

Before the Python drive ejects the cassette it automatically updates the system log, which requires a few seconds; then it rewinds and ejects the cassette. When ejected, the cassette is pushed out of the cassette insertion slot to a half-way position for easy removal.

4.6 Using a Cassette Containing Data

The sequence for writing a cassette that already contains data is similar to the blank cassette sequence except the cassette is not automatically initialized by the drive. A brief delay occurs when the cassette is first inserted as the Python drive identifies the cassette type and state, and positions to the data area.

4.7. Using a Prerecorded Audio Cassette

Inserting a prerecorded audio cassette in the Python drive automatically plays the cassette. A brief delay after you insert the cassette occurs as the drive determines that the cassette is in the audio format. After that brief delay, the audio tape is played. However, no output jack (either digital or analog) is available as a standard feature in Python DDS-DC drives to allow the drive to interface to sound reproduction equipment. Connection must be made to an internal digital audio signal point. Contact your Archive sales representative if this capability is a major component of your specific application.

4.8 Loading Revised Firmware via Archive Firmware Cassette

Python "NP" and "XP" DDS-DC drive models support flash EEPROM. Flash EEPROM enables you to download new firmware when revisions to firmware are released. Firmware revisions are released on specially encoded cassettes that are automatically recognized by these Python drives. These firmware revisions are available for qualified OEMs only from Archive Corporation through ARDAT, Inc.

To load a firmware upgrade tape, follow these steps.

1. Power on the host system with the Model 4322NP, 4542NP, or 4352XP Python drive installed.
2. Insert the Archive firmware upgrade cassette.
3. The Python drive automatically recognizes the tape and loads the new firmware. This process takes about 30 seconds. The drive automatically ejects the cassette once the firmware is loaded. No LEDs are illuminated if the firmware loaded correctly.

NOTE: *Once the firmware upgrade cassette is inserted in the drive, it is important that no power interruption occurs while the firmware is loading. DO NOT POWER OFF THE DRIVE AT THIS TIME. If a power interruption occurs, the firmware may not be loaded correctly, and the drive may not operate properly.*

If a problem occurs during the firmware loading process, the cassette is ejected, and the amber LED on the front panel flashes. In that case, the firmware upgrade cassette may be defective, or the drive may not be operating correctly.

If after a repeat loading of the firmware cassette, the same condition is observed, contact your Archive sales representative.

Firmware upgrade cassettes are available to qualified Archive OEM customers. Contact your Archive sales representative for information.

Revised firmware can also be loaded into EEPROM using the SCSI WRITE DATA BUFFER command and through the DATMON software supplied with the Archive Python Serial Port Diagnostic Kit product. Refer to Section 8.3.3 for more information about flash EEPROM. Refer to Appendix C for more information about the SCSI WRITE DATA BUFFER command.

NOTES

CHAPTER 5

SCSI INTERFACE

5.1 Introduction

The Python DDS-DC DAT drives provide an embedded Small Computer System Interface (SCSI) controller for communications between the host computer and the DAT drive. The Python drives support the SCSI-1 (ANSI Standard X3.131-1986) or SCSI-2 (ANSI X3T9.2 WORKING DRAFT Revision 10) interfaces.

The factory default interface at power up is for the SCSI-2 interface. (For Python internal drives, the SCSI interface is selected by S4 on the 8-position DIP switch. See Chapter 3 for more information about this switch.) The SCSI-2 interface provides a modified MODE SENSE page to control and report on data compression operations. Data compression can also be enabled and disabled by software (MODE SELECT command), which overrides the DIP switch setting.

The Python 4322, 4542, and 4352 drives provide a single-ended SCSI connection. This chapter summarizes the SCSI-1 and SCSI-2 status codes, message codes, and commands. Refer to Chapter 3 for specific SCSI cabling and connector information.

NOTE: *Refer to the Python Technical Reference Manual (P/N 25356-002, DDS-only) or to the Python SCSI Manual (P/N 27298-001, available in 1992) for detailed information about the SCSI interfaces. That manual includes an alphabetically arranged summary of the Python SCSI commands. Appendix C of this manual provides specific SCSI information pertaining to data compression.*

5.2 SCSI-1 Interface

The SCSI-1 interface for the premium series DAT drives conforms with the ANSI Standard X3.131-1986. Tables 5-1, 5-2, and 5-3 list the Status Codes, Message Codes, and Commands for this interface.

Table 5-1. Status Codes---SCSI-1

Bits	4-BIT STATUS CODE					DEFINITION
	4	3	2	1	0	
	0	0	0	0	X	Good Status
	0	0	0	1	X	Check Condition
	0	1	0	0	X	Busy
	1	0	0	0	X	Intermediate Status
	1	1	0	0	X	Reservation Conflict

Table 5-2. Message Codes---SCSI-1

CODE	DESCRIPTION	DIRECTION*
00h	Command Complete	In
02h	Save Data Pointer	In
04h	Disconnect	In
05h	Initiator Detected Error	Out
06h	Abort	Out
07h	Message Reject	In/Out
08h	No Operation	Out
0Ah	Linked Command Complete	In
0Bh	Linked Command Complete with Flag	In
0Ch	Bus Device Reset	Out
80h	Identify (No Disconnect/Reconnect)	In/Out
C0h	Identify (Disconnect/Reconnect)	In/Out
01h**	Extended Message	In/Out

* Direction: In = Drive to host; Out = Host to Drive.

** Supports only one extended message: SYNCHRONOUS DATA TRANSFER REQUEST.

Table 5-3. Commands---SCSI-1

CODE	TYPE	COMMAND
00h	O	TEST UNIT READY
01h	M	REWIND
02h	V	REQUEST BLOCK ADDRESS
03h	M	REQUEST SENSE
05h	E	READ BLOCK LIMITS
08h	M	READ
0Ah	M	WRITE
0Ch	V	SEEK BLOCK
10h	M	WRITE FILEMARKS
11h	O	SPACE
12h	E	INQUIRY
13h	O	VERIFY
14h	O	RECOVER BUFFERED DATA
15h	O	MODE SELECT
16h	O	RESERVE UNIT
17h	O	RELEASE UNIT
19h	O	ERASE
1Ah	O	MODE SENSE
1Bh	O	LOAD/UNLOAD
1Dh	O	SEND DIAGNOSTIC
1Eh	O	PREVENT/ALLOW MEDIUM REMOVAL
40h	*	CHANGE DEFINITION

M = Mandatory Command	O = Optional Command
E = Extended Command	V = Vendor Unique Command
* = Defined in SCSI-2	

5.3 ANSI X3.131 Conformance Statement (SCSI-1)

GENERAL FEATURES

1. Supports arbitration.
2. Disconnect/reconnect.
3. Single-ended or differential drivers.
4. Termination power supplied to cable (jumper option).
5. Supports both single and multi-initiator systems.
6. Fixed and variable block transfer lengths.
7. Hard reset.
8. Synchronous data transfer.
9. Parity implemented (switch option).
10. Space blocks, filemarks, and EOD.
11. Supports third-party reservation.

COMMANDS

- | | |
|--------------------------------|---------------------|
| 1. Change Definition | 14. Request Sense |
| 2. Erase | 15. Reserve Unit |
| 3. Inquiry | 16. Rewind |
| 4. Load/Unload | 17. Seek Block |
| 5. Mode Select | 18. Send Diagnostic |
| 6. Mode Sense | 19. Space |
| 7. Prevent/Allow Media Removal | 20. Test Unit Ready |
| 8. Read | 21. Verify |
| 9. Read Block Limits | 22. Write |
| 10. Recover Buffered Data | 23. Write Filemarks |
| 11. Release Unit | |
| 12. Request Block Address | |

MESSAGES

- | | |
|----------------------|---------------------------------------|
| 1. Save Data Pointer | 9. Linked Command Complete |
| 2. Disconnect | 10. Linked Command Complete with flag |
| 3. Message Reject | 11. Initiator Detected Error |
| 4. Identify | 12. Synchronous Data Transfer Request |
| 5. Abort | |
| 6. Bus Device Reset | |
| 7. No Operation | |
| 8. Bus Device Reset | |

VENDOR UNIQUE COMMANDS

- | | |
|---------------|--------------------------|
| 1. Seek Block | 2. Request Block Address |
|---------------|--------------------------|

5.4 SCSI-2 Interface

The SCSI-2 interface for the Python DDS-DC DAT drives conforms with the ANSI X3T9.2 Working Draft, Revision 10 standard. Tables 5-4, 5-5, and 5-6 list the Status Codes, Message Codes, and Commands for this interface.

Table 5-4. Status Codes---SCSI-2

Bits	4-BIT STATUS CODE					DEFINITION
	4	3	2	1	0	
	0	0	0	0	X	Good Status
	0	0	0	1	X	Check Condition
	0	1	0	0	X	Busy
	1	0	0	0	X	Intermediate Status
	1	1	0	0	X	Reservation Conflict

Table 5-5. Message Codes---SCSI-2

CODE	DESCRIPTION	DIRECTION*
00h	Command Complete	In
02h	Save Data Pointer	In
04h	Disconnect	In
05h	Initiator Detected Error	Out
06h	Abort	Out
07h	Message Reject	In/Out
08h	No Operation	Out
0Ah	Linked Command Complete	In
0Bh	Linked Command Complete with Flag	In
0Ch	Bus Device Reset	Out
80h	Identify (No Disconnect/Reconnect)	In/Out
C0h	Identify (Disconnect/Reconnect)	In/Out
01h**	Extended Message	In/Out

* Direction: In = Drive to host; Out = Host to Drive.

** Supports only one extended message: SYNCHRONOUS DATA TRANSFER REQUEST.

Table 5-6. Commands---SCSI-2

CODE	TYPE	COMMAND
00h	M	TEST UNIT READY
01h	M	REWIND
02h	V	REQUEST BLOCK ADDRESS
03h	M	REQUEST SENSE
05h	M	READ BLOCK LIMITS
08h	M	READ
0Ah	M	WRITE
0Ch	V	SEEK BLOCK
10h	M	WRITE FILEMARKS
11h	M	SPACE
12h	M	INQUIRY
13h	O	VERIFY
14h	O	RECOVER BUFFERED DATA
15h	M	MODE SELECT
16h	M	RESERVE UNIT
17h	M	RELEASE UNIT
19h	M	ERASE
1Ah	M	MODE SENSE
1Bh	O	LOAD/UNLOAD
1Ch	O	RECEIVE DIAGNOSTIC RESULTS
1Dh	M	SEND DIAGNOSTIC
1Eh	O	PREVENT/ALLOW MEDIUM REMOVAL
2Bh	O	LOCATE
34h	O	READ POSITION
3Bh	O	WRITE DATA BUFFER
3Ch	O	READ DATA BUFFER
40h	O	CHANGE DEFINITION
4Ch	O	LOG SELECT
4Dh	O	LOG SENSE

M = Mandatory Command	O = Optional Command
E = Extended Command	V = Vendor Unique Command

5.5 ANSI X3T9.2 Conformance Statement (SCSI-2) (Working Draft, Rev 10)

GENERAL FEATURES

1. Disconnect/reconnect, arbitration (required in SCSI-2).
2. Single-ended or differential drivers.
3. Termination power supplied to cable (jumper option).
4. Supports both single and multi-initiator systems.
5. Fixed and variable block transfer lengths.
6. Hard reset.
7. Synchronous data transfers
8. Parity implemented (switch option).
9. Space blocks, filemarks, and EOD.
10. Supports third-party reservation.
11. Log Sense and Log Select for managing soft errors reporting.
12. MODE SENSE/SELECT page to control and report operation of data compression in sequential access devices.

COMMANDS

- | | |
|---------------------------------|---------------------------|
| 1. Change Definition | 16. Recover Buffered Data |
| 2. Erase | 17. Release Unit |
| 3. Inquiry | 18. Request Block Address |
| 4. Load/Unload | 19. Request Sense |
| 5. Locate | 20. Reserve Unit |
| 6. Log Select | 21. Rewind |
| 7. Log Sense | 22. Seek Block |
| 8. Mode Select | 23. Send Diagnostic |
| 9. Mode Sense | 24. Space |
| 10. Prevent/Allow Media Removal | 25. Test Unit Ready |
| 11. Read | 26. Verify |
| 12. Read Block Limits | 27. Write |
| 13. Read Data Buffer | 28. Write Data Buffer |
| 14. Read Position | 29. Write Filemarks |
| 15. Receive Diagnostic Results | |

MESSAGES

- | | |
|----------------------|---------------------------------------|
| 1. Save Data Pointer | 9. Linked Command Complete |
| 2. Disconnect | 10. Linked Command Complete with flag |
| 3. Message Reject | 11. Initiator Detected Error |
| 4. Identify | 12. Synchronous Data Transfer Request |
| 5. Abort | |
| 6. Bus Device Reset | |
| 7. No Operation | |
| 8. Bus Device Reset | |

VENDOR UNIQUE COMMANDS

- | | |
|---------------|--------------------------|
| 1. Seek Block | 2. Request Block Address |
|---------------|--------------------------|

NOTES

CHAPTER 6

TAPE FORMATS

6.1 Introduction

The Python DAT drives with data compression capabilities use the ANSI Digital Data Storage (DDS) and Digital Data Storage Data Compression (DDS-DC) tape formats to record uncompressed and compressed data on Digital Audio Tape (DAT) media. The DDS-DC format is a superset of the DDS format, thus ensuring compatibility and allowing the Python DDS-DC drives to write compressed and uncompressed data to the same cassette.

As a superset of the DDS format, the DDS-DC format adds support for in-drive data compression and decompression. The features of this format were conceived for use with lossless compression algorithms based on substitution; however, the format is not necessarily limited to those algorithms. The implementation of data compression in Python drives uses the advanced DCLZ (data compression LempelZiv) algorithm.

The industry standard DDS-DC format has been approved by the DDS Manufacturers Group, ANSI, and ECMA.

6.1.1 DDS Format

The DDS format supports DAT devices as computer peripherals. The objectives of the DDS format are:

- To maximize storage capacity and performance
- To facilitate data interchange
- To provide compatibility with existing storage command sets
- To provide extremely fast random access
- To take advantage of the helical scan recording method
- To augment the inherent error-detection and error correction capabilities of the DAT technology

Achievement of these objectives is shown in the following list of advantages provided by the DDS format.

- Maximizes capacity and transfer rate.
- Provides compatibility with both 1/2-inch tape and QIC command sets.
- Supports SCSI sequential command set functions.
- Supports random read functions.
- Supports both variable-length and fixed-length records.

- Supports two independently-writeable partitions to support directory of stored data. (Host determines whether tape is single data space or two-partition layout.)
- Provides save-set marks to enhance segmentation.
- Allows filemarks and save-set marks without decreased capacity.
- Supports third level of ECC (C3) for optimal data reliability.
- Provides redundantly-stored, track-based checksums for increased data reliability.
- Provides Read-After-Write (RAW) and frame repetition on error.
- Provides randomizer for increased data reliability.
- Provides N-Group writing for tape duplication applications.
- Supports fast search to data, marks, and EOD.
- Does not require formatting.
- Supports short tape load sequences.
- Provides areas for Tape Error Logs.

6.1.2 DDS-DC Format

The advantages of the DDS format are retained in the DDS-DC format, which is an extension of the DDS format that supports data compression. In addition to the previously listed benefits of data compression in increasing storage capacity at fast transfer rates, the DDS-DC format provides these capabilities:

- Enables a tape drive without decompression capabilities to retrieve compressed data and transmit it to the host for decompression.
- Maintains high performance by the compression algorithm while avoiding controller-induced limitations of the transfer rate when handling multiple, small fixed-length records.
- Allows writing of compressed and uncompressed records to the same cassette.

In the DDS-DC format, a data record is the smallest data object. A record is typically the smallest distinct set of data bytes transferred between a host and a tape drive. The DDS-DC format supports compressed and uncompressed records with the compressed records recorded into objects called *entities*.

DDS-DC groups can be made up of entities and uncompressed records. Each group contains an index that describes the contents of the group.

6.1.3 Commonalities Between the Formats

Because the DDS-DC format is an extension or superset of the DDS format, many commonalities exist between the two formats. These commonalities are outlined in the following points:

- The series of transformations (randomizing, interleaving, generating and including two Reed-Solomon error-correction codes, blocking and translating bytes to channel bits) applied to each group
- The definition and use of the third, group-based Reed-Solomon error-correction code

- ❑ The method of recording a group
- ❑ The tape layout (partitioning, Device Area, System Area, Data Area, and EOD Area)
- ❑ The rules for recording multiple occurrences of a group in contiguous sequence

Basically, the only differences between the two formats are in the contents of groups.

