

INTERNATIONAL STANDARD

ISO/IEC
9318-3

First edition
1990-12-15

Information technology — Intelligent Peripheral Interface

Part 3:

Device generic command set for magnetic and optical
disk drives

Technologies de l'information — Interface pour les périphériques intelligents —

*Partie 3: Jeu de commandes génériques appareil pour les disques magnétiques et
optiques*



Reference number
ISO/IEC 9318-3 : 1990 (E)



Documents

Document Sharing Hub

Contents

	Page
Foreword	xxvii
Introduction	xxviii
1 Scope	1
2 Normative reference	2
3 Definitions and conventions	3
3.1 Definitions	3
3.2 Conventions	5
4 Logical interface characteristics	6
4.1 Operations	6
4.1.1 Commands	6
4.1.1.1 Command types	6
4.1.1.2 Command stacking	6
4.1.1.2.1 Individual	6
4.1.1.2.2 Queued	7
4.1.1.3 Command execution order	7
4.1.2 Operation responses	8
4.1.2.1 Interrupts	8
4.1.2.2 Response types	9
4.1.2.2.1 Command completion response	9
4.1.2.2.2 Transfer Notification Response (optional)	9
4.1.2.2.3 Asynchronous response	9
4.1.2.2.4 Imbedded data response (optional)	9
4.1.2.3 Response handling	9
4.1.3 Physical interface error recovery considerations	10
4.1.3.1 Recovery from unsuccessful Slave Status octet	10
4.1.3.2 Recovery from bad parity on the Slave Status octet	10
4.2 Operation sequences	11
4.2.1 Slave procedures	11
4.2.2 Basic steps	11
4.2.2.1 Transmit Command packet to slave	11
4.2.2.2 Poll interrupts	12
4.2.2.3 Receive Response packet from slave	12
4.2.2.4 Transfer of data between master and slave	12
4.2.3 Operation sequence examples	13
4.2.3.1 Example of facility selection and individual commands	13
4.2.3.2 Example of facility selection and queued commands	13
4.2.3.3 Example of slave selection	14
4.2.3.4 Example of slave selection and slave control of Bus	14
4.3 Multiplexed data transfers (optional)	15
4.3.1 Physical interface Pause and Continue (optional)	15
4.3.2 Slave pause and master Continue	15
4.3.2.1 Implicit continue	16
4.3.2.2 Explicit continue	16
4.3.3 Slave control of Pause and Continue	16
4.3.4 Master control of Pause and Continue	16
4.3.5 Uses of multiplexing	17
4.3.5.1 One paused transfer per facility	17

© ISO/IEC 1990

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

4.3.5.2 One paused transfer per slave	17
4.3.5.3 Multiple paused transfers per slave	17
4.3.6 Anticipated pause	17
4.3.6.1 Master stays selected	17
4.3.6.2 Master deselects	18
4.3.7 Unanticipated pauses by the slave	18
4.3.8 Unanticipated pauses by the master	18
4.3.9 Multiplexed transfer mode identification	18
4.4 Data groupings	18
4.4.1 PhysicalBlocks	19
4.4.2 DataBlocks	19
4.4.3 Extents	19
4.4.4 Partitions	19
4.4.4.1 Slave-defined partitions	20
4.4.4.1.1 Data partition	20
4.4.4.1.2 Maintenance partitions	20
4.4.4.2 Master-defined partitions	20
4.4.4.2.1 Data partitions	20
4.4.4.2.2 Maintenance partitions	20
4.4.5 Alternate data areas	20
4.4.6 Physical groups	20
4.5 Media addressing definitions	21
4.5.1 Absolute addressing	21
4.5.2 Physical addressing	21
4.5.3 Logical addressing	22
4.5.4 Media defect management considerations	22
4.6 Interface addressing definitions	23
4.6.1 Actual Addressscs	23
4.6.2 Selection Addresses	23
4.6.3 Command Addresses	23
4.6.4 Facility Address	24
4.6.5 Synonym Addresses (optional)	24
4.6.6 Alias Addresses (optional)	24
4.6.7 Partition parameters	25
4.6.8 Communication addresses (optional)	25
4.6.9 Address examples	26
4.7 Slave and facility conditions	26
4.7.1 Interface conditions	27
4.7.1.1 P-Available	27
4.7.1.2 Not P-Available	27
4.7.1.3 Operational	27
4.7.1.4 Not Operational	28
4.7.1.5 P-Busy	28
4.7.1.6 Not P-Busy	28
4.7.1.7 L-Available	28
4.7.1.8 Not L-Available	28
4.7.1.9 L-Busy	28
4.7.1.10 Not L-Busy	28
4.7.2 General conditions	28
4.7.2.1 Active	28
4.7.2.2 Inactive	28
4.7.2.3 Status Pending	29
4.7.2.4 Reset	29
4.7.3 Operating status	29
4.8 Multiple ports (optional)	30
4.8.1 Slave switching	30
4.8.1.1 Physical switch	31
4.8.1.2 Logical switch	31
4.8.1.3 Mixed switch types	32
4.8.2 Facility switching	32

ISO/IEC 9318-3 : 1990 (E)

4.8.3 Slave static switching	33
4.8.3.1 Disabling a slave port	33
4.8.3.2 Enabling a slave port	33
4.8.4 Facility static switching	33
4.8.4.1 Disabling a facility	34
4.8.4.1.1 Disabling a facility at a slave port	34
4.8.4.1.2 Disabling a facility at a facility port	34
4.8.4.2 Enabling a facility	34
4.8.4.2.1 Enabling a facility at a slave port	34
4.8.4.2.2 Enabling a facility at a facility port	35
4.8.5 Slave dynamic switching	35
4.8.5.1 Neutral Mode	35
4.8.5.2 Switched Mode	35
4.8.5.3 Implicitly switched	35
4.8.5.4 Explicitly switched	36
4.8.6 Facility dynamic switch	36
4.8.6.1 Facility neutral mode	36
4.8.6.1.1 Facility neutral mode at the slave port	36
4.8.6.1.2 Facility neutral mode at the facility port	36
4.8.6.2 Facility switched mode	36
4.8.6.2.1 Facility switched mode at a slave port	37
4.8.6.2.2 Facility switched mode at a facility port	37
4.8.6.3 Implicitly switched facilities	37
4.8.6.3.1 Implicitly switched facilities at a slave port	37
4.8.6.3.2 Implicitly switched facility at a facility port	37
4.8.6.4 Explicitly switched facilities	37
4.8.6.4.1 Explicitly switched facilities at a slave port	38
4.8.6.4.2 Explicitly switched facilities at a facility port	38
4.8.7 Allegiances	38
4.8.7.1 Multiple allegiances	38
4.8.7.2 Explicit group allegiance	38
4.8.8 Alternate port notification of changes	38
4.9 Reset	38
4.9.1 External reset	38
4.9.2 Internal reset	39
4.10 Bus octets	39
4.10.1 Facility selection & request facility interrupts octets	39
4.10.2 Bus Control octet	39
4.10.3 Bus Acknowledge octet	39
4.10.4 Master Status octet	40
4.10.4.1 Bit definitions	40
4.10.4.2 Valid combinations	40
4.10.5 Slave Status octet	41
4.10.5.1 Bit definitions	41
4.10.5.2 Valid combinations	42
4.10.6 Request interrupts octet	42
4.10.7 Selective Reset Control octet	42
4.11 ATTENTION IN signal	43
4.12 Information transfers	43
4.12.1 Packet transfer conventions	43
4.12.2 Bit significance conventions	44
4.12.3 Octet significance conventions	44
4.12.4 Command and Response packet conventions	44
4.12.5 Data transfer conventions	45
5 Message packet structure	46
5.1 Conventions	46
5.1.1 General organization	46
5.1.2 Parameters	46
5.1.2.1 Parameter documentation	46

5.1.2.2	Parameter length	47
5.1.2.3	Parameter ID	47
5.1.3	Message packet representation in the document	47
5.2	Operation Command packets	48
5.2.1	Fields in Command packets	48
5.2.1.1	Packet Length	48
5.2.1.2	Command Reference Number	48
5.2.1.3	Slave Address	48
5.2.1.4	Facility Address	49
5.2.1.5	Opcode	49
5.2.1.6	Modifier octet	50
5.2.1.7	Parameters	50
5.2.2	Basic Command Message packet	50
5.2.3	Command packet parameter requirements	52
5.2.3.1	Control Command packet	52
5.2.3.2	Position Command packet	52
5.2.3.3	Transfer and Other Transfer Command packet	52
5.2.3.4	Combination Transfer Command packet (optional)	52
5.2.3.5	Diagnostic Command packet	52
5.2.4	Transferring parameters as data	52
5.2.4.1	Writing	52
5.2.4.2	Reading	52
5.2.4.3	Example	53
5.2.4.3.1	General	53
5.2.4.3.2	Specific	54
5.3	Operation Response Message packets	54
5.3.1	Fields In Response packets	54
5.3.1.1	Packet Length	54
5.3.1.2	Command Reference Number	54
5.3.1.3	Slave address	54
5.3.1.4	Facility address	54
5.3.1.5	Opcode	55
5.3.1.6	Modifier	55
5.3.1.7	Response Type	55
5.3.1.8	Major Status	55
5.3.1.9	Parameters	55
5.3.2	Basic Response packet	56
5.3.3	Response packet parameter requirements	56
5.3.3.1	Control Response packet	56
5.3.3.2	Position Response packet	56
5.3.3.3	Transfer and Other Transfer Response packet	56
5.3.3.4	Combination Transfer Response packet (optional)	56
5.3.3.5	Diagnostic Response packet	56
5.3.3.6	Asynchronous Response packet	56
5.3.3.7	Transfer Notification packet (optional)	57
5.3.3.8	Imbedded Data Response packet (optional)	57
5.4	Status	58
5.4.1	Major Status	58
5.4.2	Substatus	58
5.4.2.1	Intervention Required (ID='x4')	59
5.4.2.1.1	Not P-Available	59
5.4.2.1.2	Not Ready	59
5.4.2.1.3	Not P-Available transition	59
5.4.2.1.4	Not Ready transition	59
5.4.2.1.5	Physical Link failure	59
5.4.2.1.6	Attribute Table may be corrupted	59
5.4.2.1.7	Addressee Busy	60
5.4.2.2	Alternate Port exception (ID='x5')	60
5.4.2.2.1	Priority Reserve issued	60
5.4.2.2.2	Attributes updated	60

ISO/IEC 9318-3 : 1990 (E)

5.4.2.2.3	Initialization completed	60
5.4.2.2.4	Format completed	60
5.4.2.2.5	Facility switched to another port	60
5.4.2.2.6	Slave Diagnostic in progress	61
5.4.2.2.7	Slave Diagnostic terminated	61
5.4.2.3	Machine Exception (ID='x6')	61
5.4.2.3.1	Addressee no longer busy	61
5.4.2.3.2	P-Available transition	62
5.4.2.3.3	Ready transition	62
5.4.2.3.4	Operation timeout	62
5.4.2.3.5	Physical Interface check	62
5.4.2.3.6	Slave-Initiated reset	62
5.4.2.3.7	Environmental error	62
5.4.2.3.8	Power fail alert	62
5.4.2.3.9	Data Check (on raw data)	62
5.4.2.3.10	Uncorrectable Data Check (on perfect data)	62
5.4.2.3.11	Fatal error	62
5.4.2.3.12	Hardware write protected	63
5.4.2.3.13	Queue full	63
5.4.2.3.14	Command failure	63
5.4.2.3.15	Read access violation	63
5.4.2.3.16	Write access violation	63
5.4.2.3.17	Data overrun	63
5.4.2.3.18	Reallocation space exhausted	63
5.4.2.3.19	End of media detected	63
5.4.2.3.20	End of extent detected	63
5.4.2.3.21	Unexpected master action	63
5.4.2.3.22	Error Log full	64
5.4.2.3.23	Defect Directory full	64
5.4.2.3.24	Logical link failure	64
5.4.2.3.25	Position lost	64
5.4.2.4	Command Exception (ID='x7')	64
5.4.2.4.1	Invalid Packet length	65
5.4.2.4.2	Invalid Command Reference Number	65
5.4.2.4.3	Invalid Slave Address	65
5.4.2.4.4	Invalid Facility Address	65
5.4.2.4.5	Invalid selection address	65
5.4.2.4.6	Invalid Opcode	65
5.4.2.4.7	Invalid Modifier	65
5.4.2.4.8	Invalid Extent	65
5.4.2.4.9	Out of context	65
5.4.2.4.10	Invalid parameter(s)	65
5.4.2.4.11	Missing parameter(s)	66
5.4.2.4.12	Reserved value not equal to zero	66
5.4.2.4.13	Invalid combination	66
5.4.2.4.14	Not at initial position	66
5.4.2.5	Command Aborted (ID='x8')	66
5.4.2.5.1	Command Aborted	66
5.4.2.5.2	Command sequence terminated	66
5.4.2.5.3	Unexecuted command from terminated sequence	67
5.4.2.5.4	Command chain terminated	67
5.4.2.5.5	Unexecuted command from terminated chain	67
5.4.2.5.6	Command order terminated	67
5.4.2.5.7	Unexecuted command from terminated order	67
5.4.2.6	Conditional Success (ID='x9')	68
5.4.2.6.1	Logging data appended	68
5.4.2.6.2	ABORT received: No Command Active	69
5.4.2.6.3	ABORT received: Status Pending	69
5.4.2.6.4	ABORT received: Not Operational	69
5.4.2.6.5	Anticipated error	69

5.4.2.6.6	Anticipated data error	69
5.4.2.6.7	Reallocation required	69
5.4.2.6.8	Reallocation discontinuity	69
5.4.2.6.9	Defect Directory threshold exceeded	69
5.4.2.6.10	Error retry performed	69
5.4.2.6.11	Data retry performed	69
5.4.2.6.12	Motion retry performed	70
5.4.2.6.13	Data correction performed	70
5.4.2.6.14	Soft error	70
5.4.2.6.15	Release of an unreserved addressee	70
5.4.2.6.16	Request Diagnostic Control command	70
5.4.2.6.17	Error Log request	70
5.4.2.6.18	Non-Interchange volume	70
5.4.2.6.19	Retention required	70
5.4.2.6.20	End of Media Warning (EMW)	70
5.4.2.6.21	Statistics update requested	71
5.4.2.6.22	Parameter update requested	71
5.4.2.6.23	Asynchronous event occurrence	71
5.4.2.6.24	Master-Terminated transfer	71
5.4.2.7	Incomplete (ID='xA')	71
5.4.2.7.1	Command may be resumed	71
5.4.2.7.2	COPY source space empty	72
5.4.2.7.3	Response packet truncated	72
5.4.2.7.4	Select Subservient slave	72
5.4.2.7.5	Connect unsuccessful	72
5.4.2.7.6	Disconnect unsuccessful	72
5.4.2.7.7	Connect Identifier already assigned	72
5.4.2.7.8	Link not connected	72
5.4.2.7.9	Beginning of Media (BOM) detected	72
5.4.2.7.10	End of Media Warning (EMW)	72
5.4.2.7.11	End of extent detected	72
5.4.2.7.12	Block length difference	73
5.4.2.7.13	Unrecorded media	73
5.4.2.7.14	Data length difference	73
5.4.2.7.15	Block not found	73
5.4.2.8	Successful	73
5.4.2.9	Message/Microcode Exception (ID='x3')	73
5.4.2.9.1	Microcode Data not accepted	74
5.4.2.9.2	Request Master to IML Slave	74
5.4.2.9.3	Slave Unable to IML	74
5.4.2.9.4	Message	74
5.4.2.9.5	Microcode Execution error	74
5.4.2.9.6	Failure message	74
5.4.2.9.7	Port Disable pending	74
5.4.2.9.8	Port Response	74
5.4.2.9.9	Facility status	74
5.4.3	Extended substatus	75
5.5	Common parameters	75
5.5.1	Transfer Notification parameter (optional)	75
5.5.2	Command Extent parameter	76
5.5.2.1	Count	76
5.5.2.2	Data Address	76
5.5.3	Response Extent parameter	76
5.5.3.1	Residual Count	77
5.5.3.2	Data Address	77
5.5.4	Combination Command Extent parameter (optional)	77
5.5.4.1	Slave address	78
5.5.4.2	Facility address	78
5.5.4.3	Modifiers	78
5.5.4.4	Count	78

