Draft Proposed American National Standard

X3T10 Project 1047D

Revision 6
20-July-95

Information technology - SCSI-3 Controller Commands (SCC)

This is a draft proposed American National Standard of Accredited Standards Committee X3. As such this is not a completed standard. The X3T10 Technical Committee may modify this document as a result of comments received during public review and its approval as a standard. Use of the information contained here in is at your own risk.

Permission is granted to members of X3, its technical committees, and their associated task groups to reproduce this document for the purposes of X3 standardization activities without further permission, provided this notice is included. All other rights are reserved. Any duplication of this document for commercial or for-profit use is strictly prohibited.

ASC X3T10 Technical Editor: George O. Penokie
IBM
Dept. 2B7
3605 Highway 52 N.
Rochester, MN 55901
USA

Telephone: 507-253-5208
Facsimile: 507-253-2432
Email: gop@rchvmp3.vnet.ibm.com

Reference number
ISO/IEC ***** : 199x
ANSI X3.276 - 199x

Printed Thursday, July 20, 1995 11:50 am

POINTS OF CONTACT:

X3T10 Chair
John B. Lohmeyer
Symbios Logic
1635 Aeroplaza Drive
Colo Spgs, CO 80916
Tel: (719) 573-3362
Fax: (719) 593-3037
Email: john.lohmeyer@symbios.com

X3T10 Vice-Chair
Lawrence J. Lamers
Adaptec
691 South Milpitas Blvd
Milpitus, CA 95035
Tel: (408) 975-7817
Fax: (408) 957-7193
Email: ljlamers@aol.com

X3 Secretariat
Lynn Barra
Administrator Standards Processing
X3 Secretariat
1250 Eye Street, NW Suite 200
Washington, DC 20005
Telephone: 202-626-5738
Facsimile: 202-638-4922
Email: lbarra@cbema.org

SCSI Reflector Internet address for subscription of the SCSI reflector: scsi-request@symbios.com
Internet address for distribution via SCSI reflector: scsi@symbios.com

SCSI Bulletin Board 719-574-0424

Document Distribution
Global Engineering
15 Inverness Way East
Englewood, CO 80112-5704
Telephone: 303-792-2181 or
800-854-7179
Facsimile: 303-792-2192

ABSTRACT

This standard defines commands for SCSI-3 storage array devices; commonly known as RAID devices. This standard is principally intended to be used in conjunction with, not as an alternate to, any of the SCSI-3 command standards and the SCSI-3 Architecture Model Standard. The resulting commands facilitate the control and configuration of SCSI-3 storage arrays and thus provide a common command specification for both system integrators and suppliers of SCSI-3 storage array devices.

PATENT STATEMENT

CAUTION: The developers of this standard have requested that holder's of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard. As of the date of publication of this standard, following calls for the identification of patents that may be required for the implementation of the standard, notice of one or more claims has been received.

By publication of this standard, no position is taken with respect to the validity of this claim or of any rights in connection therewith. The known patent holder(s) has (have), however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher.

No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

Contents

Page

Foreword x

0 Introduction xii

1 Scope 1

2 Normative references 2

3 Definitions, symbols, abbreviations, and conventions 2
3.1 Definitions 2
3.2 Symbols and abbreviations 5
3.3 Conventions 5

4 General 6

5 Models for systems containing arrays of devices 7
5.1 System layering model 7
5.1.1 SACL functions 7
5.1.2 Protocol conversion layer 8
5.1.3 Storage array conversion layer (SACL) 9
5.1.4 Examples of system layering variations 10
5.1.5 Branch of generic layers 11
5.1.6 Software SACL with a branch of SCSI disks 12
5.1.7 Branch with HBA SACL 13
5.1.8 Branch with bridge controller SACL 14
5.1.9 Branch with SACLs in multiple layers 15
5.2 Model for SCSI-3 storage arrays 16
5.2.1 SCSI storage array addressing 16
5.2.1.1 SCSI-3 storage array base address 16
5.2.1.2 Eight byte LUN structure 16
5.2.1.3 Component device address method 18
5.2.1.4 Logical unit address method 18
5.2.1.5 Peripheral device address method 19
5.2.1.6 Redundancy group address method 20
5.2.1.7 Spare address method 20
5.2.1.8 Volume set address method 20
5.2.2 SCSI-3 storage array objects 21
5.2.2.1 Adding objects 21
5.2.2.2 Association of objects 21
5.2.2.3 Attachment of objects 21
5.2.2.4 Covering of objects 22
5.2.2.5 Exchanging objects 22
5.2.2.6 Protected objects 22
5.2.2.7 Removing objects 22
5.2.2.8 Component device 23
5.2.2.9 Peripheral device 23
5.2.2.10 P_extent 23
5.2.2.11 Ps_extent 23
5.2.2.12 Redundancy group 24
5.2.2.12.1 No redundancy method of check data mapping 27
5.2.2.12.2 Copy redundancy method of check data mapping 27
5.2.2.12.3 XOR or P+Q redundancy method of check data mapping 27
5.2.2.12.4 Vendor specific redundancy method of check data mapping 27
5.2.2.13 Spares 27
5.2.2.14 Volume sets 28

5.2.3 SCSI-3 storage array operations 34
5.2.3.1 Rebuild operation 34
5.2.3.2 Recalculate operation 34
5.2.3.3 Regenerate operation 34
5.2.3.4 Verify operation 34
5.2.4 SCSI-3 storage array states 34
5.2.5 SCSI-3 storage array exception conditions 34

6 Commands for SCSI-3 storage array devices 36
6.1 MAINTENANCE(IN) command 37
6.1.1 MAINTENANCE(IN) command service actions 37
6.1.1.1 REPORT ASSIGNED/UNASSIGNED P_EXTENT service action 37
6.1.1.2 REPORT COMPONENT DEVICE service action 40
6.1.1.3 REPORT COMPONENT DEVICE ATTACHMENTS service 43
6.1.1.4 REPORT PERIPHERAL DEVICE service action 47
6.1.1.5 REPORT PERIPHERAL DEVICE ASSOCIATIONS service action 49
6.1.1.6 REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action 52
6.1.1.7 REPORT STATES service action 54
6.2 MAINTENANCE (OUT) commands 62
6.2.1 MAINTENANCE (OUT) command service actions 62
6.2.1.1 ADD PERIPHERAL DEVICE/COMPONENT DEVICE service action 63
6.2.1.2 ATTACH TO COMPONENT DEVICE service action 64
6.2.1.3 BREAK PERIPHERIAL DEVICE/COMPONENT DEVICE service action 65
6.2.1.4 EXCHANGE P_EXTENT service action 66
6.2.1.5 EXCHANGE PERIPHERAL DEVICE/COMPONENT DEVICE service action 68
6.2.1.6 INSTRUCT COMPONENT DEVICE service action 69
6.2.1.7 REMOVE PERIPHERAL DEVICE/COMPONENT DEVICE service action 70
6.2.1.8 SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action 71
6.3 REDUNDANCY GROUP (IN) command 73
6.3.1 REDUNDANCY GROUP (IN) command service actions 73
6.3.1.1 REPORT REDUNDANCY GROUPS service action 73
6.3.1.2 REPORT UNASSIGNED REDUNDANCY GROUP SPACE service action 77
6.4 REDUNDANCY GROUP (OUT) command 80
6.4.1 REDUNDANCY GROUP (OUT) command service actions 80
6.4.1.1 CONTROL GENERATION OF CHECK DATA service action 81
6.4.1.2 CREATE/MODIFY REDUNDANCY GROUP service action 82
6.4.1.3 DELETE REDUNDANCY GROUP service action 85
6.4.1.4 REBUILD P_EXTENT service action 86
6.4.1.5 REBUILD PERIPHERAL DEVICE service action 88
6.4.1.6 RECALCULATE CHECK DATA service action 91
6.4.1.7 VERIFY CHECK DATA service action 92
6.5 VOLUME SET (IN) command 95
6.5.1 VOLUME SET (IN) command service actions 95
6.5.1.1 REPORT VOLUME SETS service action 95
6.6 VOLUME SET (OUT) command 98
6.6.1 VOLUME SET (OUT) command service actions 98
6.6.1.1 CONTROL GENERATION OF CHECK DATA service action 99
6.6.1.2 CONTROL WRITE OPERATIONS service action 100
6.6.1.3 CREATE/MODIFY VOLUME SET service action 101
6.6.1.4 DELETE VOLUME SET service action 104
6.6.1.5 RECALCULATE VOLUME SET CHECK DATA service action 105
6.6.1.6 VERIFY VOLUME SET CHECK DATA service action 107
6.7 SPARE (IN) command 111
6.7.1 SPARE (IN) command service actions 111
6.7.1.1 REPORT P_EXTENT SPARE service action 111
6.7.1.2 REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action 114
6.8 SPARE (OUT) command 117

6.8.1 SPARE (OUT) command service actions 117
6.8.1.1 CREATE/MODIFY P_EXTENT SPARE service action 118
6.8.1.2 CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action 121
6.8.1.3 DELETE SPARE service action 122
6.9 Parameters for direct-access devices 124
6.9.1 Mode parameters 124
6.9.1.1 LUN mapping page 124

A.0 SCSI-3 storage array IDENTIFY message format 126
A.1 IDENTIFY message 126

B.0 SCSI-3 storage array addressing examples 127
B.1 Addressing Examples for the 8-byte LUN structure 127
B.2 Addressing Examples for the 6-bit LUN structure 130

C.0 Examples of check data and user data mappings 132
C.1 Example XOR redundancy mapping 132
C.2 User data mapping examples 133

Tables

Page
1 Addressing methods within a SCSI-3 storage array 16
2 Eight byte LUN structure adjustments 17
3 Eight byte LUN structure 17
4 FIRST LEVEL ADDRESSING field, SECOND LEVEL ADDRESSING field, THIRD LEVEL ADDRESSING field, and FOURTH LEVEL ADDRESSING field 18
5 ADDRESS METHOD 18
6 Logical unit addressing 19
7 Peripheral device addressing 19
8 Volume set addressing 20
9 Commands for SCSI storage array devices 36
10 Service actions for MAINTENANCE(IN) command 37
11 REPORT ASSIGNED/UNASSIGNED P_EXTENT service action 38
12 REPORT ASSIGNED/UNASSIGNED P_EXTENT parameter list 39
13 Data format of ASSIGNED/UNASSIGNED P_EXTENT DESCRIPTOR 39
14 Data format of P_EXTENT DESCRIPTOR 40
15 REPORT COMPONENT DEVICE service action 41
16 REPORT COMPONENT DEVICE parameter list 42
17 Data format of COMPONENT DEVICE DESCRIPTOR 42
18 COMPONENT DEVICE TYPES 43
19 REPORT COMPONENT DEVICE ATTACHMENTS service action 44
20 REPORT COMPONENT DEVICE ATTACHMENTS parameter list 45
21 Format of COMPONENT DEVICE ATTACHMENT DESCRIPTOR 46
22 Data format of LOGICAL UNIT DESCRIPTOR 46
23 LOGICAL UNIT types 47
24 REPORT PERIPHERAL DEVICE service action 47
25 SELECT REPORT 48
26 REPORT PERIPHERAL DEVICE parameter List 48
27 Format of PERIPHERAL DEVICE DESCRIPTOR 49
28 REPORT PERIPHERAL DEVICE ASSOCIATIONS service action 50
29 REPORT PERIPHERAL DEVICE ASSOCIATIONS parameter list 51
30 Format of PERIPHERAL DEVICE ASSOCIATIONS DESCRIPTOR 52
31 REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action 53
32 REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER parameter list 54
33 REPORT STATES service action 55
34 REPORT STATES 55
35 REPORT STATES parameter list 56
36 Format of LOGICAL UNIT STATES DESCRIPTORS 57
37 Target base device (LUN 0) states 58
38 Volume set states 59
39 Redundancy group states 60
40 Peripheral device and p_extent states 61
41 Spare states 61
42 Component device states 62
43 Service actions for MAINTENANCE (OUT) command 62
44 ADD PERIPHERAL DEVICE/COMPONENT DEVICE service action 63
45 ATTACH COMPONENT DEVICE service actions 64
46 ATTACH COMPONENT DEVICE parameter list 65
47 BREAK PERIPHERAL DEVICE/COMPONENT DEVICE service action 66
48 EXCHANGE P_EXTENT service action 67
49 EXCHANGE P_EXTENT parameters list 67
50 EXCHANGE PERIPHERAL DEVICE/COMPONENT DEVICE Service Action 68
51 INSTRUCT COMPONENT DEVICE service action 69
52 COMPONENT DEVICE INSTRUCTION field 69
53 INSTRUCT COMPONENT DEVICE parameter list 70
54 REMOVE PERIPHERAL DEVICE/COMPONENT DEVICE service action 70

55 SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action 71
56 SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER parameter list 72
57 Service actions for REDUNDANCY GROUP (IN) command 73
58 REPORT REDUNDANCY GROUPS service action 73
59 REPORT REDUNDANCY GROUPS parameter list 74
60 Format of REPORT REDUNDANCY GROUP DESCRIPTOR 75
61 REDUNDANCY GROUP IDENTIFIERS 75
62 GRANULARITY OF UNITS 76
63 REDUNDANCY GROUP P_EXTENT DESCRIPTOR 76
64 REPORT UNASSIGNED REDUNDANCY GROUP SPACE service action 77
65 REPORT UNASSIGNED REDUNDANCY GROUP SPACE parameter list 78
66 Format of REPORT UNASSIGNED REDUNDANCY GROUP SPACE DESCRIPTOR 79
67 Data format of PS_EXTENT DESCRIPTOR 80
68 Service actions for REDUNDANCY GROUP (OUT) command 81
69 CONTROL GENERATION OF CHECK DATA service action 81
70 CREATE/MODIFY REDUNDANCY GROUP service action 82
71 CREATE/MODIFY REDUNDANCY GROUP parameter list 83
72 Data format of CREATE/MODIFY P_EXTENT DESCRIPTOR 84
73 DELETE REDUNDANCY GROUP service action 85
74 REBUILD P_EXTENT service action 86
75 Rebuild types 87
76 REBUILD P_EXTENT parameter list 88
77 REBUILD PERIPHERAL DEVICE service action 89
78 Rebuild types 90
79 REBUILD PERIPHERAL DEVICE parameter list 91
80 RECALCULATE CHECK DATA service action 92
81 VERIFY CHECK DATA service action 93
82 Service actions for volume set (in) command 95
83 REPORT VOLUME SETS service action 95
84 REPORT VOLUME SETS parameter list 96
85 Format of REPORT VOLUME SET DESCRIPTOR 97
86 VOLUME SET PS_EXTENT DESCRIPTOR 98
87 Service actions for VOLUME SET (OUT) command 99
88 CONTROL GENERATION OF CHECK DATA service action 99
89 CONTROL WRITE OPERATIONS service action 101
90 CREATE/MODIFY VOLUME SET service action 102
91 CREATE/MODIFY VOLUME SET parameter list 103
92 Data format of CREATE/MODIFY PS_EXTENT DESCRIPTOR 104
93 DELETE VOLUME SET service action 105
94 RECALCULATE VOLUME SET CHECK DATA service action 106
95 RECALCULATE VOLUME SET CHECK DATA parameter list 107
96 VERIFY VOLUME SET CHECK DATA service action 108
97 VERIFY RANGE 109
98 VERIFY VOLUME SET CHECK DATA parameter list 110
99 Service actions for SPARE (IN) command 111
100 REPORT P_EXTENT SPARE service action 111
101 REPORT P_EXTENT SPARE parameter list 112
102 Format of REPORT P_EXTENT SPARE DESCRIPTOR 113
103 REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action 114
104 REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE parameter list 115
105 Format of REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE DESCRIPTOR 116
106 Data format of covered LOGICAL UNIT DESCRIPTOR 117
107 LOGICAL UNIT TYPES 117
108 Service actions for spare (out) command 118
109 CREATE/MODIFY P_EXTENT SPARE service action 118
110 CREATE/MODIFY P_EXTENT SPARE parameter list 120
111 CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action 121

112 CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE parameter list 122
113 DELETE SPARE service action 123
114 Mode page codes 124
115 LUN mapping page 125
A.1 IDENTIFY message format 126

Figures

Page
1 SCSI-3 document road map 1
2 Protocol conversion layer 8
3 SACL conversion layer 9
4 Typical system diagram 10
5 Branch of generic layers 11
6 Software SACL with a branch of SCSI disks 12
7 Branch with HBA SACL 13
8 Branch with bridge controller SACL 14
9 Branch with SACLs in multiple layers 15
10 Single redundancy group 24
11 Multiple volume sets associated with a single redundancy group 25
12 Redundancy group check data mapping flow chart 26
13 Multiple redundancy groups 29
14 Single volume set associated with multiple redundancy groups 29
15 Volume set user data mapping flow chart (part 1) 31
16 Volume set user data mapping flow chart (part 2) 32
17 Volume set user data mapping flow 33
C.1 XOR redundancy mapping example 132
C.2 User data mapping for a RAID 5 configuration 133
C.3 User data mapping for a RAID 3 configuration 134

Foreword

This foreword is not part of ANSI X3.276 - 199x.

The SCSI-3 Controller Commands (SCC) standard is divided into six clauses:

- Clause 1 is the scope;

- Clause 2 enumerates the normative references that apply to this standard;

- Clause 3 describes the definitions, symbols and abbreviations used in this standard;

- Clause 4 describes the conceptual relationship between this document and the SCSI-3 Architecture Model;

- Clause 5 describes the command model for SCSI-3 storage array devices;

- Clause 6 defines the commands that may be implemented by an SCSI-3 storage array device.

The annexes provide information to assist with implementation of the SCSI-3 Controller Commands standard. Annex A is normative and is considered part of this standard. Annexes B and C are for information only.

This standard was developed by a joint effort between the X3T10 SCSI Controller Commands working group and the RAID Advisory Board Host Interface working group during 1992-94. The standards approval process started in 1995.

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the X3 Secretariat, Computer and Business Equipment Manufacturers Association, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Processing Systems, X3. Committee approval of the standard does not necessarily imply that all committee members voted for approval.

The joint X3T10 SCSI Controller Commands working group/RAID Advisory Board Host Interface working group, which developed this standard, has the following members:

George O. Penokie, Chair

Dal Allan
John Baudrexl
Charles Binford
Paul Boulay
Allen Cole
Bill Dallas
Peter Dougherty
David Fellinger
Mike Fitzpatrick
Giles Frazier
Jerry Fredin
Bill Galloway
Mike Glass
Howard Grill
Randy Hall
Doug Hagerman
Ed Haske
Andrew Hill
Gerry Houlder
Bill Hutchinson
M. Jibbe
Gerry Johnsen
Larry Lamers
Paul Massiglia
Dick McCormick
Gene Milligan
Joe Molina
David Reilly
Herb Silverman
David Skinner
Bob Snively
Bob Solomon
George Su
Gary Watson
Ralph Weber
Jeffery Williams

Technical Committee X3T10 on Lower Level Interfaces, which approved this standard, had the following members:

John B. Lohmeyer, Chair
Lawrence J. Lamers, Vice-Chair
Ralph O. Weber, Secretary

I. Dal Allan
Paul D. Aloisi
Ron Apt
Geoffrey Barton
Robert Bellino
Charles Brill
Peter Brown
Michael Bryan
Joe Chen
Chris D'Iorio
Joe Dambach
Jan V. Dedek
Stephen G. Finch
Edward Fong
Louis Grantham
Norm Harris
Edward Haske
Dennis R. Haynes
Stephen F. Heil
Stephen Holmstead
David Hudson
Peter Johansson
Gerry Johnsen
Skip Jones
Edward Lappin
Joe Lawlor
David Lawson
Robert Liu
Bob Masterson
David McFadden
James McGrath
Pete McLean
Patrick Mercer
Gene Milligan
Charles Monia
Dennis P. Moore
Ian Morrell
John Moy
S. Nadershahi
Erich Oetting
Alan R. Olson
Dennis Pak
Duncan Penman
George Penokie
Doug Piper
Donna Pope
Robert Reiseh
Scott Smyers
Robert N. Snively
Jeff Stai
Gary R. Stephens
Clifford E. Strang Jr.
Thomas 'Rick' Tewell
Dean Wallace
Harvey Waltersdorf
Gary M. Watson
Michael Wingard
Mark Woithe
Ezra Alcudia (Alt.)
Michael Alexenko (Alt.)
Steven A. Anderson (Alt.)
David Andreatta (Alt.)
Tak Asami (Alt.)
Akram Atallah (Alt.)
Paul Boulay (Alt.)
Kevin Calvert (Alt.)
John Cannon (Alt.)
Kurt Chan (Alt.)
Shufan Chan (Alt.)
Ting Li Chan (Alt.)
Mike Chenery (Alt.)
Nancy Cheng (Alt.)
William Clemmey (Alt.)
Dan Colegrove (Alt.)
Roger Cummings (Alt.)
Zane Daggett (Alt.)
William Dallas (Alt.)
Varouj Der-Hacopian (Alt.)
Dhiru N. Desai (Alt.)
Mike Eneboe (Alt.)
Timothy Feldman (Alt.)
Edward A. Gardner (Alt.)
John Geldman (Alt.)
Chuck Grant (Alt.)
Peter Haas (Alt.)
Douglas Hagerman (Alt.)
Kenneth J. Hallam (Alt.)
William Ham (Alt.)
Tom Hanan (Alt.)
Rick Heidick (Alt.)
Mike Hetzel (Alt.)
Gerald Houlder (Alt.)
Paul Jackson (Alt.)
Kevin James (Alt.)
Brian Johnson (Alt.)
Mark Jordan (Alt.)
Richard Kalish (Alt.)
Greg Kapraun (Alt.)
Thomas J. Kulesza (Alt.)
Dennis Lang (Alt.)
Bill Mable (Alt.)
Gerald Marazas (Alt.)
John Masiewicz (Alt.)
Daniel E. Moczarny (Alt.)
E.J. Mondor (Alt.)
Jay Neer (Alt.)
Tim Norman (Alt.)
Vit Novak (Alt.)
Kevin R. Pokorney (Alt.)
Gary Porter (Alt.)
Doug Prins (Alt.)
Steven Ramberg (Alt.)
Ron Roberts (Alt.)
John P. Scheible (Alt.)
J. R. Sims (Alt.)
Michael Smith (Alt.)
Allen Spalding (Alt.)
Arlan P. Stone (Alt.)
Joe Stoupa (Alt.)
George Su (Alt.)
Nicos Syrimis (Alt.)
Pete Tobias (Alt.)
Adrienne Turenne (Alt.)
Joseph Wach (Alt.)
Roger Wang (Alt.)
Dave Weber (Alt.)
Bob Whiteman (Alt.)
Jeffrey L. Williams (Alt.)
Kurt Witte (Alt.)
Devon Worrell (Alt.)
Charles I. Yang (Alt.)
Danny Yeung (Alt.)
Mike Yokoyama (Alt.)

0 Introduction

The SCSI command set is designed to provide efficient peer-to-peer operation of input/output devices (disks, tapes, printers, etc.) by an operating system, The SCSI command set assumes a command-response protocol. Action on SCSI commands shall not be deemed completed until a response is received. The response shall include a status that indicates the final disposition of the command.

The SCSI command set provides multiple operating systems concurrent control over one or more input/output devices. However, the multiple operating systems must properly coordinate their actions or data corruption will result. This standard defines commands that assist with coordination between multiple operations systems. However, details of the coordination are beyond the scope of the SCSI command set.

This standard defines a device model for SCSI-3 storage arrays. This standard defines the SCSI commands that may apply to SCSI-3 storage arrays and the SCSI commands that are uniquely for SCSI-3 storage arrays.

With any technical document there may arise questions of interpretation as new products are implemented, The X3 Committee has established procedures to issue technical opinions concerning the standards developed by the X3 organization. These procedures may result in SCSI Technical Information Bulletins being published by X3.

Any such bulletins, while reflecting the opinion of the Technical Committee that developed the standard, are intended solely as supplementary information to other users of the standard. This standard, ANSI X3.276 - 199x, as approved though the publication and voting procedures of the American National Standards Institute, is not altered by these bulletins. Any subsequent revision to this standard may or may not reflect the contents of any such Technical Information Bulletins.

Current X3 practice is to make Technical Information Bulletins available through:

Global Engineering	Telephone:	303-792-2181 or
15 Inverness Way East		800-854-7179
Englewood, CO 80112-5704	Facsimile:	303-792-2192

1 Scope

This standard defines the command set extensions to facilitate operation of SCSI-3 storage array devices. The clause(s) of this standard pertaining to the SCSI-3 storage array device class, implemented in conjunction with the applicable clauses within any of the SCSI-3 command standards, shall specify the standard command set available for SCSI-3 storage arrays.

The objectives of the SCSI-3 Controller Commands is to provide the following:

a) Transfer commands unique to SCSI-3 Controller Command devices;
b) Control commands to manage the operation of an SCSI-3 Controller Command device;
c) Optional device mapping and pass-through support.

Figure 1 is intended to show the relationship of this document to other SCSI-3 standards. The figure is not intended to imply a relationship such as a hierarchy, protocol stack, or system architecture. It indicates the applicability of a standard to the implementation of a given transport.