This chapter describes the DDS and DDS-DC formats. The information is divided into six major topics:

- ❑ Entities (DDS-DC format only, Section 6.2)
- ❑ Tape Layout (Sections 6.3, 6.4, and 6.5)
- ❑ Data Storage Organization (Sections 6.6, 6.7, and 6.8)
- ❑ Indexing (Section 6.9)
- ❑ Vendor Group (Section 6.10)
- ❑ Read-After-Write (Section 6.11)

Information describing the DDS-DC extensions to the DDS format are tagged as DDS-DC only.

6.2 Entities (DDS-DC Format Only)

An entity is a recorded object that consists of an entity header followed by one or more compressed records. The compressed records in an entity must be based on uncompressed records of equal size. All of the compressed records in an entity depend on the same access point for decompression. An *access point* is the beginning of a sequence of compressed records at which point codewords to a decompression algorithm must start--whether or not the required data is at the beginning of the sequence. Thus, the access point may or may not exist at the start of the first compressed record in the entity; however, access points cannot appear elsewhere in an entity.

Access points are required in certain cases and are important when performing SCSI SEEK or LOCATE commands to a record within a entity. The first entity in a group must have an access point. Also, an access point is required at the beginning of the first compressed record of an entity following a filemark, a setmark, or an uncompressed record.

As long as all of the entity header and the first 8 bits of the first compressed record in the entity are in the same group, an entity may cross groups. However, the requirement that an access point occur at the first entity in a group means that all span points must occur within the last compressed record of an entity.

The entity header is made up of 8 uncompressed bytes as outlined in the following points:

- ❑ Byte 1 contains the length of the entity header (08h)
- ❑ Byte 2 specifies the compression algorithm identifier number (when an access point occurs at the beginning of the entity) or specifies that the compressed records in the entity depend on a the access point in the previous entity (all bits zero).

- Bytes 3 to 5 reflect the length of the uncompressed record or records from which the compressed record or records in the entity are derived. (If the entity contains more than one compressed record, they must all be based on uncompressed records of equal length.)
- Bytes 6 to 8 show the number of compressed records in the entity.

6.3 Tape Layout

The DDS and DDS-DC formats support two tape layouts: a single data space (one partition) and two-partitions. The data organization is the same for either tape layout -- a sequence of groups with each group having a fixed capacity.

The single data space provides a capability similar to 1/2-inch reel-to-reel tape. It allows for sequential storage and access to data and separator marks in a continuous space, which spans the entire tape volume.

The two-partition layout gives the same capability in each of two independent partitions on the tape. Data may be written to either partition without affecting the other partition; thus, data could be written alternately between partitions.

6.4 Single Data Space Layout

The single data space tape layout uses the entire tape volume to sequentially store data and to access the data and separator marks. This layout is made up of four areas:

- The Device Area
- The Reference and System Area
- The Data Area
- The End-of-Data (EOD) Area

Figure 6-1 shows this tape layout.

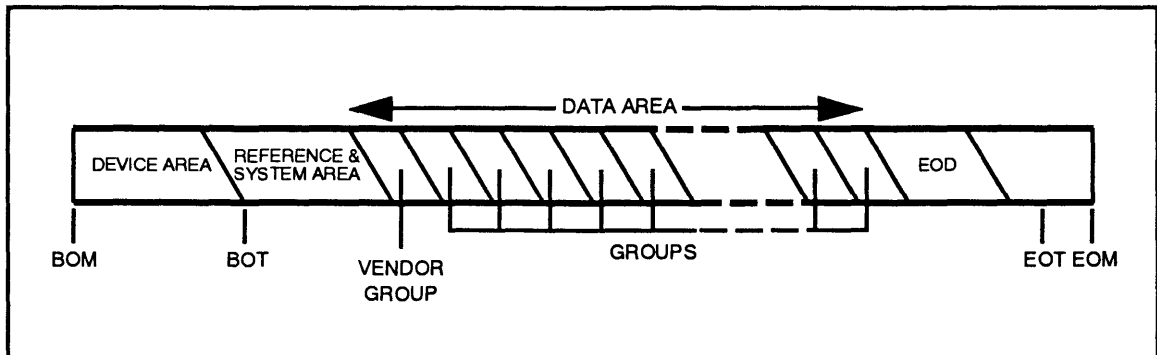


Figure 6-1
Single Data Space Tape Layout

6.4.1 Device Area

The initial section of the tape is the Device Area, which is used to load the tape and is NOT used to write any user data.

Some slight indentations to the tape may result from the clamp that holds the tape to the hub. These indentations are generally more pronounced at the beginning of the tape. Therefore, this section acts as a safeguard against writing data over these indentations.

Once a sufficient distance from the Beginning-of-Media (BOM) is attained, a device area is provided. The content of this area is not defined by the DDS format.

Immediately following the device area is another area that contains no valid signals recorded on it. This area is a guard zone between the test area and the start of recorded frames.

6.4.2 Reference and System Area

The second area on this tape layout is the Reference and System area, which provides logs of usage and soft error occurrences. The System area allows such logs to be updated by being overwritten in the same physical location. The Reference area defines the Beginning-of-Tape (BOT) point and allows efficient positioning during an update. The reference area is recorded only by the first drive to record on a blank cassette.

Figure 6-2 shows the layout of the Reference and System area.

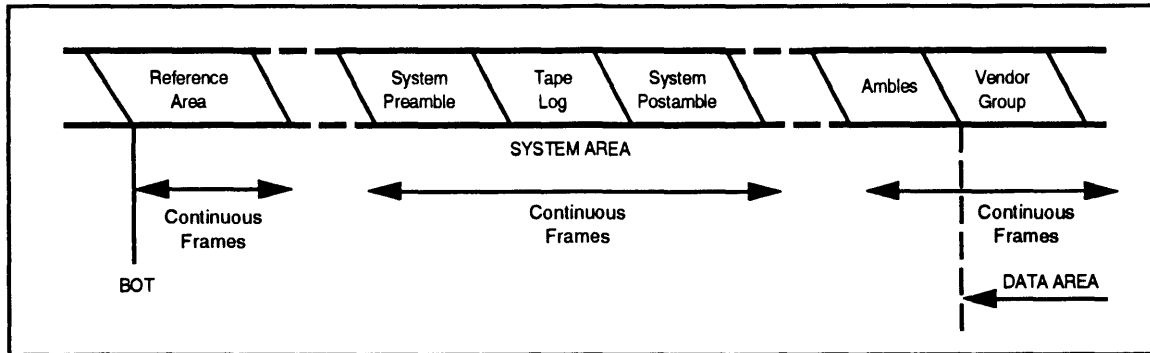


Figure 6-2
Reference and System Area Layout

Reference Area

The Reference area is the first collection of frames written on the tape after the BOT. It is a reference point when determining the tape format and updating the System area. This area consists of 35 frames followed by a band of between zero and ten frames. Because updating the System area may result in a physical position of the new System area that is not identical to its previous physical position, this band allows a sufficient tolerance to allow for variation. The Main Data area of all tracks in the Reference are set to zero.

System Area

The System area consists of:

- The *System Preamble*, which allows the servo to synchronize before reading the System Log.
- The *Tape Log*, which contains tape usage information, including current usage error count. The Tape Log in this format contains information for the entire tape.
- The *System Postamble*, which consists of ten frames. Immediately following these frames is another band of ten identical frames. These extra frames allow a position tolerance for updating the System area.

During the cassette ejection sequence, the System area is updated. Because of this updating, the physical length of the System area must remain constant.

Frame rewriting is not used on any frames in the System area. This factor could cause the Main Data area to be incorrect; thus, the Main Data area is not used and is set to zero.

Because the Sub areas are repeated many times, the Sub area is used to store the usage information.

The frames following the System area allow the servo to resynchronize before it must read the Vendor Group.

6.4.3 Data Area

The third area on this tape layout is the Data area, which is written as a sequence of groups beginning with a special Vendor Group. Each group is a fixed size and stores uncompressed records, entities (DDS-DC format only), filemarks, and save-set marks written by the host. The variable-length records and filemarks are mapped into the group by indexing.

6.4.4 End-of-Data (EOD) Area

The last area in this tape layout is the End-of-Data (EOD) area -- the point on the tape where the host ceased writing data. This area is not specifically written through host command. Rather, the drive will detect any conditions -- such as a REWIND following a WRITE command -- which comprise EOD and then generate the EOD area.

The area is physically long enough to guarantee that it is detected during fast searching. A set of at least 12 Data area amble frames, which are continuous and contiguous with the last frame of the last data group written precedes the EOD area.

In the case where existing data is partially overwritten and the overwrite ends before the existing EOD area, a new EOD is generated at that point. A tape may, therefore, contain more than one EOD area. However, the EOD closest to the BOT is the only valid one. Any data or other EOD areas beyond that point are logically inaccessible.

6.5 Two-Partition Layout

The two-partition tape layout provides two separate areas that may be written independently to be contained on a single tape volume. The host chooses the tape layout -- single data space or two-partition -- before it sends data for writing to a blank tape for the first time or when it sends a tape format command.

The two partitions are adjacent on the tape with no gap between them. The partition closer to the BOM is Partition 1; the partition closer to the EOM is Partition 0. The boundary between the partitions is defined by an upper limit of the value of the Absolute Frame Count of Partition 1.

The two-partition layout is generally made up of:

- The Device Area
- Partition 1
- Partition 0

The Device area is the same as the Device area for the single data space tape layout. Also, each partition is made up of a Reference and System area, a Data area, and an EOD area. These areas are the same as for the single data space tape layout. Additionally, Partition 1 contains a synthetic EOT and EOM; Partition 0 contains a synthetic BOT. Thus, each partition is structured identically to a single data space tape except for the Device area.

Figure 6-3 illustrates the two-partition tape layout.

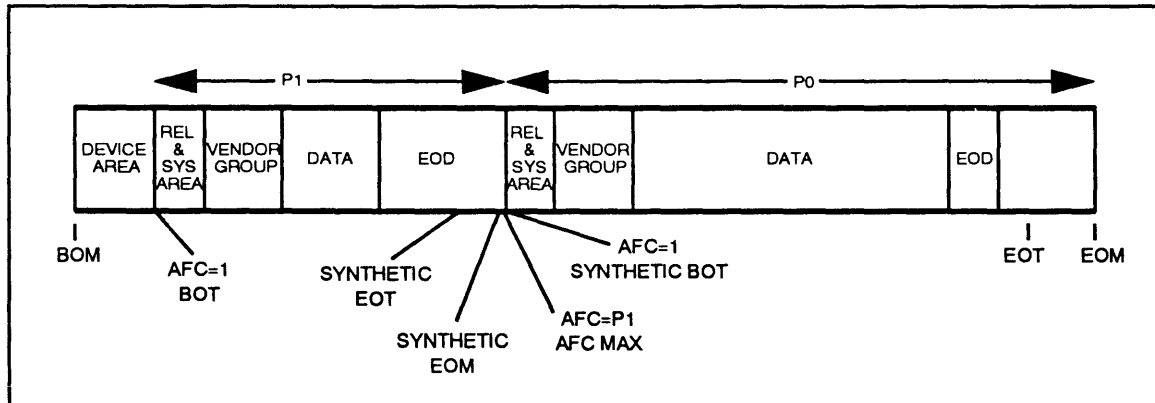


Figure 6-3
Two-Partition Tape Layout

6.6 Data Storage Organization

The data storage organization of the DDS and DDS-DC formats is based on two considerations:

- Individual track layout
- Data groups

Like the audio format, the DDS and DDS-DC formats allocate 60 percent of each track to user data and ECC. The remaining 40 percent is divided between automatic-track-finding (ATF) signals and the subcode area. The ATF and subcode areas are collectively called subareas and are used to maintain information about the location of data records, filemarks, and save set marks.

The data area on the tape is organized into *groups*. Each group is made up of a fixed number of frames and has a fixed capacity. A group stores host written data, save-set marks, and filemarks. In the DDS format, the host written data consists of uncompressed records; in the DDS-DC format, the host written information may consist of uncompressed records and entities.

Although a group has a fixed size in memory, each group does not necessarily contain the same number of physical frames. Some frames may be rewritten because an error is detected by the Read-After-Write process.

If a host record is larger than the space remaining in the group, the space is filled, and bytes not accommodated are placed in the next group.

In the case where a group is not fully occupied by user data, the group is still written as the fixed logical size, but the index reflects its actual contents.

6.7 Individual Track Layout

The layout of an individual track is made up of a main area and two subareas, as shown in Figure 6-4.

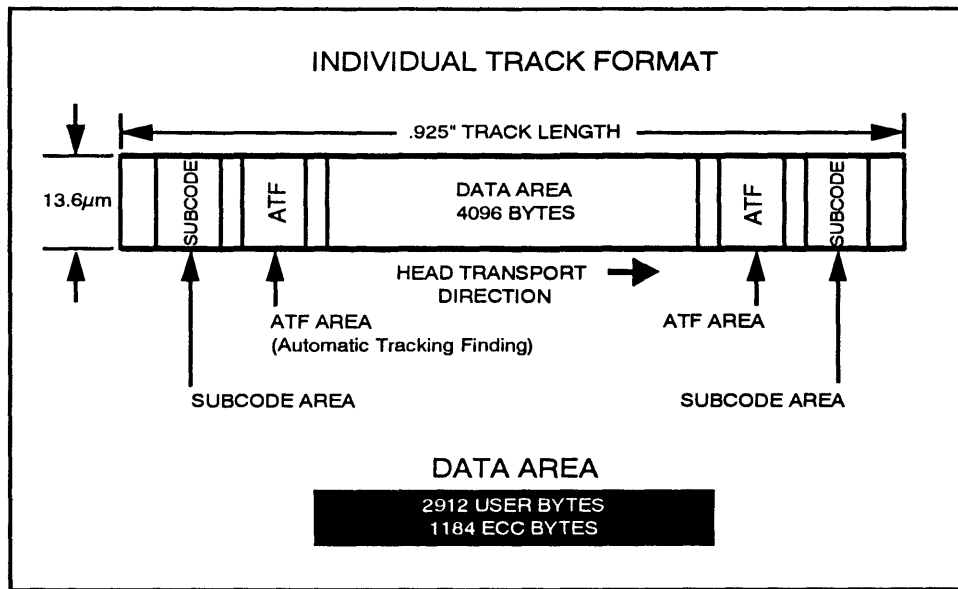


Figure 6-4
Individual Track Layout

6.7.1 Main Area

Within the individual track, the main area is made up of the 128 main data blocks. Each main data block is made up of the following:

- Synchronization (8 bits)
- Main ID (16 bits -- W1 and W2)
- Main ID parity (8 bits)
- Main data (256 bits)

Synchronization Pattern: The 10-channel bit synchronization pattern corresponds to 8 bits, which are contained in the synchronization portion of the main data block.

Main ID: The Main ID is made up of W1 (8 bits) and W2 (8 bits). These bits identify various characteristics about the main data area.

The *Block Address* is determined by W2, bit 7, which identifies the type of block. The remaining seven bits of W2 -- B6 (msb) through B0 (lsb) -- are the identify bits for the main data block address (0 to 127) in one track.

The *Frame Address* is a 4-bit code of W1 -- B3 (msb) through B0 (lsb) -- in even address blocks. The frame address repeats from 0000 to 1111.

The *Format ID* is a 2-bit code of W1 -- B7 and B8 -- which shows the application of the main ID and main data where:

- 00 = audio format
- 01 = DDS or DDS-DC format

Main ID Parity: The Main ID parity bits provide the error detection mode for W1 and W2.

Main Data: the main data block contains 32 symbols (each 8-bits long) of data. Each symbol is assigned a number 0 to 31 along the recording direction.

The two kinds of symbols are:

- Data
- Parity

The three kinds of blocks are:

- Data only
- Data and parity
- Parity only

Main data in one track may store 2912 symbols of data, 672 symbols of C2 parity, and 512 symbols of C1 parity.

The main data area is made up of:

- The header (first four bytes), which contains format information
- 5756 bytes of user data
- 64 bytes of zeros

The header contains the *Data Format ID (DFID)* for the tape with 0000 indicating the DDS format.

The *Logical Frame ID (LF-ID)* contains the logical frame number (bits 8-13), the ECC frame ID (bit 14), and the last frame ID (bit 15). If bit 14, the ECC frame ID, is 1, the current frame is the ECC frame of the group.

A *data word* is made up of 16 bits which comprise two data symbols. The data word can be for channel A or B.

6.7.2 Subareas

The two subareas per track each contain a sub data area of 8 blocks. Each block contains a sub ID and sub data. The sub data is made up of four packs or three packs and C1 parity.