ISO/IEC 9318-3 : 1990 (E)

5.5.4.5 Data Address	78
5.5.5 Combination Response Extent parameter (optional)	79
5.5.5.1 Slave address	79
5.5.5.2 Facility address	79
5.5.5.3 Modifiers	79
5.5.5.4 Residual Count	79
5.5.5.5 Data Address	79
5.5.5.6 Major Status	80
5.5.5.7 Substatus	80
5.5.6 Access Key parameter (optional)	80
5.5.7 Reserved	80
5.5.8 Reserved	80
5.5.9 Invalid Parm parameter (optional)	80
5.5.9.1 Displacement of parameter in error	80
5.5.9.2 Displacement of field in error	81
5.5.10 Missing Parm parameter (optional)	81
5.5.11 Data Address parameter (optional)	81
5.5.12 Block Size parameter (optional)	82
5.5.13 Transfer	82
5.5.13.1 Verify	82
5.5.13.2 Volume	83
5.5.13.3 Certify	83
5.5.13.4 Stop on Data Error	83
5.5.13.5 Compare	83
5.5.13.6 Threshold	83
5.5.13.7 Suppress Incorrect Length Indication	84
5.5.13.8 Response conditions (octet 3 bits 7-4)	84
5.5.14 Encapsulation parameter (optional)	84
5.5.15 Partition parameter (optional)	84
5.5.15.1 Disk Partitions	85
5.5.15.2 Tape Partitions	85
5.5.16 Stop On Discontinuity parameter (optional)	87
5.5.16.1 Cylinders	87
5.5.16.2 Tracks	87
5.5.16.3 Access boundary	87
5.5.16.4 Discontiguous Defect reallocation	87
5.5.16.5 Bands	87
5.5.16.6 Time	87
5.5.16.7 Discontinuity Time	87
5.5.17 Imbedded Data parameter (optional)	88
6 Control commands	89
6.1 NOP	89
6.1.1 Command packet	89
6.1.2 Response packet	89
6.1.3 Description	89
6.2 FACILITY OPERATION	90
6.2.1 Command packet	90
6.2.2 Response packet	90
6.2.3 Description	90
6.2.4 Parameter 3D - Encapsulation parameter	90
6.3 ATTRIBUTES	91
6.3.1 Command packet	91
6.3.2 Response packet	91
6.3.3 Description	91
6.3.4 Parameters	93
6.3.4.1 Parameters 3A, 3E, 50	93
6.3.4.1.1 Data Address (common) parameter	94
6.3.4.1.2 Partition (common) parameter	94
6.3.4.1.3 Vendor ID parameter	94

6.3.4.2 Parameters 51-58	95
6.3.4.2.1 Size of Disk DataBlocks parameter	95
6.3.4.2.2 Size of Disk PhysicalBlocks parameter	95
6.3.4.2.3 Total Number of Disk DataBlocks parameter	96
6.3.4.2.4 Total Number of Disk PhysicalBlocks parameter	96
6.3.4.2.5 DataBlock Sizes Supported parameter	96
6.3.4.2.6 PhysicalBlock Sizes Supported parameter	96
6.3.4.2.7 Size of Physical Groups parameter	96
6.3.4.2.8 Hard Disk Formats parameter	96
6.3.4.3 Parameters 59-5A	97
6.3.4.3.1 Attributes Table conditions parameter	97
6.3.4.3.2 Pad with Fill Characters parameter	97
6.3.4.4 Parameters 5B-5D	98
6.3.4.4.1 Disk Partition Definition parameter	98
6.3.4.4.2 Synonym Definition parameter	98
6.3.4.4.3 Alias Definition parameter	98
6.3.4.5 Parameters 5E-5F	99
6.3.4.5.1 Multi-Port Characteristics parameter	99
6.3.4.5.2 Physical Disk Configuration parameter	99
6.3.4.6 Parameters 60-63	100
6.3.4.6.1 DataBlock Interleave parameter	100
6.3.4.6.2 Transfer Rate parameter	101
6.3.4.6.3 PhysicalBlock Performance Characteristics Supported parameter	101
6.3.4.6.4 Current PhysicalBlock Performance Settings parameter	101
6.3.4.7 Parameters 64-65	102
6.3.4.7.1 Physical Interface Attributes parameter	102
6.3.4.7.2 Addressee Configuration parameter	102
6.3.4.8 Parameter 66 - Slave Configuration (bit significant)	103
6.3.4.9 Parameter 67 - Slave Configuration (fields)	105
6.3.4.10 Parameter 68 - Facilities Attached to Slave	106
6.3.4.11 Parameters 69-6A	108
6.3.4.11.1 Parameter 69	108
6.3.4.11.2 Command Supported parameter	108
6.3.4.12 Parameter 6B - Masks of octets supported	109
6.3.4.13 Parameters 6C-6D	110
6.3.4.13.1 Request Parm parameter	110
6.3.4.13.2 Parm Length parameter	111
6.3.4.14 Parameter 6E - Slave Reconfiguration (bit-significant)	111
6.3.4.15 Parameter 6F -	113
6.4 REPORT ADDRESSEE STATUS	114
6.4.1 Command packet	114
6.4.2 Response packet	114
6.4.3 Description	114
6.4.4 Parameters 50-53	115
6.4.4.1 Port Mask parameter	115
6.4.4.2 Condition parameter	116
6.4.4.3 Media Status parameter	116
6.4.4.4 Vendor Unique Status parameter	116
6.5 PORT ADDRESS	116
6.5.1 Command packet	116
6.5.2 Response packet	116
6.5.3 Description	116
6.6 PATH CONTROL	118
6.6.1 Command packet	118
6.6.2 Response packet	118
6.6.3 Description	118
6.6.4 Parameters 50-51	119
6.6.4.1 Port Mask parameter	119
6.6.4.2 Path Control Mask parameter	119
6.7 ATTENTION CONTROL	119

ISO/IEC 9318-3 : 1990 (E)

6.7.1 Command packet	119
6.7.2 Response packet	120
6.7.3 Description	120
6.7.4 Interrupts Mask parameter	120
6.8 OPERATING MODE	121
6.8.1 Command packet	121
6.8.2 Response packet	121
6.8.3 Description	121
6.8.4 Parameters 3E, 50-51	122
6.8.4.1 Partition (common) parameter	122
6.8.4.2 Response Conditions parameter	122
6.8.4.3 Disk Modes parameter	122
6.9 ABORT	123
6.9.1 Command packet	123
6.9.2 Response packet	123
6.9.3 Description	123
6.9.4 Parameters 50-54	124
6.9.4.1 Command Reference Number parameter	124
6.9.4.2 Alternate Port Commands parameter	124
6.9.4.3 Facility Address parameter	125
6.9.4.4 Alternate Port parameter	125
6.9.4.5 Facility Reset parameter	125
6.10 ACCESS PERMITS	125
6.10.1 Command packet	125
6.10.2 Response packet	126
6.10.3 Description	126
6.10.4 Parameters 31-32, 35, 3A, 3E, 50-51	127
6.10.4.1 Command Extent (common) parameter	127
6.10.4.2 Response Extent (common) parameter	128
6.10.4.3 Access Key (common) parameter	128
6.10.4.4 Data Address (common) parameter	128
6.10.4.5 Partition (common) parameter	128
6.10.4.6 Port Mask parameter	128
6.10.4.7 Access Protection parameter - protection modifiers	128
6.11 RESUME	129
6.11.1 Command packet	129
6.11.2 Response packet	129
6.11.3 Description	129
6.11.4 Parameters 50-51	130
6.11.4.1 Command Reference Number parameter	130
6.11.4.2 Alternate Port Commands parameter	130
6.12 PORT RESPONSE	130
6.12.1 Command packet	130
6.12.2 Response packet	131
6.12.3 Description	131
6.12.4 Parameters 50-51	131
6.12.4.1 Port Mask parameter	131
6.12.4.2 Response Information Transfer parameter	131
6.13 ANTICIPATED ACTION	131
6.13.1 Command packet	131
6.13.2 Response packet	132
6.13.3 Description	132
6.13.4 Parameters 3D, 50	132
6.13.4.1 Expected Conditions parameter	132
6.13.4.2 Encapsulation parameter	132
6.14 OPERATOR DISPLAY	133
6.14.1 Command packet	133
6.14.2 Response packet	133
6.14.3 Description	133
6.14.4 Parameters 50-53	134

6.14.4.1 Identifier parameter	135
6.14.4.2 Length parameter	135
6.14.4.3 Mode parameter	135
6.14.4.4 Timing parameter	136
6.14.4.5 Normal message parameter	137
6.14.4.6 Alternate message parameter	137
7 Position commands	138
7.1 Reserved	138
7.2 POSITION CONTROL	138
7.2.1 Command packet	138
7.2.2 Response packet	138
7.2.3 Description	138
7.2.4 Parameters 31, 32, 35, 3A, 3E	139
7.2.4.1 Command Extent (common) parameter	139
7.2.4.2 Response Extent (common) parameter	139
7.2.4.3 Access Key (common) parameter	139
7.2.4.4 Data Address (common) parameter	139
7.2.4.5 Partition (common) parameter	139
7.3 REPORT POSITION	140
7.3.1 Command packet	140
7.3.2 Response packet	140
7.3.3 Description	140
7.3.4 Parameters 32, 35, 3A, 3E	140
7.3.4.1 Response Extent (common) parameter	140
7.3.4.2 Access Key (common) parameter	141
7.3.4.3 Data Address (common) parameter	141
7.3.4.4 Partition (common) parameter	141
7.4 Reserved	141
7.5 REPORT DISCONTINUITY	141
7.5.1 Command packet	141
7.5.2 Response packet	141
7.5.3 Description	141
7.5.4 Parameters 31-32, 35, 3A, 3E-3F, 6C-6D	142
7.5.4.1 Command Extent (common) parameter	142
7.5.4.2 Response Extent (common) parameter	142
7.5.4.3 Access Key (common) parameter	143
7.5.4.4 Data Address (common) parameter	143
7.5.4.5 Partition (common) parameter	143
7.5.4.6 Stop on Discontinuity (common) parameter	143
7.5.4.7 Request Parm parameter	143
7.5.4.8 Parm Length parameter	143
8 Transfer commands	144
8.1 READ	144
8.1.1 Command packet	144
8.1.2 Response packet	144
8.1.3 Description	144
8.1.4 Parameters	146
8.1.4.1 Parameters 31-32, 35, 3A, 3C, 3E-3F	146
8.1.4.1.1 Command Extent (common) parameter	146
8.1.4.1.2 Response Extent (common) parameter	146
8.1.4.1.3 Access Key (common) parameter	146
8.1.4.1.4 Data Address (common) parameter	146
8.1.4.1.5 Transfer (common) parameter	146
8.1.4.1.6 Partition (common) parameter	147
8.1.4.1.7 Stop on Discontinuity (common) parameter	147
8.1.4.2 Parameters 50-53	147
8.1.4.2.1 Skip Mask parameter	147
8.1.4.2.2 Information Transfer Size Override parameter	147

ISO/IEC 9318-3 : 1990 (E)

8.1.4.2.3 Master Termination Permitted parameter	148
8.1.4.2.4 Boundary Gather parameter	148
8.2 READ RAW DATA	148
8.2.1 Command packet	148
8.2.2 Response packet	148
8.2.3 Description	148
8.2.4 Parameters 31-32, 35, 3A, 3C, 3E-3F	149
8.2.4.1 Command Extent (common) parameter	149
8.2.4.2 Response Extent (common) parameter	149
8.2.4.3 Access Key (common) parameter	149
8.2.4.4 Data Address (common) parameter	149
8.2.4.5 Transfer (common) parameter	149
8.2.4.6 Partition (common) parameter	149
8.2.4.7 Stop on Discontinuity (common) parameter	150
8.3 READ REPLICATED DATA	150
8.3.1 Command packet	150
8.3.2 Response packet	150
8.3.3 Description	150
8.3.4 Parameters 31-32, 35, 3A, 3C, 3E, 50	151
8.3.4.1 Command Extent (common) parameter	151
8.3.4.2 Response Extent (common) parameter	151
8.3.4.3 Access Key (common) parameter	151
8.3.4.4 Data Address (common) parameter	151
8.3.4.5 Transfer (common) parameter	151
8.3.4.6 Partition (common) parameter	151
8.3.4.7 Range Count parameter	151
8.4 SEARCH	152
8.4.1 Command packet	152
8.4.2 Response packet	152
8.4.3 Description	152
8.4.4 Parameters	153
8.4.4.1 Parameters 02, 31-32, 35, 3A, 3E	153
8.4.4.1.1 Continuation of Preceding (common) parameter	153
8.4.4.1.2 Command Extent (common) parameter	153
8.4.4.1.3 Response Extent (common) parameter	153
8.4.4.1.4 Access Key (common) parameter	154
8.4.4.1.5 Data Address (common) parameter	154
8.4.4.1.6 Partition (common) parameter	154
8.4.4.2 Parameter 50 - Set String Search parameter	154
8.4.4.3 Parameter 51 - Boolean Operator parameter	155
8.4.4.4 Parameter 52 - Set Multiple Key Search parameter	155
8.4.4.5 Parameter 53 - Report Search parameter	156
8.5 WRITE	157
8.5.1 Command packet	157
8.5.2 Response packet	157
8.5.3 Description	157
8.5.4 Parameters 31-32, 35, 3A, 3C, 3E-3F, 50-52	158
8.5.4.1 Command Extent (common) parameter	158
8.5.4.2 Response Extent (common) parameter	158
8.5.4.3 Access Key (common) parameter	158
8.5.4.4 Data Address (common) parameter	158
8.5.4.5 Transfer (common) parameter	159
8.5.4.6 Partition (common) parameter	159
8.5.4.7 Stop on Discontinuity (common) parameter	159
8.5.4.8 Skip Mask parameter	159
8.5.4.9 Information Transfer Size Override parameter	159
8.5.4.10 Master Termination Permitted parameter	159
8.6 WRITE PATTERN	159
8.6.1 Command packet	159
8.6.2 Response packet	159