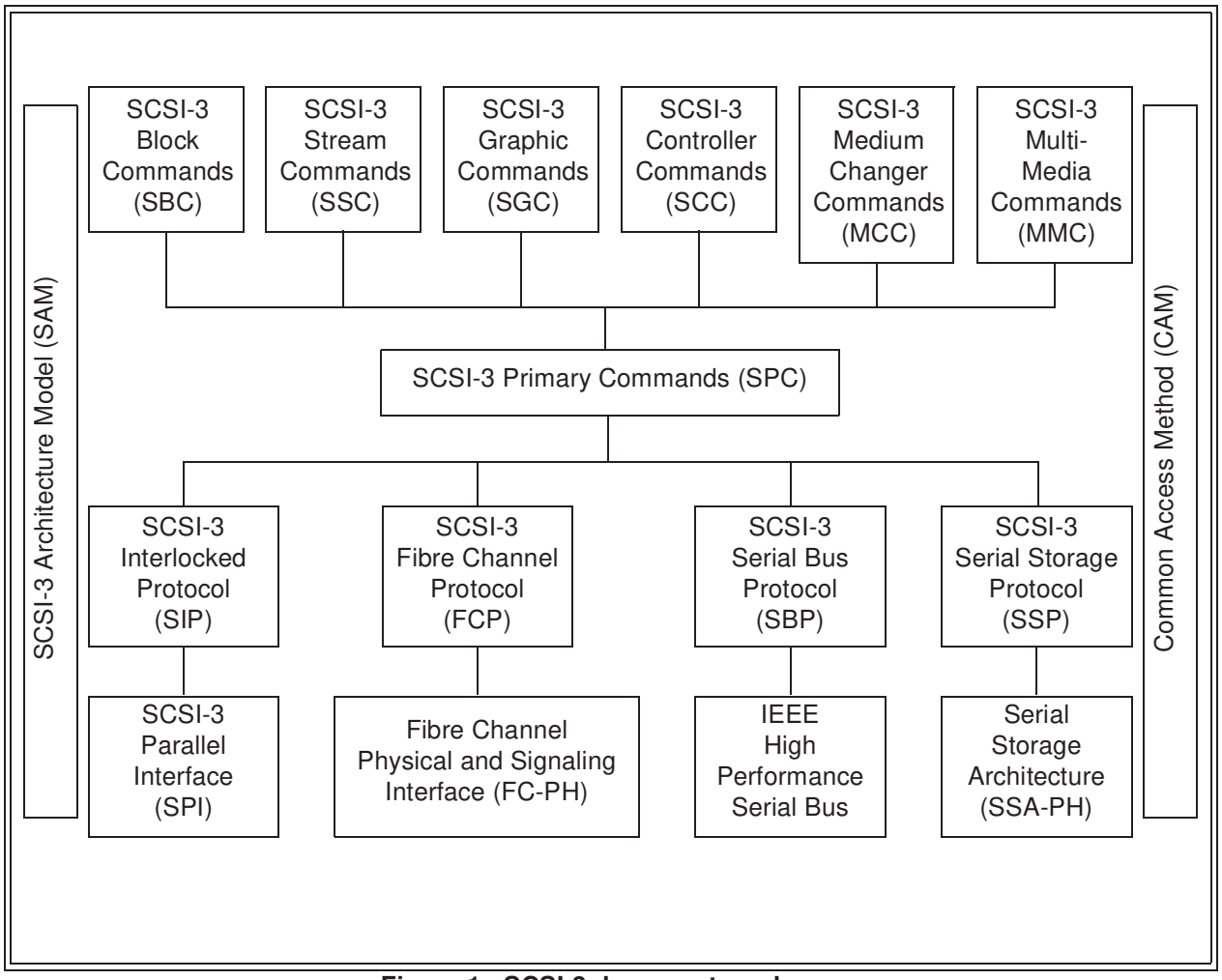

Figure 1 - SCSI-3 document road map

The term SCSI is used wherever it is not necessary to distinguish between the versions of SCSI. The Small Computer System Interface - 2 (ANSI X3.131-1994), is referred to herein as SCSI-2.

The term SCSI-3 refers collectively to the following documents:

SCSI-3 Parallel Interface Standard [X3T10/855D]

SCSI-3 Interlocked Protocol Standard [X3T10/856D]
SCSI-3 Fiber Channel Protocol Standard [X3T10/993D]
SCSI-3 Serial Bus Protocol Standard [X3T10/992D]

SCSI-3 Architecture Model Standard [X3T10/994D]

SCSI-3 Primary Commands Standard [X3T10/995D]

SCSI-3 Block Commands Standard [X3T10/996D]
SCSI-3 Stream Commands Standard [X3T10/997D]
SCSI-3 Graphic Commands Standard [X3T10/998D]
SCSI-3 Medium Changer Commands Standard [X3T10/999D]
SCSI-3 Multi-Media Commands Standard [X3T10/1048D]

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this American National Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

Members of IEC and ISO maintain registers of currently valid standards.

SCSI-3 Architecture Model Standard, X3T10/994D

SCSI-3 Primary Commands Standard, X3T10/995D

SCSI-3 Block Commands Standard, X3T10/996D
SCSI-3 Stream Commands Standard, X3T10/997D

SCSI-2 Common Access Method Transport and SCSI Interface Module Standard, X3T10/792D

SCSI-3 Parallel Interface Standard, X3T10/855D

SCSI-3 Interlocked Protocol Standard, X3T10/856D
SCSI-3 Fiber Channel Protocol Standard, X3T10/993D
SCSI-3 Serial Bus Protocol Standard, X3T10/992D
SCSI-3 Graphic Commands Standard, X3T10/998D
SCSI-3 Medium Changer Commands Standard, X3T10/999D
SCSI-3 Multi-Media Commands Standard, X3T10/1048D

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

3.1.1 application client: An object that is the source of SCSI commands. Further definition of an application client can be found in the SCSI-3 Architecture Model Standard.

3.1.2 association: The linking of SCSI-3 storage array objects in a manner explicitly defined within this standard.

3.1.3 attachment: The linking of SCSI-3 storage array objects in a vendor specific manner.

3.1.4 base address: The LUN that an application client addresses to configure an SCSI-3 storage array and to determine information about an SCSI-3 storage array and the logical units attached to it.

3.1.5 byte: Indicates an 8-bit construct.

3.1.6 check data: Information contained within a redundancy group that allows lost or destroyed user data to be recreated.

3.1.7 check data mapping: The distribution of the check data within a redundancy group.

3.1.8 command: A request describing a unit of work to be performed by a device server. See the SCSI-3 Architecture Model Standard for a detailed definition of a command.

3.1.9 command descriptor block: The structure of up to 16 bytes used to communicate commands from an initiator to a target.

3.1.10 component device: . Any physical addressable component not identifiable as an SCSI-3 peripheral device type. See table 18 for a list of components.

3.1.11 configuration: A collection of SCSI-3 storage array objects that follow the rules defined within this standard. For a list of SCSI-3 storage array objects see 5.2.2.

3.1.12 covering: The linking of spare objects to other SCSI-3 storage array objects in a manner explicitly defined within this standard.

3.1.13 field: A group of one or more contiguous bits.

3.1.14 initiator: An SCSI device containing application clients that originate device service requests to be processed by a target SCSI device. See the SCSI-3 Architecture Model Standard for a detailed definition of an initiator.

3.1.15 invalid: An illegal or unsupported field or code value.

3.1.16 groups: Objects that are independent from one another that may overlay one another.

3.1.17 logical block address: An address of a unit of data supplied or requested by an initiator.

3.1.18 logical unit: An externally addressable entity within a target that implements an SCSI device model. See the SCSI-3 Architecture Model Standard for a detailed definition of a logical unit.

3.1.19 logical unit identifier: An object that is part of the SCSI-3 Architecture Model Standard definition of a logical unit. A logical unit identifier uniquely identifies a logical unit in a SCSI domain. See the SCSI-3 Architecture Model Standard for a detailed definition of SCSI domain and logical unit identifier.

3.1.20 logical unit number: An identifier for a logical unit.

3.1.21 mandatory: The referenced item is required to claim compliance with this standard.

3.1.22 one: A true signal value or a true condition of a variable.

3.1.23 optional: The referenced item is not required to claim compliance with this standard. Implementation of an optional item shall be as defined in this standard.

3.1.24 page: Several commands use regular parameter structures that are referred to as pages. These pages are identified with a value known as a page code.

3.1.25 p_extent: All or part of the host addressable space within a single peripheral device of a SCSI-3 storage array.

3.1.26 peripheral device: Any addressable device identifiable as a SCSI-3 peripheral device type. See the IDENTIFY COMMAND description in the SCSI-3 Primary Commands Standard for the list of SCSI-3 peripheral device types.

3.1.27 protected space: The portion of a redundancy group that does not contain check data.

3.1.28 ps_extent: All or part of the protected space within a single peripheral device configured as a redundancy group.

3.1.29 rebuild operation: Re-creation of protected space contents or any check data within a p_extent using check data and protected space contents from the remaining p_extents within the redundancy group.

3.1.30 recalculate operation: Re-creation of check data from protected space contents.

3.1.31 redundancy group: A grouping of protected space and associated check data (Check data may be null) into a single logical unit that shall only have a single type of redundancy.

3.1.32 regenerate operation: Re-creation of inaccessible protected space contents from accessible check data and protected space contents.

3.1.33 reserved: Identifies bits, fields, and code values that are set aside for future standardization.

3.1.34 SCSI-3 storage array: A target that processes SCSI command descriptor blocks and performs the services of a SACL. A single SCSI-3 storage array may contain multiple SACLs.

3.1.35 service action: A request describing a unit of work to be performed by a device server. A service action is an extension of a command. See the SCSI-3 Architecture Model Standard for a detailed definition of a command.

3.1.36 set: Objects that do not intersect and are independent from one another. Sets may span more than one device. A single device may contain more than one set or may contain the entire set.

3.1.37 spare: A range of logical block addresses, a component device, or a peripheral device covered by one or more redundancy groups, component devices, or peripheral devices that can be used to replace all or part of a redundancy group or a peripheral device or all of a component device.

3.1.38 storage array conversion layer (SACL): Converts input logical unit numbers to output logical unit numbers and may convert input logical block addresses to output logical block addresses.

3.1.39 stripe: All or part of a volume set that is bounded by a number of contiguous units within a single ps_extent and by a number of ps_extents.

3.1.40 target: In this standard a target refers to an SCSI-3 storage array device that performs an operation requested by an application client.

3.1.41 underlying redundancy group: The portion of a redundancy group that contains protected space that has been mapped to specific volume set(s).

3.1.42 unit: A standard basic quantity in bits, bytes, words, logical blocks, etc., specified by the GRANULARITY OF UNITS field (see table 62).

3.1.43 user data: The addressable logical blocks that are input to the SACL. Check data is not part of the addressable logical blocks.

3.1.44 user data mapping: The distribution of user data within a volume set.

3.1.45 vendor-specific: Something (e.g., a bit, field, code value, etc.) that is not defined by this standard and may be used differently in various implementations.

3.1.46 verify operation: Re-creation of check data from protected space contents and the comparison of the recreated check data with the current check data.

3.1.47 volume set: One or more ps_extents grouped into a single LUN_V.

3.1.48 zero: A false signal value or a false condition of a variable.

3.2 Symbols and abbreviations

HBA	Host bus adapter
ITTU	I'm talking to you
LBA_P	Peripheral device logical block address
LBA_PS	Protected space logical block address
LBA_V	Volume set logical block address
LSB	Least significant bit
LUI	Logical unit identifier
LUN	Logical unit number
LUN_C	Component device logical unit number
LUN_P	Peripheral device logical unit number
LUN_R	Redundancy group logical unit number
LUN_S	Spare logical unit number
LUN_V	Volume set logical unit number
MSB	Most significant bit
SCSI	Either SCSI-2 or SCSI-3
SCSI-2	The Small Computer System Interface - 2 (ANSI X3.131)
SCSI-3	The Small Computer System Interface - 3
SIM	SCSI interface module

3.3 Conventions

Certain words and terms used in this American National Standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in 3.1 or in the text where they first appear, Names of messages, commands, statuses, sense keys, additional sense codes, and additional sense code qualifiers are in all uppercase (e.g., REQUEST SENSE), names of fields are in small uppercase (e.g., STATE OF SPARE), lower case is used for words having the normal English meaning.

Fields containing only one bit are usually referred to as the name bit instead of the name field.

Field names are in SMALL CAPS to distinguish them from normal English.

Numbers that are not immediately followed by lower-case b or h are decimal values.

Numbers immediately followed by lower-case b (xxb) are binary values.

Numbers immediately followed by lower-case h (xxh) are hexadecimal values

4 General

This standard defines a device model for SCSI-3 storage arrays and the SCSI commands that apply to SCSI-3 storage arrays. This standard assumes all interconnects between devices are SCSI interconnects.

The SCSI command set assumes a command-response protocol. The fundamental properties of the command-response protocol are defined in the SCSI-3 Architecture Model Standard. In accordance with the SCSI-3 Architecture Model Standard, the command-response protocol can be modelled as a procedure call, specifically:

Service response = execute command (task identifier, command descriptor block, [data-out buffer], task attributes, || [data-in buffer], [autosense data], [autosense return flag], status).

The SCSI-3 Architecture Model Standard defines all of the inputs and outputs in the procedure call. As they may apply to any SCSI device, this standard defines the contents of the following procedure call inputs and outputs; command descriptor block, data-out buffer, data-in buffer, and autosense data. This standard does not define all possible instances of these procedure inputs and outputs. This standard defines only those instances that may apply to SCSI-3 storage array devices.

This standard references values returned via the status output parameter. Examples of such status values are CHECK CONDITION and COMMAND TERMINATED. Status values are not defined by this standard, The SCSI-3 Architecture Model Standard defines all status values.

The entity that makes the procedure call from an initiator is an application client, as defined by the SCSI-3 Architecture Model Standard. The procedure call's representation arrives at the target in the form of a device service request. The entity that performs the work of the procedure call in a target is a device server. A device server is an object within a logical unit and is defined by the SCSI-3 Architecture Model Standard.

5 Models for systems containing arrays of devices

The first part of this clause defines a system layering model that uses the concept of SACLs to control arrays of devices. The second part of this clause defines the model for SCSI-3 storage array devices. The model assumes all the SCSI peripheral devices controlled within a SCSI-3 storage array are either fixed block or variable block devices.

5.1 System layering model

5.1.1 SACL functions

A SACL initiates several functions when an application client requests a media access. The type of media access and the configuration in effect determine which functions are used. The following are the functions available within a SACL:

a) Translation of input logical unit identifiers to output logical unit identifiers;
b) Translation of input logical block addresses to output logical block addresses;
c) Reading data from and writing data to locations based on the configuration in effect for the addressed volume set;
d) Calculating and updating the check data (if any);
e) Regeneration of protected space contents within the volume set using check data information or duplicate data;
f) Rebuilding of protected space contents associated with the redundancy group and(or) check data within the redundancy group using the contents of the redundancy group;
g) Recalculation of the check data within a redundancy group;
h) Recalculation of the check data within any redundancy group underlaying a volume set;
i) Determining when a p_extent should be disabled and/or replaced;
j) Returning a confirmation to the application client as to the success or failure of a request and, in the case of a failure, gives possible corrective actions.

5.1.2 Protocol conversion layer

A system is typically composed of many protocol conversion layers (see figure 2), and these layers may exist in hardware or software. Each of these layers has input(s) and output(s). The next layer accessed is determined by the preceding layer's output.

These protocol conversion layers include, but are not limited to: transport modules, host adapter drivers, SIMs (SCSI interface module), HBAs (host bus adapter), bridge controllers, and storage drives. In this model each of these layers will be represented by a simple block that has an input and output.

All requests to or from a SACL contain logical unit identifiers but not all requests contain logical block addresses.

NOTE 1 - The logical unit identifier is defined in the SCSI-3 Architecture Model Standard. SIMs and HBAs are defined in the SCSI-2 Common Access Method Transport and SCSI Interface Module Standard.

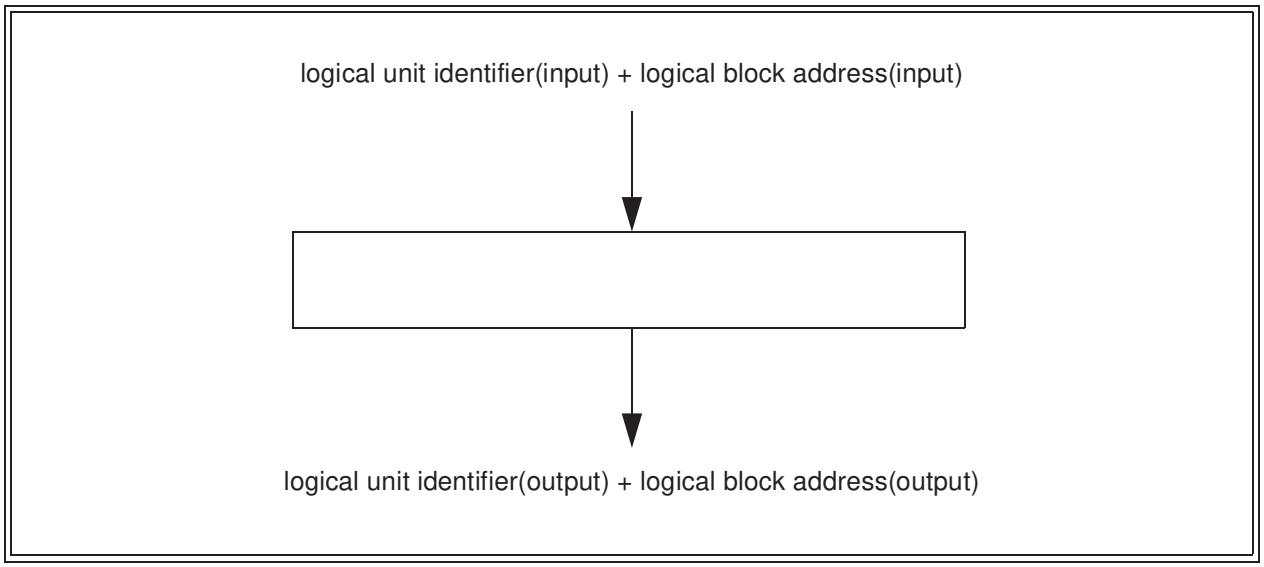

Figure 2 - Protocol conversion layer

Generic Layers do not modify the logical unit identifier or the logical block address.

NOTE 2 - There are types of layers other than generic and SACLs, however, these are not covered in this model.

5.1.3 Storage array conversion layer (SACL)

The SACL is capable of extensive manipulation on the logical unit identifier and the logical block address, based upon a consistent algorithm that follows the defined configuration. It is possible that a single logical unit identifier input or logical block address input may be converted to multiple different logical unit identifier outputs and logical block address outputs. In the following figures (figure 3, figure 4, and figure 6 through figure 9) SACLs will be shown with 'SACL' in the block.

The model does not require a one to one correspondence between logical unit identifiers and SCSI devices.

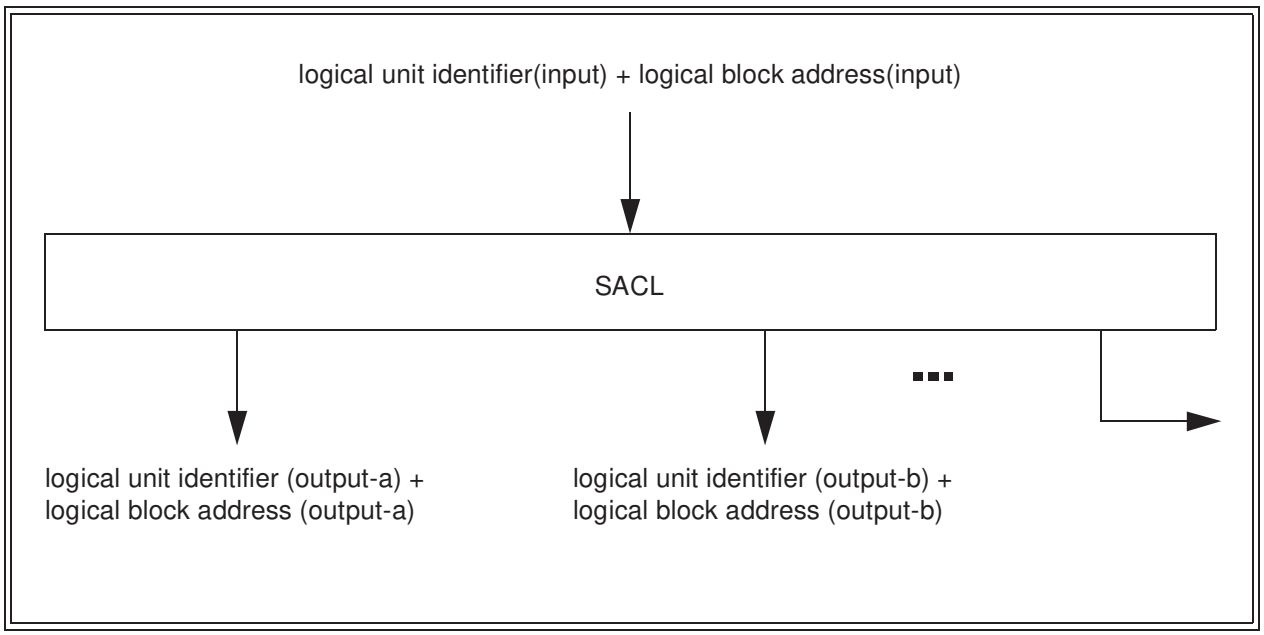

Figure 3 - SACL conversion layer

5.1.4 Examples of system layering variations

Typically a system diagram will be composed of many layers combined into a tree. For example, a driver may connect to multiple HBAs, which in turn may connect to multiple SCSI devices, etc. See figure 4 for an example of a system that consists of:

a) one initiator that has two SCSI devices attached on a single SCSI bus that is not expandable;
b) one initiator has two SCSI devices attached on a single SCSI that is expandable. One of the SCSI devices contains a SACL;
c) the SCSI device that contains the SACL has three SCSI buses with SCSI devices attached and is capable of driving more SCSI buses:
   a) two of the SCSI buses contain two SCSI devices each and these SCSI buses are not expandable;
   b) one of the SCSI buses contains one SCSI device and is expandable.

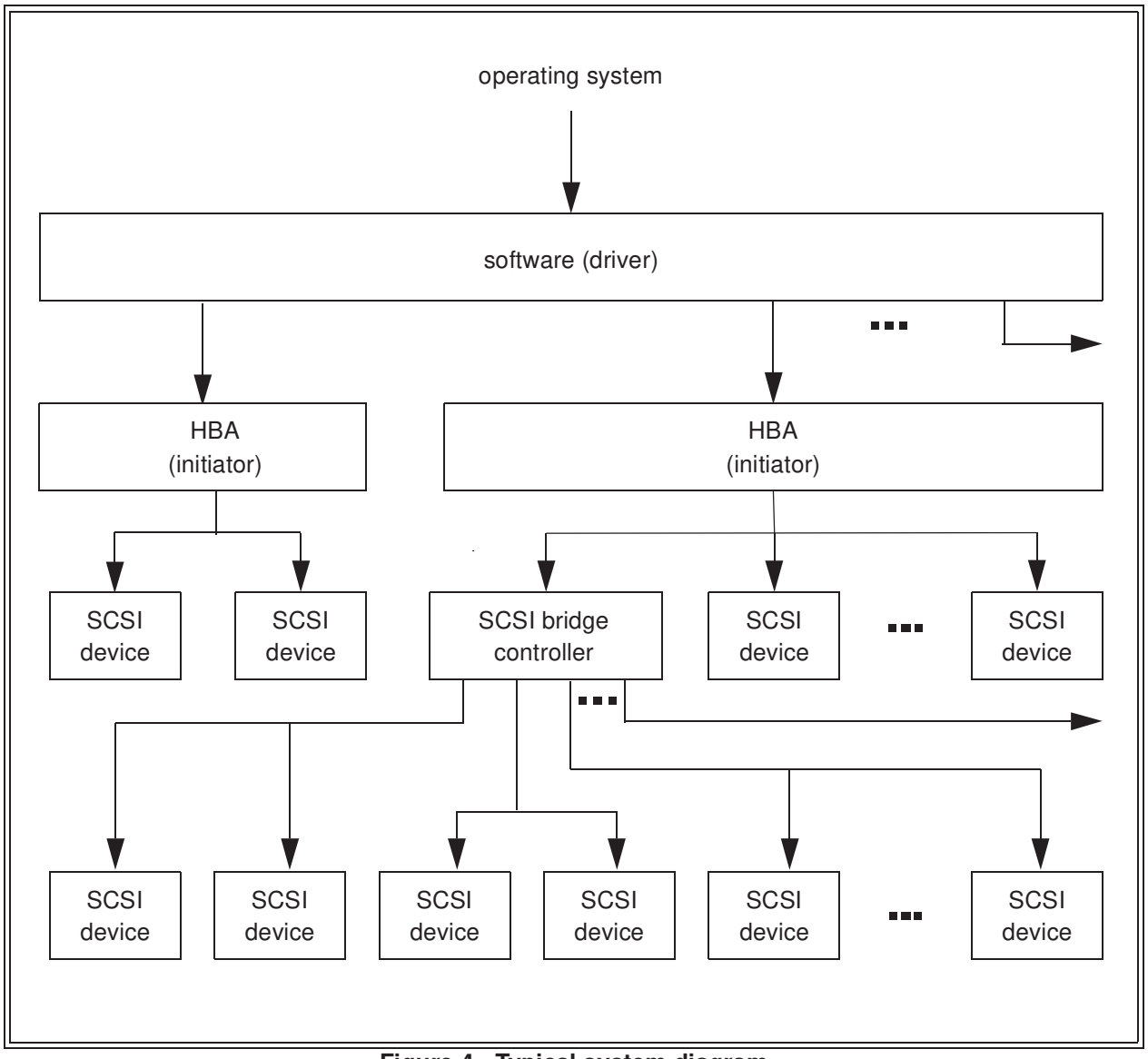

Figure 4 - Typical system diagram

5.1.5 Branch of generic layers

Figure 5 shows a system that does not contain any SCSI-3 storage arrays. In such a system all layers pass the logical unit identifier and logical block addresses directly through.

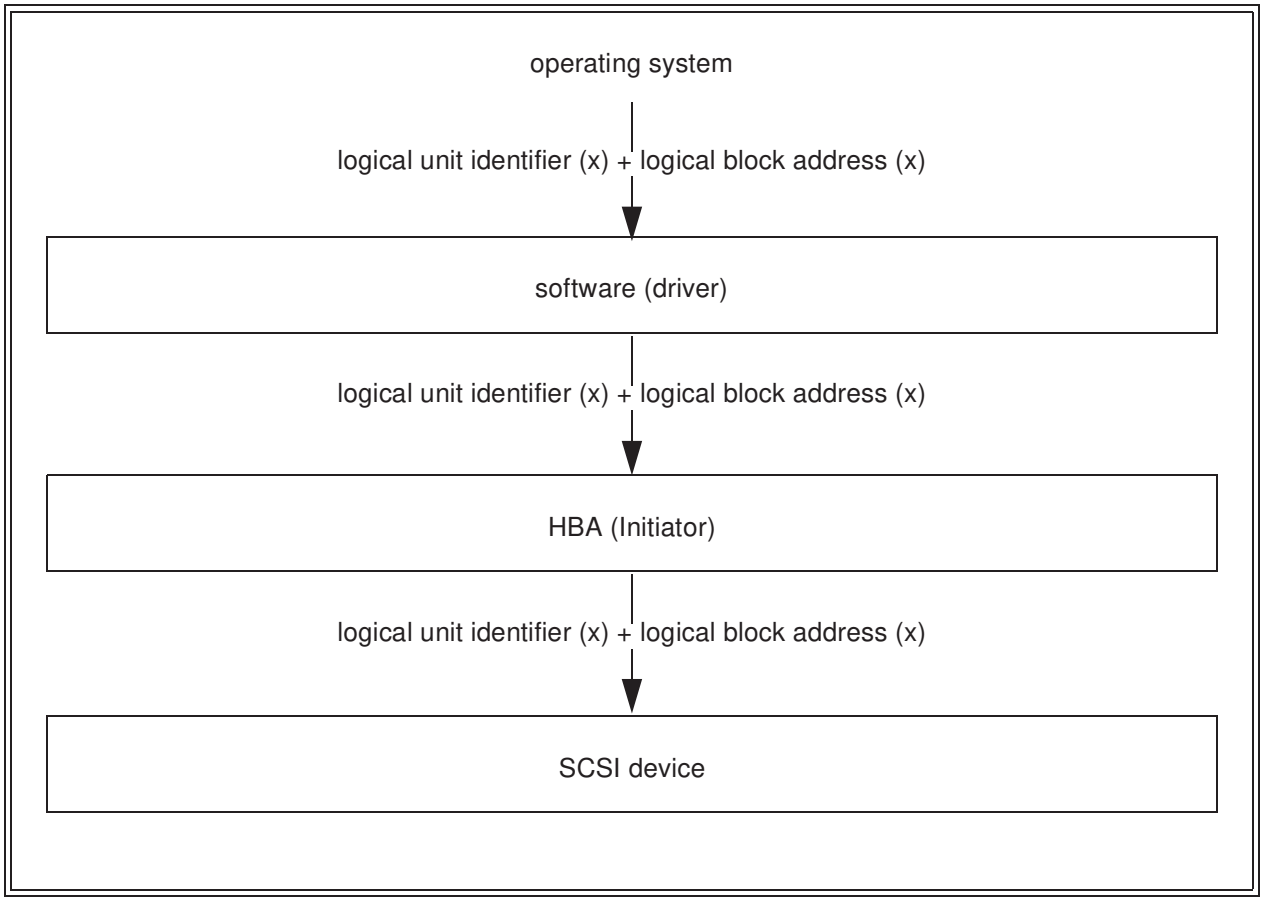

Figure 5 - Branch of generic layers

5.1.6 Software SACL with a branch of SCSI disks

Figure 6 shows system software performing SACL functions. These functions convert the input logical unit identifier(X) and the input logical block address(X) to the output logical unit identifier(Y) and output logical block address(Y). All other layers pass the logical unit identifier and logical block address through.