In any one subarea, the mix of pack items varies between different areas of the tape.

Within one sub ID for a block of the sub data area, the *area ID* identifies the current area on tape and the type of current frame, where:

0000 = Device area
 X001 = Reference area
 X010 = System area
 X100 = Data area
 X101 = EOD area
 others are reserved

Above, X is 0 for Partition 1 and 1 for Partition 0 in a two-partition tape. For a single data space tape, X is 1.

The various pack items represent different types of information such as, group count, filemark count, save-set count, the absolute frame count, checksums, N-group reading counts, and ECC retries.

6.8 Group Structure

A group is made up of 22 data frames for a size of 126632 bytes and an optional ECC frame. Groups are separated by zero or more amble frames. The 22 data frames contain host written data and index information. Figure 6-5 shows the group structure.

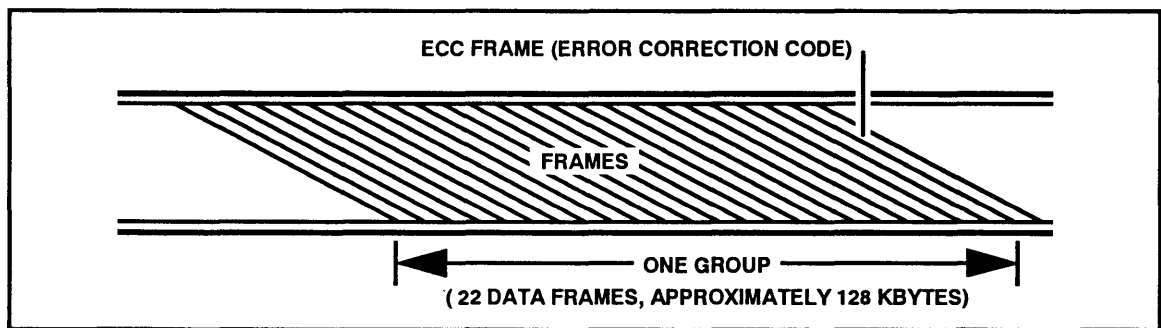


Figure 6-5
 Group Structure

Figure 6-6 compares a group in the DDS format without compression to a group in the DDS-DC format with compression. As an example, a 3.2:1 compression ratio is depicted. The size and structure of the groups are the same. However, the contents of the group are different in that the group in the DDS-DC format may contain uncompressed records and entities.

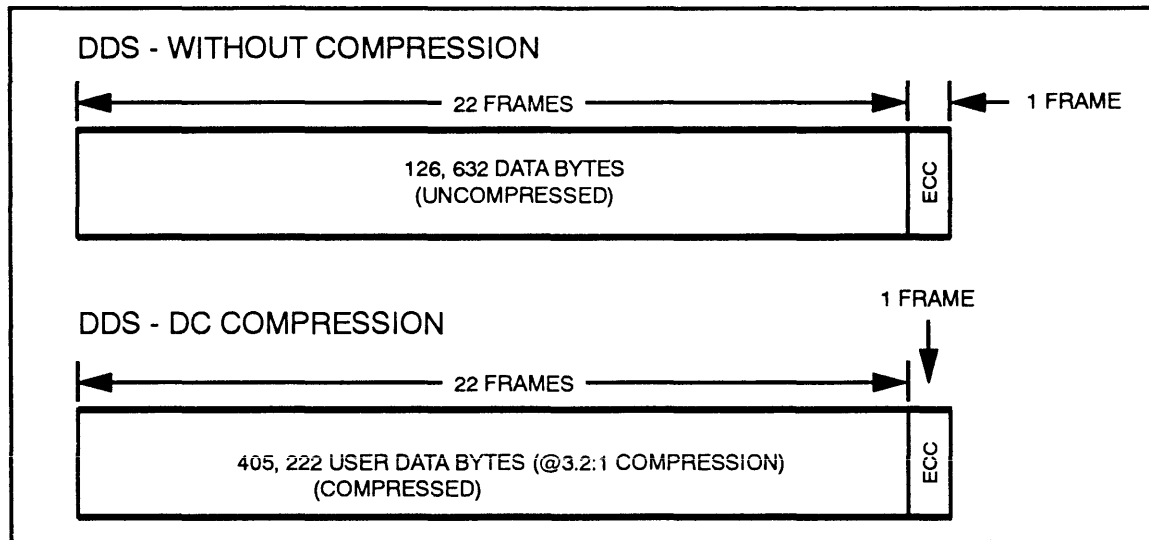


Figure 6-6
Group Structure---DDS and DDS-DC Formats

6.8.1 Data Frames

Each group can store 22 frames of data for a size of 126632 bytes. It may consist of more than 22 frames if any are rewritten using the Read-After-Write process.

6.8.2 Index

The index is stored as part of the group. its size grows as needed to describe the group contents. The index is stored at the end of the group and grows toward the front of the group. A Group Information Table (GIT) occupies bytes 126601 through 126632. A Block Access Table (BAT) is located immediately before the GIT. The first 4-byte entry of the BAT occupies bytes 126597 through 12660 with additional BAT entries extending toward the front of the group.

If necessary, records are split as needed to fill a group complete with data and index information.

A group that is not filled with user data contains an index specifying that it is not full. However, that group must still contain 126632 bytes.

Refer to Section 6.9, "Indexing" for more information about the GIT and BAT.

6.8.3 ECC Frame

The 22 data frames are followed by an ECC frame that contains the third level of error correction (C3). The first two levels of error correction (C1 and C2) are embedded in the frame itself and are identical to those used in the audio format. These levels of ECC only cover the data within a frame. C1 and C2 can correct an error in each track up to 792 bytes long.

The third level of ECC provides recovery across frames. Only the frames in the group are covered by the third level of ECC. Each group is considered independently of all other groups. The Main Data area of the ECC frame contains the actual error correction bits.

The third level of ECC can correct any two tracks that are bad in a group. These tracks may be of the same or different azimuth angles and may be in the same or different frames.

6.8.4 Amble Frames

Amble frames are only written after the last frame in a group or after another amble frame. The amble frame contains a valid Header, but the remainder of the Main Data area is set to zero. The logical frame number of an amble frame is zero.

6.9 Indexing

The indexing concept allows variable-sized host records to map onto fixed-size groups. All groups except the Vendor Group contain an index. Each index is divided into two parts:

- The Block Access Table (BAT), which describes the contents of the group and has an entry for each record and save-set mark.
- The Group Information Table (GIT), which contains a list of counters and pointers that describe the characteristics of the group. For the DDS-DC format, the definitions of three of the fields in the GIT are extended.

6.9.1 Group Information Table (GIT)

The GIT is included in each group. It is in a fixed location at the end of the group and is the same size regardless of the contents of the group (bytes 126601 through 126632). It contains a list of counters and pointers that describe the characteristics of the group. Its entries are two or four bytes wide.

For the DDS-DC format, the Record Count, Number of Records in Group, and Previous Record Group Count fields are extended as described in the following text.

Group Count: The Group count is the number of user data groups written since the beginning of the current partition, beginning with one and including the current group.

Block Access Table Count: The Block Access Table Count is the number of entries in the Block Access Table.

Record Count: The Record Count is the number of records written since the BOT of the current partition and includes any completed records in the group.

For the DDS-DC format, this 4-byte field specifies the sum of the values in the Number of Records in Group of the GIT of all groups written after the logical beginning of tape (LBOT) through the current group.

NOTE: *As in the DDS format, bytes 4 through 7 of Pack item 2 (sub data) in the DDS-DC format is the same as the Record Count fields in the GIT. The DDS-DC extension for this field also applies to Pack Item 2.*

Filemark Count: The Filemark Count is the number of filemarks written since the BOT of the current partition.

Save-Set Count: The Save-Set Count is the number of setmarks written since the beginning of the current partition up to, and including, the current group.

Number of Records in Group: The Number of Records in Group is exactly the number of entries in the Block Access Table of the current group with the Record End bit set.

For the DDS-DC format, this 2-byte field specifies the sum of the following:

- The number of mark entries in the BAT of the current group
- The number of Total Count of Uncompressed Record entries in the BAT of the current group
- The number of Entire Uncompressed Record entries in the BAT of the current group
- The sum of the numbers of compressed records within all entities for which the BAT of the current group contains Entire Entity entries
- The number minus one of compressed records in the entity for which the BAT of the current group contains a Start Part of Entity entry, if an entry exists
- The number of Total Count of Entity entries contained in the BAT of the current group

Previous Record Group Count: The Previous Record Group Count is the number of the group containing the most recent beginning of record.

For the DDS-DC format, this 2-byte field specifies the running number of the highest-numbered previous group that contains the beginning of an uncompressed record, an access point, a filemark, or a setmark. If no such group exists, it contains zero.

Number of FMs in Group: the Number of Filemarks in Group is the total number of filemarks written in the current group.

Previous FM Group Count: the Previous Filemark Group Count is the number of the group containing the previous filemark written on the tape in the current partition or is zero if no such group exists.

6.9.2 Block Access Table (BAT)

The BAT identifies the contents of the group. It contains an entry for each uncompressed record, filemark, and save-set mark. For the DDS-DC format, it additionally contains an entry for each entity in the group.

If the group contains partial uncompressed records, each of those also has an entry in this table. In the DDS-DC format, if the group contains a partial entity, each of those also has an entry in this table.

The size of the table depends on the contents of the group. Access entry for the first item in the group is located immediately before the Physical Information. The other access entries follow in a reverse direction in the order of the contents of the group.

The entries in the BAT are each a 4-byte field that contains a flag byte (byte 1) and a count field (bytes 2 through 4), which contains a number based on the flag byte.

In the DDS-DC format, all BAT entries for entities and uncompressed records are defined and used the same as in the DDS format except for bit 5 of the flag byte. Table 6-1 lists the bit configurations for the flag bytes as related to entities (DDS-DC format only).

Table 6-1. Bit Configurations of Flag Byte (DDS-DC Format Only)

Bit Configuration	Description
0111X011	The entity starts in the current group and ends in a subsequent group (Entire Entity). The count field specifies how many bytes the entity contains.
0101X010	The entity starts in the current group and ends in a subsequent group (Start Part of Entity). The count field specifies how many bytes are contained in the part of the entity that is in the current group.
0101X000	The entity starts in a previous group and ends in a subsequent group (Middle Part of Entity). The count field specifies how many bytes are contained in the part of the entity that is in the current group.
0111X000	The entity starts in a previous group and ends in the current group (Last Part of Entity). The count field specifies how many bytes are contained in the part of the entity that is in the current group. An entry for the Total Count of Entity must follow this entry in the BAT of the current group.
0001X001	This entry (Total Count of Entity) follows an entry for the Last Part of Entity in the BAT. The count field specifies the total number of bytes in the entity.

NOTE: *All other bit configurations for the flag byte relate to uncompressed records and are identical to those defined in the DDS format.*

6.10 Vendor Group

The Vendor Group is the first group in a partition. It is preceded by an amble area. The Vendor Group is not accessible via the SCSI bus.

This group contains user data area fields, which are briefly described in the following paragraphs.

The only required fields are ASCII strings that give the name of the manufacturer and the model number of the drive.

Name of Manufacturer (Bytes 0 -127): The name of the manufacturer of the drive that initialized or wrote the partition is required. This field is an ASCII string (null terminated and padded).

Model Number (Bytes 128-159): The model number or identifier of the drive that initialized or wrote the partition is required. This field is an ASCII string (null terminated and padded).

Host Interface Type and Address (Bytes 224-255): The type of host interface and its connection may appear here. (Bytes 224-239 = type; bytes 240-255 = address). This field is an ASCII string (null terminated and padded).

Bytes 400 - 5755 are reserved and are set to ASCII null character.

Vendor Unique (Bytes 5756-126631): Contents not specified.

6.11 Read-After-Write

The Read-After-Write (RAW) technique provides a means of verifying that host data was written on the tape correctly by reading it later. After each frame is written, it is examined to determine whether or not it is correctly recorded.

If a frame is identified as bad, it is rewritten later down the tape. The bad frame is not necessarily rewritten immediately. It can be rewritten after three, four, or five other frames have been written. Any frame can be rewritten multiple times to provide for skipping over bad areas on the tape. The maximum number of times a rewrite sequence can occur is 128 (original plus 127 repeats).

CHAPTER 7

DATA COMPRESSION

7.1 Introduction

Typical data streams of text, graphics, code, or other forms of data contain repeated information that is considered redundant. This redundancy occurs because the data contains groups of symbols that are repeated or exhibit a particular pattern. Storage efficiency is increased if the redundancy in the data is removed before the data is recorded to tape. Simply repeating the same information on tape wastes valuable and limited space.

With data compression, the redundant (repeated) information in a data stream is identified and then represented by codewords or symbols, which allow the data to be recorded in a fewer number of bits. These symbols or codewords encode the redundant strings or point back to the original string, thus using fewer characters to represent the strings. Because these smaller symbols are substituted for the longer strings of data, more data can be stored in the same physical space.

Some important benefits result from data compression in DAT drives:

- The same amount of information can be stored on a smaller length of tape.
- More data can be stored on a given length of tape.
- Performance can more closely parallel that of high transfer rate computers.
- More information can be transferred in the same time interval.

7.1.1 Data Compression Considerations

In an effective data compression method, several factors are important considerations:

- The amount of compression (measured by the *compression ratio*, which is a ratio comparing the amount of uncompressed data to the amount of compressed data and is obtained by dividing the size of the uncompressed data by the size of the compressed data)
- The speed with which data is compressed and decompressed in relation to the host transfer rate
- The types of data to be compressed
- The integrity of the compressed data

Considering these factors, Archive engineers concluded:

The most effective data compression method must compress as much data as possible under the following conditions:

- The transfer rate of the host computer is not impeded.
- Adaptation is made to different types of data.
- Data integrity is maintained. (Uncompressed data is compressed and then decompressed without error---*lossless* compression.)

The amount of compression possible in a data stream depends on such factors as the data pattern, the compression algorithm, the pattern repetition length, the pattern repetition frequency, the object size (chunk of information to be compressed), and the starting pattern chosen.

The transfer rate depends on such factors as the compression ratio, the drive buffer size, the host computer input/output (I/O) speed, the effective disk speeds of the host computer, and the record lengths transmitted by the host computer.

Because various types of data are encountered for compression, the data compression method must be able to *adapt* to different types---automatically providing optimum handling for all types of data. If only one type of data is to be compressed, an effective data compression method can be tailored to achieve maximum speed and storage for that type.

However, an effective data compression method for a tape drive must serve varying data types while ensuring maximum storage and transfer rates as well as maintaining data integrity. Thus, creating an effective data compression method with those stipulations requires integrating several design decisions that work together.

In Python DDS-DC DAT drives, these design decisions centered on the following:

- Supporting the DDS-DC format
- Implementing the Data Compression LempelZiv (DCLZ) algorithm, which is based on the LZ2/LZW algorithms
- Using the AHA3101 data compression chip for hardware compression in the drive

NOTE: Refer to the Chapter 6 for a description of the DDS-DC format.

7.1.2 DCLZ Algorithm

Within the computer industry, algorithms developed by Abraham Lempel and Jacob Ziv (enhanced later by Terry Welch) are popular, versatile and powerful compression methods. These LZ algorithms are basically of two types:

- LZ1** is a *sliding window* method based on the premise that if a string occurs once, it is probably repeated. The sliding window maintains a recent history of the data stream being compressed. As the data is processed, the window around a byte is searched for matching patterns. Although this algorithm provides quick response to changing data patterns because of the sliding window, it contains some inherent speed limitations.

- **LZ2** and **LZW** (LempelZivWelch) are algorithms based on the *hashed dictionary* method; these algorithms offer an acceptable compromise between speed and compression ratio. This type of algorithm builds a symbol dictionary to represent strings as the data is processed and then look up matching patterns in the dictionary. By monitoring the compression ratio in this type of algorithm, a new dictionary can be started when the ratio drops, indicating a change in the data type. Thus, this type of algorithm is responsive to changing data patterns while maintaining acceptable speed.

Although dependent on the particular implementation, the LZ2/LZW type of algorithm is generally faster than the LZ1 type because the dictionary structure promotes efficient searching.

The Python DDS-DC drives use the DCLZ algorithm, which is based on the LZ2/LZW type of algorithm. Refer to Section 7.2 for a more detailed description of the algorithm.

7.1.3 Hardware Compression

Because the DCLZ algorithm is implemented in the AHA3101 data compression chip in the Python drive itself, the compression is transparent to the host computer and the user.