8.6.3	Description	160
8.6.4	Parameters 02, 31-32, 35, 3A, 3C, 3E, 50, 6C	160
8.6.4.1	Continuation of Preceding (common) parameter	160
8.6.4.2	Command Extent (common) parameter	160
8.6.4.3	Response Extent (common) parameter	160
8.6.4.4	Access Key (common) parameter	160
8.6.4.5	Data Address (common) parameter	161
8.6.4.6	Transfer (common) parameter	161
8.6.4.7	Partition (common) parameter	161
8.6.4.8	Pattern parameter	161
8.6.4.9	Request Parm parameter	161
8.7	FORMAT	161
8.7.1	Command packet	161
8.7.2	Response packet	161
8.7.3	Description	162
8.7.3.1	Interleave considerations	163
8.7.3.2	Defect List considerations	163
8.7.4	Parameters	164
8.7.4.1	Parameters 31-32, 35, 3A-3C, 3E-3F	164
8.7.4.1.1	Command Extent (common) parameter	164
8.7.4.1.2	Response Extent (common) parameter	164
8.7.4.1.3	Access Key (common) parameter	165
8.7.4.1.4	Data Address (common) parameter	165
8.7.4.1.5	Block Size (common) parameter	165
8.7.4.1.6	Transfer (common) parameter	165
8.7.4.1.7	Partition (common) parameter	165
8.7.4.1.8	Stop On Discontinuity (common) parameter	165
8.7.4.2	Parameters 50-54	166
8.7.4.2.1	Number of PhysicalBlocks per Track parameter	166
8.7.4.2.2	PhysicalBlock Interleave Factors parameter	166
8.7.4.2.3	PhysicalBlock Interleave Table parameter	167
8.7.4.2.4	Transfer Rate parameter	167
8.7.4.2.5	DataBlock Interleave parameter	167
8.7.4.3	Parameters 56-58	168
8.7.4.3.1	Track Defects List parameter	168
8.7.4.3.2	Sector Defects List parameter	168
8.7.4.3.3	Hard Disk Formats parameter	168
8.7.4.4	Parameters 5A, 6C	169
8.7.4.4.1	Cell Defects List parameter	169
8.7.4.4.2	Request Parm parameter	169
9	Combination commands	170
9.1	COPY	170
9.1.1	Command packet	170
9.1.2	Response packet	170
9.1.3	Description	170
9.1.4	Parameters 33-35, 3A, 3C, 3E, 50	172
9.1.4.1	Combination Command Extent (common) parameter	172
9.1.4.2	Combination Response Extent (common) parameter	172
9.1.4.3	Access Key (common) parameter	172
9.1.4.4	Data Address (common) parameter	173
9.1.4.5	Transfer (common) parameter	173
9.1.4.6	Partition (common) parameter	173
9.1.4.7	Extended Modifiers parameter	173
9.2	COMPARE SLAVE DATA	174
9.2.1	Command packet	174
9.2.2	Response packet	174
9.2.3	Description	174
9.2.4	Parameters 33-35, 3A, 3E, 50	175

9.2.4.1	Combination Command Extent (common) parameter	175
9.2.4.2	Combination Response Extent (common) parameter	175
9.2.4.3	Access Key (common) parameter	175
9.2.4.4	Data Address (common) parameter	175
9.2.4.5	Partition (common) parameter	176
9.2.4.6	Extended Modifiers parameter	176
9.3	COMPARE DATA	176
9.3.1	Command packet	176
9.3.2	Response packet	176
9.3.3	Description	176
9.3.4	Parameters 31-35, 3A, 3E, 50	177
9.3.4.1	Command Extent (common) parameter	177
9.3.4.2	Response Extent (common) parameter	177
9.3.4.3	Combination Command Extent (common) parameter	177
9.3.4.4	Combination Response Extent (common) parameter	178
9.3.4.5	Access Key (common) parameter	178
9.3.4.6	Data Address (common) parameter	178
9.3.4.7	Partition (common) parameter	178
9.3.4.8	Extended Modifiers parameter	178
9.4	REALLOCATE	178
9.4.1	Command packet	178
9.4.2	Response packet	178
9.4.3	Description	178
9.4.4	Parameters	179
9.4.4.1	Parameters 31-32, 35, 3A, 3E	179
9.4.4.1.1	Command Extent (common) parameter	180
9.4.4.1.2	Response Extent (common) parameter	180
9.4.4.1.3	Access Key (common) parameter	180
9.4.4.1.4	Data Address (common) parameter	180
9.4.4.1.5	Partition (common) parameter	180
9.4.4.2	Parameters 50, 55	180
9.4.4.2.1	Defect parameter	180
9.4.4.2.2	Defective DataBlock parameter	181
9.4.4.2.3	Defective lists management	181
9.5	ALLOCATE RESTORE	181
9.5.1	Command packet	181
9.5.2	Response packet	181
9.5.3	Description	182
9.5.4	Parameters	182
9.5.4.1	Parameters 31-32, 35, 3A, 3E	182
9.5.4.1.1	Command Extent (common) parameter	182
9.5.4.1.2	Response Extent (common) parameter	182
9.5.4.1.3	Access Key (common) parameter	183
9.5.4.1.4	Data Address (common) parameter	183
9.5.4.1.5	Partition (common) parameter	183
9.5.4.2	Parameters 50, 55	183
9.5.4.2.1	Defect parameter	183
9.5.4.2.2	Defective DataBlock parameter	183
9.5.4.2.3	Defective lists management	183
9.6	SHADOW READ	184
9.6.1	Command packet	184
9.6.2	Response packet	184
9.6.3	Description	184
9.6.4	Parameters 33-35, 3A, 3C, 3E, 51, 52	185
9.6.4.1	Combination Command Extent (common) parameter	185
9.6.4.2	Combination Response Extent (common) parameter	185
9.6.4.3	Access Key (common) parameter	185
9.6.4.4	Data Address (common) parameter	185
9.6.4.5	Transfer (common) parameter	186
9.6.4.6	Partition (common) parameter	186

9.6.4.7 Information Transfer Size Override parameter	186
9.6.4.8 Master Termination Permitted parameter	186
9.7 SHADOW WRITE	186
9.7.1 Command packet	186
9.7.2 Response packet	186
9.7.3 Description	186
9.7.4 Parameters 33-35, 3A, 3C, 3E, 51, 52	187
9.7.4.1 Combination Command Extent (common) parameter	187
9.7.4.2 Combination Response Extent (common) parameter	187
9.7.4.3 Access Key (common) parameter	188
9.7.4.4 Data Address (common) parameter	188
9.7.4.5 Transfer (common) parameter	188
9.7.4.6 Partition (common) parameter	188
9.7.4.7 Information Transfer Size Override parameter	188
9.7.4.8 Master Termination Permitted parameter	188
9.8 SHADOW RESTORE	188
9.8.1 Command packet	188
9.8.2 Response packet	188
9.8.3 Description	189
9.8.4 Parameters 33-35, 3A, 3C, 3E	189
9.8.4.1 Combination Command Extent (common) parameter	189
9.8.4.2 Combination Response Extent (common) parameter	189
9.8.4.3 Access Key (common) parameter	190
9.8.4.4 Data Address (common) parameter	190
9.8.4.5 Transfer (common) parameter	190
9.8.4.6 Partition (common) parameter	190
10 Other Transfer commands	191
10.1 READ VERIFY	191
10.1.1 Command packet	191
10.1.2 Response packet	191
10.1.3 Description	191
10.1.3.1 High Margins modifier	192
10.1.3.2 Volume modifier	192
10.1.4 Parameters 31-32, 35, 3A, 3C, 3E	192
10.1.4.1 Command Extent (common) parameter	192
10.1.4.2 Response Extent (common) parameter	192
10.1.4.3 Access Key (common) parameter	192
10.1.4.4 Data Address (common) parameter	192
10.1.4.5 Transfer (common) parameter	192
10.1.4.6 Partition (common) parameter	193
10.2 READ AT FIRST AVAILABLE DATA	193
10.2.1 Command packet	193
10.2.2 Response packet	193
10.2.3 Description	193
10.2.4 Parameters 31-32, 35, 3A, 3C, 3E-3F, 50-51	194
10.2.4.1 Command Extent (common) parameter	194
10.2.4.2 Response Extent (common) parameter	194
10.2.4.3 Access Key (common) parameter	194
10.2.4.4 Data Address (common) parameter	194
10.2.4.5 Transfer (common) parameter	195
10.2.4.6 Partition (common) parameter	195
10.2.4.7 Stop On Discontinuity parameter	195
10.2.4.8 Read at First Data parameter	195
10.2.4.9 Information Transfer Size Override parameter	195
10.3 READ FROM BUFFER	195
10.3.1 Command packet	195
10.3.2 Response packet	195
10.3.3 Description	196
10.3.4 Parameters 31-32, 35, 3A, 3C, 3E, 50	196

ISO/IEC 9318-3 : 1990 (E)

10.3.4.1	Command Extent (common) parameter	197
10.3.4.2	Response Extent (common) parameter	197
10.3.4.3	Access Key (common) parameter	197
10.3.4.4	Data Address (common) parameter	197
10.3.4.5	Transfer (common) parameter	197
10.3.4.6	Partition (common) parameter	197
10.3.4.7	Buffer Address parameter	197
10.4	READ FACILITY DATA TO BUFFER	198
10.4.1	Command packet	198
10.4.2	Response packet	198
10.4.3	Description	198
10.4.4	Parameters 31-32, 35, 3A, 3C, 3E, 50	199
10.4.4.1	Command Extent (common) parameter	199
10.4.4.2	Response Extent (common) parameter	199
10.4.4.3	Access Key (common) parameter	199
10.4.4.4	Data Address (common) parameter	199
10.4.4.5	Transfer (common) parameter	199
10.4.4.6	Buffer Address parameter	199
10.5	READ PHYSICAL DATA AND ECC	200
10.5.1	Command packet	200
10.5.2	Response packet	200
10.5.3	Description	200
10.5.4	Parameters 31-32, 35, 3A, 3C, 3E	201
10.5.4.1	Command Extent (common) parameter	201
10.5.4.2	Response Extent (common) parameter	201
10.5.4.3	Access Key (common) parameter	201
10.5.4.4	Data Address (common) parameter	201
10.5.4.5	Transfer (common) parameter	201
10.5.4.6	Partition (common) parameter	201
10.6	READ PHYSICAL HEADER	202
10.6.1	Command packet	202
10.6.2	Response packet	202
10.6.3	Description	202
10.6.4	Parameters 31-32, 35, 3A, 3E	202
10.6.4.1	Command Extent (common) parameter	203
10.6.4.2	Response Extent (common) parameter	203
10.6.4.3	Access Key (common) parameter	203
10.6.4.4	Data Address (common) parameter	203
10.6.4.5	Partition (common) parameter	203
10.7	READ IPL	203
10.7.1	Command packet	203
10.7.2	Response packet	203
10.7.3	Description	203
10.7.4	Parameters 31-32, 35, 3A, 3E	204
10.7.4.1	Command Extent (common) parameter	204
10.7.4.2	Response Extent (common) parameter	204
10.7.4.3	Access Key (common) parameter	204
10.7.4.4	Data Address (common) parameter	204
10.7.4.5	Partition (common) parameter	204
10.8	READ PHYSICAL HEADER AND ECC	205
10.8.1	Command packet	205
10.8.2	Response packet	205
10.8.3	Description	205
10.8.4	Parameters 31-32, 35, 3A, 3E	206
10.8.4.1	Command Extent (common) parameter	206
10.8.4.2	Response Extent (common) parameter	206
10.8.4.3	Access Key (common) parameter	206
10.8.4.4	Data Address (common) parameter	206
10.8.4.5	Partition (common) parameter	206
10.9	WRITE TO BUFFER	207

10.9.1	Command packet	207
10.9.2	Response packet	207
10.9.3	Description	207
10.9.4	Parameters 31-32, 35, 3A, 3C, 3E, 50	208
10.9.4.1	Command Extent (common) parameter	208
10.9.4.2	Response Extent (common) parameter	208
10.9.4.3	Access Key (common) parameter	208
10.9.4.4	Data Address (common) parameter	208
10.9.4.5	Transfer (common) parameter	208
10.9.4.6	Partition (common) parameter	209
10.9.4.7	Buffer Address parameter	209
10.10	WRITE BUFFER TO FACILITY	209
10.10.1	Command packet	209
10.10.2	Response packet	209
10.10.3	Description	209
10.10.4	Parameters 31-32, 35, 3A, 3C, 3E, 50	210
10.10.4.1	Command Extent (common) parameter	210
10.10.4.2	Response Extent (common) parameter	210
10.10.4.3	Access Key (common) parameter	210
10.10.4.4	Data Address (common) parameter	210
10.10.4.5	Transfer (common) parameter	210
10.10.4.6	Partition (common) parameter	210
10.10.4.7	Buffer Address parameter	211
10.11	WRITE PHYSICAL DATA AND ECC	211
10.11.1	Command packet	211
10.11.2	Response packet	211
10.11.3	Description	211
10.11.4	Parameters 31-32, 35, 3A, 3E	212
10.11.4.1	Command Extent (common) parameter	212
10.11.4.2	Response Extent (common) parameter	212
10.11.4.3	Access Key (common) parameter	212
10.11.4.4	Data Address (common) parameter	212
10.11.4.5	Partition (common) parameter	212
10.12	WRITE PHYSICAL HEADER	213
10.12.1	Command packet	213
10.12.2	Response packet	213
10.12.3	Description	213
10.12.4	Parameters 31-32, 35, 3A, 3E	214
10.12.4.1	Command Extent (common) parameter	214
10.12.4.2	Response Extent (common) parameter	214
10.12.4.3	Access Key (common) parameter	214
10.12.4.4	Data Address (common) parameter	214
10.12.4.5	Partition (common) parameter	214
10.13	LOAD SLAVE IML	215
10.13.1	Command packet	215
10.13.2	Response packet	215
10.13.3	Description	215
10.13.4	Parameters 31-32, 3E	216
10.13.4.1	Command Extent (common) parameter	216
10.13.4.2	Response Extent (common) parameter	216
10.13.4.3	Partition (common) parameter	216
10.14	ERASE	217
10.14.1	Command packet	217
10.14.2	Response packet	217
10.14.3	Description	217
10.14.4	Parameters 31-32, 35, 3A, 3E	218
10.14.4.1	Command Extent (common) parameter	218
10.14.4.2	Response Extent (common) parameter	218
10.14.4.3	Access Key (common) parameter	218
10.14.4.4	Data Address (common) parameter	218

ISO/IEC 9318-3 : 1990 (E)