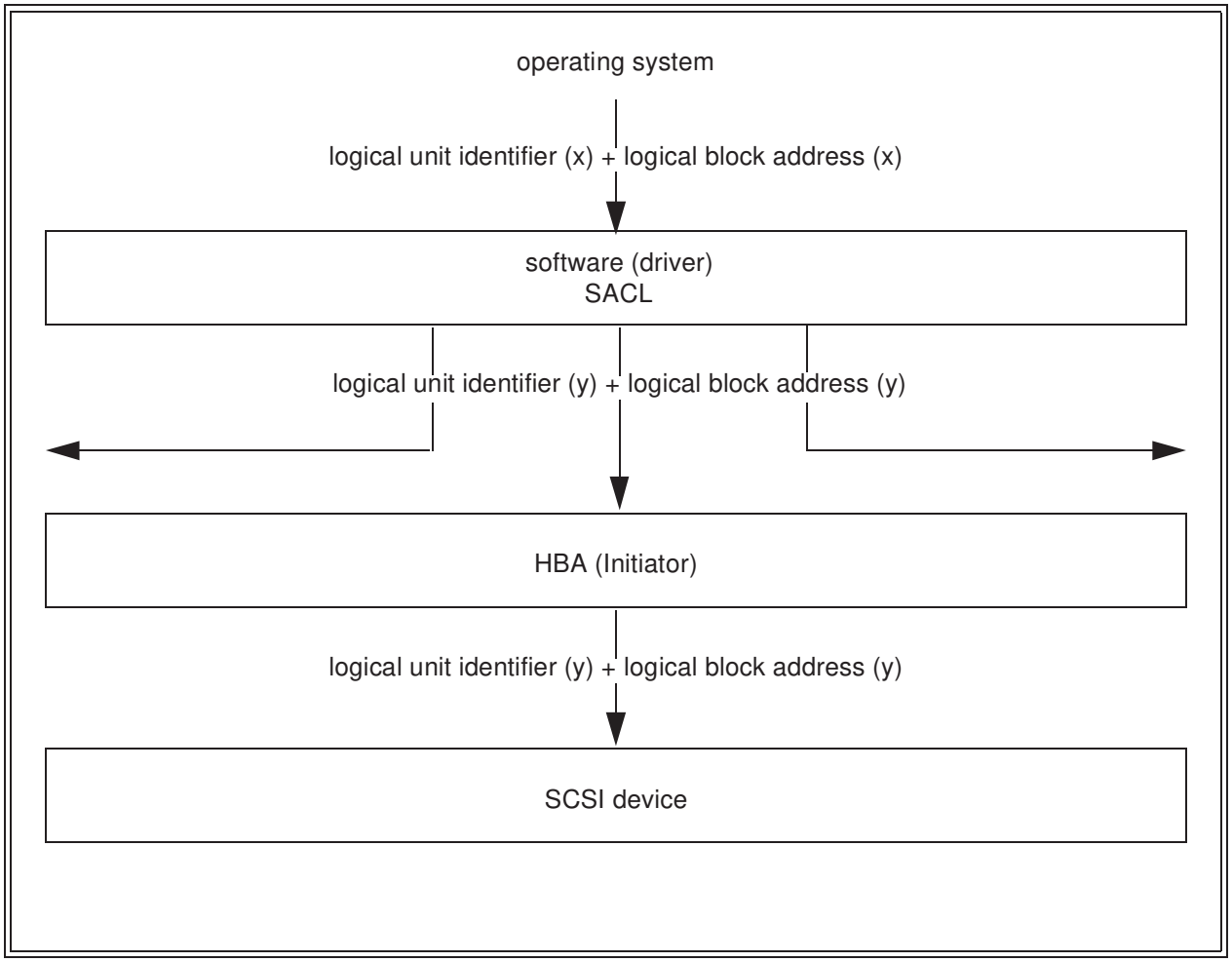

Figure 6 - Software SACL with a branch of SCSI disks

5.1.7 Branch with HBA SACL

Figure 7 shows a HBA performing SACL functions. These functions convert the input logical unit identifier(X) and the input logical block address(X) to the output logical unit identifier(Y) and output logical block address(Y). All other layers pass the logical unit identifier and logical block address through.

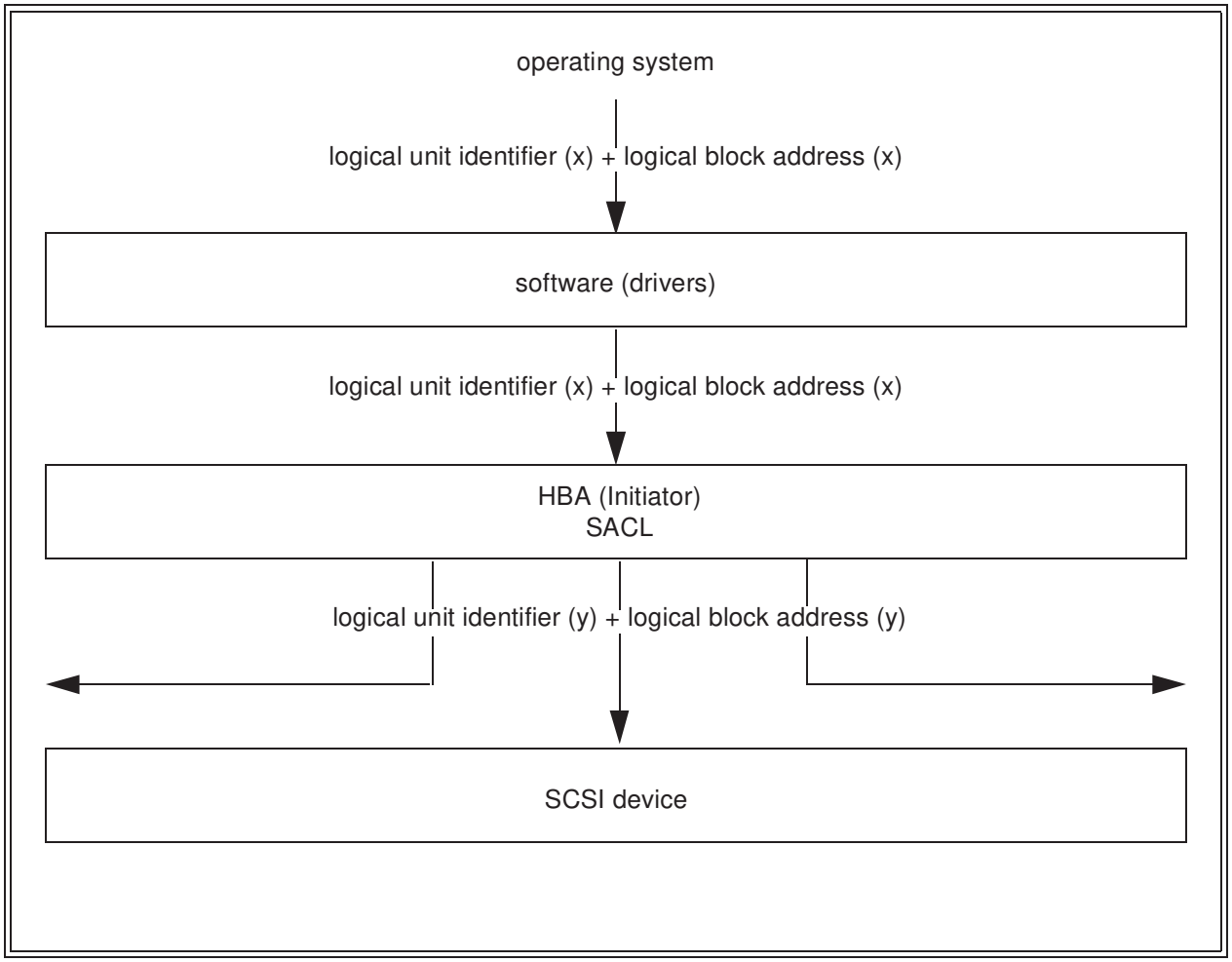

Figure 7 - Branch with HBA SACL

5.1.8 Branch with bridge controller SACL

Figure 8 shows a bridge controller performing SACL functions. These functions convert the input logical unit identifier(X) and the input logical block address(X) to the output logical unit identifier(Y) and output logical block address(Y). All other layers pass the logical unit identifier and logical block address through.

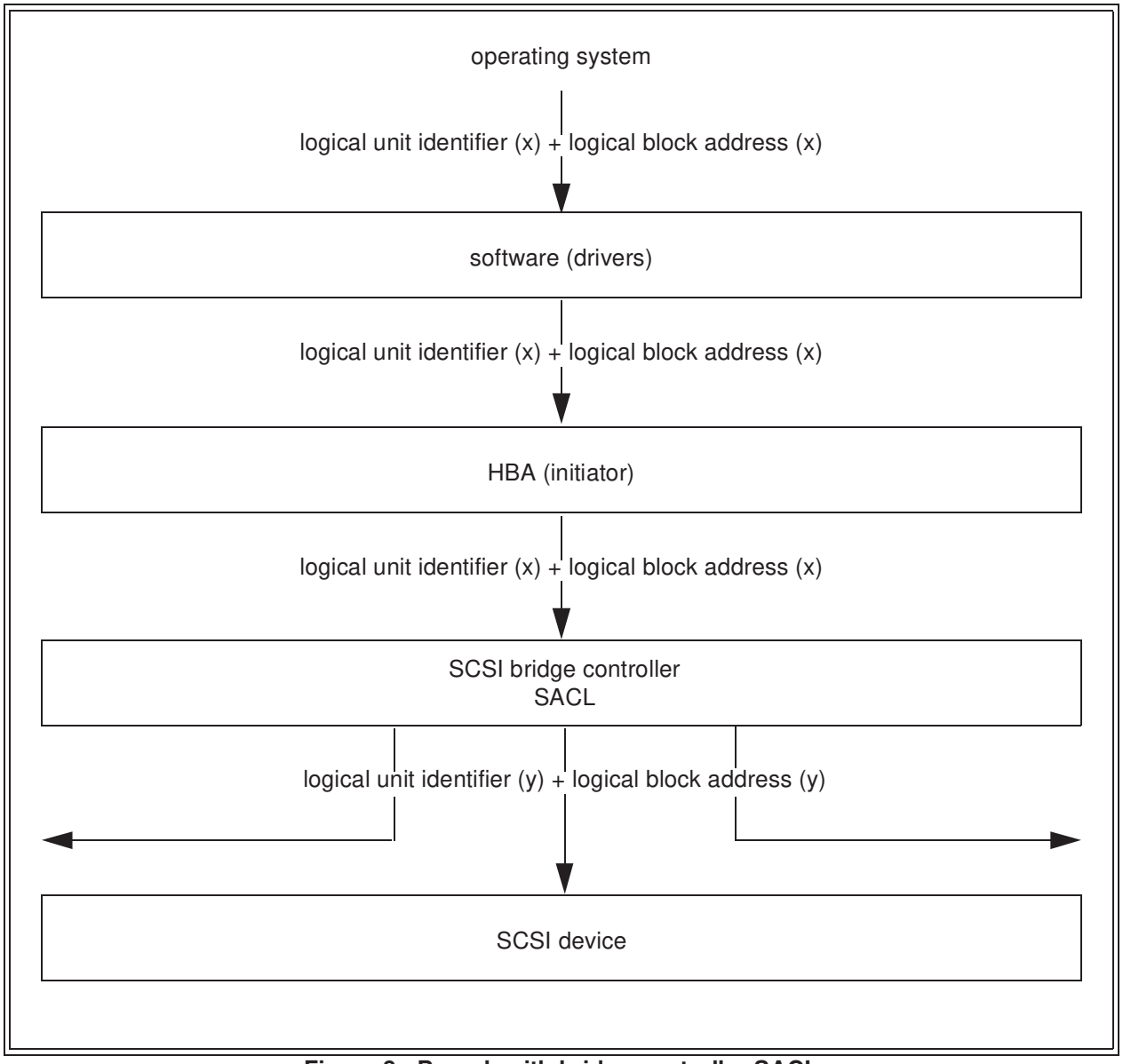

Figure 8 - Branch with bridge controller SACL

5.1.9 Branch with SACLs in multiple layers

Figure 9 shows an example with SACLs in multiple layers. This example shows a software layer performing SACL functions and a bridge controller performing SACL functions.

The software functions convert the input logical unit identifier(X) and the input logical block address(X) to the output logical unit identifier(Y) and output logical block address(Y). The bridge controller functions then convert the input logical unit identifier(Y) and the input logical block address(Y) to the output logical unit identifier(Z) and output logical block address(Z). All other layers pass the logical unit identifier and logical block address through.

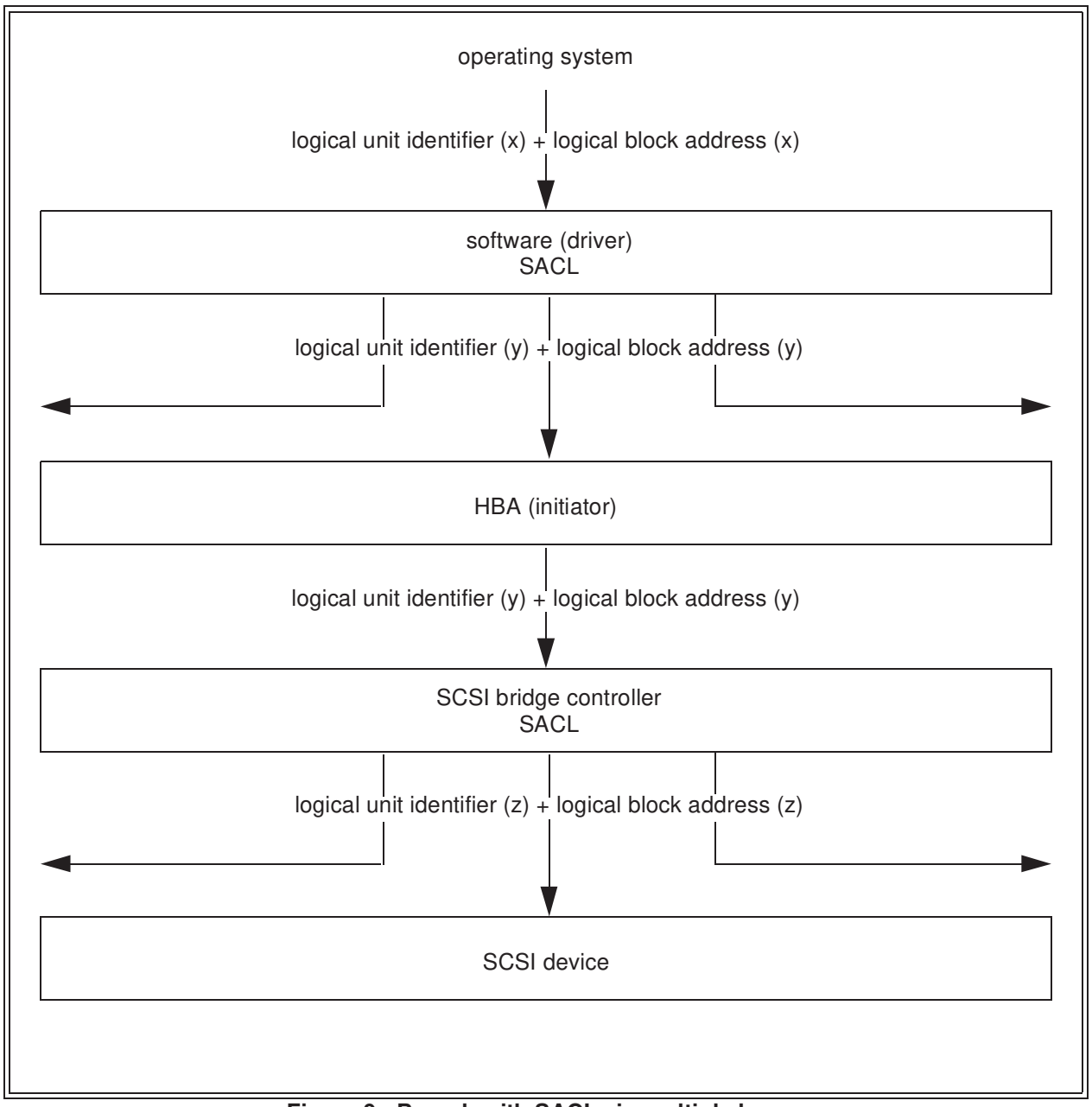

Figure 9 - Branch with SACLs in multiple layers

5.2 Model for SCSI-3 storage arrays

The SCSI-3 storage array model defines:

a) the addressing of multiple devices and the addressing of multiple types of devices through multiple layers of SCSI-3 storage arrays as a single target,
b) objects and how they are to be configured to create an operational SCSI-3 storage array; and
c) the operations that occur within SCSI-3 storage arrays.

5.2.1 SCSI storage array addressing

See table 1 for the methods used when addressing a SCSI-3 storage array.

Table 1 - Addressing methods within a SCSI-3 storage array

Method of addressing	Directly addressable by application client	Suffix	Space addressed
Component device	no	_C	none
Peripheral device	yes	_P	physical
Redundancy group	no	_R	protected
Spare	no	_S	none
Volume set	yes	_V	user data

The application client shall access redundancy groups, spares, and components by issuing commands to the SCSI-3 storage array's base address (LUN 0). The application client may directly address peripheral devices and volume sets by using eight byte LUN fields (see 5.2.1.2). For SCSI-3 storage array devices that conform to the SCSI-3 Interlocked Protocol Standard a method has been defined using a modified IDENTIFY message (see A.1) and a LUN mapping page (see 6.9.1.1) to allow direct addressing of peripheral devices and volume sets that conform to the SCSI-3 storage array addressing model.

All peripheral device addresses, except the base address, default to vendor specific values. All component device, redundancy group, spare, and volume set addresses may default to vendor specific values or may be defined by an application client during configuration.

5.2.1.1 SCSI-3 storage array base address

All SCSI-3 storage arrays shall accept a LUN value of zero as a valid address. For SCSI-3 storage arrays that have more than one logical unit, logical unit zero shall be the logical unit that an application client addresses to configure an SCSI-3 storage array and to determine information about the target and the logical units contained within the target. INQUIRY commands to an SCSI-3 storage array base address shall return a device type of array controller device.

To address the base address of the SCSI-3 storage array the peripheral device address method shall be used.

5.2.1.2 Eight byte LUN structure

The eight byte LUN structure (see table 3) allows up to four levels of devices to be addressed under a single target. Each level shall use bytes 0-1 to define the address and/or location of the SCSI device to be addressed on that level.

The eight byte LUN structure defined in this standard shall apply to SCSI-3 storage array devices that

conform to the SCSI-3 Controller Command.

If the LUN indicates that the command is to be passed to the next layer then the current layer shall use bytes 0-1 of the eight byte LUN structure to determine the address of the device to which the command is to be sent. When the command is sent to the target the eight byte LUN structure that was received shall be adjusted to create a new eight byte LUN structure (see table 2).

SCSI-3 storage arrays shall keep track of the necessary addressing information to allow reconnection to the correct task during reselection.

Table 2 - Eight byte LUN structure adjustments

Byte position		
Old		**New**
0 - 1	Moves to	Not used
2 - 3	Moves to	0 - 1
4 - 5	Moves to	2 - 3
6 - 7	Moves to	4 - 5

Bytes six and seven of each new eight byte LUN structure shall be set to zero.

Table 3 - Eight byte LUN structure

Bit / Byte	7	6	5	4	3	2	1	0
0	FIRST LEVEL ADDRESSING							
1								
2	SECOND LEVEL ADDRESSING							
3								
4	THIRD LEVEL ADDRESSING							
5								
6	FOURTH LEVEL ADDRESSING							
7								

The FIRST LEVEL ADDRESSING field indicates the first level address of a peripheral device or volume set. See table 4 for a definition of the FIRST LEVEL ADDRESSING field.

The SECOND LEVEL ADDRESSING field indicates the second level address of a peripheral device or volume set. See table 4 for a definition of the SECOND LEVEL ADDRESSING field.

The THIRD LEVEL ADDRESSING field indicates the third level address of a peripheral device or volume set. See table 4 for a definition of the THIRD LEVEL ADDRESSING field.

The FOURTH LEVEL ADDRESSING field indicates the fourth level address of a peripheral device or volume set.

See table 4 for a definition of the FOURTH LEVEL ADDRESSING field.

Table 4 - FIRST LEVEL ADDRESSING field, SECOND LEVEL ADDRESSING field, THIRD LEVEL ADDRESSING field, and FOURTH LEVEL ADDRESSING field

Bit / Byte	7	6	5	4	3	2	1	0
n-1	ADDRESS METHOD							
n	ADDRESS METHOD SPECIFIC							

The ADDRESS METHOD field defines the contents of the ADDRESS METHOD SPECIFIC field. See table 5 for the defined address methods.

Table 5 - ADDRESS METHOD

Codes	Description
10b	Logical unit addressing method
00b	Peripheral device addressing method
01b	Volume set addressing method
11b	Reserved

See 5.2.1.5 and 5.2.1.8 for the definitions of the ADDRESS METHOD SPECIFIC field.

5.2.1.3 Component device address method

A component shall only be addressed by addressing the base address of the SCSI-3 storage array that controls or will control the component device. For information on addressing the base address see 5.2.1.1.

5.2.1.4 Logical unit address method

All SCSI commands are allowed when the logical unit address method is selected, however logical units and SCSI-3 storage arrays are only required to support mandatory SCSI commands. SCSI-3 storage arrays are not required to honor pass-through requests from the application client. Any command that is not supported or passed-through shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID COMMAND OPERATION CODE.

If the logical unit addressing method is selected the SCSI-3 storage array shall relay the received command, if supported, to the addressed logical unit. See table 6 for the definition of the ADDRESS METHOD SPECIFIC field used when the logical unit addressing method is selected.

Table 6 - Logical unit addressing

Bit / Byte	7	6	5	4	3	2	1	0
n-1	1	0	TARGET					
n	BUS			LUN				

The TARGET field indicates the target address of the peripheral device to which the SCSI-3 storage array shall relay the received command. The TARGET field indicates the address of the target on the bus indicated by the BUS NUMBER field that the received command shall be relayed to.

NOTE 3 - The value of targets within the TARGET field are defined by individual standards. (e.g., SCSI-3 Parallel Interface Standard defines targets to be in the range 0-7, 0-15, and 0-31.

The BUS NUMBER field indicates the location of the bus that the SCSI-3 storage array shall use to relay the received command.

The LUN field indicates the address of the logical unit to which the SCSI-3 storage array shall relay the received command. The LUN field indicates the address of the logical unit within the target indicated by the TARGET field that the received command shall be relayed to.

5.2.1.5 Peripheral device address method

All SCSI commands are allowed when the peripheral address method is selected, however peripheral devices and SCSI-3 storage arrays are only required to support mandatory SCSI commands. SCSI-3 storage arrays are not required to honor pass-through requests from the application client. Any command that is not supported or passed-through shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID COMMAND OPERATION CODE.

If the peripheral device addressing method is selected the SCSI-3 storage array shall relay the received command, if supported, to the addressed peripheral device providing a bypass of the SACL. See table 7 for the definition of the ADDRESS METHOD SPECIFIC field used when the peripheral device addressing method is selected.

Table 7 - Peripheral device addressing

Bit / Byte	7	6	5	4	3	2	1	0
n-1	0	0	BUS NUMBER					
n	TARGET/LUN							

The BUS NUMBER field indicates the location of the bus that the SCSI-3 storage array shall use to relay the received command.

The TARGET/LUN field indicates the address of the peripheral device/LUN to which the SCSI-3 storage array shall relay the received command. If the BUS NUMBER field is set to zero the TARGET/LUN field indicates the address of the logical unit to relay the received command to, is located within the current level of the

SCSI-3 storage array. If the bus number field is not zero the TARGET/LUN filed indicates the address of the target on the bus indicated by the bus number field that the received command shall be relay to. The logical unit to relay the received command to shall be LUN zero.

The base device located within the current level shall be addressed by a BUS NUMBER field and a TARGET/LUN field of all zeros.

NOTE 4 - The value of targets within the TARGET/LUN field are defined by individual standards. (e.g., SCSI-3 Parallel Interface Standard defines targets to be in the range 0-7, 0-15, and 0-31.

5.2.1.6 Redundancy group address method

A redundancy group shall only be addressed by addressing the base address of the SCSI-3 storage array that controls or will control the redundancy group. For information on addressing the base address see 5.2.1.1.

5.2.1.7 Spare address method

A spare shall only be addressed by addressing the base address of the SCSI-3 storage array that controls or will control the spare. For information on addressing the base address see 5.2.1.1.

5.2.1.8 Volume set address method

The volume set address method points to the SACL that executes the command using the algorithms defined by the configuration.

NOTE 5 - The volume set might not be under the control of the addressed SCSI-3 storage array. It is allowed to be in an SCSI-3 storage array lower in the tree structure.

All SCSI commands are allowed when the volume set address method is used, however volume sets are not required to support all SCSI commands. Any command that is not supported shall terminate in error.

In the response to an INQUIRY command the addressed volume set shall return a valid SCSI-3 peripheral device type.(e.g., direct access device, streaming device, etc.)

When the volume set addressing method is selected the SCSI-3 storage array shall use a SACL at the current level to address peripheral devices as required to execute the received command. See table 8 for the definition of the ADDRESS METHOD SPECIFIC field used when the volume set addressing method is selected.

Table 8 - Volume set addressing

Bit Byte	7	6	5	4	3	2	1	0
n-1	0	1	(MSB)					
n	LUN							(LSB)

The LUN field indicates the address of the volume set that the SCSI-3 storage array shall direct the received command to.