If data compression is implemented in software on the host computer rather than in the hardware of the drive, the transfer rate of the host can be retarded because the host must perform compression computations in addition to its regular computations. Also, any other host that wants to retrieve (decompress) the data must have the same software resident.

The AHA3101 data compression chip is designed to facilitate a complete data compression system using the DCLZ algorithm. This chip provides support circuitry as well as the core DCLZ machine.

Refer to Section 7.3 for a more detailed description of the data compression chip.

7.1.4 Data Integrity

Generally, errors do not occur within the electronics of a tape drive. When errors do occur, the majority of these errors occur in reading or writing data to or from media where physical motion is involved and where environmental conditions are factors.

With hardware data compression in the drive, the data compression takes place within the drive electronically, which reduces the physical motion and thus further reduces the possibility of errors. For example, if the error rate without data compression is 1 in 10^{15} and a compression ratio of 4:1 is achieved, then the data compression error rate becomes 1 in 4×10^{15} . This decrease in error rate becomes a multiple of the compression ratio. However, because more data is recorded on a given physical space with data compression, more data is lost than without data compression if an error occurs in that space.

The DCLZ algorithm is itself a lossless algorithm, which ensures that the compressed data can be decompressed without error. Of course, if an error is present in the uncompressed data, that error is still present in the compressed, and then decompressed, data.

7.2 DCLZ Algorithm

The DCLZ algorithm used in the Python DDS-DC drives is based on the LZ2/LZW algorithm type described in Section 7.1.2. This algorithm has been approved by the US ANSI standards group and the European ECMA standards group. The DDS Manufacturers Group and QIC tape industry standards committees accept DCLZ as an approved standard.

The DCLZ algorithm removes redundant information in a data stream by recognizing and encoding patterns of input characters. When a unique string of characters is encountered, that string is entered into a dictionary and assigned a numeric value. After a dictionary entry is created, when that string (the dictionary entry) is again encountered, it is replaced in the data stream by its numeric value (codeword).

7.2.1 Simplified Compression Operation

The following steps describe a simplified version of operation of the algorithm for compressing data.

1. From the current position in the input data stream, the algorithm fetches bytes (characters) until a string is formed that does not have a matching entry in the dictionary.
2. The codeword for the longest string that does have an entry in the dictionary (all bytes except the last) is output.
3. A dictionary entry for the string formed in step 1 is created.
4. The current position is moved to the last byte of that string.
5. Steps 1 through 4 are repeated until the input data stream is completely processed.

Table 7-1 illustrates this simplified operation.

Table 7-1. Simplified Compression

Input Byte	Current String	Match ?	Build Entry	Output Code Value
R	R	Y		
I	RI	N	RI	(R)
	I	Y		
N	IN	N	IN	(I)
	N	Y		
T	NT	N	NT	(N)
	T	Y		
I	TI	N	TI	(T)
	I	Y		
N	IN	Y		
T	INT	N	INT	(IN)
	T	Y		
I	TI	Y		
N	TIN	N	TIN	(TI)
	N	Y		

7.2.2 Dictionary

The dictionary is built and contained logically in external RAM and is not output as a distinct item. Rather, the decompressor recreates the dictionary in order to recreate the original data exactly.

The dictionary allows up to 4096 entries with each entry made up of

- The unique string found in the data stream
- The codeword for that string

Codewords represent strings of up to 128 characters and are formed by adding a new character to an existing codeword. These codewords range from 9 through 12 bits in size and are assigned a number in the range 0 through 4095.

These codewords are either control flags, encoded bytes, or dictionary codes. The following points explain these three types of codewords.

- Control Flags, codewords 0 through 7:* These control flags are reserved codewords that flag specific conditions as follows:
 - 0 Dictionary frozen
 - 1 Dictionary reset
 - 2 Increment codeword size
 - 3 End of record (EOR)
 - 4-7 Reserved
- Encoded bytes, codewords 8 through 263:* These encoded bytes represent single bytes of the input data stream and contain the values 0 through 255.
- Dictionary codes, codewords 264 through 4095:* The dictionary codes refer to dictionary entries and represent multiple bytes (a string of characters) in the input data stream. These codes are built as the input stream is processed. These codes are pointers to other locations and eventually end by pointing to one of the byte values 0 through 255. Thus, a linked chain is created that builds up a string of characters.

Each dictionary entry is 23 bits long and comprises a logical RAM address. The information is physically stored in 8-bit wide static RAM chips that are 8K, 10K, or 16K by 22-bits. The structure of each dictionary entry is as follows:

- Bits 0 through 7* contain the byte value of the entry.
- Bits 8 through 19* contain the codeword that represents the entry or that points to a previous entry (encoded byte or dictionary code).
- Bits 20-22* are condition flag bits.

Dictionary codewords range from 9 through 12 bits in length and correspond to dictionary entries from 0 through 4095. These entries are divided as follows:

- First 512 entries are 9-bit codewords.
- Second 512 entries are 10-bit codewords.

- Next 1024 entries are 11-bit codewords.
- Final 2048 entries are 12-bit codewords.

7.2.3 Simplified Decompression Operation

The DCLZ algorithm requires that compression and decompression be tied together through

- The compression and decompression processes (requires synchronization)
- The packing and unpacking of codewords into a byte stream (requires synchronization)

That is, decompression of the data does not begin at an arbitrary point; rather, it begins at a point where the dictionary is reset---known to be empty. This stipulation is vital because the dictionary is embedded in the codewords, which saves time and space as it is not "recorded" separately.

Likewise, the packing and unpacking process require synchronization such that the compressed data is presented to the algorithm in the proper order.

The following steps describe a simplified version of the operation of the algorithm for decompressing data.

1. From a reset dictionary point (which contains only control codes and encoded bytes) codewords are fetched from the input stream and looked up in the dictionary.
2. New dictionary codes are built by combining the previously received codewords. (Thus, the dictionary created during compression is recreated, guaranteeing that any codeword received is contained in the dictionary.)

Codewords that are encoded bytes are output directly. Codewords that are dictionary codes lead the algorithm through a series of bytes and codewords that point to other dictionary entries. Bytes are stacked until an encoded byte occurs; then, the stack is output.

Table 7-2 illustrates the reverse process of compression showing simplified decompression operation.

Table 7-2. Simplified Decompression

Input Code Value	Byte Value	Pointer	Root ?	LIFO	Entry	Output Byte
(R)	R	--	Y	R	--	R
(I)	I	--	Y	I	RI	I
(N)	N	--	Y	N	IN	N
(T)	T	--	Y	T	NT	T
(IN)	N	(I)	N	N		
	I	--	Y	NI	TI	I
(TI)	I	(T)	N	I		
	T	--	Y	IT	INT	T
						I
(N)	N	--	Y	N	TIN	N

Table 7-3 shows the dictionary based on Tables 7-1 and 7-2.

Table 7-3. Dictionary

Codeword	Byte Value	Code Value (Pointer)
(RI)	I	(R)
(IN)	N	(I)
(NT)	T	(N)
(TI)	I	(T)
(INT)	T	(IN)
(TIN)	N	(TI)

7.3 AHA3101 Data Compression Chip

The AHA3101 data compression chip used in the Python DDS-DC drives provides the basis for an effective data compression system using the DCLZ algorithm. Using this chip implements hardware data compression in the drive itself.

Like other component devices used in the state-of-the-art Python drives, the AHA3101 uses the low power CMOS technology. The device is packaged in a 100-pin plastic quad flat pack (PQFP).

The AHA3101 contains compression and decompression circuitry, which forms the core of the data compression system by implementing the DCLZ algorithm. This core is called the *DCLZ machine*. In addition to the DCLZ machine, this data compression chip provides support circuitry in the form of four main interfaces:

- The *processor interface* allows a microprocessor to access information inside the chip and in the SRAM buffer.
- The *Port A and Port B DMA interfaces* are used to transfer uncompressed and compressed data through the chip.
- The *external static RAM (SRAM) interface* allows high speed access to the external SRAM, which is used to store DCLZ algorithm information and to buffer data.

These interfaces offer a means of tuning parameters provided in the DCLZ algorithm. The major parameters are

- Dictionary size selection
- Logical record control
- Parameters to use in evaluating the current compression ratio

These parameters can be used to determine the dictionary reset points in order to optimize searching and, in conjunction with the chip counters, to allow the processor to evaluate the compression ratio dynamically. This dynamic evaluation of the compression ratio enables the processor to detect a drop in the compression ratio performance and thus force a dictionary reset to adapt to the new pattern structure of the data stream.

In the Python DDS-DC drives, the AHA3101 is inserted in line between the SCSI protocol LSI and the main data buffer of the drive using the Port A and B DMA interfaces. These ports each support the 5 MB/sec burst transfer rate.

Connected to the SCSI protocol LSI, Port A transfers uncompressed data to and from the SCSI bus. Connected to the DCLZ machine by an internal connection, Port B transfers compressed data to and from the main data buffer of the drive.

The data flow through the AHA3101 data compression system for compression is generally as follows:

1. During compression, the input data stream flows into an SRAM data buffer through the Port A DMA interface.
2. From the SRAM buffer, the input data stream (uncompressed data) then flows to the DCLZ machine. In the DCLZ machine, the data is compressed and the codewords are packed into bytes.
3. The compressed data and codewords are then sent to the main data buffer of the drive through the Port B DMA interface.

The data flow through the AHA3101 data compression system for decompression is generally as follows:

1. During decompression, the data stream flows through the Port B DMA interface to the DCLZ machine. In the DCLZ machine, the data is unpacked into codewords and decompressed.
2. From the DCLZ machine, the data is sent into the SRAM buffer.
3. From the SRAM buffer, the uncompressed data flows through the Port A DMA interface to the SCSI protocol LSI.

Refer to Chapter 8 for a functional block diagram showing the AHA3101 data compression chip.

CHAPTER 8

THEORY OF OPERATION

8.1 Overview

The design of the Python DDS-DC DAT drives integrates the DAT technology (helical scan recording method) into a true computer grade data storage peripheral with industry standard data compression capability.

The Python DDS-DC DAT drives are the result of:

- Combining the economies of scale from the audio electronics market for key components such as the cylinder, heads, and audio LSIs with a computer grade drive (3.5-inch) using four direct drive motors, a "no-mode change" mechanism, and electronic tape path control for the demanding computer storage environment.
- Implementing a four-head design to provide read-after-write (RAW) error correction and to maximize the benefits of the helical scan recording method, namely: (1) high density recording (all tape space is utilized by dense, overlapping tracks at alternating azimuth angles) and (2) high speed searches.
- Using second generation audio and custom LSIs for efficient circuit layout and increased reliability with low power consumption. These LSIs are quad-flat-pack (QFP) designs using complementary metal-oxide semiconductor (CMOS) technology.
- Implementing the DDS and DDS-DC formats.
- Implementing hardware data compression in the drive using the AHA3101 chip.
- Using flash EEPROM devices for easy firmware upgrades.
- Implementing a custom ECC C3 coprocessor and other error correction techniques.
- Embedding a full-LSI SCSI controller with capability for both SCSI-1 or SCSI-2 command sets in single-ended SCSI DDS-DC models.

This chapter describes the Python data compression drive in more detail and explains implementation specific information.

8.2 Block Diagrams

The electronics of all Python data compression models that use the self-contained Python 3.5-inch DDS-DC drive (Models 4322, 4542, and 4352) are laid out on one main printed circuit board (PCB) and five small PCBs. The five small boards control specific functions:

- The RF PCB controls the head assembly.
- The Switch PCB contains the eject switch and the front panel LEDs, and other switches.
- The SCSI Controller PCB contains the SCSI controller, buffer memory, the AHA3101 data compression chip, compression memory (SRAM), and ECC processor.
- The SCSI Interface PCB contains the SCSI connector, power connector, configuration switches, serial port, terminator power jumper, and terminators.
- The Capstan/Cylinder Drive PCB contains the cylinder servo circuits and the cylinder and capstan drivers.

Figure 8-1 is a simplified block diagram that applies to all models of the Python DDS-DC drives. Figure 8-2 is a simplified block diagram of the SCSI Controller PCB for Python models 4322NT and 4542NT. Figure 8-3 is a simplified block diagram of the SCSI Controller PCB for Python models 4322NP, 4542NP, and 4352XP.

8.3 Python DDS-DC DAT Drives

This section generally describes the hardware design features of the Python data compression drives. You may want to refer to the block diagrams referenced in Section 8.2 as you read this information.

Python computer DAT drives use the helical scan recording method employing a four-head design. Four direct-drive motors and one brush-type motor are used in the drive. Also, the read and write functions are implemented using LSIs. Engineering decisions such as the modular partitioning of the electronics and use of surface mount, low power commercial and custom LSIs allow Python drives to conform to the industry accepted 3.5-inch form factor. These design features are also important contributors to the overall reliability, durability, and performance of the drive.

Additionally, the Python mechanism is designed for minimum tape wear and prevention of damage to the tape. The modes or operational states, such as stop, rewind, play, etc., are specifically implemented to reduce mechanism and tape wear. Fewer mechanical mode changes result in less wear on key drive components. In some cases, the need for a mode change is circumvented using the Pause mode, which stops the tape without activating the mechanism. Figure 8-4 shows the operational modes in relation to mode changes. All mode selection is performed by the controller firmware. The host computer does not directly control mode selection.