10.14.4.5 Partition (common) parameter	218
10.15 WRITE PHYSICAL HEADER AND ECC	219
10.15.1 Command packet	219
10.15.2 Response packet	219
10.15.3 Description	219
10.15.4 Parameters 31-32, 35, 3A, 3E	220
10.15.4.1 Command Extent (common) parameter	220
10.15.4.2 Response Extent (common) parameter	220
10.15.4.3 Access Key (common) parameter	220
10.15.4.4 Data Address (common) parameter	220
10.15.4.5 Partition (common) parameter	220
11 Diagnostic commands	221
11.1 PERFORM SLAVE DIAGNOSTICS	221
11.1.1 Command packet	221
11.1.2 Response packet	221
11.1.3 Description	221
11.1.4 Parameters 31-32, 50-53, 6C	222
11.1.4.1 Command Extent (common) parameter	222
11.1.4.2 Response Extent (common) parameter	222
11.1.4.3 Diagnostic Number parameter	222
11.1.4.4 Diagnostic Function List parameter	222
11.1.4.5 Diagnostic Action Code parameter	222
11.1.4.6 Vendor Unique parameter	223
11.1.4.7 Request Parm parameter	223
11.2 PERFORM FACILITY DIAGNOSTICS	223
11.2.1 Command packet	223
11.2.2 Response packet	223
11.2.3 Description	223
11.2.4 Parameters 31-32, 50-53, 6C	224
11.2.4.1 Command Extent (common) parameter	224
11.2.4.2 Response Extent (common) parameter	224
11.2.4.3 Diagnostic Number parameter	224
11.2.4.4 Diagnostic Function List parameter	225
11.2.4.5 Diagnostic Action Code parameter	225
11.2.4.6 Vendor Unique parameter	225
11.2.4.7 Request Parm parameter	225
11.3 READ DEFECT LIST	225
11.3.1 Command packet	225
11.3.2 Response packet	225
11.3.3 Description	225
11.3.4 Parameters	226
11.3.4.1 Parameters 31-32, 3E, 55-57	226
11.3.4.1.1 Command Extent (common) parameter	227
11.3.4.1.2 Response Extent (common) parameter	227
11.3.4.1.3 Partition (common) parameter	227
11.3.4.1.4 Defective DataBlock parameter	227
11.3.4.1.5 Track Defects List parameter	227
11.3.4.1.6 Sector Defects List parameter	227
11.3.4.2 Parameters 58, 5A, 6C-6D	228
11.3.4.2.1 Variable Size Sectors parameter	228
11.3.4.2.2 Cell Defects List parameter	228
11.3.4.2.3 Request Parm parameter	228
11.3.4.2.4 Parm Length parameter	228
11.4 WRITE DEFECT LIST	229
11.4.1 Command packet	229
11.4.2 Response packet	229
11.4.3 Description	229
11.4.4 Parameters	230
11.4.4.1 Parameters 31-32, 3E, 55-57	230

11.4.4.1.1 Command Extent (common) parameter	230
11.4.4.1.2 Response Extent (common) parameter	230
11.4.4.1.3 Partition (common) parameter	231
11.4.4.1.4 Defective DataBlock parameter	231
11.4.4.1.5 Track Defects List parameter	231
11.4.4.1.6 Sector Defects List parameter	231
11.4.4.2 Parameters 58, 5A, 6C	231
11.4.4.2.1 Variable Size Sectors parameter	231
11.4.4.2.2 Cell Defects List parameter	232
11.4.4.2.3 Request Parm parameter	232
11.5 READ ERROR LOG	232
11.5.1 Command packet	232
11.5.2 Response packet	232
11.5.3 Description	232
11.5.4 Parameters 31-32, 50, 6C-6D	233
11.5.4.1 Command Extent (common) parameter	233
11.5.4.2 Response Extent (common) parameter	233
11.5.4.3 Error Log parameter	233
11.5.4.4 Request Parm parameter	233
11.5.4.5 Parm Length parameter	233
11.6 WRITE ERROR LOG	234
11.6.1 Command packet	234
11.6.2 Response packet	234
11.6.3 Description	234
11.6.4 Parameters 31-32, 50, 6C	235
11.6.4.1 Command Extent (common) parameter	235
11.6.4.2 Response Extent (common) parameter	235
11.6.4.3 Error Log parameter	235
11.6.4.4 Request Parm parameter	235
11.7 DIAGNOSTIC CONTROL	236
11.7.1 Command packet	236
11.7.2 Response packet	236
11.7.3 Description	236
11.7.4 Parameters 31-32, 50-53, 6C	237
11.7.4.1 Command Extent (common) parameter	237
11.7.4.2 Response Extent (common) parameter	237
11.7.4.3 Diagnostic Number parameter	237
11.7.4.4 Diagnostic Function List parameter	237
11.7.4.5 Diagnostic Action Code parameter	237
11.7.4.6 Vendor Unique parameter	238
11.7.4.7 Request Parm parameter	238
12 Command summary	239
12.1 Control commands	239
12.2 Position commands	240
12.3 Transfer commands	240
12.4 Combination commands	241
12.5 Other transfer commands	242
12.6 Diagnostic commands	243

Annexes

	Page
Annex A	244
A.1 Interface levels	244
A.2 Concepts	244
A.2.1 Relationship of master, slave, and facility	244
A.2.2 Relationship of facilities and partitions	245
A.2.3 Command structure	245
A.3 Application environments	246
A.3.1 Control of facilities by the master	246
A.3.2 Shared Control of Facilities	246
A.3.3 Control of Facilities by the Slave	246
Annex B	247
B.1 Informative references	247
B.2 Equivalent ANSI standards	247
Annex C	248
C.1 FORMAT command	248
C.2 REALLOCATE command	249
C.3 ALLOCATE RESTORE command	250

Figures

	Page
Figure 1 — Hierarchy of Slave Conditions	27
Figure 2 — Physical Switch	31
Figure 3 — Logical Switch	32
Figure 4 — Valid Combinations of the Master Status Octet	40
Figure 5 — Valid Combinations of Slave Status Octet	42
Figure 6 — Octet Transfer Positions	43
Figure 7 — Bit Positions	44
Figure 8 — Octet Positions	44
Figure 9 — Sequence of Octet Transfers	44
Figure 10 — Parameter Documentation	46
Figure 11 — Modifier octet	50
Figure 12 — Basic Command packet in SOM Representation	50
Figure 13 — Basic Command packet in DOM Representation	51
Figure 14 — Parameter List for 17-Octet Transfer	51
Figure 15 — Parameter List for 18-Octet Transfer	51
Figure 16 — Example of Defect List parameters	53
Figure 17 — Example of Defect List parameters as Naked Data	54
Figure 18 — Basic Response packet	56
Figure 19 — Types of Substatus and Responses	58
Figure 20 — Command packet for NOP	89
Figure 21 — Response packet for NOP	89
Figure 22 — Command packet for FACILITY OPERATION	90
Figure 23 — Response packet for FACILITY OPERATION	90
Figure 24 — Command packet for ATTRIBUTES	91
Figure 25 — Response packet for ATTRIBUTES	91
Figure 26 — Command packet for REPORT ADDRESSEE STATUS	114
Figure 27 — Response packet for REPORT ADDRESSEE STATUS	114
Figure 28 — Port Mask parameter	115
Figure 29 — Command packet for PORT ADDRESS	116
Figure 30 — Response packet for PORT ADDRESS	116
Figure 31 — Command packet for PATH CONTROL	118
Figure 32 — Response packet for PATH CONTROL	118
Figure 33 — Mask Octet Examples	118
Figure 34 — Command packet for ATTENTION CONTROL	119
Figure 35 — Response packet for ATTENTION CONTROL	120
Figure 36 — Command packet for OPERATING MODE	121
Figure 37 — Response packet for OPERATING MODE	121
Figure 38 — Command packet for ABORT	123
Figure 39 — Response packet for ABORT	123
Figure 40 — Command packet for ACCESS PERMITS	125
Figure 41 — Response packet for ACCESS PERMITS	126
Figure 42 — Command packet for RESUME	129
Figure 43 — Response packet for RESUME	129
Figure 44 — Command packet for PORT RESPONSE	130
Figure 45 — Response packet for PORT RESPONSE	131
Figure 46 — Command packet for ANTICIPATED ACTION	131
Figure 47 — Response packet for ANTICIPATED ACTION	132
Figure 47' — Command packet for OPERATOR DISPLAY	133
Figure 48' — Response packet for OPERATOR DISPLAY	133
Figure 48 — Command packet for POSITION CONTROL	138
Figure 49 — Response packet for POSITION CONTROL	138
Figure 50 — Command packet for REPORT POSITION	140
Figure 51 — Response packet for REPORT POSITION	140

Figure 52 — Command packet for REPORT DISCONTINUITY	141
Figure 53 — Response packet for REPORT DISCONTINUITY	141
Figure 54 — Command packet for READ	144
Figure 55 — Response packet for READ	144
Figure 56 — Command packet for READ RAW DATA	148
Figure 57 — Response packet for READ RAW DATA	148
Figure 58 — Command packet for READ REPLICATED DATA	150
Figure 59 — Response packet for READ REPLICATED DATA	150
Figure 60 — Command packet for SEARCH	152
Figure 61 — Response packet for SEARCH	152
Figure 62 — Command packet for WRITE	157
Figure 63 — Response packet for WRITE	157
Figure 64 — Command packet for WRITE PATTERN	159
Figure 65 — Response packet for WRITE PATTERN	159
Figure 66 — Command packet for FORMAT	161
Figure 67 — Response packet for FORMAT	161
Figure 68 — Command packet for COPY	170
Figure 69 — Response packet for COPY	170
Figure 70 — Command packet for COMPARE SLAVE DATA	174
Figure 71 — Response packet for COMPARE SLAVE DATA	174
Figure 72 — Command packet for COMPARE DATA	176
Figure 73 — Response packet for COMPARE DATA	176
Figure 74 — Command packet for REALLOCATE	178
Figure 75 — Response packet for REALLOCATE	178
Figure 76 — Command packet for ALLOCATE RESTORE	181
Figure 77 — Response packet for ALLOCATE RESTORE	181
Figure 78 — Command packet for SHADOW READ	184
Figure 79 — Response packet for SHADOW READ	184
Figure 80 — Command packet for SHADOW WRITE	186
Figure 81 — Response packet for SHADOW WRITE	186
Figure 82 — Command packet for SHADOW RESTORE	188
Figure 83 — Response packet for SHADOW RESTORE	188
Figure 84 — Command packet for READ VERIFY	191
Figure 85 — Response packet for READ VERIFY	191
Figure 86 — Command packet for READ AT FIRST AVAILABLE DATA	193
Figure 87 — Response packet for READ AT FIRST AVAILABLE DATA	193
Figure 88 — Command packet for READ FROM BUFFER	195
Figure 89 — Response packet for READ FROM BUFFER	195
Figure 90 — Command packet for READ FACILITY DATA TO BUFFER	198
Figure 91 — Response packet for READ FACILITY DATA TO BUFFER	198
Figure 92 — Command packet for READ PHYSICAL DATA AND ECC	200
Figure 93 — Response packet for READ PHYSICAL DATA AND ECC	200
Figure 94 — Command packet for READ PHYSICAL HEADER	202
Figure 95 — Response packet for READ PHYSICAL HEADER	202
Figure 96 — Command packet for READ IPL	203
Figure 97 — Response packet for READ IPL	203
Figure 98 — Command packet for READ PHYSICAL HEADER AND ECC	205
Figure 99 — Response packet for READ PHYSICAL HEADER AND ECC	205
Figure 100 — Command packet for WRITE TO BUFFER	207
Figure 101 — Response packet for WRITE TO BUFFER	207
Figure 102 — Command packet for WRITE BUFFER TO FACILITY	209
Figure 103 — Response packet for WRITE BUFFER TO FACILITY	209
Figure 104 — Command packet for WRITE PHYSICAL DATA AND ECC	211
Figure 105 — Response packet for WRITE PHYSICAL DATA AND ECC	211
Figure 106 — Command packet for WRITE PHYSICAL HEADER	213
Figure 107 — Response packet for WRITE PHYSICAL HEADER	213
Figure 108 — Command packet for LOAD SLAVE IML	215
Figure 109 — Response packet for LOAD SLAVE IML	215
Figure 110 — Command packet for ERASE	217
Figure 111 — Response packet for ERASE	217

Figure 112 — Command packet for WRITE PHYSICAL HEADER AND ECC	219
Figure 113 — Response packet for WRITE PHYSICAL HEADER AND ECC	219
Figure 114 — Command packet for PERFORM SLAVE DIAGNOSTICS	221
Figure 115 — Response packet for PERFORM SLAVE DIAGNOSTICS	221
Figure 116 — Command packet for PERFORM FACILITY DIAGNOSTICS	223
Figure 117 — Response packet for PERFORM FACILITY DIAGNOSTICS	223
Figure 118 — Command packet for READ DEFECT LIST	225
Figure 119 — Response packet for READ DEFECT LIST	225
Figure 120 — Command packet for WRITE DEFECT LIST	229
Figure 121 — Response packet for WRITE DEFECT LIST	229
Figure 122 — Command packet for READ ERROR LOG	232
Figure 123 — Response packet for READ ERROR LOG	232
Figure 124 — Command packet for WRITE ERROR LOG	234
Figure 125 — Response packet for WRITE ERROR LOG	234
Figure 126 — Command packet for DIAGNOSTIC CONTROL	236
Figure 127 — Response packet for DIAGNOSTIC CONTROL	236

Tables

	Page
Table 1 — Different Ways of Multiplexing Transfers	18
Table 2 — Manufacturer's Original Flaw Map and Slave's Defect List	23
Table 3 — Table of Equivalents for Slave	26
Table 4 — Possible Combinations of Request Interrupts Bits	29
Table 5 — Substatus parameters for Intervention Required	59
Table 6 — Substatus parameters for Alternate Port exception	60
Table 7 — Substatus parameters for Machine Exception	61
Table 8 — Substatus parameters for Command Exception	64
Table 9 — Substatus parameters for Command Aborted	66
Table 10 — Substatus parameters for Conditional Success	68
Table 11 — Substatus parameters for Incomplete	71
Table 12 — Substatus parameters for Message/Microcode Exception	73
Table 13 — Transfer Notification parameter	75
Table 14 — Command Extent parameter	76
Table 15 — Response Extent parameter	76
Table 16 — Combination Command Extent parameter	78
Table 17 — Combination Response Extent parameter	79
Table 18 — Access Key parameter	80
Table 19 — Invalid Parm parameter	80
Table 20 — Missing Parm parameter	81
Table 21 — Data Address parameter	81
Table 22 — Block Size parameter	82
Table 23 — Transfer parameters	82
Table 24 — Encapsulation parameter	84
Table 25 — Partition parameter	84
Table 26 — Stop on Discontinuity parameters	87
Table 27 — Imbedded Data parameter	88
Table 28 — Facility Operation parameter 3D	90
Table 29 — Attributes parameters 3A, 3E, 50	93
Table 30 — Attributes parameters 51-58	95
Table 31 — Attributes parameters 59-5A	97
Table 32 — Attributes parameters 5B-5D	98
Table 33 — Attributes parameters 5E-5F	99
Table 34 — Attributes parameters 60-63	100
Table 35 — Attributes parameters 64-65	102
Table 36 — Attributes parameter 66	103
Table 37 — Attributes parameter 67	105
Table 38 — Attributes parameter 68	106
Table 39 — Attributes parameters 69-6A	108
Table 40 — Attributes parameter 6B	109
Table 41 — Attributes parameters 6C-6D	110
Table 42 — Attributes parameter 6E	111
Table 43 — Attributes parameter 6F	113
Table 44 — Report Addressee Status parameters 50-52	115
Table 45 — Path Control parameters 50-51	119
Table 46 — Attention Control parameter 50	120
Table 47 — Operating Mode parameters 3E, 50-51	122
Table 48 — Abort parameters 50-54	124
Table 49 — Access Permits parameters 31-32, 35, 3A, 3E, 50-51	127
Table 50 — Resume parameters 50-51	130
Table 51 — Port Response parameters 50-51	131
Table 52 — Anticipated Action parameters 3D, 50	132
Table 53 — Operator Display parameters 50-53	134