5.2.2 SCSI-3 storage array objects

An SCSI-3 storage array consists of several objects that are configured to create an operating device. The rules for configuring objects and their interactions are defined in the following clauses. An SCSI-3 storage array shall save the current configuration throughout all task management functions and power conditions.

NOTE 6 - The configuration becomes the current configuration on successful completion of a configuration command.

The objects are:

a) component device;
b) peripheral device;
c) p_extent;
d) ps_extent;
e) redundancy group;
f) spare;
g) volume set.

5.2.2.1 Adding objects

Objects that have been added to a SCSI-3 storage array shall be addressable (see 5.2.1) by an application client.

Peripheral devices and component devices may be logically added to a SCSI-3 storage array. The ADD PERIPHERAL DEVICE/COMPONENT DEVICE service action (see 6.2.1.1) adds objects. Adding a peripheral device or a component device may or may not automatically cause the device to become part of a configuration.

5.2.2.2 Association of objects

Objects become associated during the configuration of SCSI-3 storage arrays. The create/modify service actions for redundancy groups (see 6.4.1.2) and volume sets (see 6.6.1.3) create associations and the delete service actions (see 6.4.1.3 and 6.6.1.4) remove associations. The rules for objects that are associated are explicitly defined throughout this clause and within the create/modify service actions.

An SCSI-3 storage array may create associations through mechanisms other than the create/modify service actions defined in this document (e.g., by factory settings or by external configuration facilities). Whether such associations may be modified by the create/modify service actions specified by this standard or by other mechanisms is vendor specific. The only requirement on such associations is that they shall be reported using the report service actions defined in this standard.

NOTE 7 - One example of an association would be to associate a group of p_extents with a redundancy group using a CREATE/MODIFY REDUNDANCY GROUP service action.

5.2.2.3 Attachment of objects

Objects shall only be attached to component devices. The ATTACH TO COMPONENT DEVICE service action (6.2.1.2) establishes attachments.

The behavior of attachments and their interactions with component devices are vendor specific.

The attachment of component devices shall not be reflexive (i.e. attaching LUN_C 1 to LUN_C 2 does not attach LUN_C 2 to LUN_C 1).

NOTE 8 - One example of an attachment would be to attach a redundancy group to a specific power supply. This attachment could mean that the power supply would only supply power to that redundancy

group. Another example would be to attach a controller card to another controller card. This attachment could mean that the work load is shared between the two controllers.

An SCSI-3 storage array may create attachments through mechanisms other than the attach service actions defined in this document (e.g., by factory settings or by external configuration facilities). Whether such attachments may be modified by the attach service actions specified by this standard or by other mechanisms is vendor specific.The only requirement on such attachments is that they shall be reported using the report service actions defined in this standard.

5.2.2.4 Covering of objects

Any object or group of objects may be covered by a like object. The SPARE(OUT) command (see 6.8.1) establishes the covering of the objects.

When an object or group of objects are covered and there is a failure, the covering object shall assume all the characteristics of the failed object, in effect doing an automatic exchange. The length of time the exchange stays in effect is vendor specific.

Objects may be covered by a SCSI-3 storage array without any spare requests being issued from an application client. The SCSI-3 storage array may have covered objects based on internal requirements.

NOTE 9 - One example of covering would be for a peripheral device to cover like peripheral device(s) within a redundancy group. This would cause any peripheral device that failed within the covered redundancy group to be replaced by the peripheral device covering that redundancy group. Two controllers could also be configured to cover one another, using two spare service actions, in which case if either fails the other will take over the entire work load.

5.2.2.5 Exchanging objects

A peripheral device, p_extent, or component device may be exchanged with a like device. The exchange service actions (see 6.2.1.4 and 6.2.1.5) are used to exchange objects. When an old peripheral device, p_extent, or component device is exchanged, the new peripheral device, p_extent, or component device takes on all the characteristics of the old peripheral device, p_extent, or component device (e.g., redundancy group, volume set mappings, attachments, etc.). An exact copy of any protected space contents and/or check data on the old device shall be replicated on the new device. The characteristics of the old peripheral device are vendor specific.

5.2.2.6 Protected objects

Protected objects are objects that are able to tolerate one or more objects failing without any loss of user data or loss of SCSI-3 storage array availability.

5.2.2.7 Removing objects

Objects that have been removed from a SCSI-3 storage array shall not be addressable by an application client, until the object is re-added or recreated. The remove service actions (see 6.2.1.7, 6.4.1.3 and 6.8.1.3) are used to remove objects.

Any object may be removed from a configuration. However, some restrictions may apply as to when an object may be removed. Those restrictions are defined in 6.2.1.7, 6.4.1.3, and 6.8.1.3.

When the object is removed all attachments, associations, and covers relating to the object being removed are erased from any configurations containing the object. Any addressable logical block addresses within a removed volume set shall become unassigned protected space. Any addressable logical block addresses within a removed redundancy group or spare shall become unassigned p_extent space.

5.2.2.8 Component device

See 3.1.10 for the definition of component devices.

5.2.2.9 Peripheral device

See 3.1.26 for the definition of peripheral devices.

5.2.2.10 P_extent

See 3.1.25 for the definition of p_extents.

P_extents are used by the application client to create and modify redundancy groups and spares. A single assigned p_extent or unassigned p_extent that contains no check data may be configured into one or more redundancy groups. A single assigned p_extent or unassigned p_extent may be configured into one or more spares. A single p_extent shall not be associated with a redundancy group and configured as a spare at the same time. After a p_extent has been configured into a redundancy group or as a spare it becomes an assigned p_extent.

Any addressable logical block addresses within an operating peripheral device connected within a SCSI-3 storage array that have not been configured into a redundancy group shall be unassigned p_extents. Before assignment, all the consecutive addressable logical block addresses on a single peripheral device shall be grouped into a single unassigned p_extent.

The application client uses a REPORT ASSIGNED/UNASSIGNED P_EXTENT service action (see 6.1.1.1) to determine the unassigned p_extents.

A single unassigned p_extent may be subdivided into one or more p_extents by the application client. The subdivided p_extents may be used within one or more CREATE/MODIFY REDUNDANCY GROUP service actions.

5.2.2.11 Ps_extent

See 3.1.28 for the definition of ps_extent.

Ps_extents are used by the application client to create and modify volume sets. Ps_extents are created on the successful completion of a CREATE/MODIFY REDUNDANCY GROUP service action (6.4.1.2.) A single ps_extent shall only be configured into one volume set. After a ps_extent has been configured into a volume set it becomes an assigned ps_extent.

Any addressable logical block addresses, not defined as containing check data, within an operating peripheral device connected within a SCSI-3 storage array that have not been configured into a volume set shall be unassigned ps_extents. All the consecutive addressable logical block addresses on a single peripheral device, excluding logical blocks defined as containing check data, shall be grouped into a single unassigned ps_extent.

The application client uses a REPORT UNASSIGNED REDUNDANCY GROUP SPACE service action (see 6.3.1.2) to determine the unassigned ps_extents.

A single unassigned ps_extent may be subdivided into one or more ps_extents by the application client. The subdivided ps_extents may be used within one or more CREATE/MODIFY VOLUME SET service actions.

A ps_extent is different from a p_extent in that the ps_extent does not include any logical block addresses that have been mapped as check data where the p_extent includes all addressable logical blocks within the selected range on a peripheral device.

5.2.2.12 Redundancy group

See 3.1.31 for the definition of redundancy group.

Redundancy groups are created by the application client to protect user data contained within volume sets by mapping check data within a redundancy group. Redundancy groups may overlap, however, the underlaying p_extents within the overlap shall not contain any check data.

As a result of a successful creation of a redundancy group ps_extents are formed. The ps_extents then may be used by the application client to create volume sets.

Figure 10 shows the relationship between the check data and protected space before any volume sets have been defined.

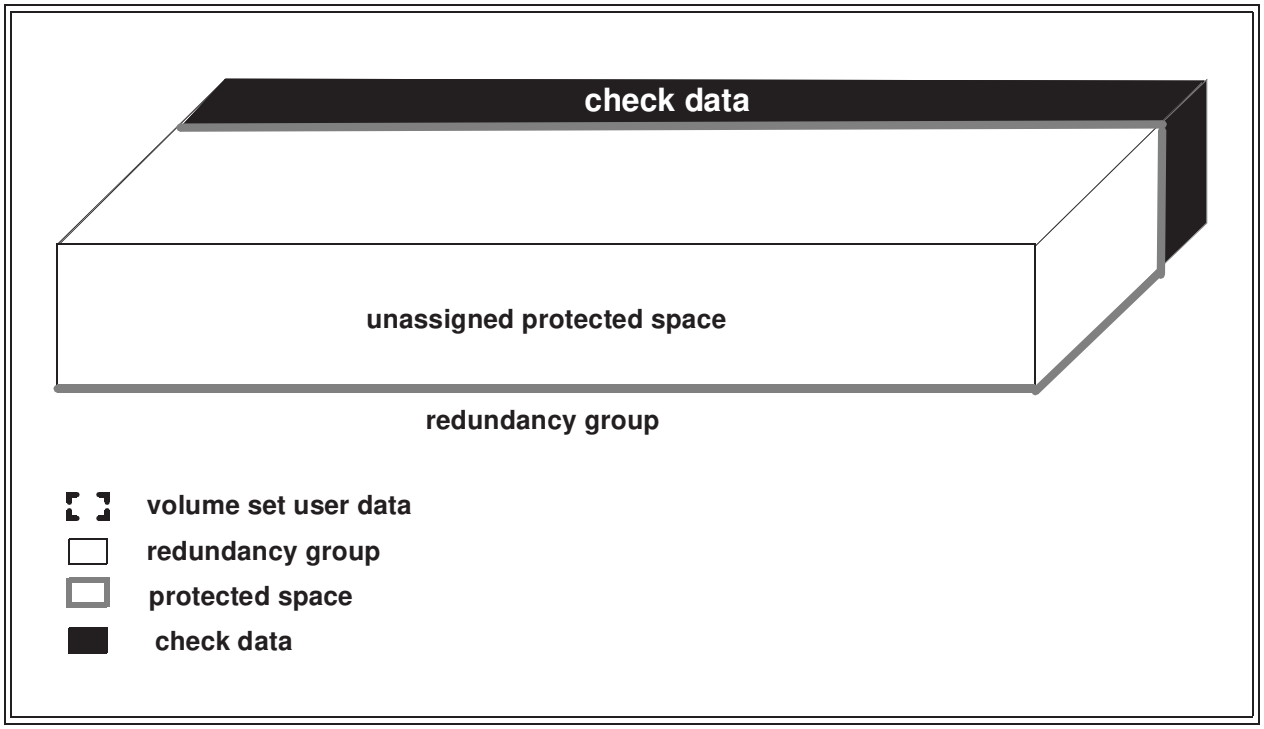

Figure 10 - Single redundancy group

Figure 11 shows the relationship between the check data and protected space after two volume sets have been defined within the redundancy group.

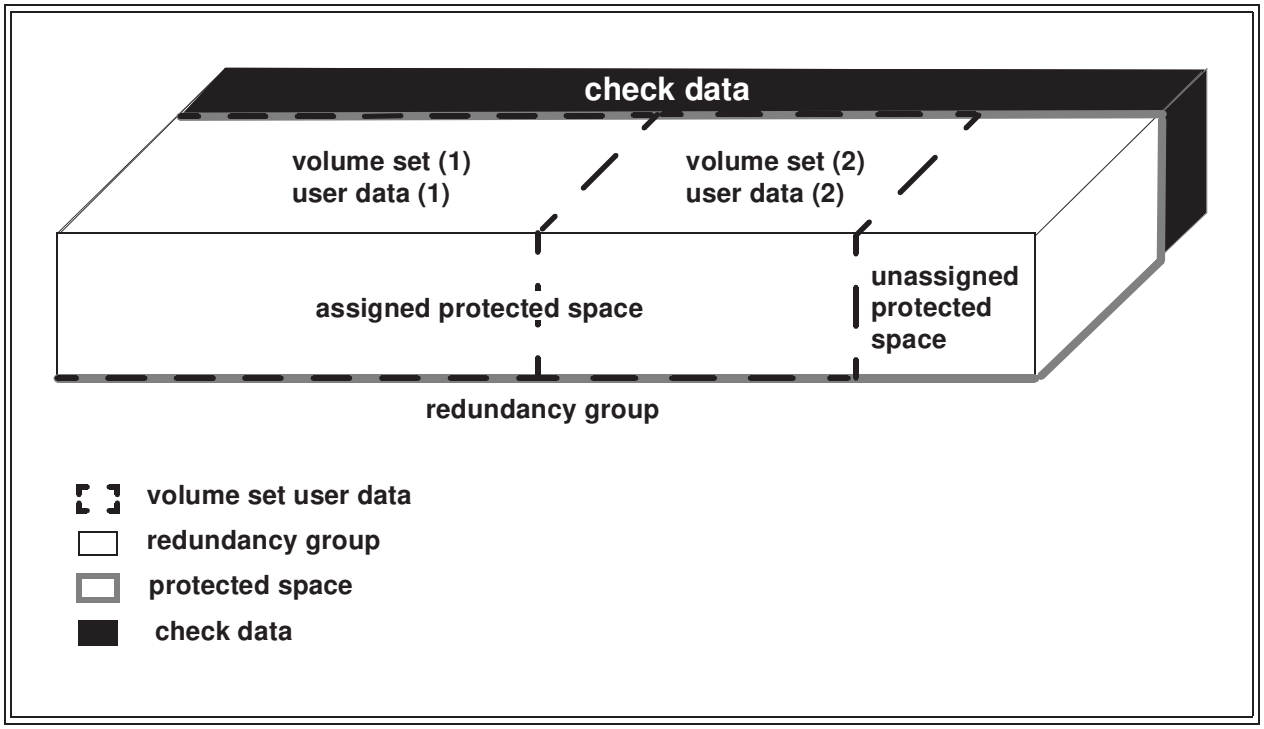

Figure 11 - Multiple volume sets associated with a single redundancy group

The application client may map check data within a redundancy group using the following three parameters:

a) START CHECK DATA INTERLEAVE UNIT field;
b) NUMBER OF UNITS OF CHECK DATA field;
c) NUMBER OF UNITS OF USER DATA field.

The START CHECK DATA INTERLEAVE UNIT field contains the location of the first unit of check data within the p_extent. There is a unique START CHECK DATA INTERLEAVE UNIT field for each p_extent.

The NUMBER OF UNITS OF CHECK DATA field contains the number of consecutive units to be reserved for check data within the p_extent. There is a unique NUMBER OF UNITS OF CHECK DATA field for each p_extent.

The NUMBER OF UNITS OF USER DATA field contains the number of consecutive units to be reserved for protected space within the p_extent. There is a unique NUMBER OF UNITS OF USER DATA field for each p_extent.

The flow chart in figure 12 is an example of an implementation of a SCSI-3 storage array redundancy group being configured with a check data mapping of XOR redundancy. The check data mapping routine uses the START CHECK DATA INTERLEAVE UNIT field, NUMBER OF UNITS OF CHECK DATA field, and NUMBER OF UNITS OF USER DATA field and assumes a non-zero value for the NUMBER OF UNITS OF CHECK DATA field and the NUMBER OF UNITS USER DATA field.

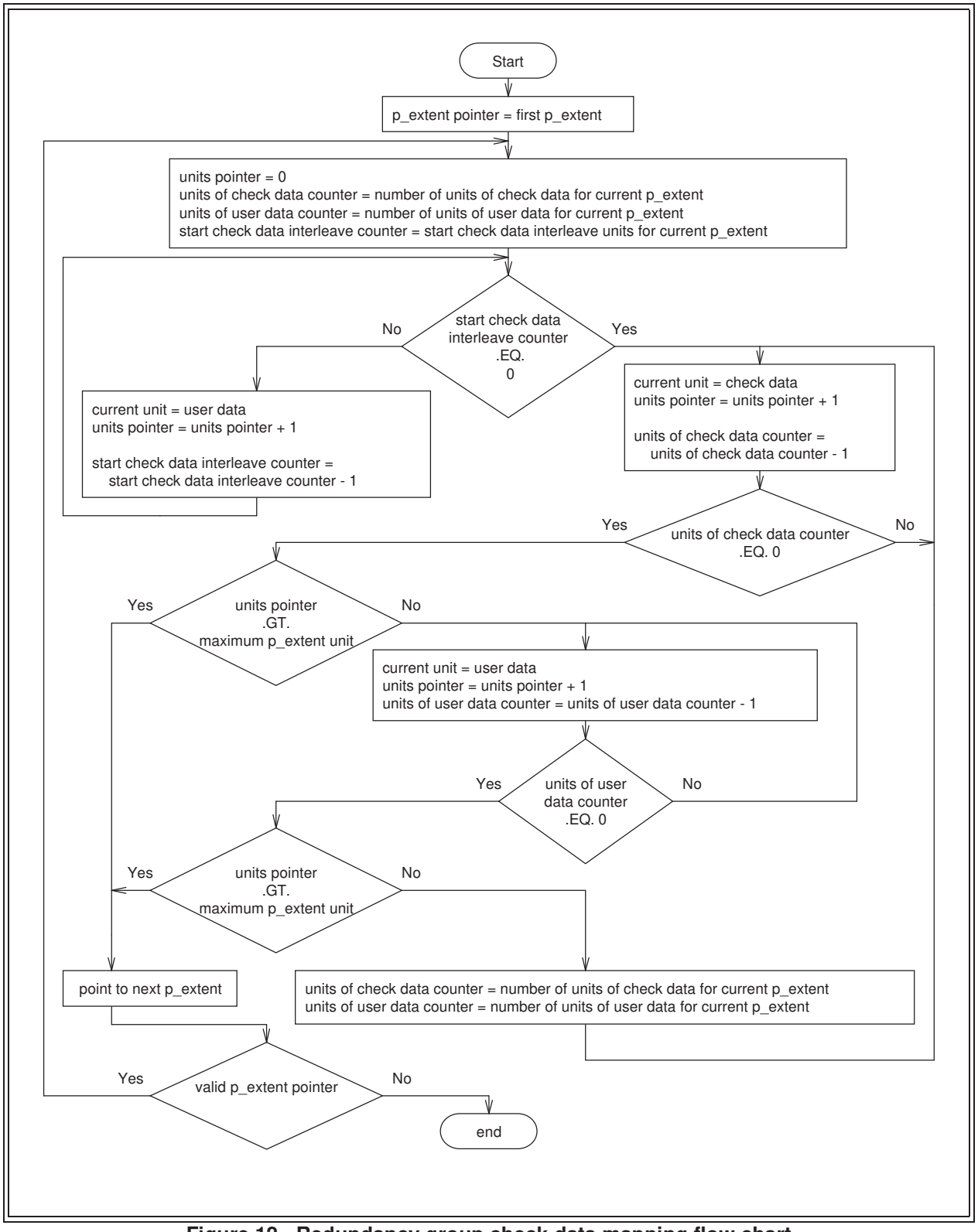

Figure 12 - Redundancy group check data mapping flow chart

The application may choose any one of the following methods of check data mapping within a redundancy group:

a) No redundancy;

b) Copy redundancy;
c) XOR redundancy;
d) P+Q redundancy;
e) Vendor specific redundancy.

5.2.2.12.1 No redundancy method of check data mapping

In a redundancy group that the application client requests no redundancy, user data is not protected. If this method is requested then the following fields in the CREATE/MODIFY REDUNDANCY GROUP service action (see 6.4.1.2) shall all be set as indicated:

a) GRANULARITY OF UNITS field = 0;
b) START CHECK DATA INTERLEAVE UNIT field = 0;
c) NUMBER OF UNITS OF CHECK DATA field = 0;
d) NUMBER OF UNITS OF USER DATA field = set to desired value.

5.2.2.12.2 Copy redundancy method of check data mapping

In a redundancy group that the application client requests copy redundancy all user data is replicated on all the p_extents listed in the CREATE/MODIFY REDUNDANCY GROUP service action (6.4.1.2). If this method is requested then the following fields in the CREATE/MODIFY REDUNDANCY GROUP service action (see 6.4.1.2) shall be set as indicated:

a) GRANULARITY OF UNITS field = logical block;
b) START CHECK DATA INTERLEAVE UNIT field = 0;
c) NUMBER OF UNITS OF CHECK DATA field = 0;
d) NUMBER OF UNITS OF USER DATA field = set to desired value.

For copy redundancy the NUMBER OF UNITS OF USER DATA field shall be equal for all the p_extents within a single CREATE/MODIFY REDUNDANCY GROUP service action.

5.2.2.12.3 XOR or P+Q redundancy method of check data mapping

In a redundancy group that the application client requests XOR redundancy, or P+Q redundancy the user data is protected by use of check data located within the check data areas. The check data mapping is defined by the check data interleave fields. If this method is requested then the following fields in the CREATE/MODIFY REDUNDANCY GROUP service action (see 6.4.1.2) shall be set as indicated:

a) GRANULARITY OF UNITS field = set to desired value;
b) START CHECK DATA INTERLEAVE UNIT field = set to desired value;
c) NUMBER OF UNITS OF CHECK DATA field = set to desired value;
d) NUMBER OF UNITS OF USER DATA field = set to desired value.

5.2.2.12.4 Vendor specific redundancy method of check data mapping

In a redundancy group for which the application client requests a redundancy type method of vendor specific redundancy, the user data is protected in a vendor specific manner.

5.2.2.13 Spares

See 3.1.37 for the definition of spares.

Spares are created by the application client to allow a failed p_extent, peripheral device, or component device to be automatically exchanged (see 5.2.2.5). P_extents shall only be exchanged with p_extents, peripheral devices shall only be exchanged with peripheral devices, and component devices shall only be exchanged with component devices. A redundancy group or peripheral device that contains a failed range of LBA_P(s), a failed peripheral device, or a failed component shall be covered by a spare before an

automatic exchange is allowed to occur.

After an automatic exchange the spare takes on all the characteristics of the failed p_extent, peripheral device, or component device. After the automatic exchange the covering p_extent, peripheral device, or component device shall no longer be available to cover another object. The failed p_extent, peripheral device, or component device shall be marked as broken.

The method for replacing the failed p_extent, peripheral device, or component device and restoring the spare is vendor specific.

NOTE 10 - One method of replacing a failed p_extent, peripheral device, or component device would be to automatically exchange a new p_extent, peripheral device, or component device with the spare that covered the failure. After a successful exchange the spare is available to cover failures and the failed p_extent, peripheral device, or component device is as it was before the failure.

NOTE 11 - Another method would be to automatically add the new p_extent, peripheral device, or component device into the SCSI-3 storage array, delete the original covering spare, then create a new spare with the new p_extent, peripheral device, or component device that would cover the same p_extent, peripheral device, or component device as the original spare.

5.2.2.14 Volume sets

See 3.1.47 for the definition of volume sets.

Volume sets are created by the application client to provide a contiguous range of logical block addresses for reading and writing user data. Volume sets shall not overlap and shall be independent from one another.

The application client issues a REPORT UNASSIGNED REDUNDANCY GROUP SPACE service action (see 6.3.1.2) to determine the boundaries of any unassigned protected space. The unassigned protected space may be used to create volume sets or to expand existing volume sets. This service action provides information to the application client so it can place volume sets on specific peripheral devices.

Figure 13 shows a relationship between the check data and protected space before any volume sets have been defined.

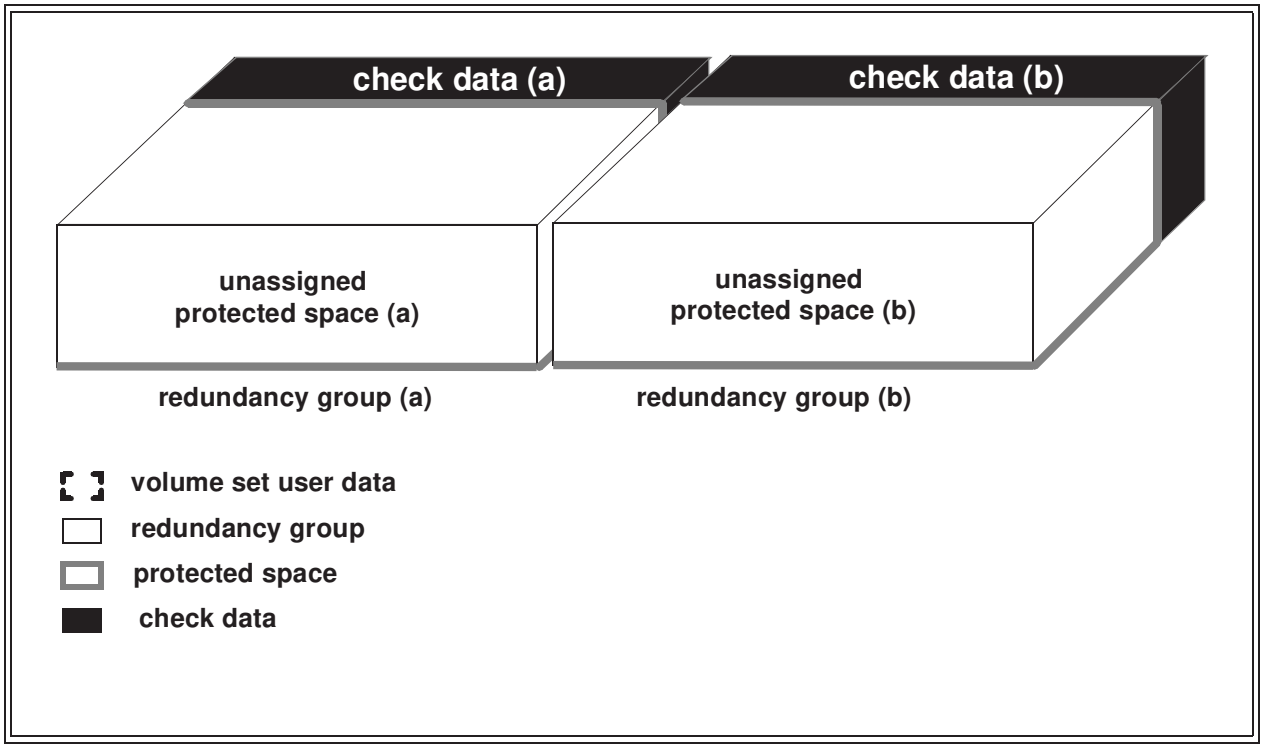

Figure 13 - Multiple redundancy groups

Figure 14 shows the relationship between check data, user data and protected space after a single volume set has been defined across the two redundancy groups.