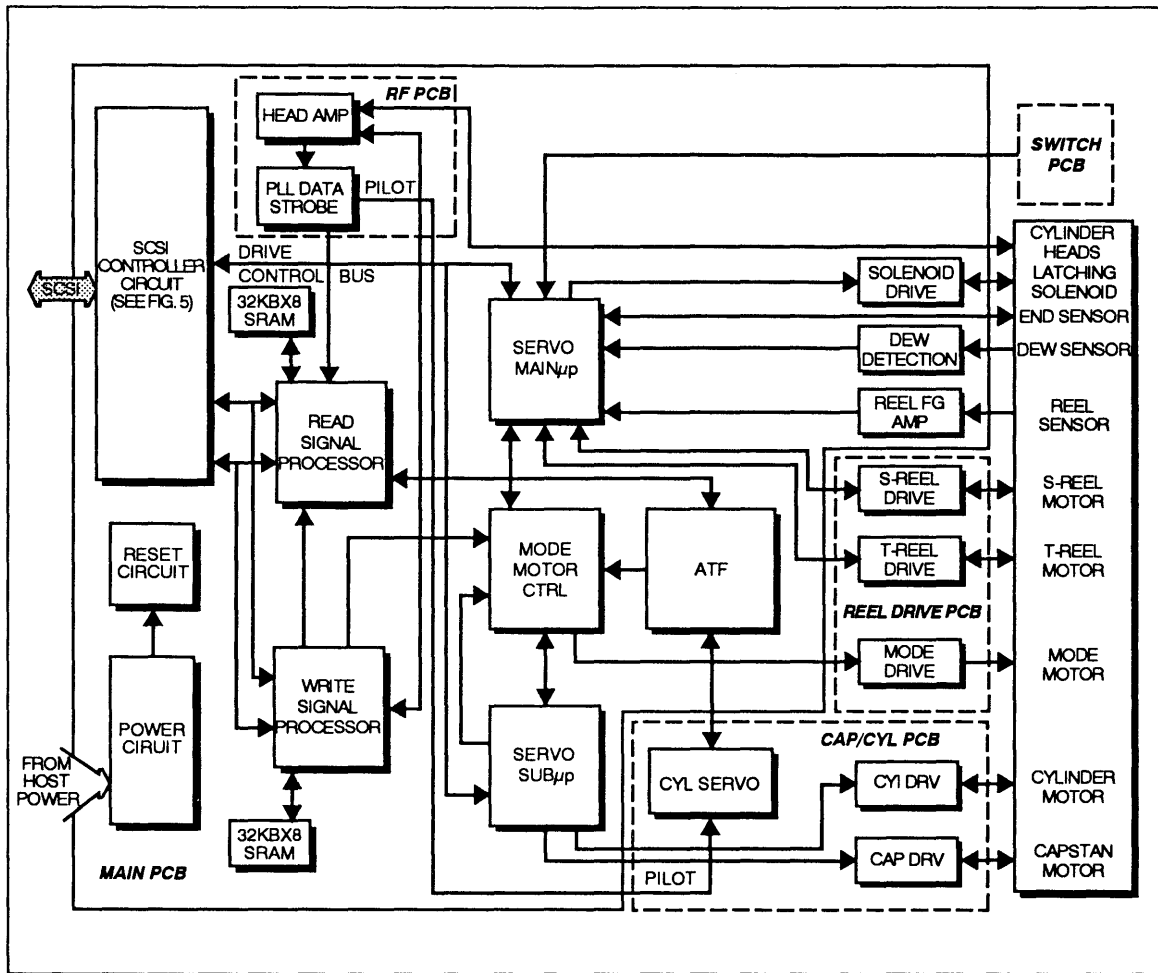


Figure 8-1
Simplified Block Diagram -- Python DDS-DC Drives

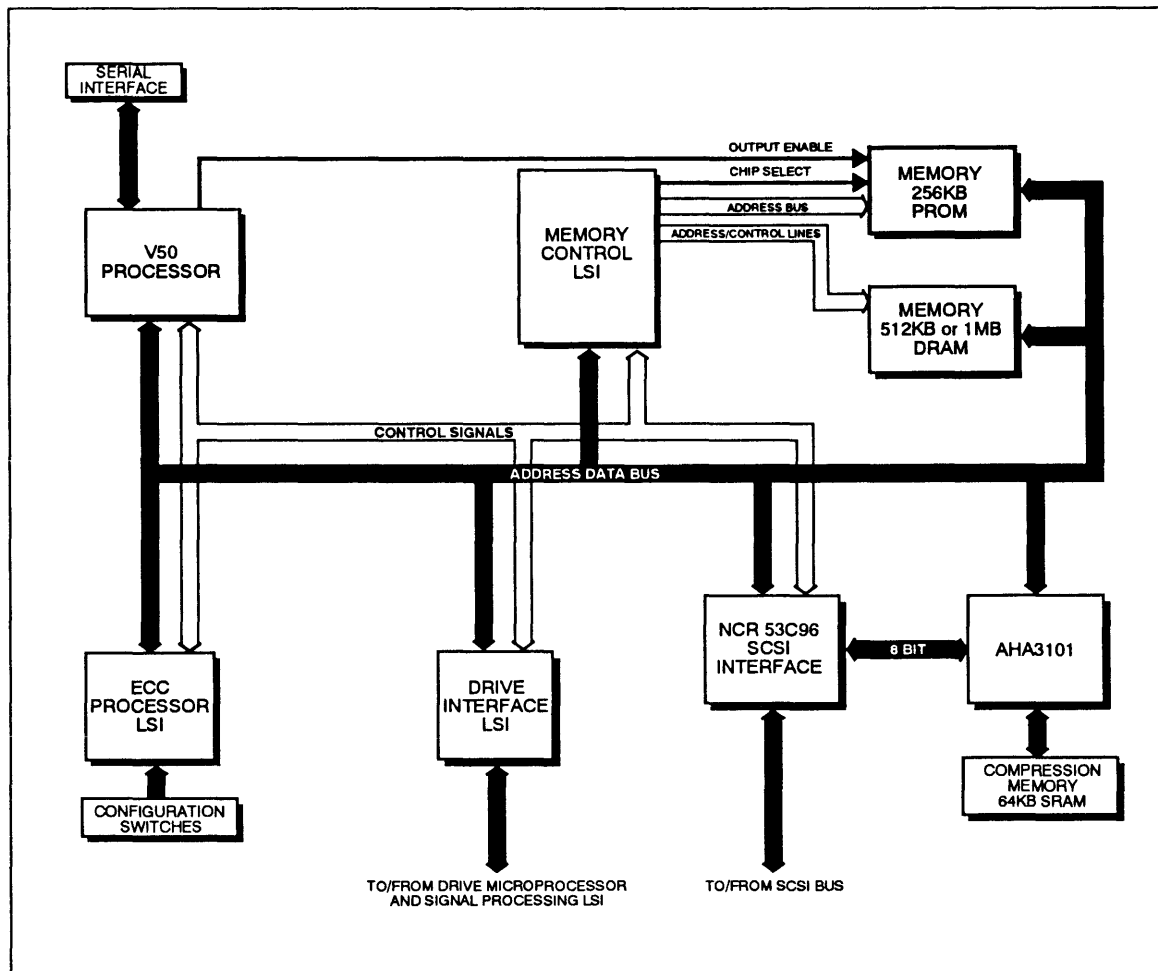


Figure 8-2
Block Diagram -- SCSI Controller Python Models 4322NT and 4542NT DDS-DC Drives

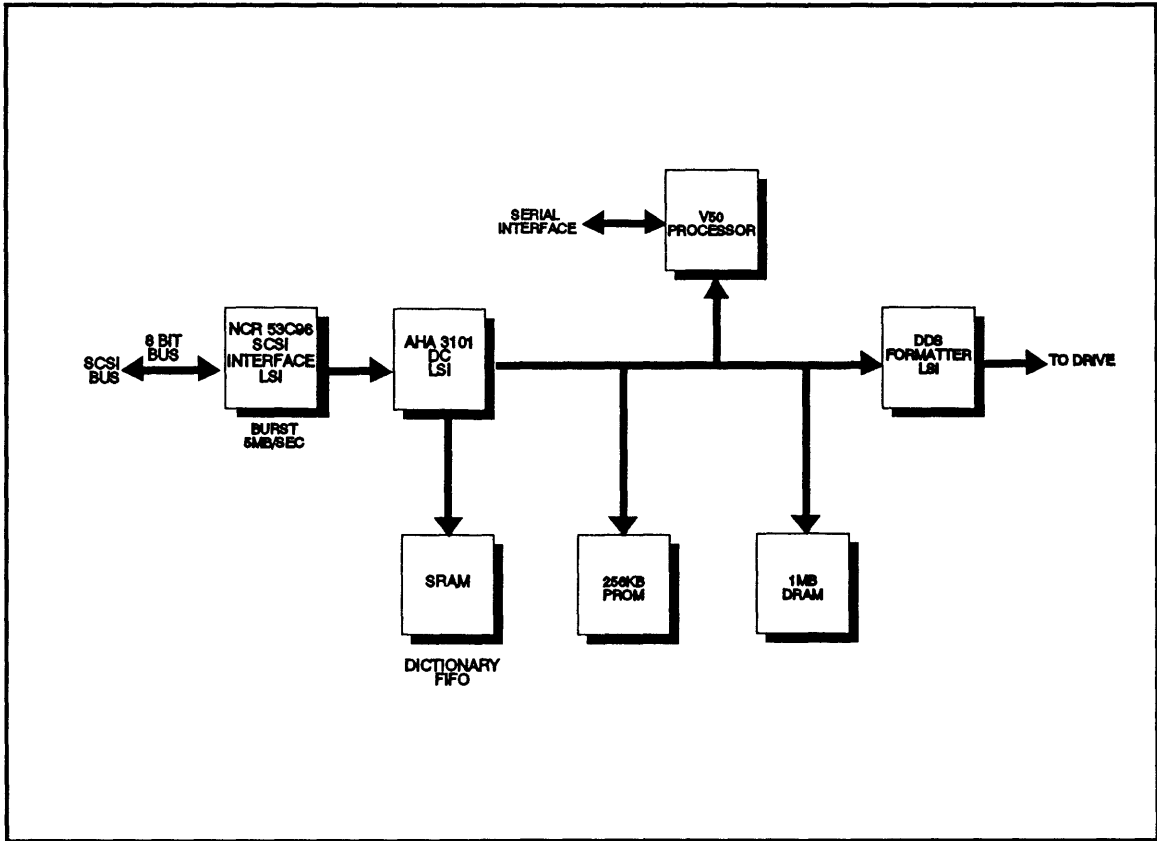


Figure 8-3
Block Diagram -- SCSI Controller Python Models 4322NP, 4542NP,
and 4352XP DDS-DC Drives

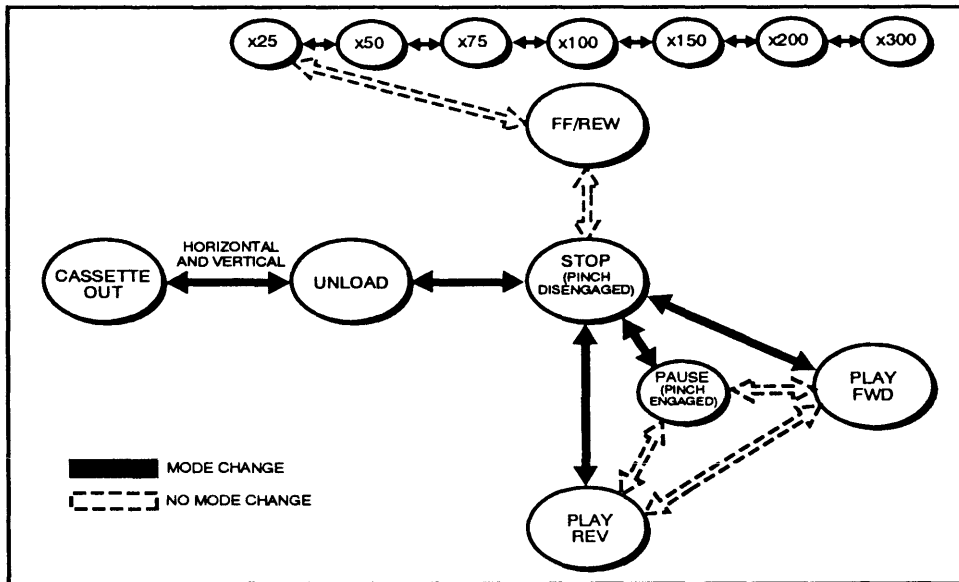


Figure 8-4
Python State Diagram.

The custom automatic track finding (ATF) LSI combined with the four-head design implements the ANSI DDS standard specifications and provides the precision required to perform *seamless appends*, the ability to add information after a stop immediately adjacent to the last data written on the tape, with no significant loss of tape storage capacity.

In Python internal models, an accessible switch bank lets you select certain operational options, such as a default at power up of SCSI-1 or SCSI-2 interface, SCSI parity enable, DDS pass-through (uncompressed) operation, and a power-on diagnostic self-test. A serial port is provided for access to firmware that provides internal drive access. Also, a 6-pin header allows remote SCSI ID selection. Refer to Section 3.5.1 for information about setting these operational switches.

Additionally, the internal models provide an accessible jumper at the rear of the drive that allows you to enable terminator power if needed. (The default for Python internal models is with terminator power disabled. For external Python drives, the default is with terminator power enabled.)

NOTE: Also, a terminator power fuse is located at the rear of internal drives to provide protection from component damage in case the SCSI cable is connected upside down (with terminator power enabled).

The drive provides several sensors, both optical and mechanical, to conform to the ANSI Helical Scan Tape Cassette specification; to effectively implement mode changes; and to insert and eject the cassette.

Two rectangular front panel light-emitting-diodes (LEDs) indicate drive busy status and tape cassette in place status. When blinking, these LEDs also function as fault indicators. (Refer to Section 4.3 for a summary of the function of these LEDs.) The 4352 external subsystem also provides a round, green LED on the front panel to indicate power on.

The general information above is discussed in more detail in the following sections.

8.3.1 Helical Scan Recording -- Four-Head Design

In helical scan recording, the heads are positioned opposite one another on a cylinder, which is tilted approximately 6 degrees from the vertical plane and rotates counterclockwise at 2000 revolutions per minute (rpm). At the same time, the tape moves slowly (0.32 inches per second) in a horizontal path around part of the cylinder. This simultaneous motion of cylinder and tape results in the head traveling across the width of the tape in a helix-shaped motion.

The Python drive is designed with four, long-life ferrite heads -- two read and two write heads. These heads are set opposite one another on the cylinder with a rotation sequence of: write A, read B, write B, read A (or write A new, read B old, write B new, read A old). The advantage of this design is that a RAW check can be performed immediately after the data is written.

The cylinder rotates rapidly (2000 RPM) in the same direction that the tape moves. The wrap angle of the tape on the cylinder is approximately 90 degrees. This wrap angle plus the slow tape speed minimize tape and head wear. The combined movement of the tape and cylinder results in a relative head-tape speed of 123 ips.

Figure 8-5 illustrates a helix track, the four-head design, and shows the 90 degree wrap angle.

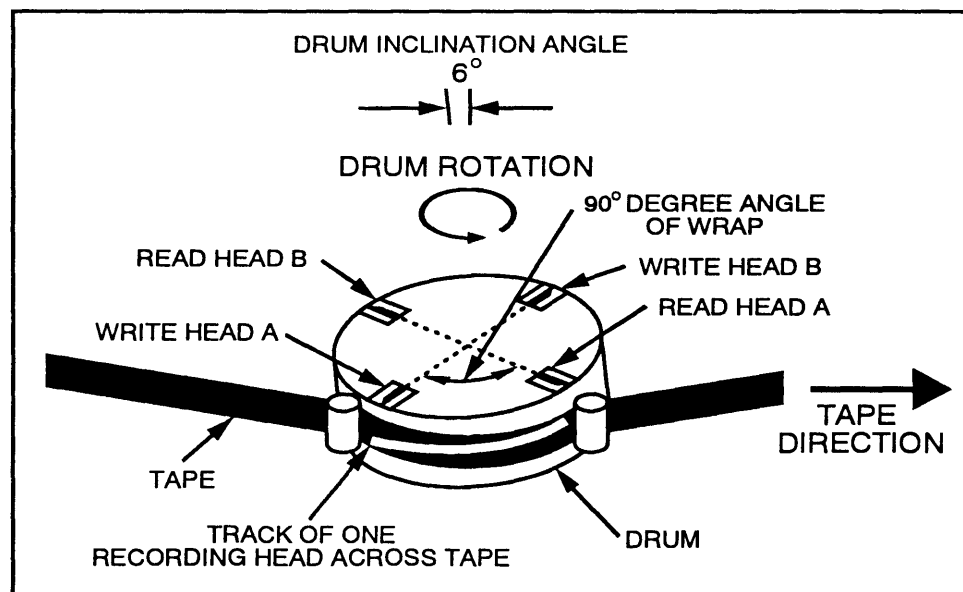


Figure 8-5
Four-Head Design

The recorded tracks are written diagonally across the tape from bottom to top by each write head. Because the head is wider than the track written, tracks overlap with no tape space between them. In conventional recording, such overlap or even "close proximity" would result in *crosstalk* (signals from adjacent tracks interfering with signals from another track).

However, in helical scan recording, the heads are set at a different azimuth angles so that alternate tracks on the tape are written at alternate azimuth angles. (See Figure 8-6.) Because the read head is set to the same angle as its corresponding write head, it picks up a stronger signal from data written in the same azimuth angle from itself. Thus, it reads the track with minimal crosstalk. At the same time, the head is maintained centered in the track by using the ATF reference bursts.

Figure 8-6 shows alternate tracks and alternate azimuth angles.

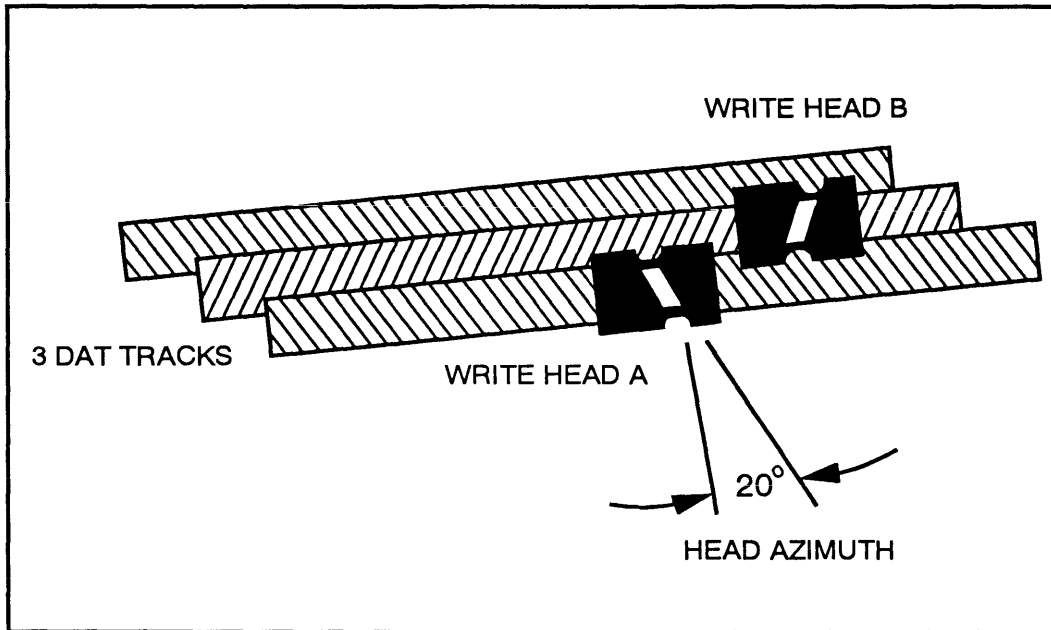


Figure 8-6
Alternate Azimuth Angles

8.3.2 Motors and Control Circuits

The Python products use four direct-drive, brushless motors: the capstan, cylinder, and two reel motors. Using these small, direct-drive motors provides maximum reliability. The cylinder motor rotates the cylinder. The capstan motor moves the tape and loads and ejects the cassette. The two reel motors turn the tape reels.