Table 53a — Position Control parameters 31, 32, 35, 3A, 3E	139
Table 54 — Report Position parameters 32, 35, 3A, 3E	140
Table 55 — Report Discontinuity Params 31-32, 35, 3A, 3E-3F, 6C-6D	142
Table 56 — Read parameters 31-32, 35, 3A, 3C, 3E-3F	146
Table 57 — Read parameters 50-53	147
Table 58 — Read Raw Data parameters 31-32, 35, 3A, 3C, 3E-3F	149
Table 59 — Read Replicated Data parameters 31-32, 35, 3A, 3C, 3E, 50	151
Table 60 — Search parameters 02, 31-32, 35, 3A, 3E	153
Table 61 — Search parameter 50	154
Table 62 — Search parameter 51	155
Table 63 — Search parameter 52	155
Table 64 — Search parameter 53	156
Table 65 — Write parameters 31-32, 35, 3A, 3C, 3E-3F, 50-52	158
Table 66 — Write Pattern parameters 02, 31-32, 35, 3A, 3C, 3E, 50, 6C	160
Table 67 — Format parameters 31-32, 35, 3A-3C, 3E-3F	164
Table 68 — Format parameters 50-54	166
Table 69 — Format parameters 56-58	168
Table 70 — Format parameters 5A, 6C	169
Table 71 — Copy parameters 33-35, 3A, 3C, 3E, 50	172
Table 72 — Compare Slave Data parameters 33-35, 3A, 3E, 50	175
Table 73 — Compare Data parameters 31-35, 3A, 3E, 50	177
Table 74 — Reallocate parameters 31-32, 35, 3A, 3E	179
Table 75 — Reallocate parameters 50, 55	180
Table 76 — Allocate Restore parameters 31-32, 35, 3A, 3E	182
Table 77 — Allocate Restore parameters 50, 55	183
Table 78 — Shadow Read parameters 33-35, 3A, 3C, 3E, 51, 52	185
Table 79 — Shadow Write parameters 33-35, 3A, 3C, 3E, 51, 52	187
Table 80 — Shadow Restore parameters 33-35, 3A, 3C, 3E	189
Table 81 — Read Verify parameters 31-32, 35, 3A, 3C, 3E	192
Table 82 — Read at First Avail Data Parameters 31-32, 35, 3A, 3C, 3E-3F, 50-51	194
Table 83 — Read From Buffer parameters 31-32, 35, 3A, 3C, 3E, 50	196
Table 84 — Read Facility Data To Buffer Parameters 31-32, 35, 3A, 3C, 3E, 50	199
Table 85 — Read Physical Data And ECC parameters 31-32, 35, 3A, 3C, 3E	201
Table 86 — Read Physical Header parameters 31-32, 35, 3A, 3E	202
Table 87 — Read IPL parameters 31-32, 35, 3A, 3E	204
Table 88 — Read Physical Header And ECC parameters 31-32, 35, 3A, 3E	206
Table 89 — Write to Buffer parameters 31-32, 35, 3A, 3C, 3E, 50	208
Table 90 — Write Buffer To Facility Parameters 31-32, 35, 3A, 3C, 3E, 50	210
Table 91 — Write Physical Data and ECC parameters 31-32, 35, 3A, 3E	212
Table 92 — Write Physical Header parameters 31-32, 35, 3A, 3E	214
Table 93 — Load Slave IML parameters 31-32, 3E	216
Table 94 — Erase parameters 31-32, 35, 3A, 3E	218
Table 95 — Write Physical Header And ECC parameters 31-32, 35, 3A, 3E	220
Table 96 — Perform Slave Diagnostics parameters 31-32, 50-53, 6C	222
Table 97 — Perform Facility Diagnostics parameters 31-32, 50-53, 6C	224
Table 98 — Read Defect List parameters 31-32, 3E, 55-57	226
Table 99 — Read Defect List parameters 58, 5A, 6C-6D	228
Table 100 — Write Defect List parameters 31-32, 3E, 55-57	230
Table 101 — Write Defect List parameters 58, 5A, 6C	231
Table 102 — Read Error Log parameters 31-32, 50, 6C-6D	233
Table 103 — Write Error Log parameters 31-32, 50, 6C	235
Table 104 — Diagnostic Control parameters 31-32, 50-53, 6C	237

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 9318-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 9318 consists of the following parts, under the general title *Information technology — Intelligent Peripheral Interface*:

- *Part 1: Physical level*
- *Part 2: Device specific command set for magnetic disk drives*
- *Part 3: Device generic command set for magnetic and optical disk drives*
- *Part 4: Device generic command set for magnetic tape drives*

Annex A forms an integral part of this part of ISO/IEC 9318. Annexes B and C are for information only.

Introduction

This part of ISO/IEC 9318 does not replace any existing standard, but it does complement other Intelligent Peripheral Interface (IPI) standards (see clause 2).

This part of ISO/IEC 9318 provides a definition of the device-generic portion of a series of standards called the Intelligent Peripheral Interface (IPI), a high performance, general-purpose parallel peripheral interface. This part of ISO/IEC 9318 responds to an industry market need (expressed both by users and manufacturers) to limit the increasing costs in hosts associated with changes in peripherals.

The first five clauses of this part of ISO/IEC 9318 contain material that is useful across all classes of devices that the device-generic command sets can support. Clauses 6 to 12 are oriented to particular device classes and in this document are intended for use with Magnetic and Optical disks.

- Clause 1 describes the scope.
- Clause 2 lists the normative reference.
- Clause 3 provides descriptions and conventions.
- Clause 4 describes the environment-of-use and projected application areas.
- Clause 5 describes the Message Packet structure used for commands and responses.
- Clause 6 describes Control commands.
- Clause 7 describes Position commands.
- Clause 8 describes the most generic Transfer commands.
- Clause 9 describes the Combination Transfer commands, which require a minimum of two sets of extents.
- Clause 10 describes the other Transfer commands, which are more device specific than those in clause 8.
- Clause 11 describes the Diagnostic commands.
- Clause 12 summarizes the commands defined in the document.

Information technology - Intelligent Peripheral Interface -

Part 3:

Device generic command set for magnetic and optical disk drives

1 Scope

This part of ISO/IEC 9318 describes the Logical Level 3 (generic level) Interface for magnetic and optical disk drives. See clause 6 of ISO/IEC 9318-1 for an explanation of the levels.

The physical, electrical, and configuration characteristics and the transmission protocol of this interface are in accordance with ISO/IEC 9318-1. The interface is capable of handling data rates from 0 to at least 10 Mbytes/s, depending on driver and receiver classes.

The purpose of this part of ISO/IEC 9318 is to facilitate the development and utilization of an intelligent interface which permits the interconnection of multiple peripheral types such as disk, tape, communications, to a controller.

This part of ISO/IEC 9318 does not replace any existing standard, but it does complement other Intelligent Peripheral Interface (IPI) standards (see clause 2).

This part of ISO/IEC 9318 provides a definition of the device-generic portion of a family of standards called the Intelligent Peripheral Interface (IPI), a high performance, general-purpose parallel peripheral interface.

The intent of the IPI is to isolate the host (CPU), both hardware and software, from changes in peripherals by providing a "function-generic" command set to allow the connection of multiple types of peripherals (disks, printers, tapes, communications). To smooth the transition from the current methods to the generic approach, the IPI supports device-specific command sets to aid in bridging the gap between the two approaches.

To accomplish this set of goals, the design of the IPI includes device-specific and device-generic command sets, both utilizing a common physical bus. The device-specific command set provides:

- device-oriented control;
- physical data addressing;
- timing critical operations;
- lower device cost.

The device-generic command set provides a higher level of functionality and portability. It includes:

- host/device independence;
- logical data addressing;
- timing independence;
- command queuing capability.

A system is not restricted to the use of one level of command set or the other. It is possible that both levels of command sets will be utilized with a given system's architecture to balance such parameters as system performance, cost, and peripheral availability. It is also possible for the host to provide for migration from device-specific to device-generic levels while still retaining the same physical interface.

2 Normative reference

The following standard contains provisions which, through reference in this text, constitutes provisions of this part of ISO/IEC 9318. At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9318 are encouraged to investigate the possibility of applying the most recent edition of the standard indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 9318-1:—¹⁾ - *Information technology - Intelligent Peripheral Interface*
 - *Part 1: Physical Level*

1) To be published.

3 Definitions and conventions

3.1 Definitions

For the purpose of this part of ISO/IEC 9318, the following definitions apply:

3.1.1 actual address: The address of the facility, which is unique and is used by the slave to accomplish selection (the interface by which facilities are attached to the slave may or may not be an IPI).

3.1.2 addressee: The slave, the facility, or the combination of the two that are identified by the combination of Slave Address and Facility Address.

3.1.3 alias: The name of a partition within a facility.

3.1.4 attributes: Each slave (or facility) has a level of functionality that can be described by a table that contains entries for characteristics that require definition to the master. The entries are self-defining and are termed attributes.

3.1.5 burst: The maximum number of octets to be transferred in any one Information Transfer. The transfer of a number of bursts may be implicit within a single command request.

3.1.6 command: A sequence issued by the master to initiate some specific operation.

3.1.7 check sum disk operation: A high availability solution for a disk cluster in which a check sum is used to allow a failure in one device to be transparent to the host (i.e., the data in the failed device is reconstructed from the remaining devices).

NOTE — If a device fails, although processing may continue, it is without protection.

3.1.8 command address: The two octets of Slave Address and Facility Address in the basic Command packet which uniquely identify the addressee (or optionally partition) to which the command is issued.

3.1.9 data: Any Information Transfer over the interface not associated with either a command or a response.

3.1.10 DataBlock: A term which is uniquely defined in this part of ISO/IEC 9318 to mean the logical representation of data on the media. A DataBlock may or may not have a relationship to PhysicalBlocks (i.e., it may be equal to, less than, or greater than PhysicalBlocks in size). If commands are issued with DataBlock addresses to a slave that supports logical addressing, the slave is responsible to locate the physical representation of data by logical address correctly, and present it to the master in DataBlocks.

3.1.11 extent: A range of contiguous blocks, defined as the count of a number of blocks beginning at a data address (PhysicalBlock or DataBlock) within (and limited to) a partition.

3.1.12 facility address: The octet value in the Command Address that may be an Actual Address, Selection Address, Synonym, or Alias. It is used to identify the facility (or partition) to which the command is addressed. Valid values are from 0 to 255.

3.1.13 housekeeping: A procedure which is required at initial usage of a slave (or facility) because the master must obtain the attributes in order to adapt to, or configure, its functionality.

3.1.14 individual: This term refers to the execution of commands that must be completed before the next command can be issued.

3.1.15 information transfer: The transferring of octets on the Physical Interface of commands, responses, and data that are framed by a Bus Exchange.

3.1.16 interrupts: The ability at the Physical Interface for the slave to advise the master of which types of Response are available in the slave. The slave uses interrupts to initiate assertion of the ATTENTION IN signal.

ISO/IEC 9318-3 : 1990 (E)

3.1.17 L-available (logically available): See 4.7.1.7.

3.1.18 L-busy (logically busy): See 4.7.1.9.

3.1.19 level 2 (device-specific): This term refers to commands that may be timing critical and that are used to define the execution of device-dependent operations. (See ISO/IEC 9318-2 for a further explanation and description of these commands).

3.1.20 level 3 (device-generic): This term refers to commands that are not timing critical and that are in an intelligent environment in which the slave has functional control (which may or may not be overridden by the master) over the attached facility or facilities.

3.1.21 Logical Address: The address of a cluster configuration by which the master would address the cluster.

3.1.22 logical interface: This term refers collectively to all protocols higher than the Physical Interface.

3.1.23 mandatory: To conform to the standard, all functions described as mandatory shall be implemented as defined in this part of ISO/IEC 9318.

3.1.24 multiplex: The ability of a master to intersperse the execution of commands between addressees; or of a slave to intersperse the execution of commands between different facilities; or of a slave to intersperse transfer information in bursts that are less than the requested transfer size.

3.1.25 operational: See 4.7.1.3.

3.1.26 optional: Features that are not required by the standard. However, if any feature defined by the standard is implemented, it shall be done in the same way as defined by the standard.

3.1.27 parity disk operation: A high availability solution for a disk cluster in which a parity device contains redundancy for the remaining devices such that a single device failure is transparent to the host (i.e., the data on the failed device is reconstructed from the remaining devices).

NOTE — The use of parity usually implies that the devices are also in parallel.

3.1.28 P-available (physically available): See 4.7.1.1.

3.1.29 P-busy (physically busy): See 4.7.1.5.

3.1.30 partition: A recording area that may be logically addressed. A partition may be slave defined (e.g., data area, CE area, IML area) or may be master defined (e.g., an addressable set of contiguous blocks within the data area). See also alias.

3.1.31 PhysicalBlock: This term is uniquely defined in this part of ISO/IEC 9318 as meaning the physical representation of data on the media (e.g., sectors or records on disk and blocks or records on tape). It is used to prevent confusion between industry usage of terms.

3.1.32 physical interface: This term refers to the mechanical, electrical, and bus protocol characteristics specified in ISO/IEC 9318-1.

3.1.33 queued: The ability of a slave to accept multiple commands per Facility Address from the master and execute them in a sequence according to slave-defined or master-defined algorithms.

3.1.34 ready: This term is used to indicate that a slave or facility can execute its intended functions.

3.1.35 response: A sequence issued by a slave to advise the master of the results of a command, or of conditions within the slave.

3.1.36 selection address: The address used by the master at the Physical Interface to select a slave, a facility, or both. (This may not be the same as the Actual Address if Synonyms are used.)

3.1.37 shadow disk operation: A high availability solution for a disk cluster in which duplicate copies of a disk exist. This allows continued operation even though one device fails.

NOTE — If one device fails, processing may continue, but it is without protection.