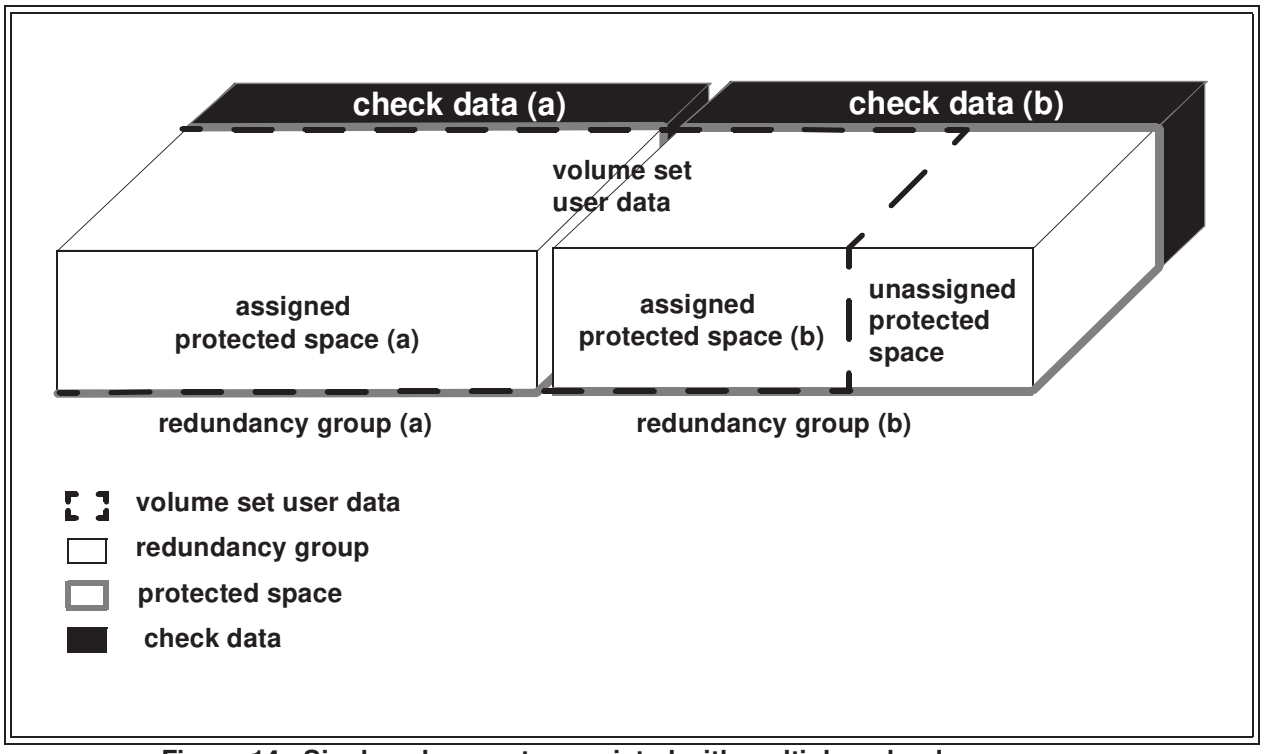

Figure 14 - Single volume set associated with multiple redundancy groups

NOTE 12 - The volume set in figure 14 is ready to receive user data.

The application client may map user data within a volume set using the following three parameters:

a) USER DATA STRIPE DEPTH field;
b) PS_EXTENT STRIPE LENGTH field;
c) PS_EXTENT INTERLEAVE DEPTH field.

The USER DATA STRIPE DEPTH field contains the number of contiguous units to count within a ps_extent before proceeding to the next ps_extent. There is a USER DATA STRIPE DEPTH field for each ps_extent.

The PS_EXTENT STRIPE LENGTH field contains the number of contiguous ps_extents to count before looping back to the first ps_extent of the current stripe.

The PS_EXTENT INTERLEAVE DEPTH field contains the number of stripes to count before continuing onto the next consecutive ps_extent beyond the current stripe.

The ps_extent interleave depth is only used if the ps_extent stripe length is not equal to the number of ps_extents.

The flow chart in figure 15 and figure 16 is an example of an implementation of a SCSI-3 storage array volume set user data mapping routine using the USER DATA STRIPE DEPTH field, PS_EXTENT STRIPE length field, and ps_extent interleave depth field. For simplicity the user data stripe depth field is assumed to be the same value for all the ps_extents in the flow chart.

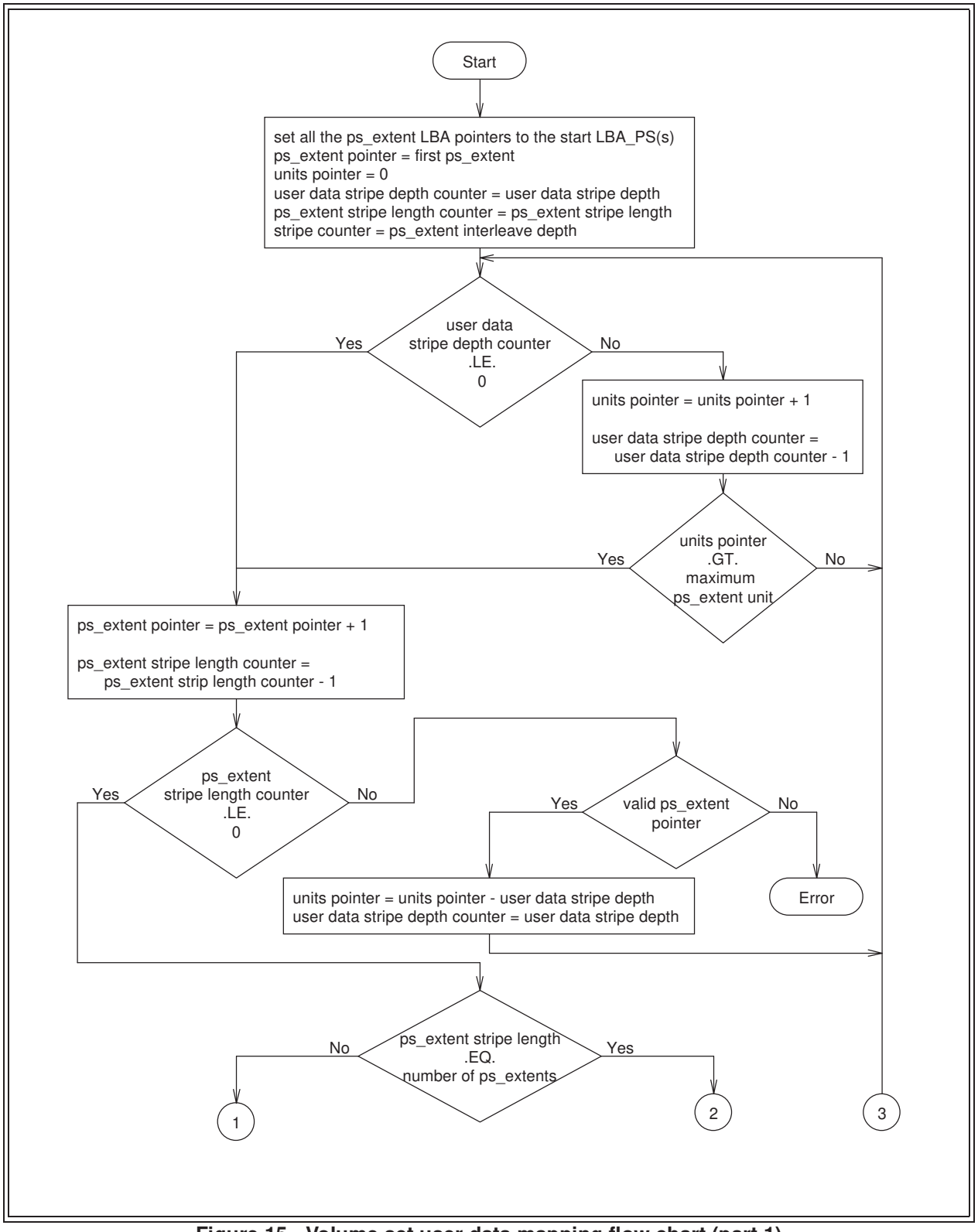

Figure 15 - Volume set user data mapping flow chart (part 1)

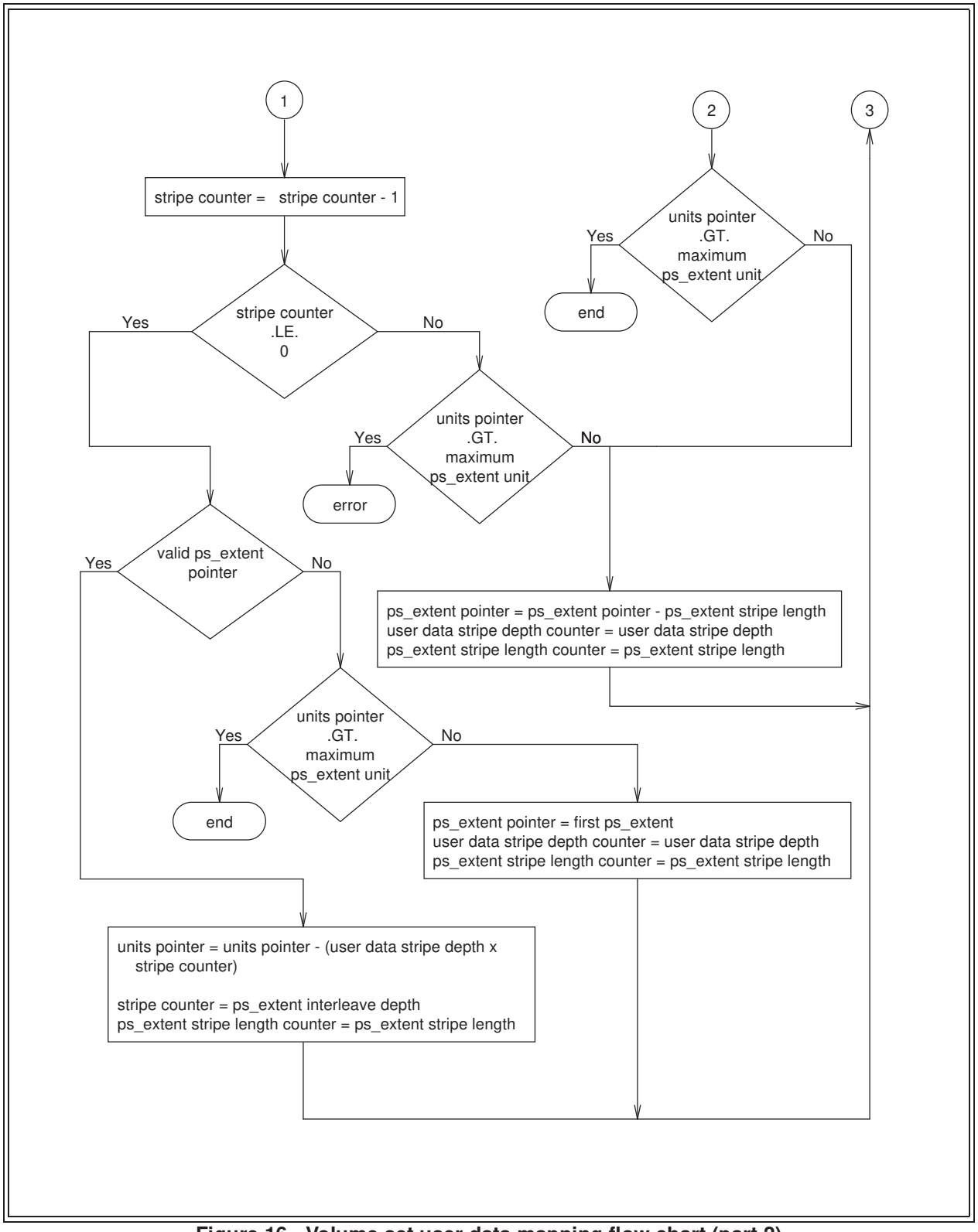

Figure 16 - Volume set user data mapping flow chart (part 2)

Figure 17 shows the most general implementation of the three parameters used to map the user data.

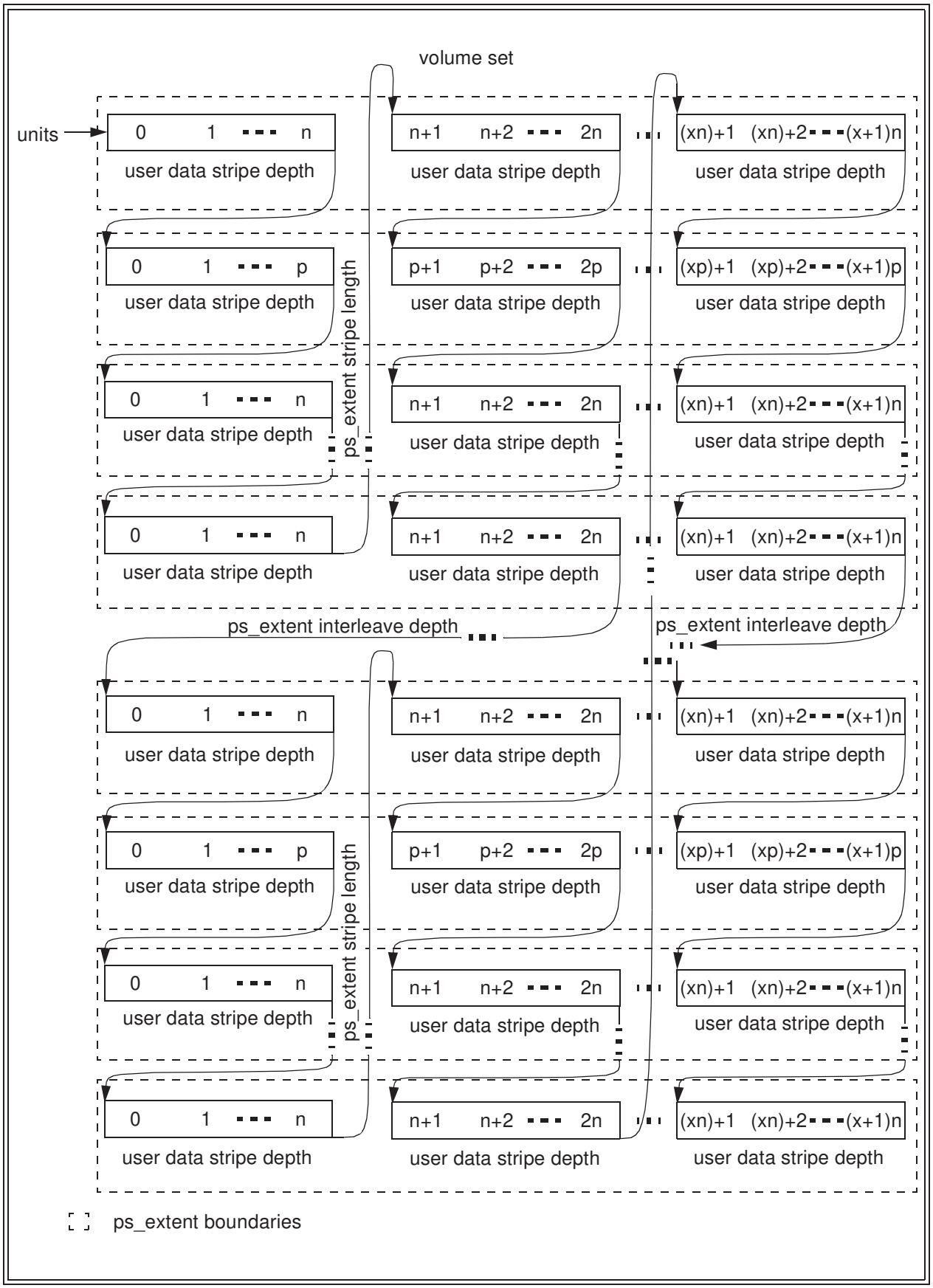

Figure 17 - Volume set user data mapping flow

5.2.3 SCSI-3 storage array operations

5.2.3.1 Rebuild operation

The rebuild operation recreates protected space contents or any check data within a p_extent using check data and protected space contents from the remaining p_extents within the redundancy group. The regenerated protected space contents or any recalculated check data shall be written to the p_extent being rebuilt.

5.2.3.2 Recalculate operation

The recalculate operation recreates check data from protected space contents. The recreated check data shall be written to the check data location being recalculated.

5.2.3.3 Regenerate operation

The regenerate operation recreates inaccessible protected space contents from accessible check data and protected space contents. The recreated protected space contents is not saved after it is transferred to the application client.

5.2.3.4 Verify operation

The verify operation recreates check data from protected space contents and compares the recreated check data with the current check data. If the recreated check data does not match the current check data an exception condition shall be created. See 6.6.1.6 and 6.4.1.7 for exception handling information.

5.2.4 SCSI-3 storage array states

A SCSI-3 storage array defines states for each type of logical unit that may be connected within the SCSI-3 storage array or configured by an application client. The state gives an application client information on the current operating condition of selected logical unit(s).

The state of a SCSI-3 storage array (LUN 0), a peripheral device, or a volume set may be determined by using the same methods as any other SCSI device (e.g., a TEST UNIT READY command followed by a REQUEST SENSE command). However, more detailed state information may be obtained by issuing a REPORT STATES service action.

The REPORT STATES service action (see 6.1.1.7) reports all available state information for the selected logical unit(s) (e.g., If a logical unit has two states, both of those states are reported in the REPORT STATES service action.). States are also returned to the application client in many of the report service actions and only one state shall be returned per report service action. The priority of reporting multiple states is vendor specific.

5.2.5 SCSI-3 storage array exception conditions

Exception conditions inform an application client that:

a) a change occurred in the physical configuration;
b) a change occurred in a volume set configuration;
c) a change occurred in a redundancy group configuration;
d) a change occurred in a spare;
e) a change occurred in the operation state of the SACL;
f) a repair action is requested (e.g., device is predicting failure);
g) a repair action is required to restore the volume sets availability (e.g., power supply failure);
h) a repair action is required to restore the volume sets level of integrity (e.g., device fails); or
i) an error occurred.

Which exception conditions are returned and how often they are returned is based on configuration requests received from an application client.

Not all exception conditions reported from an SCSI-3 storage array indicate a task failed. Some are informational, in that, they provide the application client information on the condition of the SCSI-3 storage array. (e.g., the SCSI-3 storage array is predicting a failure of a logical unit, something is broken but the SCSI-3 storage array is still operational, etc.). The method of reporting informational exception conditions is defined in the SCSI-3 Primary Commands Standard.

6 Commands for SCSI-3 storage array devices

The operation codes for commands that apply only to SCSI-3 storage array devices are listed in table 9.

Table 9 - Commands for SCSI storage array devices

Command Name	Operation code	Type	Subclause
INQUIRY	12h	M	SPC
LOG SELECT	4Ch	O	SPC
LOG SENSE	4Dh	O	SPC
MAINTENANCE(IN)	A3h	M	6.1
MAINTENANCE(OUT)	A4h	O	6.2
MODE SELECT(6)	15h	O	SPC
MODE SELECT(10)	55h	O	SPC
MODE SENSE(6)	1Ah	O	SPC
MODE SENSE(10)	5Ah	O	SPC
PERSISTENT RESERVE IN	5Eh	O	SPC
PERSISTENT RESERVE OUT	5Fh	O	SPC
PORT STATUS	1Fh	O	SPC
READ BUFFER	3Ch	O	SPC
RECEIVE DIAGNOSTICS RESULTS	1Ch	O	SPC
REDUNDANCY GROUP(IN)	BAh	M	6.3
REDUNDANCY GROUP(OUT)	BBh	O	6.4
RELEASE(6)	17h	O	SPC
RELEASE(10)	57h	O	SPC
REPORT LUNS	A0h	M	SPC
REQUEST SENSE	03h	M	SPC
RESERVE(6)	16h	O	SPC
RESERVE(10)	56h	O	SPC
SEND DIAGNOSTIC	1Dh	O	SPC
SPARE(IN)	BCh	M	6.7
SPARE(OUT)	BDh	O	6.8
START STOP UNIT	1Bh	O	SPC
TEST UNIT READY	00h	M	SPC
VOLUME SET(IN)	BEh	M	6.5
VOLUME SET(OUT)	BFh	O	6.6
WRITE BUFFER	3Bh	O	SPC

Key: M = Command implementation is mandatory.
O = Command implementation is optional.
SPC = SCSI-3 Primary Commands Standard.

6.1 MAINTENANCE(IN) command

6.1.1 MAINTENANCE(IN) command service actions

The service actions for the MAINTENANCE(IN) command are listed in table 10.

Table 10 - Service actions for MAINTENANCE(IN) command

Service name	Service actions	Type	Subclause
REPORT ASSIGNED/UNASSIGNED P_EXTENT	00h	M	6.1.1.1
REPORT COMPONENT DEVICE	01h	M	6.1.1.2
REPORT COMPONENT DEVICE ATTACHMENTS	02h	M	6.1.1.3
REPORT PERIPHERAL DEVICE	03h	M	6.1.1.4
REPORT PERIPHERAL DEVICE ASSOCIATIONS	04h	M	6.1.1.5
REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER	05h	M	6.1.1.6
REPORT STATES	06h	M	6.1.1.7
Reserved	07h-17h		
Vendor specific	18h-1Fh		
Key: M = Service action implementation is mandatory. O = Service action implementation is optional.			

6.1.1.1 REPORT ASSIGNED/UNASSIGNED P_EXTENT service action

The REPORT ASSIGNED/UNASSIGNED P_EXTENT service action (see table 11) requests that information regarding assigned p_extents or unassigned p_extents within the target be sent to the application client. See 5.2.2.10 for a definition of assigned and unassigned p_extents.

Table 11 - REPORT ASSIGNED/UNASSIGNED P_EXTENT service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	RESERVED			SERVICE ACTION (00h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_P				
5								(LSB)
6	(MSB)			ALLOCATION LENGTH				
7								
8								
9								(LSB)
10	RESERVED					ASSIGN	RESERVED	RPTSEL
11	CONTROL							

The LUN_P field specifies the address of the peripheral device from which the p_extent(s) shall be reported per table 12. If the requested logical unit has not been added to the target the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

A report selected bit (RPTSEL) of zero indicates the target shall report all the assigned p_extent(s) or all the unassigned p_extents within the target. The LUN_P field shall be ignored when the RPTSEL bit is zero. A RPTSEL bit of one indicates the target shall report only the assigned p_extent(s) or the unassigned p_extents on the peripheral device indicated in the LUN_P field.

A report assigned p_extents bit (ASSIGN) of zero indicates the target shall report unassigned p_extents. A ASSIGN bit of one indicates the target shall report assigned p_extents.

The REPORT ASSIGNED/UNASSIGNED P_EXTENT parameter list (see table 12) contains a four-byte header that contains the length in bytes of the parameter list and a list of ASSIGNED/UNASSIGNED P_EXTENT DESCRIPTORS.

Table 12 - REPORT ASSIGNED/UNASSIGNED P_EXTENT parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1		ASSIGNED/UNASSIGNED P_EXTENT(S) LIST LENGTH (n-3)						
2								
3								(LSB)
	ASSIGNED/UNASSIGNED P_EXTENT(s) (if any)							
4		ASSIGNED/UNASSIGNED P_EXTENT DESCRIPTOR 0						
19								
	. . .							
n-15		ASSIGNED/UNASSIGNED P_EXTENT DESCRIPTOR X						
n								

The ASSIGNED/UNASSIGNED P_EXTENT(S) LIST LENGTH field specifies the length in bytes of the following ASSIGNED/UNASSIGNED P_EXTENT DESCRIPTOR(s).

The ASSIGNED/UNASSIGNED P_EXTENT DESCRIPTOR is defined in table 13.

Table 13 - Data format of ASSIGNED/UNASSIGNED P_EXTENT DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0		P_EXTENT DESCRIPTOR						
11								
12	RESERVED							
13	RESERVED							
14	PERIPHERAL DEVICE TYPE							
15	RESERVED	P_EXTENT STATE						

The P_EXTENT DESCRIPTOR is defined in table 14.

Table 14 - Data format of P_EXTENT DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				LUN_P				(LSB)
2	(MSB)							
3								
4				START LBA_P				
5								(LSB)
6	(MSB)							
7								
8				NUMBER OF LBA_P(S)				
9								(LSB)
10	(MSB)							
11				NUMBER OF BYTES PER LBA_P				(LSB)

The LUN_P field contains the address of the peripheral device that contains the p_extent.

The START LBA_P field contains the first addressable logical block address of the p_extent.

The NUMBER OF LBA_P(S) field contains the capacity of the p_extent in blocks.

The NUMBER OF BYTES PER LBA_P field contains the size, in bytes, of the blocks in the p_extent. A value of zero in the NUMBER OF BYTES PER LBA_P field shall indicate the NUMBER OF BYTES PER LBA_P is variable.

The PERIPHERAL DEVICE TYPE field contains the type of SCSI device that contains the p_extent defined by the P_EXTENT DESCRIPTOR. See the SCSI-3 Primary Commands Standard for a list of SCSI peripheral device types.

The P_EXTENT STATE field is defined in table 40.

6.1.1.2 REPORT COMPONENT DEVICE service action

The REPORT COMPONENT DEVICE service action (see table 15) requests that information regarding component device(s) within the target be sent to the application client.

Table 15 - REPORT COMPONENT DEVICE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	RESERVED			SERVICE ACTION (01h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_C				
5								(LSB)
6	(MSB)							
7				ALLOCATION LENGTH				
8								
9								(LSB)
10	RESERVED							RPTSEL
11	CONTROL							

The LUN_C field specifies the address of the component device that shall be reported per table 16. If the requested logical unit has not been added to the target, the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

A report selected bit (RPTSEL) of zero indicates the target shall report on all the component device(s) within the target. The LUN_C field shall be ignored when the RPTSEL bit is zero. A RPTSEL bit of one indicates the target shall report only on the component device indicated in the LUN_C field.

The REPORT COMPONENT DEVICE parameter list (see table 16) contains a four-byte header that contains the length in bytes of the parameter list and a list of COMPONENT DEVICE DESCRIPTORS.

Table 16 - REPORT COMPONENT DEVICE parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				COMPONENT DEVICE LIST LENGTH (n-3)				
2								
3								(LSB)
	COMPONENT DEVICE(S) (if any)							
4				COMPONENT DEVICE DESCRIPTOR 0				
7								
	. . .							
n-3				COMPONENT DEVICE DESCRIPTOR x				
n								

The COMPONENT DEVICE LIST LENGTH field specifies the length in bytes of the following COMPONENT DEVICE DESCRIPTOR(s).

The COMPONENT DEVICE DESCRIPTOR is defined in table 17.

Table 17 - Data format of COMPONENT DEVICE DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	COMPONENT DEVICE TYPE							
1	REPLACE	COMPONENT DEVICE STATE						
2	(MSB)			LUN_C				
3								(LSB)

The COMPONENT DEVICE TYPE field contains the type of component device. See table 18 for a list of component device types.

The component device types table (see table 18) contains a list of non-data type devices the target may address as component devices.

TABLE 18 - COMPONENT DEVICE TYPES

Code	Description
00h	Controller electronics that contain a SACL
01h	Non-volatile cache
02h	Power supply
03h	Uninterruptable power supply
04h	Display
05h	Key pad entry
06h	Fan
07h-7Fh	Reserved
80h-FFh	Vendor specific

The component device STATE field is defined in table 42.

A replace bit (REPLACE) of zero indicates the component device indicated in the LUN_C field is not a replacable unit. A replace bit of one indicates the component device indicated in the LUN_C field is a replaceable unit.

The LUN_C field contains the address of the component device.