The Cylinder Servo LSI on the Capstan/Cylinder Drive PCB communicates with the ATF LSI, which, in turn, sends servo data to the Servo Sub LSI. The Servo Sub controls the Cylinder and Capstan Drivers. The Supply-Reel and Takeup-Reel motors are controlled by the Servo Main LSI.

The fifth motor in the mechanism is a brush-type mode motor. This motor controls (selects) the mechanism mode. Because the mode motor is not frequently used, the brush type motor is best suited to this application. The mode motor performs the mode changes as directed; for example, this motor conditions the mechanism to eject the cassette. The motor is controlled by a driver that receives instructions from mode motor controller.

Servo Main Microprocessor and Servo Sub Microprocessor LSIs

Two custom LSIs (84-pin, QFPs)--the Servo Main Microprocessor and Servo Sub Microprocessor---provide the processing power for the various motion control circuits within the Python drives. The Drive Control Bus connects these two LSIs with the SCSI controller circuits. These two microprocessor LSIs control all the drive motors.

Read and Write LSIs

Two single-chip signal processor and audio DAT-formatter LSIs provide the read and write signals for the drive. Each LSI is supported by a 32-KB X 8 static RAM. These chips are controlled by the controller microprocessor.

ATF Circuitry

The ATF circuitry of the Python drive is designed to provide high precision tracking and head positioning in compliance with the ANSI DDS standard. The main component for ATF is a custom ATF LSI, which, in conjunction with the four-head read-after-write (RAW) design, allows *seamless appends*, also in compliance with the ANSI DDS standard.

A seamless append is the continuation of writing frames on the end-of-media (EOM) side of existing frames (after a STOP) such that the tracks are placed as if they were in a continuous sequence. That is, the servo system must be able to read the appended tracks without encountering discontinuity or gaps between tracks.

Performing a seamless append requires the highest precision and almost absolute accuracy in repositioning the head assembly. In the Python products, this level of precision is attained through the combined accuracy of the mechanical design and the ATF signal information as implemented through custom circuits.

SCSI Controller

The embedded SCSI controller circuitry in the Python drive is made up of several components. In the 3.5-inch based 4322, 4542, and 4352 models, a single chip DDS formatter LSI communicates with the Servo Main Microprocessor, Servo Sub Microprocessor, and Read and Write LSIs. The C3 ECC coprocessing capability and the Memory Control function are also included in this single chip. Other components vital to this circuitry are the high-performance SCSI LSI (NCR-53C96), the microprocessor (NEC V50), and the EPROM (Intel 27C210). The standard dynamic RAM (DRAM) buffer is 512 KB; a 1-MB buffer is optional on most models. Buffer parity checking is standard.

The AHA3101 data compression chip, which is a 100-pin PQFP, is linked to the SCSI LSI (NCR-53C96) by the Port A DMA interface of the AHA3101 chip. The SRAM (compression memory) external to the AHA3101 chip is part of the SCSI controller circuitry. Refer to Section 7.3 for a more detailed discussion of the AHA3101 data compression chip.

8.3.3 Flash EEPROM

Because the 4322NP, 4542NP, and 4352NP drives use flash EEPROM (electronically erasable, programmable read-only memory), the drive firmware can be easily upgraded when new revisions of the firmware are released. The SCSI controller circuitry includes 256 KB of flash EEPROM on these models.

Loading new firmware can be accomplished in one of three ways:

- Using a specially encoded Archive firmware upgrade cassette.
- Issuing a WRITE DATA BUFFER SCSI command to download the firmware to the EEPROM. (Refer to Appendix C for information on this SCSI command.)
- Using the DATMON software that is included with the Archive Serial Port Diagnostic Kit product. (The Archive Python Serial Port Diagnostic Kit is a separate product available to qualified OEMs for testing Python DAT drives. Contact your Archive sales representative for information.)

Refer to Section 4.8 for information about loading new firmware using an Archive firmware upgrade cassette.

8.3.4 Sensors

A number of mechanical and optical sensors are integrated in the Python design. The cassette in and cassette loading sensors are mechanical sensors to determine the position of the loading mechanism. The other mechanical sensors determine specific information based on detecting the open or closed state of four recognition holes in the DAT cassette. The open/closed state of these holes determine tape thickness, cleaning cassette, and whether the tape is prerecorded (write-protect) or unrecorded. These mechanical sensors plus the sensor for "cassette in" status are designed to comply with the ANSI DAT cassette specification.

The beginning-of-tape (BOT) and end-of-tape (EOT) sensors are optical sensors designed to detect the light path transmissivity of leader and trailer tapes as specified in the ANSI DDS cassette standard. The reel sensors for the two reels are optical. Also, three optical sensors detect mechanism position during mode changes.

The capstan sensor is a magneto-resistive Hall sensor that detects a magnetic field. The cylinder sensors are coil and magnet sensors. Each reel motor contains a high-resolution, optical speed encoder.

The dew sensor is a humidity sensor that uses carbon-composition film formed by a hygroscopic resin to detect humidity.

8.3.5 Audio Output Jack

The Python drives can read prerecorded 44.1 KHz wide-track DAT audio tapes or 48 KHz normal-track tapes. When an audio DAT cassette is inserted, the drive automatically plays the tape.

The Python internal models provide pads on the rear panel for digital audio output so that mixed media applications have the capability for audible stereo output through a connected audio device when used with an external digital-to-analog converter, preamplifier, amplifier, and speakers.

8.4 DAT Cassette

The Python DDS-DC products are designed to use data-grade DDS DAT cassettes, which comply with the specifications in the 3.81 mm Helical-Scan Digital Computer Tape Cassette for Information Interchange, ANSI X3B5/89-156 standard. Archive recommends Archive-qualified, data-grade DDS DAT cassettes (Archive Model M31300, 60 meters; Archive Model M32000, 90 meters) to ensure optimal data integrity and reliability.

Archive also recommends the use of an Archive-qualified DDS DAT head-cleaning cassette (Archive Model M7301).

NOTE: *Proper maintenance of the Python drive requires that the DDS head-cleaning cassette be used every 25 hours of read/write operation and whenever the rectangular green, cassette-in-place LED flashes during operation.*

Chapter 9 discusses the DDS DAT head-cleaning cassette in detail.

Both DAT data and head-cleaning cassettes can be ordered from Archive and are packaged in multiples of five.

These small (approximately 2 inches x 3 inches x 0.4 inch) cassettes house magnetic tape that is 3.81 mm (0.150 inch) wide. The DAT cassettes are slightly bigger than a credit card. Figure 8-7 shows the DAT cassette.

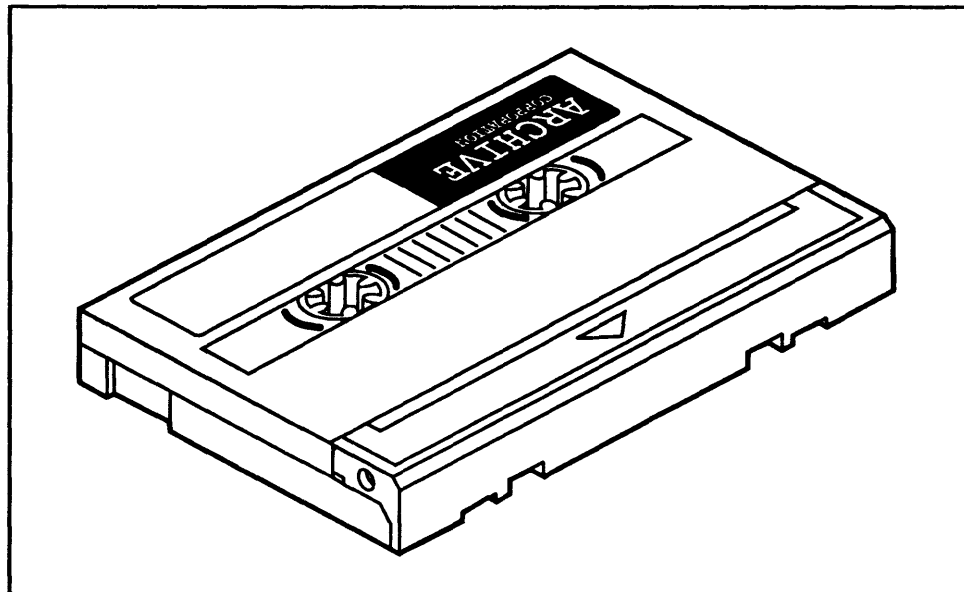


Figure 8-7
DAT Cassette

Qualified DAT cassettes are designed with specific file protect, lid, and other features for information interchange and are tested to comply with the ANSI DDS specifications.

Figure 8-8 points out the four recognition holes that allow the drive sensors to identify the type of tape, its magnetic thickness, and to determine whether the tape is prerecorded or unrecorded or is a cleaning cassette. Other cassette features to allow the drive to optically sense "cassette in", BOT, and EOT.

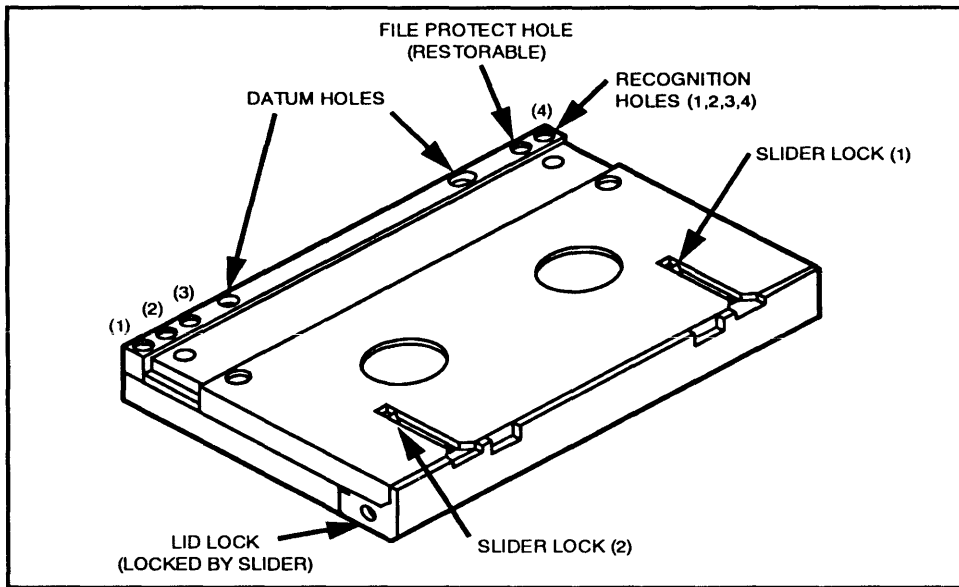


Figure 8-8
Cassette Design Features.

The cassette also provides for write protection so that existing data on the cassette is not overwritten. A write-protected cassette allows the existing data to be read but does not allow new data to be written to the tape.

NOTE: *A write-protected cassette does not allow the error log (in the System Area) to be updated.*

Figure 8-9 shows the sliding write-protect tab on the DAT cassette and its positions for write protect and write permit. When the tab is pushed into the closed position, it allows writing to the cassette tape.

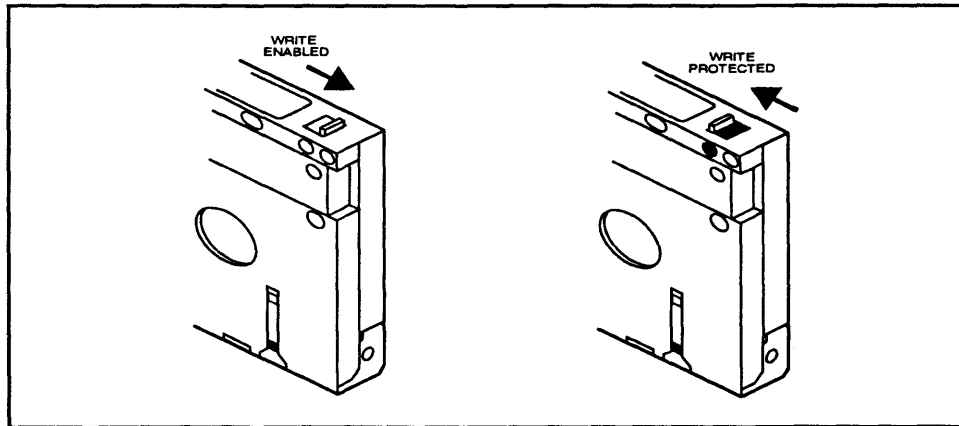


Figure 8-9
Write-Protect Tab on the DAT Cassette

Table 8-1 lists the five cassette tape lengths that may be used with Python products. Contact your Archive sales representative for information about other tape lengths not listed here.

Table 8-1. Tape Lengths and Capacities

LENGTH (Meters)	TIME (Minutes) (at play or constant, streaming backup)	TAPE CAPACITY (Gigabytes)		
		Uncompressed	Compressed Typical Maximum	
90	180	2.00	4.00	8.00
60	120	1.32	2.64	5.28
45	90	.99	1.98	3.96
30	60	.66	1.32	2.64
23	45	.49	.98	1.96

NOTE: *With data compression, tape capacity depends on the characteristics of the data being compressed. "Typical capacity" reflects a 2:1 compression ration, typical of a mix of general purpose data files such as spreadsheet and word processing files. "Maximum capacity" is not an absolute maximum but rather a nominal maximum, such as with types of conventional files which can contain more redundant data, such as certain database files.*

NOTES

CHAPTER 9

MAINTENANCE AND RELIABILITY

9.1 Maintenance

If excessive magnetic dust or debris collects at one or more of the heads, magnetic media may become unreadable or unwriteable. This situation may occur infrequently, or not at all, depending on the media used.

9.1.1 Head Cleaning

Whenever the Cassette In Place status LED (green) flashes, you should clean the drive heads with a cleaning cassette.

Also, as routine maintenance, the drive heads should be cleaned after every 25 hours of read/write operation.

NOTE: *The slowly flashing green LED may also refer to a damaged tape or a tape nearing the end of its life. If cleaning the head does not correct the flashing LED condition, replace the cassette. The slowly flashing LED does not indicate a loss of data nor does it affect SCSI operation. (A slowly flashing green LED in conjunction with the amber LED indicates the presence of a prerecorded audio tape.) Also see Section 3.7 for a description of LED operation.*

To clean the heads of the Python drive, use only an Archive-qualified DDS DAT cleaning cassette designed for DDS drives. Archive offers a cleaning cassette, Model M7301 that you may order. Cleaning cassettes are ordered (and packaged) in multiples of five.

The DDS cleaning cassette contains the correct recognition holes to allow the Python drive to recognize that it is a cleaning cassette. Follow these general guidelines to use the cleaning cassette:

- Insert the cleaning cassette. The Python immediately detects that the cassette is a cleaning cassette.

The drive loads and runs the cassette for about 20 seconds; then ejects the cassette.

NOTE: *Each time the cleaning cassette is used, the tape advances over an unused portion. Eventually, the entire tape is used, and a new cleaning cassette is required. (A cleaning cassette provides approximately 60 uses.) The Python drive will not rewind the cassette.*

Do not use an audio DAT cleaning cassette. It will not be properly recognized by the Python drive.

9.1.2 Automatic Drive Spin-Down and Write

In order to maximize tape and drive mechanism life, the Python drive automatically executes the following functions when no tape READ or WRITE activity occurs:

- After 30 seconds, the capstan and pinch roller are released. Tape tension is removed.
- After 90 seconds, the tape is pulled away from the cylinder, and the cylinder is stopped.

If a READ or WRITE operation occurs, normal operation resumes with no affect on host operation.

If tape WRITE operations cease, a partially full data buffer may remain. After one minute with no activity, the Python drive automatically writes the partial buffer to the tape. This automatic action minimizes the possibility of lost data if the power fails.

If data to be written remains in the buffer and the eject button is pushed, the data is written to tape before the tape is rewound and ejected.

9.1.3 Guidelines for High Temperature or Humidity Conditions (Outside the Specified Operating Environment)

If the drive detects moisture (dew), the amber rectangular LED on the front panel flashes rapidly. Following the guidelines listed below can minimize the possibility of extreme temperature or humidity conditions causing problems with the Python drive.