3.1.38 synonym: The ability to redefine the Facility Address of a Facility. There may be more than one synonym to address the same facility.

3.1.39 vendor unique: Those features that can be defined by a vendor in a specific implementation. Caution should be exercised in defining and using such features since they may or may not be standard between vendors.

3.2 Conventions

In this part of ISO/IEC 9318, certain terms that are proper names of signals are printed in uppercase to avoid possible confusion with other uses of the same words (e.g., ATTENTION IN). Any lowercase uses of these words have the normal English meaning.

A number of conditions, sequence parameters, events, English text, states or similar terms are printed with the first letter of each word in uppercase and the rest lowercase (e.g., In, Out, Selective Reset, Bi-directional, Bus Control, Operation Response). Any lowercase uses of these words have the normal English meaning.

4 Logical interface characteristics

4.1 Operations

The Logical Interface uses a packet structure to transfer Commands from the master to the slave and Responses from the slave to the master. A Bus Exchange at the Physical Interface requires a Bus Control sequence and its associated Ending Status sequence to frame the Information Transfer of Commands, Responses, and data.

The Commands and Responses vary in length and only a single packet of data is transferred for each Information Transfer.

4.1.1 Commands

Commands are issued by the master to instruct the slave, facility, or both to perform an operation. The slave returns a Response when the command has been completed (unless inhibited by the "Inhibit Operation Response on Success" attribute set by the master).

4.1.1.1 Command types

The command types include:

- Control. The Control commands provide for control of the slave and facility or facilities.
- Position. The Position commands cause the positioning of the facility or facilities.
- Transfer. The Transfer commands may cause multiple blocks of data to be transferred between the master and the facility. Before data is actually transferred, the slave activates the Class 2 Interrupt to inform the master that it is ready to transfer data. The complete data transfer may be broken up into several Information Transfers. A slave with Command Queuing shall generate a Transfer Notification to inform the master of the identity of the command for which the transfer is pending.
- Combination. The Combination commands provide for operations between two facilities attached to the same slave, or between two facilities attached to different slaves if Slave-to-Slave Information Transfers are supported at the Physical Interface. In an operation between two different slaves, the master shall designate a dominant slave and a subservient slave. The dominant slave assumes the role of the master for the purpose of initiating Information Transfers to carry out slave-to-slave operations.
- Diagnostic. The Diagnostic commands provide for maintenance and diagnostic operations between a master and a slave or facility. These commands may be product specific, vendor specific, or both.

4.1.1.2 Command stacking

A slave that can accept more than one command is capable of command stacking. The number of commands that may be stacked is defined by the slave's attributes. When a master attempts to send more commands to a slave than can be accepted, rejection occurs at the Physical Interface. Commands that are stacked may be Individual or Queued.

4.1.1.2.1 Individual

The slave can accept only one operation for every facility under its control (i.e., a queue of one).

4.1.1.2.2 Queued

The master can have more than one operation for every facility concurrently active under control of the slave, and the slave is responsible to execute them. This permits the slave the freedom to optimize the sequence of command execution to enhance performance (e.g., seek ordering algorithms). The master has the ability to override slave optimization via Attributes.

4.1.1.3 Command execution order

The order in which commands are executed is controlled by the command modifier bits that are common to all commands. Commands that are identified as Chained, Sequential, or Ordered cannot be intermixed for a given addressee. The last command of a Chain, Sequence, or Order has no encoding of its identity in order to identify it as being the last one - the slave shall be capable of recognizing it as being the last one, rather than treating it as an Individual or Queued command. The execution scenarios possible are:

- a) Individual/Queued Commands. Commands other than those labeled by modifier bits as Chained, Sequential, Ordered, or Priority are executed in a slave-dependent order or a facility-dependent order or both.
- b) Chained Commands. Commands are executed in the order received by the slave (first in, first out (FIFO) order) (i.e., a sequence of commands (not necessarily stacked) to a single addressee). The addressee is implicitly reserved as long as any command of the Chain is being executed. If a command in the Chain is unsuccessful, the Chain shall be terminated and the remaining commands are not executed. The Chained modifier encoding is not set for the last command in a Chain. The slave may multiplex operations for other addressees during execution of the Chain.
- c) Sequential Commands. Commands are executed in the order received by the slave (FIFO order). There may be more than one addressee in a Sequence and the slave may multiplex other operations during execution of the Sequence. If a command in the Sequence is unsuccessful, the Sequence shall be terminated and the remaining commands are not executed. The Sequential modifier encoding is not set for the last command in a Sequence. There is no implicit Reserve of the addressee beyond the command being executed.
- d) Ordered Commands. Commands are executed in the order received by the slave (FIFO order). There may be more than one addressee in an Order, but the slave shall not multiplex other operations during execution of the Order. If a command in the Order is unsuccessful, the Order shall be terminated and the remaining commands are not executed. The Ordered modifier encoding is not set for the last command in an Order. There is no implicit Reserve of the addressee beyond the command being executed.
- e) Priority Commands. Priority commands and Priority Chains/Sequences/Orders are indicated by the command modifier bits. Only the first command in a Chain/Sequence/Order shall be designated as Priority, but the Priority shall apply to the entire Chain/Sequence/Order. A Priority command causes the slave to change the order of execution of stacked commands and also causes changes in interpreting the Command packet transmission for each addressee.

When a Priority Command packet is received for an addressee, any Chain/Sequence/Order that was in the process of being received shall be ended (i.e., the last packet received is interpreted as the last of the Chain/Sequence/Order regardless of the command modifier settings in that packet. This ending applies regardless of whether the slave is operating in a queued or non-queued environment.

NOTE — This applies only to commands received over the same port.

Priority commands, whether Individual or part of a Chain/Sequence/Order, are executed before non-Priority commands. Priority commands are executed in Last In First Out (LIFO) order. With the exception of individual ABORT commands, the receipt of a Priority command does not affect the operation of any Individual command or commands in a Chain/Sequence/Order, except for the possible ending noted previously. When a Priority, Individual ABORT command is received, the slave shall suspend the command executing (if possible), and process the ABORT.

ISO/IEC 9318-3 : 1990 (E)

4.1.2 Operation responses

Each Command has an associated Response packet. Command Completion Response packets contain status that notifies the master whether or not the command was successful, and if not successful, why not. The response is not generated or transferred if the "Inhibit Operation Response on Success" attribute is set by the master.

Response packets are also generated to notify the master of commands that are ready for data transfer, or for asynchronous events that have occurred in the slave or facilities.

4.1.2.1 Interrupts

Physical Interface interrupts are used by the slave to request service from the master.

The master may poll any or all of the interrupt classes by setting the appropriate bits in the Request Interrupts octet of the Physical Interface. It is the responsibility of the master to prioritize the interrupts in the event that more than one slave is interrupting.

Optionally, the master may poll the interrupts from the facilities attached to a slave by the Request Facilities Interrupts octet of the Physical Interface if this feature is supported by the slave.

The Physical Interface ATTENTION IN signal is a logical OR of the interrupts from all but the addressee selected on the Physical interface. The interrupts may be enabled and disabled from generating Attention.

The interrupt classes for slaves and facilities are defined in descending order of priority. Slaves shall present interrupts in order of their priority, except when the response stack is full of lower order interrupt responses (i.e., if a Class 2 could be presented because data is available in a buffer, but the stack is full of Class 1 responses because the master has not requested them, the slave shall not respond to a poll for Class 2 interrupts).

An interrupt is cleared when the condition that caused it to be presented is no longer present.

The three classes of interrupts are as follows:

- a) Class 3 (Critical Status Pending). This interrupt shall be used to alert the master to events or conditions existing in a slave or facility that require immediate attention from the master and that are not associated with command completion. An Asynchronous response shall be used to alert the master, and the conditions initiating same are defined by the Major Status, associated Substatus, and Extended Substatus (if any).
- b) Class 2 (Transfer Status Pending). This interrupt shall be used by slaves to inform the master that data is ready to be transferred. A Transfer Notification to identify the command involved shall be generated by the slave unless the master is selecting facilities and executing Individual commands; or a Paused transfer implies the Transfer Notification.
- c) Class 1 (Status Pending). This interrupt shall be used to identify command completion responses, successful or otherwise, of commands issued by the master, and to inform the master of Asynchronous responses that are not critical. The Major Status and associated Substatus, if any, identify either the conditions under which the command terminated or the cause of the Asynchronous response.

An optional use of this interrupt is to report that the slave, which was previously reported Busy during a selection sequence, is no longer Busy. When the interrupt is used in this manner, the master is responsible for recognizing that there is not necessarily a Command Completion or Asynchronous Response packet already in the slave. If there is no Response packet pending in the slave, the interrupt shall be cleared by a successful selection sequence.

4.1.2.2 Response types

The response types do not directly correspond to the three interrupt classes. An Asynchronous response may be Class 3 if critical (e.g., power failure warning), or Class 1 if not critical (e.g., transition from Ready to Not Ready).

4.1.2.2.1 Command completion response

This response shall be generated by the slave when a command has been completed, unless the response has been disabled by Attributes being set to "Inhibit Operation Response on Success."

The Class 1 interrupt is activated to inform the master that a Command Completion Response is available.

4.1.2.2.2 Transfer Notification Response (optional)

This response shall be generated by a slave capable of stacking Individual or Queued commands. It is used to identify the command with which the data transfer to follow is associated.

Transfer Notification Responses are not required under the following conditions:

- When commands are Individual, and the command was preceded by a facility selection
- For continuation of paused data transfers

The Class 2 Interrupt is activated to inform the master that a Transfer Notification Response is available.

4.1.2.2.3 Asynchronous response

This response shall be generated by the slave to advise the master of an unanticipated event not associated with command completion.

Either the Class 3 or Class 1 Interrupt is activated to inform the master that this response is available (depending upon whether or not it is considered critical by the slave).

4.1.2.2.4 Imbedded data response (optional)

This response shall be generated by the slave to send small amounts of data in a parameter field of the Response packet. No more than 254 octets can be transferred in this manner.

The objective of this response is to permit peripherals with very low transfer rates and small transfer needs to gather data in a manner that does not interfere with high-performance peripherals. This response shall not terminate the command, and many of them can be received in answer to a single command.

The Class 2 Interrupt is activated to inform the master that an Imbedded Data Response is available.

4.1.2.3 Response handling

When there is more than one Response packet to be transferred by the slave, it shall transfer Response packets to the master in the order of interrupt priority. Within an interrupt class, the responses shall be presented in order of command completion.

If facility selection precedes the request for a Response packet, the slave shall only transfer the highest priority response among those for the selected facility.

The master shall properly handle any Response packet that it receives, even if it is not necessarily the response type that it expected.

ISO/IEC 9318-3 : 1990 (E)

The master should not attempt to initiate the transfer of a Response packet before interrupts indicate the availability of a response ready to transfer. If the slave has no Response packet to transfer upon receiving a request for one, it shall terminate the attempted Information Transfer (without transferring any information) and not post the Successful bit in Slave Status.

When a slave that supports Control of Bus at the Physical Interface is given control, it may elect to transfer a Response packet or transfer data.

4.1.3 Physical interface error recovery considerations

Errors detected by the Physical Interface protocol require recovery action be taken via a Device-Generic protocol. Recovery procedures depend on the packet type (Command, Data, or Response), and on whether or not the error is detected prior to the Slave Status octet being presented.

4.1.3.1 Recovery from unsuccessful Slave Status octet

The Slave Status octet contains the Successful Information Transfer bit, which is set to 0 if any errors are detected by the slave prior to transmission of the octet. This includes not posting Successful if the slave recognized bad parity on the Master Status octet.

NOTE — The slave does not change its Slave Status octet contents based on the contents of the Master Status octet.

Following a command transfer, if the Success bit in either the Master or the Slave Status octet is set to 0, the slave shall ignore the command. The master shall retransmit the command.

Following a data transfer, if the Success bit in either the Master or the Slave Status octet is set to 0, the slave shall terminate the command in progress. A response indicating the failure shall be generated.

Following a response transfer, if the Success bit in either the Master or the Slave Status octet is set to 0, the slave shall retain the response in its buffer and attempt re-transmittal under the master's control. The slave shall assert the appropriate interrupt upon deselection.

4.1.3.2 Recovery from bad parity on the Slave Status octet

If the master detects a parity error or any other invalid condition on the Slave Status octet, the slave has no way to determine what has happened (as this is presented in the last state before deselection may occur).

Whenever an invalid condition is detected in the Slave Status octet following a command, data, or Transfer Notification packet transfer, the master shall use an ABORT command or a selective reset (either Logical Interface Reset or Slave Reset) to force the slave to discontinue execution of the command.

If an invalid condition is detected in the Slave Status octet following a Response packet other than Transfer Notification (typically Command Completion), the master shall retry execution of the entire command because the slave is not aware of its failure, and has released the buffer containing the Response packet.

NOTE — In the case of a slave that queues commands, the master may have to issue no more commands until it can match valid responses with all the issued commands in order to identify the one that was unsuccessful.

If an invalid condition is detected in the Slave Status octet following an Asynchronous Response packet, the master has no way of advising the slave that the Slave Status octet was not received correctly. Master-specific recovery procedures may choose whether or not to accept the contents of the Information Transfer. If it is a condition that keeps recurring, the slave shall generate another response.

NOTE — Under this circumstance the error recovery action cannot be certain to obtain the same information as was originally presented.

4.2 Operation sequences

Operation sequences of the logical interface are controlled by the master, since it establishes the Bus Exchanges that transfer information between the master and slave. Optionally, the master may give control of the bus to a slave; actual control of the Information Transfer at the Physical Interface would then be performed by the slave.

The sequence of actions taken to execute operations between master and slave(s) is done as a series of Information Transfers across the Physical Interface. Slave selection and deselection is done as necessary to communicate with the desired addressee.

4.2.1 Slave procedures

The following procedures are not necessarily supported by every slave. The slave's Attributes indicate those which are supported.

- a) Facility Selection and Facility Interrupts. When the slave supports facility selection and interrupts, the master may make use of them to exert control of the facility. The use of the facility interrupts allows the master to determine the facility requiring service and the facility selection allows the master to direct an operation to the desired facility.
- b) Slave Control. When the slave Control of Bus is supported (Bus Acknowledge octet), the master may give control of the subsequent Information Transfer to the slave to allow it to determine whether to transfer a Response packet or data. The slave shall not set the Operation Out bits in the Bus Acknowledge octet (i.e., attempt to cause a command to be transferred).
- c) Queued Facility. When the slave's Attributes indicate that a facility can operate in a queued environment, the master may issue more than one command for a facility.

4.2.2 Basic steps

The basic steps performed by the master in carrying out operations are

- a) transmit Command packet to slave;
- b) poll interrupts;
- c) receive response from slave;
- d) transfer data between master and slave.

These steps shall be as described in 4.2.2.1 through 4.2.2.4.

4.2.2.1 Transmit Command packet to slave

The master establishes a Bus Exchange to transfer a single Command packet to the selected slave. The number of commands that can be stacked at an addressee is defined in the Attributes.