6.1.1.3 REPORT COMPONENT DEVICE ATTACHMENTS service

The REPORT COMPONENT DEVICE ATTACHMENTS service action (see table 19) requests that information regarding logical units that are attached to component device(s) be sent to the application client.

Table 19 - REPORT COMPONENT DEVICE ATTACHMENTS service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	RESERVED			SERVICE ACTION (02h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_C				
5								(LSB)
6	(MSB)			ALLOCATION LENGTH				
7								
8								
9								(LSB)
10	RESERVED							RPTSEL
11	CONTROL							

The LUN_C field specifies the address of the component device for which the target shall report information per table 20. If the requested logical unit has not been added to the target the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

A report selected bit (RPTSEL) of zero indicates the target shall report on all the component device(s) within the target. The LUN_C field shall be ignored when the RPTSEL bit is zero. A RPTSEL bit of one indicates the target shall report only on the component device indicated in the LUN_C field.

The REPORT COMPONENT DEVICE ATTACHMENTS parameter list (see table 20) contains a four-byte header that contains the length in bytes of the parameter list and a list of COMPONENT DEVICE ATTACHMENT DESCRIPTORs.

Table 20 - REPORT COMPONENT DEVICE ATTACHMENTS parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1		COMPONENT DEVICE ATTACHMENT LIST LENGTH (n-3)						
2								
3								(LSB)
	COMPONENT DEVICE ATTACHMENT(S) (if any)							
4		COMPONENT DEVICE ATTACHMENT DESCRIPTOR (First) (Length x)						
x+3								
	. . .							
n-y+1		COMPONENT DEVICE ATTACHMENT DESCRIPTOR (Last) (Length y)						
n								

The COMPONENT DEVICE ATTACHMENT LIST LENGTH field specifies the length in bytes of the following COMPONENT DEVICE ATTACHMENT DESCRIPTOR(s).

The COMPONENT DEVICE ATTACHMENT DESCRIPTOR is defined in table 21.

Table 21 - Format of COMPONENT DEVICE ATTACHMENT DESCRIPTOR

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			LUN_C				
1								(LSB)
2	(MSB)			LOGICAL UNIT LIST LENGTH (n-3)				
3								(LSB)
	LOGICAL UNIT (S) (if any)							
4				LOGICAL UNIT DESCRIPTOR 0				
7								
	. . .							
n-3				LOGICAL UNIT DESCRIPTOR X				
n								

The LUN_C field specifies the address of the component device to which the LOGICAL UNIT DESCRIPTOR(s) listed in the COMPONENT DEVICE ATTACHMENT DESCRIPTOR list are attached.

The LOGICAL UNIT LIST LENGTH field specifies the length In bytes of the following LOGICAL UNIT DESCRIPTOR(s).

The LOGICAL UNIT DESCRIPTOR(S) contain a list of logical units that are attached to the component device addressed in the LUN_C field of the COMPONENT DEVICE ATTACHMENT DESCRIPTOR. See table 22 for the format of the LOGICAL UNIT DESCRIPTOR.

Table 22 - Data format of LOGICAL UNIT DESCRIPTOR

Bit / Byte	7	6	5	4	3	2	1	0
0	RESERVED							
1	RESERVED				LOGICAL UNIT TYPE			
2	(MSB)			LUN				
3								(LSB)

The LOGICAL UNIT TYPE field (see table 23) indicates the type of logical unit addressed in the LUN field.

Table 23 - LOGICAL UNIT types

Codes	Description
0h	Physical logical unit (peripheral device)
1h	Volume set
2h-3h	Reserved
4h	Component logical unit (component device)
5h	Redundancy group
6h	Spare
7h-Bh	Reserved
Ch-Fh	Vendor specific

The LUN field contains the logical unit number of the logical unit indicated by the LOGICAL UNIT TYPE field.

6.1.1.4 REPORT PERIPHERAL DEVICE service action

The REPORT PERIPHERAL DEVICE service action (see table 24) requests that information regarding peripheral device(s) within the target be sent to the application client.

Table 24 - REPORT PERIPHERAL DEVICE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	RESERVED			SERVICE ACTION (03h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_P				
5								(LSB)
6	(MSB)							
7				ALLOCATION LENGTH				
8								
9								(LSB)
10	RESERVED				RPTMBUS	RESERVED	SELECT REPORT	
11	CONTROL							

The LUN_P field specifies the address of the peripheral device that shall be reported per table 26. If the requested logical unit has not been added to the target, the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

The SELECT REPORT field contains the information on which peripheral device(s) the target shall report. See table 25 for the defined states.

TABLE 25 - SELECT REPORT

Codes	Description
00b	The target shall report on all the peripheral device(s) within the target. The LUN_P field shall be ignored if this option is selected.
01b	The target shall report only on the peripheral device indicated in the LUN_P field.
10b	The target shall report all the peripheral device(s) within the target that have the state of not available. The LUN_P field shall be ignored if this option is selected.
11b	Reserved

A report multiple buses (RPTMBUS) bit of zero indicates only one LUN_P shall be reported for each peripheral device indicated by the SELECT REPORT field. A RPTMBUS bit of one indicates all LUN_P(s) shall be reported for each peripheral device indicated by the SELECT REPORT field.

The REPORT PERIPHERAL DEVICE parameter list (see table 26) contains a four-byte header that contains the length in bytes of the parameter list and a list of PERIPHERAL DEVICE DESCRIPTORS.

Table 26 - REPORT PERIPHERAL DEVICE parameter List

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				PERIPHERAL DEVICE LIST LENGTH (n-3)				
2								
3								(LSB)
	PERIPHERAL DEVICE(S) (if any)							
4				PERIPHERAL DEVICE DESCRIPTOR 0				
7								
	. . .							
n-3				PERIPHERAL DEVICE DESCRIPTOR X				
n								

The PERIPHERAL DEVICE LIST LENGTH field specifies the length in bytes of the following PERIPHERAL DEVICE DESCRIPTOR(s).

The PERIPHERAL DEVICE DESCRIPTOR is defined in table 27.

Table 27 - Format of PERIPHERAL DEVICE DESCRIPTOR

Bit / Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL DEVICE TYPE							
1	REPLACE	PERIPHERAL DEVICE STATE						
2	(MSB)	LUN_P						
3								(LSB)

The PERIPHERAL DEVICE TYPE field contains the type of SCSI device. See the SCSI-3 Primary Commands Standard for a list of SCSI peripheral device types.

The peripheral device STATE field is defined in table 40.

A replace bit (REPLACE) of zero indicates the peripheral device indicated in the LUN_P field is not a replacable unit. A replace bit of one indicates the peripheral device indicated in the LUN_P field is a replaceable unit.

The LUN_P field contains the address of the peripheral device.

6.1.1.5 REPORT PERIPHERAL DEVICE ASSOCIATIONS service action

The REPORT PERIPHERAL DEVICE ASSOCIATIONS service action (see table 28) requests that information regarding logical units that are associated with peripheral device(s) be sent to the application client.

Table 28 - REPORT PERIPHERAL DEVICE ASSOCIATIONS service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	RESERVED			SERVICE ACTION (04h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_P				
5								(LSB)
6	(MSB)			ALLOCATION LENGTH				
7								
8								
9								(LSB)
10	RESERVED				RPTMBUS	RESERVED	RESERVED	RPTSEL
11	CONTROL							

The LUN_P field specifies the address of the peripheral device that the target shall report information as to which logical unit(s) are associated with the peripheral device per table 30. If the requested logical unit has not been added to the target the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

A report selected bit (RPTSEL) of zero indicates the target shall report on all the peripheral device(s) within the target. The LUN_P field shall be ignored when the RPTSEL bit is zero. A RPTSEL bit of one indicates the target shall report only on the peripheral device indicated in the LUN_P field.

A report multiple buses (RPTMBUS) bit of zero indicates only one LOGICAL UNIT DESCRIPTOR (see table 30) shall be reported for each object associated with the peripheral device listed in the LUN_P field (see table 30). A RPTMBUS bit of one indicates all LOGICAL UNIT DESCRIPTOR(s) (see table 30) shall be reported for each object associated with the peripheral device listed in the LUN_P field (see table 30).

The REPORT PERIPHERAL DEVICE ASSOCIATIONS parameter list (see table 29) contains a four-byte header that contains the length in bytes of the parameter list and a list of PERIPHERAL DEVICE ASSOCIATION DESCRIPTORs.

Table 29 - REPORT PERIPHERAL DEVICE ASSOCIATIONS parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1		PERIPHERAL DEVICE ASSOCIATIONS LIST LENGTH (n-3)						
2								
3								(LSB)
	PERIPHERAL DEVICE ASSOCIATION(s) (if any)							
4		PERIPHERAL DEVICE ASSOCIATIONS DESCRIPTOR (First) (Length x)						
x+3								
	. . .							
n-y+1		PERIPHERAL DEVICE ASSOCIATIONS DESCRIPTOR (Last) (Length y)						
n								

The PERIPHERAL DEVICE ASSOCIATIONS DESCRIPTOR LIST LENGTH field specifies the length in bytes of the following PERIPHERAL DEVICE ASSOCIATIONS DESCRIPTOR(s).

The PERIPHERAL DEVICE ASSOCIATIONS DESCRIPTOR is defined in table 30.

Table 30 - Format of PERIPHERAL DEVICE ASSOCIATIONS DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
2				LUN_P				(LSB)
3	(MSB)							
4				LOGICAL UNIT LIST LENGTH (n-3)				(LSB)
				LOGICAL UNIT(S) (if any)				
5								
8				LOGICAL UNIT DESCRIPTOR 0				
				. . .				
n-3								
n				LOGICAL UNIT DESCRIPTOR X				

The LUN_P field specifies the address of the peripheral device to which the LOGICAL UNIT DESCRIPTOR(s) listed in the PERIPHERAL DEVICE ASSOCIATIONS DESCRIPTOR list are associated.

The LOGICAL UNIT LIST LENGTH field specifies the length in bytes of the following LOGICAL UNIT DESCRIPTOR(s).

The LOGICAL UNIT DESCRIPTOR(s) contain a list of logical units that are associated with the peripheral device addressed in the LUN_P field of the PERIPHERAL DEVICE ASSOCIATIONS DESCRIPTOR. See table 22 for the format of the LOGICAL UNIT DESCRIPTOR.

6.1.1.6 REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action

The REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action (see table 31) requests the location of the selected logical unit be sent to the application client.

Table 31 - REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	RESERVED			SERVICE ACTION (05h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN				
5								(LSB)
6	(MSB)			ALLOCATION LENGTH				
7								
8								
9								(LSB)
10	RESERVED						PORCLU	RESERVED
11	CONTROL							

The LUN field contains the logical unit number of the peripheral device or component device that the target shall report per table 32. If the requested logical unit has not been added to the target the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

A report physical or component logical unit bit (PORCLU) of zero indicates the LUN field shall contain the address of a peripheral device. A PORCLU bit of one indicates the LUN field shall contain the address of a component device.

The REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER parameter list (see table 32) contains a four-byte field that contains the length in bytes of the parameter list and the location of the selected logical unit.

Table 32 - REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1								
2				LOCATION IDENTIFIER LIST LENGTH				
3								(LSB)
4								
n				LOCATION IDENTIFIER				

The LOCATION IDENTIFIER LIST LENGTH field specifies the length in bytes of the IDENTIFIER field.

The LOCATION IDENTIFIER field shall be an ASCII value that indicates the position of the peripheral device or component device within the target. The ASCII value within the IDENTIFIER field is vendor specific.

6.1.1.7 REPORT STATES service action

The REPORT STATES service action (see table 33) requests that state information about the selected logical unit(s) within the target be sent to the application client.

Table 33 - REPORT STATES service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	RESERVED			SERVICE ACTION (06h)				
2	RESERVED							
3	RESERVED				LOGICAL UNIT TYPE			
4	(MSB)			LUN				
5								(LSB)
6	(MSB)			ALLOCATION LENGTH				
7								
8								
9								(LSB)
10	RESERVED		REPORT STATES		RESERVED			
11	CONTROL							

The LOGICAL UNIT TYPE field contains the type of logical unit designated by the LUN field. See table 23 for the list of logical unit types. If the requested logical unit has not been added to the target the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

The LUN field specifies the address of the logical unit the target shall report state information per table 35.

The REPORT STATES field contains the information on which logical unit(s) the target shall report state information. See table 34 for the defined states.

Table 34 - REPORT STATES

Codes	Description
00b	Report all states for all logical units within the selected target. The LOGICAL UNIT TYPE and the LUN fields shall be ignored if this option is selected.
01b	Report all states for all of the logical unit(s) of the type listed in the LOGICAL UNIT TYPE field within the selected target. The LUN field shall be ignored if this option is selected.
10b	Report all states for the selected logical unit. The LOGICAL UNIT TYPE and the LUN field shall designate the address of the logical unit if this option is selected.
11b	Reserved

The REPORT STATES parameter list (see table 35) contains a four-byte header that indicates the length in bytes of the parameter list plus a list of LOGICAL UNIT STATE DESCRIPTORs.

Table 35 - REPORT STATES parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				LOGICAL UNIT STATES LIST LENGTH (n-3)				
2								
3								(LSB)
	LOGICAL UNIT STATE(S) (if any)							
4				LOGICAL UNIT STATE(S) DESCRIPTOR (First) (Length x)				
x+3								
	. . .							
n-y+1				LOGICAL UNIT STATE(S) DESCRIPTOR (Last) (Length y)				
n								

The LOGICAL UNIT LIST LENGTH field specifies the length in bytes of the following LOGICAL UNIT STATE(S) DESCRIPTOR(s).

The LOGICAL UNIT STATE(S) DESCRIPTOR is defined in table 36.

Table 36 - Format of LOGICAL UNIT STATES DESCRIPTORS

Bit / Byte	7	6	5	4	3	2	1	0
0	DEVICE TYPE							
1	RESERVED				LOGICAL UNIT TYPE			
2	(MSB)			LUN				
3								(LSB)
4	RESERVED							
5	RESERVED							
6	(MSB)			STATE LIST LENGTH (n-7)				
7								(LSB)
	STATE DESCRIPTORS(S) (if any)							
8	REPLACE	STATE OF THE LOGICAL UNIT						
	. . .							
n	REPLACE	STATE OF THE LOGICAL UNIT						

The DEVICE TYPE field shall contain a component device type if the LOGICAL UNIT TYPE field indicates a component device type. See table 18 for a list of component device types. If the LOGICAL UNIT TYPE field does not indicate a component device type the DEVICE TYPE field shall contain a peripheral device type. See the SCSI-3 Primary Commands Standard for a list of SCSI peripheral device types.

The LOGICAL UNIT TYPE field contains the type of logical unit contained within the LUN field. See table 23 for the list of logical unit types.

The LUN field contains the address of the logical unit the target is reporting state information about.

The STATE LIST LENGTH field specifies the length in bytes of the following STATE DESCRIPTOR(s).

The STATE OF THE LOGICAL UNIT field specifies the state of logical unit addressed by the LUN field. The contents of the STATE OF THE LOGICAL UNIT field depends on the logical unit type field. See table 37, table 38, table 39, table 40, table 41, and table 42 for a definition of the states by logical unit type that shall be reported to the application client.

The order in which states are reported is vendor specific.

A replace bit (REPLACE) of zero shall indicate the logical unit is not a replacable unit if the LOGICAL UNIT TYPE field indicates the logical unit is a peripheral device or component device. A REPLACE bit of one shall indicate the logical unit is a replacable unit if the LOGICAL UNIT TYPE field indicates the logical unit is a peripheral device or component device. The target shall not set the REPLACE bit to one unless the LOGICAL UNIT TYPE field is either peripheral device or component device. The definition of replacable is vendor specific.

Table 37 - Target base device (LUN 0) states

Bit	7	6	5	4	3	2	1	0
	RESERVED	VS	RESERVED			ABNORMAL	NONAFAIL	READYING

A readying bit (READYING) of zero indicates there are no logical units within the target that have a state of readying. A READYING bit of one indicates one or more logical units within the target are being initialized and access to the target is limited.

The amount of accessibility of a target during the readying state is vendor-specific.

A non-addressable component failure bit (NONAFAIL) of zero indicates that all non-addressable parts are operational. A NONAFAIL bit of one indicates one or more non-addressable part(s) have failed. (e.g., power supply failure, LED failure, cache failure, etc,. that are not defined as component devices).

NOTE 13 - More information on the failure may be available within the sense data from a REQUEST SENSE command issued to the target's base device address.

An abnormal bit (ABNORMAL) of zero indicates that all addressable devices within the target have a state of available. An ABNORMAL bit of one indicates that one or more addressable devices within the target are indicating a state other than available.

NOTE 14 - To determine which device is indicating it is not available, issue a state request to each addressable device within the target.

The vendor specific (VS) bit indicates a vendor specific state.

Table 38 - Volume set states

Codes	States	Description
00h	Available	The addressed volume set is operational.
01h	Broken	The addressed volume set is capable of being supported but it has failed.
02h	Data lost	Within the addressed volume set data has been lost.
03h	Exposed	Within the addressed volume set data is not protected. In this state all data is still valid, however, a failure causes a loss of data or a loss of data availability.
0Eh	Fractionally exposed	Within the addressed volume set data is not protected. In this state all data is still valid, however, a failure may causes a loss of data or a loss of data availability.
04h	Partially exposed	Within the addressed volume set one or more logical unit(s) have failed. In this state all data is still protected.
05h	Protected rebuild	One or more of the redundancy groups underlaying the addressed volume set is in the process of a rebuild operation. In this state all data is protected.
0Ch	Protection disabled	Within the addressed volume set the generation of check data has been disabled. In this state all data is still valid, however, a failure causes a loss of data or a loss of data availability.
06h	Not available	The addressed volume set is capable of being supported but has not been configured.
07h	Not supported	The addressed volume set is not capable of being configured.
08h	Readying	The addressed volume set is being initialized and access to the volume set is limited. NOTE 15 - The amount of accessibility to the volume set during the readying state is vendor-specific. This state should be indicated if any of the logical units within the volume set are not ready. Intervention may be required by the application client to remove this state. (e.g., START UNIT command)
09h	Rebuild	One or more of the underlaying redundancy groups of the addressed volume set is in the process of a rebuild operation. In this state data is not protected.
0Ah	Recalculate	The addressed volume set is in the process of a recalculate operation. NOTE 16 - The recalculate operation may involve one 0r more underlaying redundancy groups.
0Bh	Spare in use	Within the addressed volume set a spare is being used. In this state all data is still protected.
0Dh	Verify in progress	Within the addressed volume set data is being verified. NOTE 17 - The verify operation may involve one or more underlaying redundancy groups.
0Eh-3Fh	Reserved	
40h-7Fh	Vendor specific	

Table 39 - Redundancy group states

Codes	States	Description
00h	Available	The addressed redundancy group is configured.
01h	Exposed	Within the addressed redundancy group data is not protected. In this state all data is still valid, however, a failure causes a loss of data or a loss of data availability.
02h	Invalidated protected space	Within the addressed redundancy group data has been lost. In this state the protected space is no longer intact.
03h	Not available	The addressed redundancy group is capable of being supported but has not been configured.
04h	Not supported	The addressed redundancy group is not capable of being configured.
05h	Partially exposed	Within the addressed redundancy group one or more logical unit(s) have failed. In this state the protected space is protected.
06h	Present	The addressed redundancy group is present but no other status is available.
07h	Protected rebuild	The addressed redundancy group is in the process of a rebuild operation. In this state the protected space is protected.
0Ah	Protection disabled	Within the addressed redundancy group the generation of check data has been disabled. In this state all data is still valid, however, a failure causes a loss of data or a loss of data availability.
08h	Rebuild	The addressed redundancy group is in the process of a rebuild operation. In this state the protected space is not protected.
09h	Recalculate	The addressed redundancy group is in the process of a recalculate operation.
0Bh	Verify in progress	Within the addressed redundancy group data is being verified.
0Ch-3Fh	Reserved	
40h-7Fh	Vendor specific	

Table 40 - Peripheral device and p_extent states

Codes	States	Description
00h	Available	The addressed peripheral device or p_extent is operational.
01h	Broken	The addressed peripheral device or p_extent is capable of being supported but it has failed.
02h	Not available	The addressed peripheral device or p_extent is capable of being supported but no device is connected.
03h	Not supported	The target is not capable of supporting a device at the addressed peripheral device or p_extent.
04h	Present	The addressed peripheral device or p_extent is present but no other status is available.
05h	Readying	The addressed peripheral device or p_extent is being initialized and access to the peripheral device or p_extent is limited.
06h	Rebuild	The addressed peripheral device or p_extent is being rebuilt.
07h-3Fh	Reserved	
40h-7Fh	Vendor specific	

Table 41 - Spare states

Codes	States	Description
00h	Available	The addressed spare is operational.
01h	Broken	The addressed spare is capable of being supported but it has failed.
02h	Not available	The addressed spare is capable of being supported but has not been configured.
03h	Not supported	The addressed spare is not capable of being configured.
04h	Present	The addressed spare is present but no other status is available.
05h	Spare in use	The addressed spare has covered an object (see 5.2.2.4).
06h-3Fh	Reserved	
40h-7Fh	Vendor specific	

Table 42 - Component device states

Codes	States	Description
00h	Available	The addressed component device is fully operational.
01h	Broken	The addressed component device is capable of being supported but it has failed.
02h	Reserved	
03h	ITTU	The addressed component device is the reporting component device. This state shall not be reported unless the command allows the reporting of multiple states. More that one component device may report an ITTU state in a single state request.
04h	Not available	The addressed component device is capable of being supported but no component is present.
05h	Not supported	The target is not capable of supporting a component at the given address.
06h	Present	The addressed component device is present but no other status is available.
07h	Readying	The addressed component device is being initialized and access to the component device is limited.
08h-3Fh	Reserved	
40h-7Fh	Vendor specific	

6.2 MAINTENANCE (OUT) commands

6.2.1 MAINTENANCE (OUT) command service actions

The service actions for the MAINTENANCE(OUT) command are listed in table 43.

Table 43 - Service actions for MAINTENANCE (OUT) command

Service name	Service actions	Type	Subclause
ADD PERIPHERAL DEVICE/COMPONENT DEVICE	00h	O	6.2.1.1
ATTACH TO COMPONENT DEVICE	01h	O	6.2.1.2
BREAK PERIPHERAL DEVICE/COMPONENT DEVICE	07h	O	6.2.1.3
EXCHANGE P_EXTENT	02h	O	6.2.1.4
EXCHANGE PERIPHERAL DEVICE/COMPONENT DEVICE	03h	O	6.2.1.5
INSTRUCT COMPONENT DEVICE	04h	O	6.2.1.6
REMOVE PERIPHERAL DEVICE/COMPONENT DEVICE	05h	O	6.2.1.7
SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER	06h	O	6.2.1.8
Reserved	08h-17h		
Vendor specific	18h-1Fh		
Key: M = Service action implementation is mandatory. O = Service action implementation is optional.			

6.2.1.1 ADD PERIPHERAL DEVICE/COMPONENT DEVICE service action

The ADD PERIPHERAL DEVICE/COMPONENT DEVICE service action (see table 44) requests a peripheral device or a component device be added to the target. If the add operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to ADD LOGICAL UNIT FAILED.

Table 44 - ADD PERIPHERAL DEVICE/COMPONENT DEVICE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A4h)							
1	RESERVED			SERVICE ACTION (00h)				
2	DEVICE TYPE							
3	RESERVED							
4	(MSB)			LUN				
5								(LSB)
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
10	RESERVED	SETLUN	RESERVED				ADDPORC	RESERVED
11	CONTROL							

The DEVICE TYPE field contains the peripheral device type or the component device type of the device that the target shall add. See the SCSI-3 Primary Commands Standard for a list of valid peripheral device types. See table 18 for a list of valid component device types.

The LUN field contains the logical unit number to assign to the peripheral device or component device that shall be added to the target if the SETLUN bit is set to zero. If the requested logical unit cannot be added to the target the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

An add physical or component logical unit bit (ADDPORC) of zero indicates the DEVICE TYPE field shall contain a valid peripheral device type. See the SCSI-3 Primary Commands Standard for a list of valid peripheral device types. An ADDPORC bit of one indicates the DEVICE TYPE field shall contain a valid component device type. See table 18 for a list of valid component device types.

A set LUN (SETLUN) bit of zero indicates the target shall assign to the peripheral device or component device the logical unit number contained in the LUN field. A SETLUN bit of one indicates that the target shall assign a logical unit number to the peripheral device or component device. The LUN field shall be ignored when the SETLUN bit is set to one.

6.2.1.2 ATTACH TO COMPONENT DEVICE service action

The ATTACH TO COMPONENT DEVICE service action (see table 45) requests the target logically attach one or more logical unit(s) to a component device. If the attach operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to ATTACHMENT OF LOGICAL UNIT FAILED.

NOTE 18 - The behavior of attached logical units and their interactions with component devices are vendor specific.

To detach a logical unit from a component device, issue the ATTACH TO COMPONENT DEVICE service action with the logical unit(s) to be detached removed from the parameter list. If the LIST LENGTH field is set to zero then all logical unit(s) attached to the component device shall be detached.

NOTE 19 - If LUN_C 1 is attached to LUN_C 2 and LUN_C 2 is attached to LUN_C 1 then a detach request to LUN_C 2 shall only detach LUN_C 1 from LUN_C 2. The attachment of LUN_C 2 to LUN_C 1 remains intact.

Table 45 - ATTACH COMPONENT DEVICE service actions

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A4h)							
1	RESERVED			SERVICE ACTION (01h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_C				
5								(LSB)
6	(MSB)			LIST LENGTH				
7								
8								
9								(LSB)
10	RESERVED							
11	CONTROL							

The LUN_C field specifies the address of the component device to which the target shall attach the logical unit(s) listed in the ATTACH COMPONENT DEVICE parameter list (see table 46). If the requested logical unit has not been added to the target the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

Table 46 - ATTACH COMPONENT DEVICE parameter list

Bit Byte	7	6	5	4	3	2	1	0
	LOGICAL UNIT(S) (if any)							
0	LOGICAL UNIT DESCRIPTOR 0							
3								
	. . .							
n-3	LOGICAL UNIT DESCRIPTOR X							
n								

The LOGICAL UNIT DESCRIPTOR(s) contain a list of logical units that shall be attached to the component device addressed by the ATTACH COMPONENT DEVICE service action. See table 22 for the format of the LOGICAL UNIT DESCRIPTOR.

6.2.1.3 BREAK PERIPHERIAL DEVICE/COMPONENT DEVICE service action

The BREAK PERIPHERAL DEVICE/COMPONENT DEVICE service action (see table 47) requests a peripheral device or a component device be placed into a broken state.

Table 47 - BREAK PERIPHERAL DEVICE/COMPONENT DEVICE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A4h)							
1	RESERVED			SERVICE ACTION (07h)				
2	DEVICE TYPE							
3	RESERVED							
4	(MSB)			LUN				
5								(LSB)
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
10	RESERVED						BRKPORC	RESERVED
11	CONTROL							

The DEVICE TYPE field contains the peripheral device type or the component device type of the device that the target shall break.