- Use DAT cassettes only at temperatures between 5°C (40°F) and 40°C (113°F). The cassettes can be stored at temperatures down to -40°C (-40°F). Although the storage specifications range from 5°C to -40°C, do not leave cassettes in severe temperature conditions---like in a car in bright sunlight. Avoid extreme changes in temperature or humidity whenever possible.
- If cassettes are exposed to temperatures or humidities outside the specified operating environment, condition the cassettes by exposure to the operating environment for a time at least equal to the period the cassettes were exposed to the out-of-spec environment (to a maximum of 24 hours).
- Place the Python drive in a position that provides stable temperatures. Do not place the drive near open windows, fans, heaters, or doors.
- Do not read from or write to cassettes when a temperature change of 10°C per hour is occurring.

9.2 Reliability

The Python products are designed for maximum reliability and data integrity. Table 9-1 summarizes the reliability specifications.

Table 9-1. Reliability Specifications

FEATURE	SPECIFICATION
Nonrecoverable error rate	Less than 1 in 10 ¹⁵ bits
Error recovery and control	Error Correction Code techniques (C1, C2, and C3 ECC) Read-after-write (RAW) N-Group writing Error monitoring and reporting (Error Log) Media specification Retry on read Data randomizer Track Checksum
Mean-Time-Between-Failures (MTBF)	More than 40,000 hours at 30% duty cycle
Mean-Time-To-Repair (MTTR)	Less than 0.5 hour

9.2.1 Mean-Time-Between Failures

The Mean-Time-Between Failures (MTBF) is greater than 40,000 hours at 30% duty cycle. This specification includes all power-on and operational time but excludes maintenance periods. Operational time is assumed to be 30% of the power-on time. Operational time is the time the tape is loaded on the cylinder (tape moving or cylinder rotating).

NOTE: *Archive does not warrant the stated MTBF as representative of any particular unit installed for customer use. The failure rate quoted here is derived from a large database of test samples. Actual rates may vary from unit to unit.*

9.2.2 Mean-Time-To-Repair

The Mean-Time-To-Repair (MTTR) is the average time required by a qualified service technician to diagnose a defective drive and install a replacement drive. The MTTR for the Python DAT products is less than 0.5 hour (30 minutes).

The Python DDS-DC drive is a field replaceable unit. If a problem occurs with a subassembly or component in the drive, the entire unit should be replaced. The faulty drive should be returned to the factory in its original packaging. Contact your distributor, dealer, your computer system company, or your Archive sales representative to arrange the return.

NOTES

APPENDIX A

GLOSSARY OF TERMS

access point--Beginning of a sequence of compressed records at which point codewords to a decompression algorithm must start---whether or not the required data is at the beginning of the sequence.

Automatic Track Finding (ATF)--A method of ensuring the head is in the center of the track being read.

Azimuth--The angular deviation, in minutes of arc, of the mean flux transition line from the line normal to the tape reference edge.

Backup--Copy of a file or collection of files on fixed disk, diskette, or tape. Ensures against data loss.

Beginning of Media (BOM)--Equal to the physical beginning of the tape, where the leader tape is jointed to the magnetic tape.

Beginning of Tape (BOT)--Equal to the logical beginning of the tape.

Bezel--Front panel of a drive.

Bit--A single digit in the binary numbering system.

Bit Error Rate--The number of errors divided by the total number of bits written or read.

Byte--A group of 8 binary bits operated on as a unit.

Cassette--An enclosure containing magnetic tape wound on two coplanar hubs and driven by an external drive.

Compression Ratio---The ratio comparing the amount of uncompressed data to the amount of compressed data. It is obtained by dividing the size of the uncompressed data by the size of the compressed data.

Crosstalk--Signals from adjacent tracks interfering with signals from another track.

Data Area---The third section of a tape on a DAT cassette, which is written as a series of groups beginning with the special Vendor Group.

Data Compression---The process of removing redundant data from a data stream before recording the data to tape. Compressed data requires less storage space than uncompressed data.

Data Density--The number of single-byte characters stored per unit length of track. Usually expressed as bits-per-inch (bpi).

Decompression---The process of restoring compressed data to its original state.

Dew---Collection of moisture in a tape drive.

Dictionary---The collection of representations (numeric values) of unique character strings encountered in compressing data.

Disk Drive--A peripheral storage device that rotates the disk, writes data onto it, and reads data from it as instructed by a program.

DDS Format--The Digital Data Storage format for tape cassettes developed by Sony and Hewlett Packard for DAT computer peripherals.

DDS-DC Format--The Digital Data Storage Data Compression format for tape cassettes that is approved by the American National Standards Institute and the European Computer Manufacturers Association for DAT computer peripherals with data compression.

End-of-Data (EOD)--Indicates the point where the host stopped writing data to the tape.

End-of-Media (EOM)--Equal to the physical end of tape where the trailer tape is jointed to the magnetic tape.

End of Tape (EOT)--Equal to the logical end of the tape.

Entity---A recorded object that consists of an entity header followed by one or more compressed records.

Error Correction Codes (ECC)--Information written on tape during the recording operation that can later be used to reconstruct errors during the data reading operation.

File--A logical unit of information.

Fixed Disk--A non-removable hard disk. All data must be transferred to and from the disk via the computer.

Frame--Two adjacent tracks, one A channel (positive azimuth) and one B channel (negative azimuth).

Full-high (or full-height)--Usually refers to a tape drive fitting in a vertical space of 3-1/2 inches.

Group--A fixed capacity set of frames written to or read from the tape. For the DDS and DDS-DC formats, 22 frames comprise a group.

Half-high (or half-height)--Refers to the size of tape drive occupying a vertical space of about 1-1/2 inches.

Head Clog--Particles from the tape or from outside the drive adhere to the head gap on a read or write head and obstruct the reading or writing of data.

Helical Scan Recording--A method of magnetically recording a tape in which the tape wraps around a rotating cylinder with 2 or 4 read/write heads writing at different azimuth angles across the width of the tape in a helix-shaped track.

Interleaving--The process of shuffling the order of data bytes before writing them to tape so the consecutive bytes are recorded as far away from each other as possible.

Magnetic Tape--A tape that accepts and retains magnetic signals intended for input, output, and storage of data for information processing.

N-Group Writing--Sometimes called multiple group writing. This technique repeats each group of data so that there are N consecutive copies of each group on the tape.

Noise--A disturbance of the signal caused by the read channel, write channel, head/tape interaction, or conducted or radiated sources.

Randomizing--A recoding of data symbols before they are written to tape in order to provide a consistently uniform RF envelope level.

Read-After-Write (RAW)--Reading data immediately after it is written and writing the frame again if an error is found.

Reference & System Area---The second section of the tape on a DAT cassette, which provides logs of usage and soft error occurrences.

Track--A storage channel on recording tape. For DAT, specifically a diagonally positioned area on the tape on which a series of magnetic transitions is recorded.

Uncorrected Bit Error Rate--The probability of a bit being in error, without using any error correction techniques.

NOTES:

APPENDIX B

ACRONYMS AND MEASUREMENTS

B.1 Acronyms and Abbreviations

ACRONYM	MEANING
4DD	4 Direct Drive
ANSI	American National Standards Institute
ATF	Automatic Track Finding
BAT	Block Access Table
BIOS	Basic Input Output System
BOM	Beginning of Media
BOT	Beginning OF Tape
BPI	Bits Per Inch
CD	Compact Disc
CMOS	Complementary Metal-Oxide Semiconductor
CSA	Canadian Standard Association
DAT	Digital Audio Tape
DCLZ	Data Compression LempelZiv
DDS	Digital Data Storage
DDS-DC	Digital Data Storage Data Compression
DMA	Direct Memory Access
ECC	Error Correction Code
ECMA	European Computer Manufacturers Association
EEPROM	Electrically Erasable, Programmable Read-Only Memory
EOD	End of Data
EOM	End of Media
EOT	End Of Tape
FCC	Federal Communications Commission
FTPI	Flux Transitions Per Inch
GIT	Group Information Table
IEC	International Electrotechnical Commission
IPS	Inches Per Second
LED	Light Emitting Diode
LSI	Large Scale Integration
LZ1	LempelZiv 1 (algorithm)
LZ2	LempelZiv 2 (algorithm)
LZW	LempelZivWelch (algorithm)
MFM	Modified Frequency Modulation
MTBF	Mean Times Between Failures
MTTR	Mean Time To Repair

ACRONYM	MEANING
OEM	Original Equipment Manufacturer
PCB	Printed Circuit Board
PQFP	Plastic Quad Flat Pack
QFP	Quad Flat Pack
QIC	Quarter Inch Cartridge Drive Standards, Incorporated
RAM	Random Access Memory
RAW	Read-After-Write
SCSI	Small Computer System Interface
TTL	Transistor-transistor logic
UL	Underwriters' Laboratories, Inc.
VAC	Volts Alternating Current
VCR	Video Cassette Recorder
VDC	Volts Direct Current
VDE	Verband Deutscher Electrotechniker
VTR	Video Tape Recorder

B.2 Measurements

MEASUREMENT	MEANING
A	Amp
C	Celsius or Centigrade
cm	centimeter
dBa	decibels, A-weighted sound power reference one picowatt
F	Fahrenheit
ft	foot or feet
g	acceleration of a free-falling body; equal to 32.17 feet per second ²
GB	gigabyte
Hz	Hertz
in.	inch
k	kilo
KB	kilobyte
kg	kilogram
KHz	kilohertz
lb(s)	pound(s)
m	meter
M	mega
Mbits	megabits
MB	megabyte
MHz	megaHertz
mm	millimeter
ms	millisecond
RPM	revolutions per minute
V	Volt
W	Watt

NOTES:

APPENDIX C

DATA COMPRESSION---SCSI INFORMATION

C.1 Introduction

This appendix contains information specific to the SCSI-2 interface for the Python DDS-DC drives. This SCSI information is additional to the generic SCSI interface described in the *Python Technical Reference Manual* (P/N 25356-002, DDS-only) and in the *Python SCSI Manual* (P/N 27298-001), available in 1992.

This appendix describes the SCSI data compression control pages (MODE SELECT and MODE SENSE pages) as implemented on the Python DDS-DC drives. This appendix also includes information about the REQUEST SENSE command and implementation notes regarding data compression.

Finally, this appendix describes the WRITE DATA BUFFER command, which can be used to download revised firmware in the Python DDS-DC drives that contain flash EEPROM (Models 4322NP, 4542NP, and 4352XP).

C.2 MODE SELECT Page

The layout of the MODE SELECT page for data compression control is shown in Table C-1.

Table C-1. Data Compression MODE SELECT Page

BYTE	BITS							
	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	1	1
1	Page Length (0Eh)							
2	DCE	0	0	0	0	0	0	0
3	DDE	0	RED	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	Compression Algorithm							
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	XX							
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0

C.2.1 Compression Control---Byte 02

The Compression Control Byte (02) of the MODE SELECT page contains the Data Compression Enable (DCE) bit (07), which controls data compression as follows:

- If the DCE bit is 1, the Python drive compresses data received from the host during a WRITE command before it writes the data to tape in the DDS-DC format.
- If the DCE bit is 0, the Python drive does not compress data sent during a WRITE command, and the host data is written to tape in the uncompressed DDS format.

C.2.2 Decompression Control---Byte 03

The Decompression Control Byte (03) of the MODE SELECT page contains the Data Decompression Enable (DDE) bit (07) and Report Error on Decompression (RED) bit (05). The following subsections briefly describe the function of these two bits.

C.2.2.1 Data Decompression Enable (DDE) Bit (07)

The Data Decompression Enable (DDE) bit (07) of the Decompression Control Byte (03) of the MODE SELECT page controls data decompression as follows:

- If the DDE bit is 1, the Python drive decompresses data that has been compressed on the tape before it sends the data to the host during a READ command.
- If the DDE bit is 0, the Python drive does not decompress any data that it sends to the host during a READ command, and any compressed data read is sent to the host in the entity (DDS-DC) format.

C.2.2.2 Report Error on Decompression (RED) Bit (05)

The Report Error on Decompression (RED) bit (05) of the Decompression Control Byte (03) of the MODE SELECT page specifies when Check Conditions are reported to the host, as follows:

- If the RED bit is 0, the Python drive generates a Check Condition for every compressed entity that it cannot decompress.
- If the RED bit is 1, the Python drive generates Check Conditions only when the format of the SCSI data changes. A SCSI format change occurs when data changes from non-DCLZ compressed to uncompressed, from uncompressed to non-DCLZ compressed or from compressed with one non-DCLZ algorithm to compressed with a different non-DCLZ algorithm.

A RED bit of 1 is intended to be used when reading a tape to minimize the number of Check Conditions received by the host when data is present that cannot be decompressed by the drive. Any media-access command, other than the READ command, causes the drive to return to an initial state for determining when to report the Check Condition.

When the RED bit is first set and decompression is enabled, the drive is in an initial state in which it expects to read either uncompressed data or data compressed with DCLZ. When the RED bit is first set and decompression is disabled, the drive is in an initial state in which it expects to read uncompressed data. In either case, if the expected data is not read, a Check Condition is reported.

The Request Sense information indicates further information about the format change.

Table C-2 outlines the DDE and RED bit configurations.

Table C-2. DDE and RED Bit Configurations

DDE	RED	Definition	Sense Key
0	0	Disable decompression and report a Check Condition for every compressed entity encountered.	MEDIUM ERROR
0	1	Disable decompression and report a Check Condition for the following format changes: <input type="checkbox"/> from compressed to uncompressed <input type="checkbox"/> from uncompressed to compressed <input type="checkbox"/> from compressed with one algorithm to compressed with a different algorithm	NO SENSE MEDIUM ERROR MEDIUM ERROR
1	0	Enable decompression and report a Check Condition for every entity compressed with an unsupported algorithm	MEDIUM ERROR
1	1	Enable decompression and report a Check Condition for the following format changes: <input type="checkbox"/> from uncompressed to compressed with an unsupported algorithm <input type="checkbox"/> from compressed with DCLZ to compressed with an unsupported algorithm <input type="checkbox"/> from compressed with unsupported algorithm to uncompressed <input type="checkbox"/> from compressed with an unsupported algorithm to compressed with DCLZ <input type="checkbox"/> from compressed with an unsupported algorithm to compressed with a different unsupported algorithm	MEDIUM ERROR MEDIUM ERROR NO SENSE RECOVERED ERROR MEDIUM ERROR
NOTE:		<i>No Check Condition is reported for the following format changes:</i> <input type="checkbox"/> <i>from uncompressed to compressed with DCLZ</i> <input type="checkbox"/> <i>from compressed with DCLZ to uncompressed</i>	

C.2.2.3 Compression Algorithm (Byte 07) and Decompression Algorithm (Byte 11)

The Compression Algorithm byte (07) allows the host to specify the algorithm that is to be used to compress data. If the drive does not support the algorithm specified in the Compression Algorithm byte, a Check Condition is returned with the Sense Key set to Illegal Request.

Table C-3 shows the DCE bit configuration and the supported algorithms. An algorithm value of 01, which is the default, can be used to determine the supported algorithm of the drive---for Python drives, DCLZ (20). The value of 20 is returned by the corresponding MODE SENSE command.

Table C-3. Algorithms---MODE SELECT Page

DCE	Algorithm	Description
0	XX	Compression is disabled.
1	00	Compression is disabled.
1	01	Compression is enabled using the default algorithm (DCLZ).
1	02-1F	Illegal Request.
1	20	Compression is enabled using the DCLZ algorithm.
1	21-FF	Illegal Request.

C.3 MODE SENSE Page

The layout of the MODE SENSE page for data compression control is shown in Table C-4.

Table C-4. Data Compression MODE SENSE Page

BYTE	BITS							
	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	1	1
1	Page Length (0Eh)							
2	DCE	1	0	0	0	0	0	0
3	DDE	0	RED	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	Compression Algorithm							
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	Decompression Algorithm							
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0

C.3.1 Compression Control---Byte 02

The Compression Control Byte (02) of the MODE SENSE page contains the Data Compression Enable (DCE) bit (07), which is set by the host with a MODE SELECT command. The MODE SENSE command returns the value last set by the host.

- If the value is 1, the target compresses data received from the host during a WRITE command before writing the data to tape.
- If the value is 0, the target does not compress data sent during a WRITE command, and the host data is written to tape uncompressed.

The DCE bit defaults to a value of 1 if switch 6 of the DIP switch located at the rear of internal Python DDS-DC drives is set to OFF. If switch 6 is set to ON, the Python internal drive defaults to data compression disabled, and the value of the DCE bit is 0.

C.3.2 Decompression Control---Byte 03

The Decompression Control Byte (03) of the MODE SENSE page contains the Data Decompression Enable (DDE) bit (07) and Report Error on Decompression (RED) bit (05). The following subsections briefly describe the function of these two bits.