Commands consist of variable length transfers, and some slaves may terminate the transfer based on the Packet Length defined in the first two octets. If slaves require master termination of transfers, the length of the transfer is decided by the master.

If the Master Status octet in the Ending Status sequence indicates an unsuccessful transfer, then the slave shall ignore the received command. No Response packet shall be generated.

If the Slave Status octet in the Ending Status sequence indicates a successful transfer, the master need not retain the entire Command packet.

ISO/IEC 9318-3 : 1990 (E)

4.2.2.2 Poll interrupts

The master polls interrupts to know when to perform data transfers or receive Responses. The polling of a specific interrupt from each of the slaves is performed by the Request Interrupt sequence of the Physical Interface. The presence of an interrupt in any of the slaves is typically provided by the ATTENTION IN signal at the Physical Interface. The master can choose to mask out interrupts by class from setting Attention (via the ATTENTION CONTROL command).

Even if a master has masked out all interrupts from setting Attention, it is still necessary to recognize that unsolicited Attentions could occur (e.g., if an attached slave executed a Power On reset, the mask information would be lost, and an Attention caused by powering on would be generated when its drivers were enabled).

If facility interrupts are supported by the slave, the master may poll facilities attached to a slave after receiving an interrupt from the slave.

NOTE — The slave generates interrupts on behalf of the attached facility or facilities.

4.2.2.3 Receive Response packet from slave

Upon receiving an interrupt from a slave, the master establishes a Bus Exchange to transfer the slave's Response. There is one exception to this action by the master: when the interrupt is Class 2 and the master is executing in a non-queued environment with facility selection.

The master may give control of the bus to the slave (if this function is supported) so that the slave can decide whether to transfer a Response packet or to transfer data.

If the Master Status in the Ending Status sequence indicates a successful transfer, the slave releases the Response packet and deactivates the interrupt; otherwise, it retains the packet until successfully transferred.

The Response packet may indicate

- a) the successful or unsuccessful completion of a command;
- b) a notice that data is to be transferred for a particular command;
- c) the occurrence of an asynchronous event.

4.2.2.4 Transfer of data between master and slave

After receiving a Transfer Notification [or when there is a Class 2 interrupt in a non-queued environment with facility selection (see 4.2.2.3)], the master initiates a Bus Exchange to transfer data to or from the slave as required by the command identified in the response. (If the slave is given Control of Bus, it sets up the direction of transfer.)

The amount of data intended to be transferred shall be determined by the slave (because it generates SYNC IN at the Physical Interface). Whether or not the ending Master Status indicates a successful input transfer, the slave may release its input data buffer. If the Slave Status indicates a successful output transfer, the master may release its output data buffer.

If the master terminates a data transfer, the characteristics of the slave shall determine whether or not it is an unsuccessful transfer. To a fixed block class of device such as disk, the transfer would be considered unsuccessful unless the transfer was being Paused (optional feature) by the master. To a variable block class of device such as tape, master termination would normally be considered successful and a short PhysicalBlock would be written.

If not Paused by the master, the Command Completion response shall identify the termination to the master (whether or not the transfer was successful) as "Unexpected Master Status" in Machine Exception Substatus.

If the transfer requested by the command was completed successfully, unless overridden by Attributes, the slave shall generate a Class 1 interrupt and a Command Completion response. If the transfer is unsuccessful, the slave shall generate a Class 1 interrupt and report the appropriate status in the Command Completion Response.

4.2.3 Operation sequence examples

The four steps described in the preceding subclauses may be used in a number of ways, and can best be illustrated by a set of examples executing the same scenario in different implementations. In these examples, the data transfer for Facility 1 requires two transmissions because the slave paused (either unanticipated, or multiplexed) during transfer.

The examples attempt to define only major steps in the flow (e.g., deselection is implicit). To clarify direction of transfer, the terms "transmit" are used for transfers out to the slave, and "receive" is used for transfers in from the slave. The term "recognize" is used for polling because the master has a choice of alternatives when polling (between slaves, facilities and masking).

4.2.3.1 Example of facility selection and individual commands

In this example, transfer commands are sent to facilities 0, 1 on slave 0.

- a) Select Slave 0/Facility 0 and transmit command packet
- b) Select Slave 0/Facility 1 and transmit Command packet
- c) Poll interrupts and recognize Class 2 Interrupt from Slave 0
- d) Poll Slave 0 and recognize Class 2 Interrupt from Facility 1
- e) Select Slave 0/Facility 1 and continue to perform data transfer until paused by Slave 0
- f) Poll interrupts and recognize Class 2 Interrupt from Slave 0
- g) Poll Slave 0 and recognize Class 2 Interrupt from Facility 0
- h) Select Slave 0/Facility 0 and perform data transfer until terminated by Slave 0
- i) Poll interrupts and recognize Class 1 Interrupt from Slave 0
- j) Poll Slave 0 and recognize Class 1 Interrupt from Facility 0
- k) Select Slave 0/Facility 0 and receive Command Completion Response with "Successful" Status
- l) Poll interrupts and recognize Class 2 Interrupt from Slave 0
- m) Poll Slave 0 and recognize Class 2 Interrupt from Facility 1
- n) Select Slave 0/Facility 1 and perform data transfer until terminated by Slave 0
- o) Poll interrupts and recognize Class 1 Interrupt from Slave 0
- p) Poll Slave 0 and recognize Class 1 Interrupt from Facility 1
- q) Select Slave 0/Facility 1 and receive Command Completion Response with "Successful" Status

4.2.3.2 Example of facility selection and queued commands

In this example, Transfer commands are sent to Facilities 0, 1 on Slave 0.

- a) Select Slave 0/Facility 0 and transmit Command packet
- b) Select Slave 0/Facility 1 and transmit Command packet
- c) Poll interrupts and recognize Class 2 Interrupt from Slave 0
- d) Poll Slave 0 and recognize Class 2 Interrupt from Facility 1
- e) Select Slave 0/Facility 1 and receive Transfer Notification identifying command
- f) Perform data transfer until paused by Slave 0
- g) Poll interrupts and recognize Class 2 Interrupt from Slave 0
- h) Poll Slave 0 and recognize Class 2 Interrupt from Facility 0
- i) Select Slave 0/Facility 0 and receive Transfer Notification identifying command
- j) Perform data transfer until terminated by Slave 0
- k) Poll interrupts and recognize Class 1 Interrupt from Slave 0
- l) Poll Slave 0 and recognize Class 1 Interrupt from Facility 0
- m) Select Slave 0/Facility 0 and receive Command Completion Response with "Successful" Status
- n) Poll interrupts and recognize Class 2 Interrupt from Slave 0
- o) Poll Slave 0 and recognize Class 2 Interrupt from Facility 1
- p) Select Slave 0/Facility 1 and receive Transfer Notification identifying command
- q) Perform data transfer until terminated by Slave 0
- r) Poll interrupts and recognize Class 1 Interrupt from Slave 0
- s) Poll Slave 0 and recognize Class 1 Interrupt from Facility 1

ISO/IEC 9318-3 : 1990 (E)

- t) Select Slave 0/Facility 1 and receive Command Completion Response with "Successful" Status

NOTE — Applications that implement multiple commands per facility in a non-ordered sequence, although not included in this example, require the use of Transfer Notification Packets. This example represents the same scenario as the other examples in this clause.

4.2.3.3 Example of slave selection.

In this example, Transfer commands are sent to Facilities 0, 1 on Slave 0.

- a) Select Slave 0 and transmit Command packet for Facility 0
- b) Remain selected and transmit Command packet for Facility 1
- c) Poll interrupts and recognize Class 2 Interrupt from Slave 0
- d) Select Slave 0 and receive Transfer Notification identifying command for Facility 1
- e) Perform data transfer until paused by Slave 0
- f) Poll interrupts and recognize Class 2 Interrupt from Slave 0
- g) Select Slave 0 and receive Transfer Notification identifying command for Facility 0
- h) Perform data transfer until terminated by Slave 0
- i) Poll interrupts and recognize Class 1 Interrupt from Slave 0
- j) Select Slave 0 and receive Command Completion Response for Facility 0 with "Successful" Status
- k) Poll interrupts and recognize Class 2 Interrupt from Slave 0
- l) Select Slave 0 and receive Transfer Notification identifying command for Facility 1
- m) Perform data transfer until terminated by Slave 0
- n) Poll interrupts and recognize Class 1 Interrupt from Slave 0
- o) Select Slave 0 and receive Command Completion Response for Facility 1 with "Successful" Status

4.2.3.4 Example of slave selection and slave control of Bus

In this example, Transfer commands are sent to Facilities 0 and 1 on Slaves 0 and 1, respectively. This introduces implicit Transfer Notification (i.e., only one data transfer is paused at a slave, so the Transfer Notification can be implied when the transfer is continued).

- a) Select Slave 0 and transmit Command packet for Facility 0
- b) Select Slave 1 and transmit Command packet for Facility 1
- c) Poll interrupts and recognize Class 2 Interrupt from Slave 0
- d) Select Slave 0 and with Bus Control give Control of Bus to slave
- e) Receive Bus Acknowledge octet indicating Response (Operation In)
- f) Receive Transfer Notification identifying command for Facility 0
- g) Master sets Bus Control (Data Direction)
- h) Perform data transfer until paused by Slave 0
- i) Poll interrupts and recognize Class 2 Interrupt from Slave 1
- j) Select Slave 1 and with Bus Control give Control of Bus to slave
- k) Receive Bus Acknowledge octet indicating Response (Operation In)
- l) Receive Transfer Notification identifying command for Facility 1
- m) Master sets Bus Control (Data Direction)
- n) Perform data transfer until terminated by Slave 1
- o) Poll interrupts and recognize Class 2 Interrupt from Slave 0
- p) Select Slave 0 and with Bus Control give Control to slave
- q) Receive Bus Acknowledge octet with implicit Transfer Notification and Data Direction
- r) Perform data transfer until terminated by Slave 0
- s) Poll interrupts and recognize Class 1 Interrupt from Slave 0
- t) Select Slave 0 and with Bus Control give Control of Bus to slave
- u) Receive Bus Acknowledge octet indicating Response (Operation In)
- v) Receive Command Completion Response for Facility 0 with "Successful" Status
- w) Poll interrupts and recognize Class 1 Interrupt from Slave 1
- x) Select Slave 1 and with Bus Control give Control of Bus to slave
- y) Receive Bus Acknowledge octet indicating Response (Operation In)
- z) Receive Command Completion Response for Facility 1 with "Successful" Status

4.3 Multiplexed data transfers (optional)

Different computer system architectures will utilize the IPI, and a wide range of products may be intermixed on a single cable. There are occasions in which a transfer may be terminated for either an unanticipated reason or a deliberate event. The interface provides for both to occur via use of the Pause function at the Physical Interface. Command, and Response packet transfers cannot be paused - only data transfers may be paused and continued.

It is necessary that a master be able to control some of the ways in which different products from different vendors coexist. One important area is in ensuring equitable use of the bandwidth. A deliberate event to pause a transfer occurs when a master wishes to prevent a particular peripheral from dominating the use of the interface during Information Transfers.

In an application in which the master requests large data transfers, it is preferred that the master be able to predict the use of the interface. The master can accomplish this by multiplexing transfers between addressees on a predictable basis. The master may define (in Attributes) the maximum number of octets to be transferred in any one data transfer (a burst), even though any single command may itself define considerably more.

A command for a 20Koctets data transfer to a slave with a defined 8Koctets maximum burst size would require three data transfers in order to complete. The slave would pause after each 8Koctets burst and complete after the remainder of 4K was transferred.

4.3.1 Physical interface Pause and Continue (optional)

The Physical Interface provides the ability to Pause and Continue transfers from both the master and the slave. The master and the slave may use this function to interrupt a continuous data transfer. The slave shall transfer data in bursts no longer than those defined or set in Attributes.

The master cannot indicate any time dependency to the slave, only whether or not the transfer is paused. However, the slave can indicate to the master some idea of the expected time period that the transfer shall be paused. This is done through use of the Time-Dependent Operation bit in the Slave Status octet.

The meaning of the bits shall be as follows:

Pause		Time-Dependent Operation
0	1	
0	0	No Delay
1	0	Short Delay
1	1	Long Delay
0	1	No More Data

If for some reason both the master and slave present the Pause bit in their respective ending status octets, the slave Pause shall override the master Pause.

In the following descriptions (4.3.2 to 4.3.9), the term P/TDO refers to the setting of the Pause and Time-Dependent Operation bits in the Slave Status octet.

4.3.2 Slave pause and master Continue

The master controls the slaves that receive service. The master shall use Attributes to establish the size of transfer, defined as a burst, to be multiplexed. During data transfers, when the burst defined by the master has been transmitted, the slave shall terminate. If information is immediately available to resume transfer with another burst, it shall so indicate by P/TDO=00. This condition can only occur on multiplexed burst boundaries.

If a slave Pauses with no information remaining in its buffer to be transferred, but it anticipates only a short delay before information shall be available, it shall set P/TDO=10. If, however, the delay is expected to exceed the value defined in Attributes, it shall set P/TDO=11.

ISO/IEC 9318-3 : 1990 (E)

If it is deselected, the slave shall generate a Class 2 interrupt when it has more information to transfer (immediately when P/TDO=00).

When a slave transferring information pauses at the end of a burst, the master may choose to deselect and poll the other slaves, or remain selected and continue the transfer.

To continue a paused data transfer, the master shall re-select (if necessary), and resumption of the data transfer may or may not be implicit, based on the mode of interface operation.

The slave shall always have a Transfer Notification packet prepared (if they are being used) so that if the master issues a Response request, the slave can respond with same.

4.3.2.1 Implicit continue

If the master is executing Individual commands with Facility Selection, it can reissue the same Bus Control octet as for the initial transfer.

If the master is executing Queued commands to a slave that can have more than one paused transfer, the master has to receive a Transfer Notification Response packet to know which one is being continued. To do so, the master may issue the Bus Control octet for a Response, or permit the slave Control of Bus and the slave shall set up the data transfer of a Response.

4.3.2.2 Explicit continue

If the master is executing Queued commands to a slave, but only one transfer can be paused per slave, the master may reissue the same Bus Control Octet as for the initial data transfer.

Alternatively, the master may provide the slave Control of Bus, and the slave shall respond with the Data, Direction, and Control of Bus Accepted bits set in the Bus Acknowledge octet. The explicit continue prevents a slave, that can have more than one paused transfer and has been given Control of Bus, from starting a second paused transfer.

NOTE — The paused slave is locked to the same transfer command until it is either successfully or unsuccessfully completed. It is possible for a master to issue commands or request responses from a paused slave, but based on internal design a slave may or may not be able to comply. The only way in which a master can free a paused slave is to continue the data transfer or abort the transfer command that is Paused.

4.3.3 Slave control of Pause and Continue

A slave may choose to pause data transfer due to an unanticipated delay encountered (e.g., repositioning, error correction). If so, the slave shall terminate the transfer and set P/TDO=11. The master can deselect the slave and await a Class 2 interrupt to advise that transfer can be continued.