The LUN field contains the logical unit number of the peripheral device or component device that shall be broken. If the requested logical unit has not been added to the target the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

A break physical or component logical unit bit (BRKPORC) of zero indicates the LUN field contains the address of a peripheral device and the DEVICE TYPE field contains a valid peripheral device type. See the SCSI-3 Primary Commands Standard for a list of valid peripheral device types. A BRKPORC bit of one indicates the LUN field shall contain the address of a component device and the DEVICE TYPE field contains a valid component device type. See table 18 for a list of valid component device types.

6.2.1.4 EXCHANGE P_EXTENT service action

The EXCHANGE P_EXTENT service action (see table 48) requests the target replace a p_extent with another p_extent. If the exchange operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to EXCHANGE OF LOGICAL UNIT FAILED.

Table 48 - EXCHANGE P_EXTENT service action

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A4h)							
1	RESERVED			SERVICE ACTION (02h)				
2	RESERVED							
3	RESERVED							
4	RESERVED							
5	RESERVED							
6	(MSB)							
7				LIST LENGTH				
8								
9								(LSB)
10	RESERVED							IMMED
11	CONTROL							

An immediate (IMMED) bit of zero indicates that status shall be returned after the exchange operation has completed. An IMMED bit value of one indicates that the target shall return status as soon as the command descriptor block has been validated, and the entire EXCHANGE P_EXTENT parameters list has been transferred.

The EXCHANGE P_EXTENT parameters list (see table 49) contains the addresses of the logical units that shall be exchanged.

Table 49 - EXCHANGE P_EXTENT parameters list

Bit / Byte	7	6	5	4	3	2	1	0
0				OLD P_EXTENT DESCRIPTOR				
11								
12				NEW P_EXTENT DESCRIPTOR				
23								

The OLD P_EXTENT DESCRIPTOR contains the p_extent that shall be replaced by a new p_extent. See table 14 for the format of the OLD P_EXTENT DESCRIPTOR.

The NEW P_EXTENT DESCRIPTOR contains the p_extent that shall replace an old p_extent. See table 14 for

the format of the NEW P_EXTENT DESCRIPTOR. If the old p_extent and the new p_extent are not the same size or have different peripheral device types the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

It is not an error for the old p_extent and new p_extenT to address the same p_extent.

6.2.1.5 EXCHANGE PERIPHERAL DEVICE/COMPONENT DEVICE service action

The EXCHANGE PERIPHERAL DEVICE/COMPONENT DEVICE service action (see table 50) requests that the target replace one logical unit with another logical unit.

Table 50 - EXCHANGE PERIPHERAL DEVICE/COMPONENT DEVICE Service Action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A4h)							
1	RESERVED			SERVICE ACTION (03h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			OLD LUN				
5								(LSB)
6	RESERVED							
7	RESERVED							
8	(MSB)			NEW LUN				
9								(LSB)
10	RESERVED						EXPORC	IMMED
11	CONTROL							

The OLD LUN field contains the logical unit number of the peripheral device or component device that the target shall exchange with the logical unit addressed in the NEW LUN field.

The NEW LUN field contains the logical unit number of the peripheral device or component device that the target shall exchange with the logical unit addressed in the OLD LUN field. If the old logical unit or the new logical unit has not been added to the target the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED

It is not an error for the OLD LUN and NEW LUN fields to address the same logical unit.

An immediate (IMMED) bit of zero indicates that status shall be returned after the exchange operation has completed. An IMMED bit value of one indicates that the target shall return status as soon as the command descriptor block has been validated.

An exchange physical or component logical unit bit (EXPORC) of zero indicates the OLD LUN and NEW LUN fields shall contain the addresses of peripheral devices. A EXPORC bit of one indicates the OLD LUN and NEW LUN fields shall contain the addresses of component devices.

6.2.1.6 INSTRUCT COMPONENT DEVICE service action

The INSTRUCT COMPONENT DEVICE service action (see table 51) requests the target to take the requested action on the addressed component device.

Table 51 - INSTRUCT COMPONENT DEVICE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A4h)							
1	RESERVED			SERVICE ACTION (04h)				
2	COMPONENT DEVICE INSTRUCTION							
3	RESERVED							
4	(MSB)	LUN_C						
5								(LSB)
6	(MSB)	LIST LENGTH						
7								
8								
9								(LSB)
10	RESERVED							
11	CONTROL							

The COMPONENT DEVICE INSTRUCTION field contains the action to be taken. See table 52 for a list of actions.

Table 52 - COMPONENT DEVICE INSTRUCTION field

Codes	Description
00h	Turn selected component device off
01h	Turn selected component device on
02h-7Fh	Reserved
80h-FFh	Vendor specific

The LUN_C field specifies the address of the component device to which the action shall be applied. If the requested logical unit has not been added to the target the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

A list length of zero shall indicate the INSTRUCT COMPONENT DEVICE service action contains no parameter list.

The INSTRUCT COMPONENT DEVICE parameter list (see table 53) contains vendor specific actions to be applied to the component device addressed in the INSTRUCT COMPONENT DEVICE service action command descriptor block.

Table 53 - INSTRUCT COMPONENT DEVICE parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	VENDOR SPECIFIC							
n								

6.2.1.7 REMOVE PERIPHERAL DEVICE/COMPONENT DEVICE service action

The REMOVE PERIPHERAL DEVICE/COMPONENT device service action (see table 54) requests a peripheral device or a component device be removed from the target.

Table 54 - REMOVE PERIPHERAL DEVICE/COMPONENT DEVICE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A4h)							
1	RESERVED			SERVICE ACTION (05h)				
2	RESERVED							
3	RESERVED							
4	(MSB)	LUN						
5								(LSB)
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
10	RESERVED						REMPORC	RESERVED
11	CONTROL							

The LUN field contains the logical unit number of the peripheral device or component device that shall be removed from the target. If the requested logical unit has not been added to the target the command shall

be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

A remove physical or component logical unit bit (REMPORC) of zero indicates the LUN field shall contain the address of a peripheral device. A REMPORC bit of one indicates the LUN field shall contain the address of a component device.

If the peripheral device contains any assigned p_extents or if any logical units are attached to the component device being removed the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to REMOVE OF LOGICAL UNIT FAILED.

6.2.1.8 SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action

The SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action (see table 55) requests the target identify the addressed logical unit with the location received in the SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER parameter list.

Table 55 - SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A4h)							
1	RESERVED			SERVICE ACTION (06h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN				
5								(LSB)
6	(MSB)			LIST LENGTH				
7								
8								
9								(LSB)
10	RESERVED						IDPORC	RESERVED
11	CONTROL							

The LUN field contains the logical unit number of the peripheral device or component device that shall be identified by the LOCATION IDENTIFIER field received in the SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER parameter list. If the requested logical unit has not been added to the target the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

A set identification peripheral device or component device bit (IDPORC) of zero indicates the LUN field shall contain the address of a peripheral device. A IDPORC bit of one indicates the LUN field shall contain the address of a component device.

The SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER parameter list (see table 56) contains the location identifier of the addressed logical unit.

Table 56 - SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	LOCATION IDENTIFIER							
n								

The LOCATION IDENTIFIER field shall be an ASCII value that indicates the position of the peripheral device or component device within the target. The ASCII value within the IDENTIFIER field is vendor specific.

6.3 REDUNDANCY GROUP (IN) command

6.3.1 REDUNDANCY GROUP (IN) command service actions

The service actions for the REDUNDANCY GROUP (IN) command are listed in table 57.

Table 57 - Service actions for REDUNDANCY GROUP (IN) command

Service name	Service actions	Type	Subclause
REPORT REDUNDANCY GROUPS	00h	M	6.3.1.1
REPORT UNASSIGNED REDUNDANCY GROUP SPACE	01h	M	6.3.1.2
RESERVED	09h-17h		
VENDOR SPECIFIC	18h-1Fh		
Key: M = Service action implementation is mandatory. O = Service action implementation is optional.			

6.3.1.1 REPORT REDUNDANCY GROUPS service action

The REPORT REDUNDANCY GROUPS service action (see table 58) requests that information regarding redundancy groups within the target be sent to the application client.

NOTE 20 - The information returned by the REPORT REDUNDANCY GROUPS service action may be used by the application client to determine the boundaries of all the current redundancy groups to allow more redundancy groups to be configured or to allow a redundancy group to be expanded.

Table 58 - REPORT REDUNDANCY GROUPS service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BAh)							
1	RESERVED			SERVICE ACTION (00h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_R				
5								(LSB)
6	(MSB)							
7				ALLOCATION LENGTH				
8								
9								(LSB)
10	RESERVED							RPTSEL
11	CONTROL							

The LUN_R field specifies the address of the redundancy group for which information shall be reported per table 59. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED.

A report selected bit (RPTSEL) of zero indicates the target shall report on all the redundancy group(s) within the target. The LUN_R field shall be ignored when the RPTSEL bit is zero. A RPTSEL bit of one indicates the target shall only report information on the redundancy group addressed by the LUN_R field.

The REPORT REDUNDANCY GROUPS parameter list (see table 59) contains a four-byte header that defines the length in bytes of the parameter list and a list of REPORT REDUNDANCY GROUP DESCRIPTORS.

Table 59 - REPORT REDUNDANCY GROUPS parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				REPORT REDUNDANCY GROUP LIST LENGTH (n-3)				
2								
3								(LSB)
	REPORT REDUNDANCY GROUP DESCRIPTORS(S) (if any)							
4				REPORT REDUNDANCY GROUP DESCRIPTOR (First)				
x+3				(Length x)				
	. . .							
n-y+1				REPORT REDUNDANCY GROUP DESCRIPTOR (Last)				
n				(Length y)				

The REPORT REDUNDANCY GROUP LIST LENGTH field specifies the length in bytes of the following REPORT REDUNDANCY GROUP DESCRIPTOR(s).

The REPORT REDUNDANCY GROUP DESCRIPTOR is defined in table 60.

Table 60 - Format of REPORT REDUNDANCY GROUP DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)			LUN_R				
1								(LSB)
2	RESERVED							
3	REDUNDANCY GROUP IDENTIFIER							
4	RESERVED				GRANULARITY OF UNITS			
5	RESERVED	REDUNDANCY GROUP STATE						
6	(MSB)			REDUNDANCY GROUP P_EXTENT LIST LENGTH (n-7)				
7								(LSB)
	REDUNDANCY GROUP P_EXTENT DESCRIPTOR(S)							
8				REDUNDANCY GROUP P_EXTENT DESCRIPTOR (First)				
31								
	. . .							
n-23				REDUNDANCY GROUP P_EXTENT DESCRIPTOR (Last)				
n								

The LUN_R field specifies the address of the redundancy group to which the information listed in this REPORT REDUNDANCY GROUP DESCRIPTOR is associated.

The REDUNDANCY GROUP IDENTIFIER field (see table 61) indicates the type of protection being used within the redundancy group.

Table 61 - REDUNDANCY GROUP IDENTIFIERS

Codes	Description
00h	No redundancy
01h	Copy redundancy
02h	XOR redundancy
03h	P+Q redundancy
04h-7Fh	Reserved
80h-FFh	Vendor specific

The GRANULARITY OF UNITS field (see table 62) indicates units being used within the redundancy group.

Table 62 - GRANULARITY OF UNITS

Codes	Description
0h	Bit
1h	Byte
2h	2-Byte word
3h	4-Byte word
4h	Logical block
5h-Bh	Reserved
Ch-Fh	Vendor specific

The REDUNDANCY GROUP STATE field is defined in table 39.

The REDUNDANCY GROUP P_EXTENT LIST LENGTH field specifies the length in bytes of the following REDUNDANCY GROUP P_EXTENT DESCRIPTOR(s).

The REDUNDANCY GROUP P_EXTENT DESCRIPTOR filed(s) contain a list of p_extents and the p_extents protected space mapping (see 5.2.2.12) for the addressed redundancy group. See table 63 for the format of the REDUNDANCY GROUP P_EXTENT DESCRIPTOR field.

Table 63 - REDUNDANCY GROUP P_EXTENT DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	P_EXTENT DESCRIPTOR							
11								
12	(MSB)	START CHECK DATA INTERLEAVE UNIT						
13								
14								
15								(LSB)
16	(MSB)	NUMBER OF UNITS OF CHECK DATA						
17								
18								
19								(LSB)
20	(MSB)	NUMBER OF UNITS OF USER DATA						
21								
22								
23								(LSB)

See table 14 for a description of the P_EXTENT DESCRIPTOR field.

The START CHECK DATA INTERLEAVE UNIT field contains the location of the first unit of check data within the p_extent.

The NUMBER OF UNITS OF CHECK DATA field contains the number of consecutive units to be reserved for check data within the p_extent.

The NUMBER OF UNITS OF USER DATA field contains the number of consecutive units to be reserved for protected space within the p_extent.

6.3.1.2 REPORT UNASSIGNED REDUNDANCY GROUP SPACE service action

The REPORT UNASSIGNED REDUNDANCY GROUP SPACE service action (see table 64) requests that information regarding the redundancy groups within the target that have protected space not yet configured into to any volume sets be sent to the application client.

Table 64 - REPORT UNASSIGNED REDUNDANCY GROUP SPACE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BAh)							
1	RESERVED			SERVICE ACTION (01h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_R				
5								(LSB)
6	(MSB)							
7				ALLOCATION LENGTH				
8								
9								(LSB)
10	RESERVED							RPTSEL
11	CONTROL							

The LUN_R field specifies the address of the redundancy group that shall be examined for any unassigned ps_extents. Any unassigned ps_extents within the addressed redundancy group shall be reported to the application client per table 65. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED.

A report selected bit (RPTSEL) of zero indicates the target shall report all the redundancy group(s) within the target. The LUN_R field shall be ignored when the RPTSEL bit is zero. A RPTSEL bit of one indicates the target shall report only on the redundancy group indicated in the LUN_R field.

The REPORT UNASSIGNED REDUNDANCY GROUP SPACE parameter list (see table 65) contains a four-byte header that defines the length in bytes of the parameter list and a list of REPORT UNASSIGNED REDUNDANCY GROUP SPACE DESCRIPTORS.

Table 65 - REPORT UNASSIGNED REDUNDANCY GROUP SPACE parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1	REPORT UNASSIGNED REDUNDANCY GROUP SPACE LIST LENGTH (n-3)							
2								
3								(LSB)
	REPORT UNASSIGNED REDUNDANCY GROUP SPACE DESCRIPTORS(S) (if any)							
4	REPORT UNASSIGNED REDUNDANCY GROUP SPACE DESCRIPTOR (First) (Length x)							
x+3								
	. . .							
n-y+1	REPORT UNASSIGNED REDUNDANCY GROUP SPACE DESCRIPTOR (Last) (Length y)							
n								

The REDUNDANCY GROUP LIST LENGTH field specifies the length in bytes of the following REPORT UNASSIGNED REDUNDANCY GROUP SPACE DESCRIPTOR(S).

The REPORT UNASSIGNED REDUNDANCY GROUP SPACE DESCRIPTOR(s) contains information on all unassigned protected space within the addressed redundancy group. See table 66 for the format of the REPORT UNASSIGNED REDUNDANCY GROUP SPACE DESCRIPTOR.

Table 66 - Format of REPORT UNASSIGNED REDUNDANCY GROUP SPACE DESCRIPTOR

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				LUN_R				(LSB)
2	RESERVED							
3	REDUNDANCY GROUP IDENTIFIER							
4	RESERVED							
5	RESERVED			STATE OF THE REDUNDANCY GROUP				
6	(MSB)							
7				PS_EXTENT LIST LENGTH (n-7)				(LSB)
	PS_EXTENT DESCRIPTOR(S)							
8								
19				PS_EXTENT DESCRIPTOR 0				
	. . .							
n-11								
n				PS_EXTENT DESCRIPTOR x				

See table 61 for a description of the REDUNDANCY GROUP IDENTIFIER field.

The LUN_R field specifies the address of the redundancy group that shall be examined for any unassigned ps_extents. Any unassigned ps_extents within the addressed redundancy group shall be listed in the PS_EXTENT DESCRIPTOR list.

The REDUNDANCY GROUP IDENTIFIER field indicates the type of protection being used within the redundancy group that contains the unassigned ps_extents. See table 61 for a list of the types of protection.

The REDUNDANCY GROUP STATE field is defined in table 39.

The PS_EXTENT LIST LENGTH field specifies the length in bytes of the following PS_EXTENT DESCRIPTOR(s).

The PS_EXTENT DESCRIPTOR(s) contain a list of unassigned ps_extents from the addressed redundancy group. The PS_EXTENT DESCRIPTOR is defined in table 67.

Table 67 - Data format of PS_EXTENT DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				LUN_P				(LSB)
2	(MSB)							
3								
4				START LBA_PS				
5								(LSB)
6	(MSB)							
7								
8				NUMBER OF LBA_PS(s)				
9								(LSB)
10	(MSB)							
11				NUMBER OF BYTES PER LBA_PS				(LSB)

The LUN_P field contains the address of the peripheral device that contains the ps_extent.

The START LBA_PS field contains the first unassigned addressable logical block address of protected space within the addressed ps_extent.

The NUMBER OF LBA_PS(s) field contains the capacity of the protected space ps_extent in blocks.

NOTE 21 - In ps_extents the number of LBA_PS(s) does not include any logical blocks that have been configured to contain check data; in contrast to p_extents in which the NUMBER OF LBA_P(s) includes all addressable logical blocks on a peripheral device.

The NUMBER OF BYTES PER LBA_PS field contains the size, in bytes, of the blocks in the ps_extent.

6.4 REDUNDANCY GROUP (OUT) command

6.4.1 REDUNDANCY GROUP (OUT) command service actions

The service actions for the REDUNDANCY GROUP (OUT) command are listed in table 68.

Table 68 - Service actions for REDUNDANCY GROUP (OUT) command

Service name	Service actions	Type	Subclause
CONTROL GENERATION OF CHECK DATA	00h	O	6.4.1.1
CREATE/MODIFY REDUNDANCY GROUP	01h	O	6.4.1.2
DELETE REDUNDANCY GROUP	02h	O	6.4.1.3
REBUILD P_EXTENT	03h	O	6.4.1.4
REBUILD PERIPHERAL DEVICE	04h	O	6.4.1.5
RECALCULATE CHECK DATA	05h	O	6.4.1.6
VERIFY CHECK DATA	06h	O	6.4.1.7
RESERVED	09h-17h		
VENDOR SPECIFIC	18h-1Fh		
Key: M = Service action implementation is mandatory. O = Service action implementation is optional.			

6.4.1.1 CONTROL GENERATION OF CHECK DATA service action

The CONTROL GENERATION OF CHECK DATA service action (see table 69) requests that the generation of check data within a redundancy group be enabled or disabled.

Table 69 - CONTROL GENERATION OF CHECK DATA service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BBh)							
1	RESERVED			SERVICE ACTION (00h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_R				
5								(LSB)
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
10	RESERVED				DISCHK	RESERVED	ALLRG	RESERVED
11	CONTROL							

The LUN_R field specifies the address of the redundancy group that shall have the generation of check data enabled or disabled. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED.

An all redundancy group bit (ALLRG) of zero indicates that the check data generation being enabled/disabled shall only be enabled/disabled on the addressed redundancy group. An ALLRG bit of one indicates that the check data generation being enabled/disabled shall be enabled/disabled on all the redundancy groups within the target. The LUN_R field shall be ignored if the ALLRG bit is one.

A disable check data bit (DISCHK) of zero indicates the generation of check data shall be enabled on the selected redundancy group(s). A DISCHK bit of one indicates the generation of check data shall be disabled on the selected redundancy group(s). Generation of check data shall be disabled until a CONTROL GENERATION OF CHECK DATA service action is requested with the DISCHK bit set to zero and the ALLRG bit set to one or the ALLRG bit set to zero and the LUN_R field set to the address of the redundancy group that is equal to the LUN_R from the original CONTROL GENERATION OF CHECK DATA service action.

A VOLUME SET (OUT) command's CONTROL GENERATION OF CHECK DATA service action shall not cause the generation of check data to be enabled if the generation of check data was disabled using the REDUNDANCY GROUP (OUT) command's CONTROL GENERATION OF CHECK DATA service action.

6.4.1.2 CREATE/MODIFY REDUNDANCY GROUP service action

The CREATE/MODIFY REDUNDANCY GROUP service action (see table 70) requests the creation of a new redundancy group or the modification of an existing redundancy group. This service action contains all the information required for the target to define a redundancy group and the check data mapping within that redundancy group (see 5.2.2.12). If the create operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to CREATION OF LOGICAL UNIT FAILED. If the modification operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to MODIFICATION OF LOGICAL UNIT FAILED.

Table 70 - CREATE/MODIFY REDUNDANCY GROUP service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BBh)							
1	RESERVED			SERVICE ACTION (01h)				
2	REDUNDANCY TYPE IDENTIFIER							
3	RESERVED				GRANULARITY OF UNITS			
4	(MSB)	LUN_R						
5								(LSB)
6	(MSB)	LIST LENGTH						
7								
8								
9								(LSB)
10	RESERVED	SETLUN	RESERVED					IMMED
11	CONTROL							

The REDUNDANCY GROUP IDENTIFIER field indicates the type of protection that shall be used within the redundancy group being created or modified. See table 61 for the format of the REDUNDANCY GROUP IDENTIFIER field.

The GRANULARITY OF UNITS field indicates if bits, bytes, words, or logical blocks shall be used within the redundancy group being created or modified. See table 62 for the format of the GRANULARITY OF UNITS field.

The LUN_R field specifies the address of the redundancy group that shall be created or modified. If the addressed redundancy group already exists within the target the target shall modify the existing redundancy group as requested in the CREATE/MODIFY REDUNDANCY GROUP service action. If the requested logical unit is not configurable the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED.

An immediate (IMMED) bit of zero indicates that status shall be returned after the create/modify redundancy group operation has completed. An IMMED bit value of one indicates that the target shall return status as soon as the command descriptor block has been validated, and the entire CREATE/MODIFIY REDUNDANCY GROUP parameters list has been transferred.

A set LUN (SETLUN) bit of zero indicates the target shall assign to the redundancy group being created the logical unit number contained in the LUN_R field. A SETLUN bit of one indicates that the target shall assign a logical unit number to the redundancy group per the addressing rules (see 5.2.1). The LUN_R field shall be ignored when the SETLUN bit is set to one.

The CREATE/MODIFY REDUNDANCY GROUP parameter list (see table 71) contains a list of CREATE/MODIFY P_EXTENT DESCRIPTORS that shall be combined to create or modify the addressed redundancy group.

Table 71 - CREATE/MODIFY REDUNDANCY GROUP parameter list

Bit Byte	7	6	5	4	3	2	1	0
	CREATE/MODIFY P_EXTENT DESCRIPTORS(S) (if any)							
0 27	CREATE/MODIFY P_EXTENT DESCRIPTOR 0							
	. . .							
n-27 n	CREATE/MODIFY P_EXTENT DESCRIPTOR X							

The CREATE/MODIFY P_EXTENT DESCRIPTOR contains information the target shall use to control the protected space mapping of the p_extent. See table 72 for the format of the CREATE/MODIFY P_EXTENT DESCRIPTOR.

Table 72 - Data format of CREATE/MODIFY P_EXTENT DESCRIPTOR

Bit / Byte	7	6	5	4	3	2	1	0
0	P_EXTENT DESCRIPTOR							
11								
12	SETPAT	RESERVED		PRESERVE	RESERVED			DEFERCAL
13	RESERVED							
14	RESERVED							
15	PROTECTED SPACE PATTERN							
16	(MSB)	START CHECK DATA INTERLEAVE UNIT						
17								
18								
19								(LSB)
20	(MSB)	NUMBER OF UNITS OF CHECK DATA						
21								
22								
23								(LSB)
24	(MSB)	NUMBER OF UNITS OF USER DATA						
25								
26								
27								(LSB)

The P_EXTENT DESCRIPTOR defines the boundaries of the protected space mapping information contained in the CREATE/MODIFY P_EXTENT DESCRIPTOR. See table 14 for a description of the P_EXTENT DESCRIPTOR.

All bits and fields within the CREATE/MODIFY P_EXTENT DESCRIPTOR shall be bounded by the p_extent. It is not an error for a group of p_extents that define a redundancy group to contain different parameters within the CREATE/MODIFY P_EXTENT DESCRIPTORS.

A defer recalculate bit (DEFERCAL) of zero indicates the target shall recalculate check data before the CREATE/MODIFY REDUNDANCY GROUP service action returns status. A DEFERCAL bit of one indicates the target shall recalculate check data after the CREATE/MODIFY REDUNDANCY GROUP service action returns status.

A preserve protected space bit (PRESERVE) of zero indicates the protected space information shall be vendor specific on completion of the CREATE/MODIFY REDUNDANCY GROUP service action. A PRESERVE bit of one indicates the protected space information shall be preserved during the modification of a redundancy group. If the PRESERVE bit is set to one and a create redundancy group is requested the

target shall terminate the command with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN PARAMETER LIST.

A set pattern bit (SETPAT) of zero indicates the protected space information contained within the protected space shall be vendor specific. The PROTECTED SPACE PATTERN field shall be ignored when the SETPAT bit is zero. A SETPAT bit of one indicates the pattern contained within the PROTECTED SPACE PATTERN field shall be replicated throughout the protected space.

The START CHECK DATA INTERLEAVE UNIT field contains the location of the first unit of check data within the p_extent.

It is not required that the START CHECK DATA INTERLEAVE UNIT be located at the beginning of the first logical block address of the p_extent. All units between the beginning of the first logical block address of the p_extent and the START CHECK DATA INTERLEAVE UNIt value shall be protected space.

The NUMBER OF UNITS OF CHECK DATA field contains the number of consecutive units to be reserved for check data within the p_extent.

The NUMBER OF UNITS OF USER DATA field contains the number of consecutive units to be reserved for protected space within the p_extent.

6.4.1.3 DELETE REDUNDANCY GROUP service action

The DELETE REDUNDANCY GROUP service action (see table 73) requests that the selected redundancy group be deleted.

Table 73 - DELETE REDUNDANCY GROUP service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BBh)							
1	RESERVED			SERVICE ACTION (02h)				
2	RESERVED							
3	RESERVED							
4	(MSB)							
5	LUN_R							(LSB)
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
10	RESERVED							
11	CONTROL							

The LUN_R field specifies the address of the redundancy group that the target shall delete. If the requested

logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED. If the selected redundancy group contains any assigned ps_extents the command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with an additional sense code of REMOVE OF LOGICAL UNIT FAILED.

6.4.1.4 REBUILD P_EXTENT service action

The REBUILD P_EXTENT service action (see table 74) requests the rebuild of all or part of one or more redundancy group(s).

Table 74 - REBUILD P_EXTENT service action

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BBh)							
1	RESERVED			SERVICE ACTION (03h)				
2	RESERVED							
3	RESERVED							
4	RESERVED							
5	RESERVED							
6	(MSB)							
7				LIST LENGTH				
8								
9								(LSB)
10	RESERVED		REBUILD		RESERVED			IMMED
11	CONTROL							

An immediate (IMMED) bit of zero indicates that status shall be returned after the rebuild operation has completed. An IMMED bit value of one indicates that the target shall return status as soon as the command descriptor block has been validated, and the entire REBUILD P_EXTENT parameters list has been transferred.