C.3.2.1 Data Decompression Enable (DDE) Bit (07)

The Data Decompression Enable (DDE) bit (07) of the Decompression Control Byte (03) of the MODE SENSE page is set by the host. The MODE SENSE command returns the value last set by the host.

- If the value is 1, the Python drive decompresses data that has been compressed on the tape before sending the data to the host during a READ command.
- If the value is 0, the Python drive does not decompress data sent to the host during a READ command, and any compressed data read is sent to the host in the entity (DDS-DC) format.

The DDE bit defaults to a 1.

C.3.2.2 Report Error on Decompression (RED) Bit (05)

The Report Error on Decompression (RED) bit (05) of the Decompression Control Byte (03) of the MODE SENSE page is set by the host. The MODE SENSE command returns the value last set by the host. The RED bit specifies when Check Conditions are reported to the host, as follows:

- If the RED bit is 0, the Python drive generates a Check Condition for every compressed entity that it cannot decompress.
- If the RED bit is 1, the Python drive generates Check Conditions only when the format of the SCSI data changes. A SCSI format change occurs when data changes from non-DCLZ compressed to uncompressed, from uncompressed to non-DCLZ compressed or from compressed with one non-DCLZ algorithm to compressed with a different non-DCLZ algorithm.

NOTE: Refer to Section C.2.2.2 for additional details.

The RED bit defaults to 0.

C.3.3 Compression Algorithm---Byte 07

The Compression Algorithm byte (07) of the MODE SENSE page specifies the algorithm used to compress data when the DCE bit is 1.

- A value of 20h in this byte specifies the DCLZ algorithm, which is the default. If the host selects a value of 01h, for the default algorithm, a value of 20h is returned by the MODE SENSE data.
- A value of 00h specifies that data compression is disabled.

C.3.4 Decompression Algorithm---Byte 11

The Decompression Algorithm byte (11) of the MODE SENSE page is specified by the Python drive to indicate the compression algorithm used on the data that was last read to the host. This byte is valid whether or not the drive decompresses the data. The byte defaults to a value of 20h to indicate DCLZ decompression.

Table C-5 shows the algorithm values and meaning.

Table C-5. Algorithms---MODE SENSE Page

Algorithm	Description
00	Data last sent to the host was uncompressed.
20	Data last sent to the host was compressed using the DCLZ algorithm.
01-1F 21-FF	Data last sent to the host was compressed using an algorithm other than the DCLZ algorithm. The contents of the DDS-DC entity header algorithm byte is returned.

C.4 Request Sense Data

During reading, several exception conditions can occur that cause the Python DDS-DC drive to send data to the host in the compressed DDS-DC entity format. The following cases cause compressed data to be sent to the host.

- Decompression is disabled and the Python drive encounters compressed data.
- The Python drive encounters data that was compressed with an algorithm other than the DCLZ algorithm.

In the above cases, the Python DDS-DC drive sends the compressed entity to the host and terminates the READ command with a Check Condition status. An additional Sense Code of 70h is used to indicate a Decompression Exception Condition. The Additional Sense Code Qualifier field is used to indicate the algorithm in the entity header that was used to compress the data.

The entity is returned as a single block. The Valid and Incorrect Length Indicator (ILI) bits are set appropriately.

Table C-6 lists the Sense Data values.

Table C-6. Sense Data Values and Descriptions

Sense Field	Sense Data	Description
Command-Specific Information	XX	The length of the entity in blocks
Additional Sense Code (Byte 12)	70h	Decompression exception
Additional Sense Qualifier (Byte 13)	00h	Uncompressed data transferred
	20h	Entity compressed using DCLZ algorithm
	XX	Entity compressed using an algorithm other than the DCLZ algorithm

If the DDE bit is not set (0) and the RED bit is set (1), a Check Condition is also returned when the format changes from compressed to uncompressed, and the Additional Sense Code Qualifier is set to 00h.

C.5 Implementation Notes

This implementation information explains how the Python DDS-DC drive operates when it is in a nondecompressing mode and compressed data is encountered. A Python DDS-DC drive may be in a nondecompressing mode for the following reasons:

- Decompression is disabled. (DDE = 0)
- The drive encounters an entity that was compressed with a non-DDS-DC algorithm. (Although the drive may be unable to decompress entities, it may be reading a tape with compressed entities.)

This implementation information deals with the following:

- How to retrieve compressed data from a drive that is unable to decompress the entities.
- Operational differences in some of the SCSI-2 commands when the drive is unable to decompress entities.

C.5.1 Retrieval of Compressed Data from Nondecompressing Modes

A tape may contain a random mix of compressed and uncompressed records. Additionally, the compressed entities are of variable and unpredictable length. A host that supports decompression must use a special strategy to retrieve the compressed data from a nondecompressing drive.

When a drive is positioned at the beginning of an entity and the host issued a READ command in fixed-block mode, a nondecompressing drive returns either the number of bytes in one SCSI block or the total number of bytes in the entity, whichever is smaller. If the host issues a READ command in variable-block mode, a nondecompressing drive returns either the number of bytes requested in the Command Descriptor Block (CDB) or the number of bytes in the entity, whichever is smaller. The drive then positions to the end of the entity.

If the RED bit is 0, or if the RED bit is 1 and this entity is the first one encountered since the last READ command was issued, the drive reports a Check Condition. The additional Sense code is set to Decompression Exception.

If the Suppress Incorrect Length Indicator (SILI) bit is not set (0), or if the SILI bit is set (1) but the number of bytes in the entity is greater than the number of bytes actually transferred, an Illegal Length condition occurs, and the ILI bit in the Sense Data is set (1).

NOTE: *If an Illegal Length condition exists, and the setting of the RED bit is such that a Decompression Exception is not reported, a Check Condition is reported. The ILI bit will be set (1) and the Sense Key will be set to NO SENSE.*

A strategy for reading compressed entities would be to set the RED bit to 1 and issue READ commands in variable-length mode with a maximum transfer size and the SILI bit set. A Check Condition would only be reported if the entity is larger than the transfer size or for the first entity encountered. The maximum transfer size allowed by SCSI is FFFFFFFF hex (16 MB). The host should allocate as much internal buffer space as possible to allow a large transfer size.

C.5.2 SCSI Command Operation for Noncompressing Modes

An entity must always be regarded as one or more contiguous blocks rather than one large variable-length block. In order to guarantee accurate tape positioning, this method of handling entities must be maintained whether or not the drive supports compression. This method of handling, however, limits drives that are unable to decompress entities. Such drives are unable to position to a point within an entity. This limitation affects the READ, SPACE Blocks, LOCATE, and SEEK commands.

For example, assume a tape contains a mix of compressed and uncompressed blocks. This tape is inserted into a drive that does not support compression, and the drive is sending the data to a host that may or may not have the ability to decompress the data. Figure C-1 illustrates this situation and the Logical Block Addresses (LBAs) on the tape.

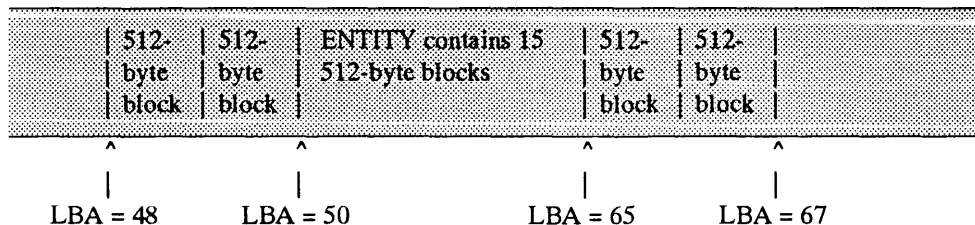


Figure C-1. LBAs of Tape in Example

C.5.2.1 READ Variable Command

If a nondecompressing drive encounters an entity when a READ Variable command is issued, the drive returns either the number of bytes specified in the CDB or the total number of bytes in the entity, whichever is smaller. The drive then positions to the end of the entity. The drive is never positioned within the entity.

Based on the setting of the RED bit, a Check Condition may be reported. The Additional Sense Code is set to Decompression Exception, the Command Specific Sense Information is set to the number of blocks in the entity, and the Residual Sense information is set to the difference between the requested number of bytes and the size of the entity in bytes.

Once the READ Variable command completes, the host can determine the tape position by issuing a READ POSITION command.

Referring to the previous example and to Figure C-1, if the tape is positioned at Logical Block Address (LBA) 50 before the READ Variable command is issued, the tape is positioned at LBA 65 after completion of the READ command. If the host supports decompression and the Residual Sense information indicates that not all compressed data was transferred, the host may then get the number of blocks in the entity from the Command Specific Sense Information. In this example, the host would then issue a SPACE Reverse 15 to be positioned at LBA 50. The host could then issue a READ Variable command with sufficient count to retrieve all of the compressed data.

If the host does not support decompression or if all was retrieved on the first READ Variable command, the host can continue from LBA 65.

C.5.2.2 READ Fixed Command

If a nondecompressing drive encounters an entity when a READ Fixed command is issued, the drive returns either the number of bytes in one block or the total number of bytes in the entity, whichever is smaller. The drive then positions to the end of the entity. The drive is never positioned within the entity.

Based on the setting of the RED bit, a Check Condition may be reported. The Additional Sense Code is set to Decompression Exception, the Command Specific Sense Information is set to the number of blocks in the entity, and the Residual Sense information is set to the difference between the requested number of blocks and the actual number of blocks transferred.

Once the READ Fixed command completes, the host can determine the tape position by issuing a READ POSITION command.

Referring to the previous example and to Figure C-1, if the tape is positioned at Logical Block Address (LBA) 50 before the READ Fixed command is issued, the tape is positioned at LBA 65 after completion of the READ command. If the host supports decompression and the Residual Sense information indicates that not all compressed data was transferred, the host may then get the number of blocks in the entity from the Command Specific Sense Information. In this example, the host would then issue a SPACE Reverse 15 to be positioned at LBA 50. The host could then issue a READ Fixed command with sufficient count to retrieve all of the compressed data.

If the host does not support decompression or if all was retrieved on the first READ Fixed command, the host can continue from LBA 65.

C.5.2.3 SPACE Blocks, LOCATE, and SEEK Commands

If a SPACE Blocks, LOCATE, or SEEK command is issued to a nondecompressing drive and the target location for the command falls within an entity, the drive reports a Check Condition and positions to the *beginning* of the entity. The Additional Sense code is set to Decompression Exception and the Command Specific Sense Information is set to the number of blocks in the entity. The host can determine the tape position by issuing a READ POSITION command.

Referring to the previous example and to Figure C-1, if the tape is positioned at beginning-of-tape (BOT) and a LOCATE to block 55 command is issued, a Check Condition is reported and the tape is positioned at the beginning of the entity---at LBA 50. The host can retrieve the number of blocks in the entity from the Command Specific Sense Information. In this example, the host can then issue a SPACE Forward 15 blocks to position past the entity.

C.6 WRITE DATA BUFFER (3Bh)

The WRITE DATA BUFFER Command is used in conjunction with the READ DATA BUFFER Command as a diagnostic function for testing the data buffer memory of the Python drive and confirming the SCSI bus integrity. The medium is not accessed during the execution of this command.

The WRITE DATA BUFFER command can also be used to download the controller firmware if the Python drive is equipped with flash EEPROM (Models 4322NP, 4542NP, or 4352NP). Only firmware supplied by Archive should be downloaded. Once the valid firmware is downloaded to the buffer, the flash EEPROM is programmed. Then, within 30 seconds, control is transferred to the new firmware and a power-on reset occurs. The drive is then ready to accept further commands.

C.6.1 WRITE DATA BUFFER Command Descriptor Block

Table C-7 shows the layout of the Command Descriptor Block.

Table C-7. Command Descriptor Block: WRITE DATA BUFFER

BYTE	BITS							
	7	6	5	4	3	2	1	0
0	0	0	1	1	1	0	1	1
1	0	0	0	0	0		MODE	
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6		MSB -- Byte Transfer Length						
7		Byte Transfer Length						
8		Byte Transfer Length -- LSB						
9	X	X	0	0	0	0	Flag	Link

C.6.2 Command Descriptor Block Field Descriptions

Table C-8 defines the fields in the Command Descriptor Block.

Table C-8. Field Descriptions: WRITE DATA BUFFER Command Descriptor Block

NAME	BYTES	BITS	DESCRIPTION
Transfer Length	6-8		<p>The Transfer Length specifies the maximum number of bytes transferred to the Python. If Mode = 0, it contains a four-byte header; thus, the data length to be stored in the buffer of the Python drive is Transfer Length - 4. If Mode = 5, the header is not used.</p> <p>A Transfer Length of zero indicates that no data are transferred. This condition is not an error. It is not an error to request a Transfer Length less than the Available Length (reported by the READ DATA BUFFER command).</p> <p>If Mode = 0, the initiator should ensure that the Transfer Length is not greater than 4 plus the Available Length that is returned in the header of the READ DATA BUFFER command. If the Transfer Length is greater than the Available Length plus 4, the drive returns a Check Condition status with a Sense Key of Illegal Request.</p>
Mode	1	0-2	<p>If Mode = 0, only the data buffer is loaded.</p> <p>If Mode = 5, the data is transferred to the controller's flash EEPROM, and the firmware is restarted. The data transferred must be a binary image of the executable firmware. The Transfer Length should = 40000h for a 256-KB flash EEPROM.</p>

C.6.3 WRITE DATA BUFFER Data Header

Table C-9 shows the layout of the Data Header for WRITE DATA BUFFER.

Table C-9. Data Header: WRITE DATA BUFFER

BYTE	BITS							
	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0

C.6.4 Completion Status

Table C-10 lists the completion status for the WRITE DATA BUFFER command.

Table C-10. Completion Status: WRITE DATA BUFFER

CODE	MESSAGE	DESCRIPTION												
00h	Good Status	<p>--The tape position is not changed. The Python drive remains in any previously set mode unless a valid firmware has been downloaded (and Mode = 5).</p> <p>If Mode = 5 and a valid firmware has been downloaded, good status is returned. The drive then accepts commands until the flash EEPROM is programmed (< 30 seconds). Then a power-on reset occurs, and commands are accepted. If the firmware file is not valid, a Check Condition is generated.</p> <p>---The Python drive is immediately ready to accept any appropriate command if Mode = 0.</p>												
02h	Check Condition	Extended Sense Byte 02h												
		<table border="1"> <thead> <tr> <th>SENSE KEY</th> <th>MESSAGE</th> <th>DESCRIPTION</th> </tr> </thead> <tbody> <tr> <td>04h</td> <td>Hardware Error</td> <td>Parity error on SCSI Bus or Python hardware failure detected.</td> </tr> <tr> <td>05h</td> <td>Illegal Request</td> <td> <p>--The CDB contains an invalid bit.</p> <p>--Transfer Length exceeds the maximum (if Mode = 0).</p> <p>---Transfer Length is not modulo 512 plus 4 (if Mode = 0).</p> <p>---The download file is invalid (if Mode = 5). The additional Sense Code and Qualifier is set to 26/02.</p> </td> </tr> <tr> <td>06h</td> <td>Unit Attention</td> <td> <p>--The cassette was changed prior to accepting this command.</p> <p>--Python drive was reset prior to accepting this command.</p> </td> </tr> </tbody> </table>	SENSE KEY	MESSAGE	DESCRIPTION	04h	Hardware Error	Parity error on SCSI Bus or Python hardware failure detected.	05h	Illegal Request	<p>--The CDB contains an invalid bit.</p> <p>--Transfer Length exceeds the maximum (if Mode = 0).</p> <p>---Transfer Length is not modulo 512 plus 4 (if Mode = 0).</p> <p>---The download file is invalid (if Mode = 5). The additional Sense Code and Qualifier is set to 26/02.</p>	06h	Unit Attention	<p>--The cassette was changed prior to accepting this command.</p> <p>--Python drive was reset prior to accepting this command.</p>
SENSE KEY	MESSAGE	DESCRIPTION												
04h	Hardware Error	Parity error on SCSI Bus or Python hardware failure detected.												
05h	Illegal Request	<p>--The CDB contains an invalid bit.</p> <p>--Transfer Length exceeds the maximum (if Mode = 0).</p> <p>---Transfer Length is not modulo 512 plus 4 (if Mode = 0).</p> <p>---The download file is invalid (if Mode = 5). The additional Sense Code and Qualifier is set to 26/02.</p>												
06h	Unit Attention	<p>--The cassette was changed prior to accepting this command.</p> <p>--Python drive was reset prior to accepting this command.</p>												

Archive Corporation
1650 Sunflower Avenue
Costa Mesa, CA 92626
(714)641-1230/FAX:(714)641-2590