If the master had deselected, when the slave is ready to continue the transfer, the slave shall generate a Class 2 interrupt. The master shall re-select and set the Control of Bus bit in the Bus Control octet. If the master did not deselect, the master can remain in a loop presenting the Bus Control octet, and being rejected by the slave until the slave is ready to continue.

If the master used the Control of Bus bit, the slave shall set the Data, Direction, and Control of Bus Accepted bits in the Bus Acknowledge octet, and transfer shall begin after the master asserts the MASTER OUT signal.

4.3.4 Master control of Pause and Continue

When a master encounters a temporary interruption, sufficient to stop a data transfer, it shall initiate termination of the transfer and set the Pause bit in the Master Status octet. The master may choose to deselect or remain selected to the slave.

To continue a Paused data transfer, the master shall re-select (if necessary), along with the same settings in the Bus Control octet used for the original command, then assert the MASTER OUT signal to restart the data transfer.

The slave shall not generate a Class 2 interrupt, but if it encounters any error while Paused, it may generate either a Class 3 or a Class 1 interrupt.

4.3.5 Uses of multiplexing

The three basic uses that are made of multiplexing between addressees are described in 4.3.5.1 through 4.3.5.3.

4.3.5.1 One paused transfer per facility

When Facility Selection is used by the master to issue Individual commands, a slave may have up to 16 concurrent data transfers being multiplexed. It is the master's responsibility to manage the multiplexing of information via selection and re-selection of the facility.

4.3.5.2 One paused transfer per slave

When Slave Selection is used by the master, a slave may be able to multiplex only one transfer (defined by Attributes). Continuing a paused transfer does not require the use of a Transfer Notification (unless enabled by Attributes) since the master can reissue the same Bus Control octet as for the initial transfer to indicate an implicit Transfer Notification.

4.3.5.3 Multiple paused transfers per slave

When Slave Selection is used by the master, a slave may be able to multiplex more than one transfer. This application requires the use of Transfer Notification packets unless a master remains selected. Upon selection, the slave shall present a Transfer Notification packet to advise the master which paused data transfer is going to be continued.

4.3.6 Anticipated pause

The master uses Attributes to advise the slave of the burst size to be transferred during a multiplexed transfer.

After the slave has terminated a burst data transfer, it posts Slave Status with the Successful and the requisite P/TDO bits.

- a) P/TDO=00 Immediate Continuation Possible. On Read, next burst is already buffered and ready to transfer. On Write, the buffer is free to accept next burst from master.
- b) P/TDO=10 Continuation Possible after Short Delay. The slave expects a delay of less than the period defined by the master in Attributes.
- c) P/TDO=11 Continuation Possible after Long Delay. The slave expects a delay longer than the period defined by the master in Attributes.
- d) P/TDO=01 No More Data. The entire data transfer is now complete.

The Time-Dependent delay posted by the slave is an indication, and not a guarantee, because many factors are involved (e.g., a Short Delay could actually be a long one if an unforeseen error condition occurred).

4.3.6.1 Master stays selected

If the master chooses to stay selected to the addressee, it continues by issuing a Bus Control octet with the same settings as on the original. If the slave cannot continue, it shall reject the Bus Control octet and use encoded Slave Status to advise why.

4.3.6.2 Master deselects

If the master deselects, the addressee shall assert its Class 2 Interrupt (and Attention if not overridden by ATTENTION CONTROL) when it is capable of continuing the burst transfer.

The actions to be taken are a function of the application (see 4.3.2.1 and 4.3.2.2).

4.3.7 Unanticipated pauses by the slave

If the slave encounters a condition that causes interruption of a data transfer (before reaching a burst boundary), it shall terminate the transfer and advise P/TDO=1x in the Slave Status octet (continuation possible after Delay).

NOTE — P/TDO=00 is invalid, because if it could continue immediately, it would not have paused in the first place.

The master may choose to remain selected or not. The same actions that are defined in 4.3.6 for anticipated pauses shall be followed by the slave and the master in order to continue the transfer.

4.3.8 Unanticipated pauses by the master

The master cannot pause more than one data transfer to the slave. The next operation to the same addressee has to be the continuation of the paused data transfer.

If a slave cannot accept a master-initiated Pause, it shall set the Not Successful bit in the Slave Status octet.

NOTE — Close examination of a vendor's specification is recommended because there are many conditions possible that could cause unpredictable results, or require extensive error recovery.

4.3.9 Multiplexed transfer mode identification

The different ways of multiplexing transfers are defined in Attributes. The various combinations shall be as shown in table 1.

Table 1 — Different Ways of Multiplexing Transfers

Who Paused	Who Continued	Attribute ID/Octet/Bit (Pause)	Settings ID (Other)
M	M	6B/4/5	
S	M	6B/5/5, 6B/5/4	66, 67
S	S		6B/2/4, 6B/3/4

4.4 Data groupings

The basic unit of recorded data is the eight-bit octet. Octets are combined to form a PhysicalBlock, which is the recording unit on the media, or may be combined to form a DataBlock, which is the master-defined unit of preference. Data transfer between the slave or facility and the master may be in either PhysicalBlocks or DataBlocks.

4.4.1 PhysicalBlocks

The PhysicalBlock size may be preset by the manufacturer of the device, determined by the slave, or by the master by means of the appropriate command (FORMAT for disks and OPERATING MODE or ATTRIBUTES for tapes). Once established, the PhysicalBlock size becomes an attribute of the facility until changed by another command. If a PhysicalBlock size has not been established by the master, the default value preset by the manufacturer shall be used by the facility.

Each PhysicalBlock on disks and on tapes that support the Update function shall be capable of being written (updated) by a write operation without requiring the master to access or read any adjacent PhysicalBlocks.

For magnetic disks, this is typically provided by separation of recorded fields. One PhysicalBlock may be updated without reading and rewriting any other PhysicalBlock contained in the physical space. If PhysicalBlocks are recorded in a field (e.g., sector) without intervening gaps, then an uncorrectable error in one PhysicalBlock may cause a data integrity exposure to one of the other PhysicalBlocks.

For magnetic tapes, if the Update function is not supported, all data beyond the last write shall be considered invalid.

4.4.2 DataBlocks

The DataBlock size may be predefined by the slave, or it may be set by the master via the appropriate command (FORMAT or ATTRIBUTES for disks and OPERATING MODE or ATTRIBUTES for tapes). If a DataBlock size has not been established by the master, the default value preset by the manufacturer shall be used for the facility.

For disks, a reset or loss of power from the slave or facility shall not cause the DataBlock size to change. Once established, the DataBlock size remains unchanged until a new FORMAT, ATTRIBUTES, or OPERATING MODE command is issued.

For tapes, a reset or loss of power from a slave or facility may cause the DataBlock size to change. The DataBlock size may need to be re-established by the master (see vendor specification).

4.4.3 Extents

An extent is a contiguous number of blocks beginning at a specified starting address. An extent may be as small as one block, or as large as the entire addressable area of a partition (or volume, if absolute addresses are used).

Extents are primarily used in association with transfer and positioning commands. However, they are also used to describe master-definable partitions.

4.4.4 Partitions

These are addressable areas on a volume, and may be either slave-defined or master-defined.

For disks, each partition is a contiguous range of blocks that begin on a physical boundary (on disks, this is typically a cylinder boundary).

For tapes, a partition is defined as a contiguous addressable area. On tapes that are not preformatted, no extent range is normally possible, and data shall be written from the beginning of a partition. If a tape uses preformatted addressable areas, then a contiguous range of blocks defines a partition.

ISO/IEC 9318-3 : 1990 (E)

4.4.4.1 Slave-defined partitions

4.4.4.1.1 Data partition

The slave shall define for each facility an addressable data area known as the default data partition, which is available to the master for data storage. There is only one default data partition per facility.

For tapes, the slave may define additional data partitions that define storage space in addition to the default data partition.

4.4.4.1.2 Maintenance partitions

The facility may include space that is dedicated for functions other than user data storage. These may include areas for assignment to describe media defects; diagnostic read and write operations; support functions, such as the storage of internal microcode routines; *et cetera*. The placement of non-user data storage areas and the algorithms for accessing them are implementation dependent and shall be described in the vendor specifications.

The slave-defined maintenance areas are available to the master via the OPERATING MODE command.

4.4.4.2 Master-defined partitions

4.4.4.2.1 Data partitions

The master may choose to subdivide the default data partition into smaller ones.

For disks, each partition, other than the default data partition, requires that a Partition ID be associated with any commands that address data (this may be implied by an Alias address).

For tapes, once a partition is addressed, the slave or facility shall continue processing in that partition until a new partition is chosen by use of a Partition Parameter, or until a volume is de-mounted (if the partition is on the volume). Subsequent commands after a partition change need not supply a partition ID.

4.4.4.2.2 Maintenance partitions

The master may have the need to define areas that are excluded from normal data accessing. Up to eight maintenance partitions may be defined by the master. These shall be allocated by the slave from the default data partition.

The master-defined maintenance areas are only accessible via the OPERATING MODE command.

4.4.5 Alternate data areas

The slave may map defective PhysicalBlocks to alternate blocks in order to create a defect-free data area for the master. If a PhysicalBlock has been assigned to an alternate data area, it shall be accessible during normal operations in a manner transparent to the master.

4.4.6 Physical groups

It is possible for a manufacturer to organize a disk with variable block sizes (as in IPI-2 Format 2 in ISO/IEC 9318-2) such that there is more than one PhysicalBlock per identification field. The identification field shall be read in order to access any of the PhysicalBlocks within the Physical Group.

4.5 Media addressing definitions

IPI disk transfer commands are multiple block transfers across physical boundaries. The addressee positions itself to the Data Address given in the Command packet, locates the block, and begins execution of the command. All read and write transfer operations utilize physical (PhysicalBlock) or logical (DataBlock) addresses, unless overridden by the Absolute Addressing modifier in the Data Address parameter. The absolute method of addressing is available for master-specific use when there is a need for a device-unique addressing mode of operation.

IPI tape transfer commands may be multiple block transfers across physical boundaries. Both explicit positioning (using a Data Address) and implicit positioning are supported. If a Data Address is supplied in the transfer command and the addressee supports explicit positioning for transfer commands, the addressee positions itself to the Data Address in the Command packet, and begins execution. Tapes that support implicit positioning begin execution at the next recorded element following the last operation. The command modifier determines which recorded element is next since direction may be changed between any two commands.

For tapes which support both types of positioning, the presence of a Data Address field in the Command Extent parameter indicates that the master intends explicit positioning; otherwise, implicit positioning is used.

4.5.1 Absolute addressing

Absolute addressing uniquely identifies a location by specifying address values that are implementation dependent (as on disks), or by using Position commands to properly position the media (as on tapes). On tape, positioning would typically be accomplished by using the SPACE BLOCK/FILE MARK command to reach the desired position.

For disks, typical values include cylinder address, head address, and sector address. Cylinder addresses run from 0 to C-1, where C is the number of tracks per disk surface; each possible value represents a particular cylinder. Head addresses run from 0 to H-1, where H is the number of read/write heads on the drive; each possible value represents a particular head. Sector addresses run from 0 to S-1, where S is the number of sectors per track; each possible value represents a definite sector, and the sectors are numbered consecutively.

For tapes, values may include track number, relative block number, *et cetera*. When used, track numbers are unique. Block numbers run from 0 to n and may be either PhysicalBlock or DataBlock values.

Absolute addresses are associated with the media, and references may occur across partition boundaries.

4.5.2 Physical addressing

Physical addressing uniquely identifies a location by specifying address values that are implementation dependent.

On disks, PhysicalBlocks use the same format (typically cylinder, head, sector) as Absolute addressing, but the PhysicalBlock address of a given sector may be different from that sector's absolute address. Factors leading to such differences include partitions, multiple PhysicalBlocks per sector, multiple PhysicalBlocks per identification field, defect mapping, and interleaving.

PhysicalBlock addresses are associated with the partition, and references cannot occur across partition boundaries.

On tapes, PhysicalBlocks use the same format as Absolute addressing (Blocks and File Marks), except that positioning may be explicit with Data Address or implicit (no Data Address). Alternatively, the SPACE BLOCK/FILE MARK command is used to reach the desired position on the media.

4.5.3 Logical addressing

For disks, logical addressing uses DataBlock addresses to reference data in partitions, within which all DataBlocks are the same size. DataBlocks in each partition are addressable through a linear address space numbered from 0 to $n-1$, where n is communicated to the master via Attributes.

For tapes, logical addressing uses DataBlock addresses to reference data in partitions. DataBlock sizes may vary within a partition, and are addressable through a linear address space. The DataBlock address may be augmented by a track reference to speed positioning, which is vendor specific.

Logical addressing permits the master to use DataBlocks that are a multiple or sub-multiple of PhysicalBlocks. It is the responsibility of the slave to manage the necessary blocking and unblocking of data, and the slave attributes shall identify whether or not it is capable of providing this function. There are many performance criteria that must be considered when DataBlock and PhysicalBlock sizes are different.

LogicalBlock addresses are associated with the partition, and references cannot occur across partition boundaries.

4.5.4 Media defect management considerations

Media defect avoidance is managed in IPI slaves and facilities using several types of entries in the defect list (see 11.3 and 11.4 for command implementation and annex C for a tabular representation of defect list management).

The facility manufacturer's flaw map is specified by the device-level interface. It is used to initialize the slave's Working Permanent defect entries the first time a facility is formatted.

For each facility, a slave shall retain information that allows it to avoid defective media in real time. The representation and use of this Working set of defects is slave specific. The combined contents of the Working Permanent and Working Temporary defect entries reflect the condition of the slave's Working set of defects.

Manufacturer's defect list entries in the Working Permanent defect list cannot be removed by any action of the master defined in the IPI command set. Working Temporary defect list entries are removed by a FORMAT command with the Initialize modifier set.

Suspect Permanent and Suspect Temporary entries are provided to contain the identity of defects between the time when they are identified to the slave via a WRITE DEFECT LIST command, and the actual substitution of replacement media during a FORMAT or REALLOCATE command. Suspect entries are moved to the corresponding Working type at that time.

Defects identified as Suspect entries are still in the master's DataBlock addressing space. If automatic reallocation is in effect, there are no Suspect defect entries.

Entries can be added to the Working Temporary defect list by specifying their addresses in defect list parameters of the FORMAT or REALLOCATE commands.

Any of the defect entries (Working Permanent, Working Temporary, Suspect Permanent, and Suspect Temporary) can be read using the Read Defect List command. Only Suspect Permanent and Suspect Temporary defect entries may be created or appended using the Write Defect List command.

The following example is intended to illustrate the condition of the various defect management structures in a hypothetical slave that implements all of the available features after the following actions:

- a) FORMAT the drive for the first time since its manufacture;
- b) Issue a WRITE DEFECT LIST command with the Permanent modifier set, and a parameter identifying address 5 as defective;
- c) Execute another FORMAT command;
- d) Issue a REALLOCATE command identifying addresses 50 and 150 as defective;