The REBUILD field specifies the rebuild instructions the target shall use. See table 75 for the format of the REBUILD field.

Table 75 - Rebuild types

Codes	Description
00b	All assigned space associated with the selected p_extent shall be rebuilt. The list of redundancy group(s) in the REBUILD P_EXTENTS parameter list shall be ignored. Protected space associated with overlapping redundancy groups shall be successfully rebuilt multiple times for a successful completion of the REBUILD P_EXTENT service action. The order of the rebuilds is vender specific.
01b	All assigned space association with the selected p_extent shall be rebuilt using any associated redundancy group. The list of redundancy group(s) in the REBUILD P_EXTENTS parameter list shall be ignored. Any protected space associated with overlapping redundancy groups shall only be successfully rebuilt one time for the successful completion of the REBUILD P_EXTENT service action. There is no indication of a failure if an overlapped redundancy group fails to rebuild.
10b	All assigned space associated with the selected p_extent shall be rebuilt. Any redundancy group(s) listed in the REBUILD P_EXTENT parameter list shall not be used to rebuild any part of the selected p_extent. Any protected space associated with overlapping redundancy groups shall be successfully rebuilt at least one time for a successful completion of the REBUILD P_EXTENT service action.
11b	Reserved

If the rebuild operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to REBUILD FAILURE OCCURED.

The REBUILD P_EXTENT parameter list (see table 76) contains the p_extent to be rebuilt and a list of redundancy groups to not be used to rebuild the p_extent.

Table 76 - REBUILD P_EXTENT parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	P_EXTENT DESCRIPTOR							
11								
	LUN_R(S) (if any)							
12	RESERVED							
13	RESERVED							
14	(MSB)	LUN_R 0						
15								(LSB)
	. . .							
n-3	RESERVED							
n-2	RESERVED							
n-1	(MSB)	LUN_R X						
n								(LSB)

The P_EXTENT DESCRIPTOR defines the boundaries of the rebuild. See table 14 for a description of the P_EXTENT DESCRIPTOR.

The LUN_R field(s) specifies the address(es) of the redundancy group(s) that shall not be used to rebuild the p_extent.

6.4.1.5 REBUILD PERIPHERAL DEVICE service action

The REBUILD PERIPHERAL DEVICE service action (see table 77) requests the rebuild of all or part of one or more redundancy group(s).

Table 77 - REBUILD PERIPHERAL DEVICE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BBh)							
1	RESERVED			SERVICE ACTION (04h)				
2	RESERVED							
3	RESERVED							
4	RESERVED							
5	RESERVED							
6	(MSB)							
7				LIST LENGTH				
8								
9								(LSB)
10	RESERVED		REBUILD		RESERVED			IMMED
11	CONTROL							

An immediate (IMMED) bit of zero indicates that status shall be returned after the rebuild operation has completed. An IMMED bit value of one indicates that the target shall return status as soon as the command descriptor block has been validated, and the entire REBUILD PERIPHERAL DEVICE parameters list has been transferred.

The REBUILD field specifies the rebuild instructions the target shall use. See table 78 for the format of the REBUILD field.

Table 78 - Rebuild types

Codes	Description
00b	All assigned space associated with the selected peripheral device shall be rebuilt. The list of redundancy group(s) in the REBUILD PERIPHERAL DEVICES parameter list shall be ignored. Protected space associated with overlapping redundancy groups shall be successfully rebuilt multiple times for a successful completion of the REBUILD PERIPHERAL DEVICE service action. The order of the rebuilds is vender specific.
01b	All assigned space associated with the selected peripheral device shall be rebuilt using any associated redundancy group. The list of redundancy group(s) in the REBUILD PERIPHERAL DEVICES parameter list shall be ignored. Any protected space associated with overlapping redundancy groups shall only be successfully rebuilt one time for the successful completion of the REBUILD PERIPHERAL DEVICE service action. There is no indication of a failure if an overlapped redundancy group fails to rebuild.
10b	All assigned space associated with the selected peripheral device shall be rebuilt. Any redundancy group(s) listed in the REBUILD PERIPHERAL DEVICE parameter list shall not be used to rebuild any part of the selected peripheral device. Any protected space associated with overlapping redundancy groups shall be successfully rebuilt at least one time for a successful completion of the REBUILD PERIPHERAL DEVICE service action.
11b	Reserved

If the rebuild operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to REBUILD FAILURE OCCURED.

The REBUILD PERIPHERAL DEVICE parameter list (see table 79) contains the peripheral device to be rebuilt and a list of redundancy groups to not be used to rebuild the peripheral device.

Table 79 - REBUILD PERIPHERAL DEVICE parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	RESERVED							
1	RESERVED							
2	(MSB)			LUN_P				
3								(LSB)
	LUN_R(S) (if any)							
4	RESERVED							
5	RESERVED							
6	(MSB)			LUN_R 0				
7								(LSB)
	. . .							
n-3	RESERVED							
n-2	RESERVED							
n-1	(MSB)			LUN_R x				
n								(LSB)

The LUN_P FIELD specifies the address of the peripheral device to be rebuilt.

The LUN_R field(s) specifies the address(es) of the redundancy group(s) that shall not be used to rebuild the peripheral device.

6.4.1.6 RECALCULATE CHECK DATA service action

The RECALCULATE CHECK DATA service action (see table 80) requests the target to recalculate check data within a redundancy group. If the recalculate operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to RECALCULATE FAILURE OCCURRED.

Table 80 - RECALCULATE CHECK DATA service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BBh)							
1	RESERVED			SERVICE ACTION (05h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_R				
5								(LSB)
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
10	RESERVED						ALLRG	IMMED
11	CONTROL							

The LUN_R field specifies the address of the redundancy group for which the target shall recalculate the check data. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED.

An immediate (IMMED) bit of zero indicates that status shall be returned after the recalculate operation has completed. An IMMED bit value of one indicates that the target shall return status as soon as the command descriptor block has been validated.

An all redundancy group bit (ALLRG) of zero indicates the recalculation of check data shall only occur on the addressed redundancy group. An ALLRG bit of one indicates the recalculation of check data shall occur on all redundancy groups within the target. The LUN_R field shall be ignored when the ALLRG bit is one.

6.4.1.7 VERIFY CHECK DATA service action

The VERIFY CHECK DATA service action (see table 81) requests that check data be verified consistent with the protected space within a redundancy group.

Table 81 - VERIFY CHECK DATA service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BBh)							
1	RESERVED			SERVICE ACTION (06h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_R				
5								(LSB)
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
10	RESERVED				CONTVER	RESERVED	ALLRG	IMMED
11	CONTROL							

The LUN_R field specifies the address of the redundancy group that shall have its check data verified. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED.

An immediate (IMMED) bit of zero indicates that status shall be returned after the verification operation has completed. An IMMED bit value of one indicates that the target shall return status as soon as the command descriptor block has been validated.

An all redundancy group bit (ALLRG) of zero indicates the verification of check data shall only occur on the addressed redundancy group. An ALLRG bit of one indicates the verification of check data shall occur on all redundancy groups within the target. The LUN_R field shall be ignored when the ALLRG bit is one.

A continuous verification bit (CONTVER) of zero indicates the check data shall be verified only once. A CONTVER bit of one indicates the check data shall be continuously verified. Verification shall continue until a VERIFY CHECK DATA service action is requested with the CONTVER bit set to zero and the ALLRG bit set to one or the ALLRG bit set to zero and the LUN_R field set to the address of the redundancy group that is equal to the LUN_R from the original VERIFY CHECK DATA service action.

NOTE 22 - If continuous verification is selected the verification executes as a background operation within the target in a vendor specific manner. Continuously may be defined as only verifying check data associated with the LBA range of any write commands that occur within the range on LBA_Vs that overlay a redundancy group that has continuous verification enabled.

Any verification failures occurring before the VERIFY CHECK DATA service action has completed shall cause the target to terminate the command with a CHECK CONDITION status. The sense key shall be set to MEDIUM ERROR, and the additional sense code shall be set to MISCOMPARE DURING VERIFY

OPERATION.

Any verification failures occurring after the VERIFY CHECK DATA service action has completed shall cause the target to generate a unit attention condition for all initiators. When reporting the unit attention condition the additional sense code shall be set to MISCOMPARE DURING VERIFY OPERATION.

The relationship between the VERIFY VOLUME SET CHECK DATA service action and the VERIFY CHECK DATA service action when both service actions have requested the same area be verified is vendor specific. This standard only requires the requested area be verified.

6.5 VOLUME SET (IN) command

6.5.1 VOLUME SET (IN) command service actions

The service actions for the VOLUME SET(IN) command are listed in table 82.

Table 82 - Service actions for volume set (in) command

Service name	Service actions	Type	Subclause
REPORT VOLUME SETS RESERVED VENDOR SPECIFIC	00h 01h-17h 18h-1Fh	M	6.5.1.1
Key: M = Service action implementation is mandatory. O = Service Action implementation is optional.			

6.5.1.1 REPORT VOLUME SETS service action

The REPORT VOLUME SETS service action (see table 83) requests that information regarding volume sets within the target be sent to the application client.

Table 83 - REPORT VOLUME SETS service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BEh)							
1	RESERVED			SERVICE ACTION (00h)				
2	RESERVED							
3	RESERVED							
4	(MSB)	LUN_V						
5								(LSB)
6	(MSB)	ALLOCATION LENGTH						
7								
8								
9								(LSB)
10	RESERVED							RPTSEL
11	CONTROL							

The LUN_V field specifies the address of the volume set for which information shall be reported per table 84. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set

to LOGICAL UNIT NOT CONFIGURED.

A report selected bit (RPTSEL) of zero indicates the target shall report on all the volume set(s) within the target. The LUN_V field shall be ignored when the RPTSEL bit is zero. A RPTSEL bit of one indicates the target shall only report information on the volume set addressed by the LUN_V field.

The REPORT VOLUME SETS parameter list (see table 84) contains a four-byte header that contains the length in bytes of the parameter list and a list of REPORT VOLUME SET DESCRIPTORS.

Table 84 - REPORT VOLUME SETS parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				REPORT VOLUME SET LIST LENGTH (n-3)				
2								
3								(LSB)
	REPORT VOLUME SET DESCRIPTORS(S) (if any)							
4				REPORT VOLUME SET DESCRIPTOR (First)				
x+3				(Length x)				
				. . .				
n-y+1				REPORT VOLUME SET DESCRIPTOR (Last)				
n				(Length y)				

The REPORT VOLUME SET LIST LENGTH field specifies the length in bytes of the following REPORT VOLUME SET DESCRIPTOR(s).

The REPORT VOLUME SET DESCRIPTOR is defined in table 85.

Table 85 - Format of REPORT VOLUME SET DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)			LUN_V				
1								(LSB)
2	RESERVED							
3	RESERVED							
4	RESERVED				GRANULARITY OF UNITS			
5	RESERVED	STATE OF THE VOLUME SET						
6	(MSB)			PS_EXTENT STRIPE LENGTH				
7								
8								
9								(LSB)
10	(MSB)			PS_EXTENT INTERLEAVE DEPTH				
11								
12								
13								(LSB)
14	(MSB)			REPORT VOLUME SET DESCRIPTOR LIST LENGTH				
15								(LSB)
	VOLUME SET PS_EXTENT DESCRIPTOR(S)							
16				VOLUME SET PS_EXTENT DESCRIPTOR (First)				
36								
	. . .							
n-20				VOLUME SET PS_EXTENT DESCRIPTOR (Last)				
n								

The LUN_V field specifies the address of the volume set to which the information listed in this REPORT VOLUME SET DESCRIPTOR is associated.

The GRANULARITY OF UNITS field indicates if bits, bytes, words, or logical blocks shall be used within the volume set being created or modified. See table 62 for the format of the GRANULARITY OF UNITS field.

The VOLUME SET STATE field is defined in table 38.

The PS_EXTENT STRIPE LENGTH field contains the number of contiguous ps_extents counted before looping back to the first ps_extent of the current stripe.

The PS_EXTENT INTERLEAVE DEPTH field contains the number of stripes counted before continuing onto the next consecutive ps_extent beyond the current stripe.

The VOLUME SET PS_EXTENT LIST LENGTH field specifies the length in bytes of the following VOLUME SET PS_EXTENT DESCRIPTOR(s).

The VOLUME SET PS_EXTENT DESCRIPTOR contains a list of assigned ps_extents and the assigned ps_extents user data mapping for the addressed volume set. See table 86 for the format of the VOLUME SET PS_EXTENT DESCRIPTOR field.

TABLE 86 - VOLUME SET PS_EXTENT DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	PS_EXTENT DESCRIPTOR							
11								
12	RESERVED							INCDEC
13	RESERVED							
14	LUN_R							
15								
16	(MSB)							
17	USER DATA STRIPE DEPTH							
18								
19								(LSB)

See table 67 for a description of the PS_EXTENT DESCRIPTOR field.

An increment/decrement bit (INCDEC) of zero indicates logical blocks within the ps_extent have been mapped in ascending order. An INCDEC bit of one indicates the logical blocks within the ps_extent have been mapped in descending order.

The LUN_R field specifies the address of the redundancy group that caused the formation of the ps_extent.

The USER DATA STRIPE DEPTH field contains the number of contiguous units counted within a ps_extent before proceeding to the next ps_extent.

6.6 VOLUME SET (OUT) command

6.6.1 VOLUME SET (OUT) command service actions

The service actions for the VOLUME SET(OUT) command are listed in table 87.

Table 87 - Service actions for VOLUME SET (OUT) command

Service name	Service actions	Type	Subclause
CONTROL GENERATION OF CHECK DATA	00h	O	6.6.1.1
CONTROL WRITE OPERATIONS	01h	O	6.6.1.2
CREATE/MODIFY VOLUME SET	02h	O	6.6.1.3
DELETE VOLUME SET	03h	O	6.6.1.4
RECALCULATE VOLUME SET CHECK DATA	04h	O	6.6.1.5
VERIFIY VOLUME SET CHECK DATA	05h	O	6.6.1.6
RESERVED	06h-17h		
VENDOR SPECIFIC	18h-1Fh		
Key: M = Service action implementation is mandatory. O = Service action implementation is optional.			

6.6.1.1 CONTROL GENERATION OF CHECK DATA service action

The CONTROL GENERATION OF CHECK DATA service action (see table 88) requests that the generation of check data within the underlaying redundancy group(s) of a volume set be enabled or disabled.

Table 88 - CONTROL GENERATION OF CHECK DATA service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BFh)							
1	RESERVED			SERVICE ACTION (00h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_V				
5								(LSB)
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
10	RESERVED				DISCHK	RESERVED	ALLVLU	RESERVED
11	CONTROL							

The LUN_V field specifies the address of the volume set that shall have the generation of check data contained within the underlaying redundancy group(s) enabled or disabled. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT

CONFIGURED.

An all volume set bit (ALLVLU) of zero indicates that the check data generation being enabled/disabled shall only be enabled/disabled within the underlaying redundancy group(s) of the addressed volume set. An ALLVLU bit of one indicates that the check data generation being enabled/disabled shall be enabled/disabled within the underlaying redundancy group(s) of all volume set(s) within the target. The LUN_V field shall be ignored if the ALLVLU bit is one.

A disable check data bit (DISCHK) of zero indicates the generation of check data contained within all of the underlaying redundancy group(s) of the selected volume(s) shall be enabled. A DISCHK bit of one indicates the generation of check data contained within all of the underlaying redundancy group(s) of the selected volume set(s) shall be disabled. Generation of check data shall be disabled until a CONTROL GENERATION OF CHECK DATA service action is requested with the DISCHK bit set to zero and the ALLVLU bit set to one or the ALLVLU bit set to zero and the LUN_V field set to the address of the volume set that is equal to the LUN_V from the original CONTROL GENERATION OF CHECK DATA service action.

A REDUNDANCY GROUP (OUT) command's CONTROL GENERATION OF CHECK DATA service action shall not cause the generation of check data to be enabled if the generation of check data was disabled using the VOLUME SET (OUT) command's CONTROL GENERATION OF CHECK DATA service action.

6.6.1.2 CONTROL WRITE OPERATIONS service action

The CONTROL WRITE OPERATIONS service action (see table 89) requests that write operations to a volume set be enabled or disabled.

This service action shall cause the following commands to be enabled/disabled:

a) ERASE command;
b) FORMAT command;
c) WRITE command;
d) WRITE FILE MARKS command;
e) WRITE LONG command;
f) WRITE SAME command;
g) WRITE & VERIFY command.

Table 89 - CONTROL WRITE OPERATIONS service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BFh)							
1	RESERVED			SERVICE ACTION (01h)				
2	RESERVED							
3	RESERVED							
4	(MSB)	LUN_V						
5								(LSB)
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
10	RESERVED				DISWR	RESERVED	ALLVLU	RESERVED
11	CONTROL							

The LUN_V field specifies the address of the volume set that shall have write operations enabled or disabled. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED.

An all volume set bit (ALLVLU) of zero indicates the write operation control shall only apply to the addressed volume set. An ALLVLU bit of one indicates the write operation control shall apply to all volume set(s) within the target. The LUN_V field shall be ignored if the ALLVLU bit is one.

A disable write operations bit (DISWR) of zero indicates write operations shall be enabled on the selected volume set(s). A DISWR bit of one indicates write operations shall be disabled on the selected volume set(s). Write operations shall disabled until a CONTROL WRITE OPERATIONS service action is requested with the DISWR bit set to zero and the ALLVLU bit set to one or the ALLVLU bit set to zero and the LUN_V field set to the address of the volume set that is equal to the LUN_V from the original CONTROL WRITE OPERATIONS service action.

6.6.1.3 CREATE/MODIFY VOLUME SET service action

The CREATE/MODIFY VOLUME SET service action (see table 90) requests the creation of a new volume set or the modification of an existing volume set. If the create operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to CREATION OF LOGICAL UNIT FAILED. If the modification operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to MODIFICATION OF LOGICAL UNIT FAILED.

Table 90 - CREATE/MODIFY VOLUME SET service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BFh)							
1	RESERVED			SERVICE ACTION (02h)				
2	RESERVED							
3	RESERVED				GRANULARITY OF UNITS			
4	(MSB)			LUN_V				
5								(LSB)
6	(MSB)			LIST LENGTH				
7								
8								
9								(LSB)
10	RESERVED	SETLUN	RESERVED					IMMED
11	CONTROL							

The GRANULARITY OF UNITS field indicates if bits, bytes, words, or logical blocks shall be used within the volume set being created or modified. See table 62 for the format of the GRANULARITY OF UNITS field.

The LUN_V field specifies the address of the volume set that shall be created or modified. If the addressed volume set already exists within the target the target shall modify the existing volume set as requested in the CREATE/MODIFIY VOLUME SET service action. If the requested logical unit is not configurable the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

An immediate (IMMED) bit of zero indicates that status shall be returned after the create/modify volume set operation has completed. An IMMED bit value of one indicates that the target shall return status as soon as the command descriptor block has been validated, and the entire CREATE/MODIFIY VOLUME SET parameters list has been transferred.

A set LUN (SETLUN) bit of zero indicates the target shall assign to the volume set being created the logical unit number contained in the LUN_V field. A SETLUN bit of one indicates that the target shall assign a logical unit number to the volume set per the addressing rules (see 5.2.1). The LUN_V field shall be ignored when the SETLUN bit is set to one.

The CREATE/MODIFY VOLUME SET parameter list (see table 91) contains user data mapping information and a list of CREATE/MODIFY PS_EXTENT DESCRIPTORS that shall be combined to create or modify the addressed volume set.

Table 91 - CREATE/MODIFY VOLUME SET parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				PS_EXTENT STRIPE LENGTH				
2								
3								(LSB)
4	(MSB)							
5				PS_EXTENT INTERLEAVE DEPTH				
6								
7								(LSB)
	CREATE/MODIFY PS_EXTENT DESCRIPTORS(S) (if any)							
8				CREATE/MODIFY PS_EXTENT DESCRIPTOR 0				
27								
	. . .							
n-19				CREATE/MODIFY PS_EXTENT DESCRIPTOR X				
n								

The PS_EXTENT STRIPE LENGTH field specifies the number of contiguous ps_extents the target shall count before looping back to the first ps_extent of the current stripe. The looping shall continue until all the units of a ps_extent are used up or the value in the PS_EXTENT INTERLEAVE DEPTH field is reached.

If the number of ps_extents within the current create volume set request is not an exact multiple of the PS_EXTENT STRIPE LENGTH field the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST. It is not an error if the PS_EXTENT STRIPE LENGTH field is not an even multiple of stripes and a modify volume set was requested.

The PS_EXTENT INTERLEAVE DEPTH field specifies the number of stripes the target shall count before continuing the mapping into the next consecutive ps_extent beyond the current stripe. If the current stripe is the last ps_extent the target shall continue the mapping at the first ps_extent. The mapping shall continue until all the units of all the ps_extents are mapped.

If the value in the PS_EXTENT STRIPE LENGTH field is equal to the number of ps_extents within the current modify/create volume set request, the PS_EXTENT INTERLEAVE DEPTH field shall be ignored.

The CREATE/MODIFY PS_EXTENT DESCRIPTOR contains information the target shall use to control the user data mapping (see 5.2.2.14) within the ps_extent. See table 92 for the format of the CREATE/MODIFY PS_EXTENT DESCRIPTOR.

Table 92 - Data format of CREATE/MODIFY PS_EXTENT DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	PS_EXTENT DESCRIPTOR							
11								
12	RESERVED							INCDEC
13	RESERVED							
14	LUN_R							
15								
16	(MSB) USER DATA STRIPE DEPTH							
17								
18								
19								(LSB)

The PS_EXTENT DESCRIPTOR defines the boundaries of the user data mapping information contained in the CREATE/MODIFY PS_EXTENT DESCRIPTOR. See table 67 for a description of the PS_EXTENT DESCRIPTOR.

All bits and fields within the CREATE/MODIFY PS_EXTENT DESCRIPTOR shall be bounded by the ps_extent. It is not an error for a group of ps_extents that define a volume set to contain different parameters within the CREATE/MODIFY PS_EXTENT DESCRIPTORS.

An increment/decrement bit (INCDEC) of zero indicates the target shall map logical blocks within the ps_extent in ascending order. When INCDEC is zero logical blocks after the START LBA_PS field value in the PS_EXTENT DESCRIPTOR shall be assigned in ascending order. An INCDEC bit of one indicates the target shall map the logical blocks within the ps_extent in descending order. When INCDEC is one logical blocks starting with the START LBA_PS field value in the PS_EXTENT DESCRIPTOR shall be assigned in descending order.

NOTE 23 - When the INCDEC bit is one the START LBA_PS field contains the largest logical block in the p_extent.

The LUN_R field specifies the address of the redundancy group that caused the formation of the ps_extent.

The USER DATA STRIPE DEPTH field contains the number of contiguous units to count within a ps_extent before proceeding to the next ps_extent. Any units defined as check data shall not be counted to determine when the user data stripe depth is reached.

6.6.1.4 DELETE VOLUME SET service action

The DELETE VOLUME SET service action (see table 93) requests that the selected volume set be deleted. If the remove operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to REMOVE OF LOGICAL UNIT FAILED.

Table 93 - DELETE VOLUME SET service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BFh)							
1	RESERVED			SERVICE ACTION (03h)				
2	RESERVED							
3	RESERVED							
4	(MSB)	LUN_V						
5								(LSB)
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
10	RESERVED							
11	CONTROL							

The LUN_V field specifies the address of the volume set that the target shall delete. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED.

6.6.1.5 RECALCULATE VOLUME SET CHECK DATA service action

The RECALCULATE VOLUME SET CHECK DATA service action (see table 94) requests the target to recalculate check data within the portion of any redundancy group(s) overlaid by the selected a volume set. If the recalculate operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to RECALCULATE FAILURE OCCURRED.

Table 94 - RECALCULATE VOLUME SET CHECK DATA service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BFh)							
1	RESERVED			SERVICE ACTION (04h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_V				
5								(LSB)
6	(MSB)			LIST LENGTH				
7								
8								
9								(LSB)
10	RESERVED						ALLVLU	IMMED
11	CONTROL							

The LUN_V field specifies the address of the volume set that shall have all or part of the check data associated with protected space contained within any underlaying redundancy group(s) recalculated. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED.

An immediate (IMMED) bit of zero indicates that status shall be returned after the recalculate operation has completed. An IMMED bit value of one indicates that the target shall return status as soon as the command descriptor block has been validated, and the entire RECALCULATE VOLUME SET CHECK DATA parameters list has been transferred.

An all volume set bit (ALLVLU) of zero indicates that only check data associated with protected space within any underlaying redundancy group(s) of the selected range of LBA_Vs shall be recalculated. An ALLVLU bit of one indicates that check data associated with protected space within any underlaying redundancy group(s) of the selected volume set shall be recalculated. The RECALCULATE VOLUME SET CHECK DATA parameter list shall be ignored when the ALLVLU bit is one.

The RECALCULATE VOLUME SET CHECK DATA parameter list (see table 95) contains the range of LBA_Vs to be recalculated.

Table 95 - RECALCULATE VOLUME SET CHECK DATA parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				START LBA_V				
2								
3								(LSB)
4	(MSB)							
5				NUMBER OF LBA_V(S)				
6								
7								(LSB)

The START LBA_V field specifies the LBA_V(s) the target shall use to begin the recalculation.

The NUMBER OF LBA_V(S) field specifies the number of consecutive LBA_V(S) the target shall use for the recalculation.

6.6.1.6 VERIFY VOLUME SET CHECK DATA service action

The VERIFY VOLUME SET CHECK DATA service action (see table 96) requests that check data be verified consistent with the protected space within any redundancy group(s) overlaid by the selected range of LBA_V(S).

Table 96 - VERIFY VOLUME SET CHECK DATA service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BFh)							
1	RESERVED			SERVICE ACTION (05h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_V				
5								(LSB)
6	(MSB)							
7				LIST LENGTH				
8								
9								(LSB)
10	RESERVED				CONTVER	VERIFY RANGE		IMMED
11	CONTROL							

The LUN_V field specifies the address of the volume set that shall have any check data associated with protected space contained within any underlaying redundancy group(s) be verified. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED.

An immediate (IMMED) bit of zero indicates that status shall be returned after the verification operation has completed at least one time regardless of the CONTVER fields value. An IMMED bit value of one indicates that the target shall return status as soon as the command descriptor block has been validated, and the entire VERIFY VOLUME SET CHECK DATA parameters list has been transferred.

The VERIFY RANGE field is defined in table 97.

Table 97 - VERIFY RANGE

Codes	Description
00b	Verify all volume sets within the selected target. The LUN_V field and the VERIFY VOLUME SET CHECK DATA parameter list shall be ignored.
01b	Verify the entire selected volume set. The VERIFY VOLUME SET CHECK DATA parameter list shall be ignored.
10b	Only verify the selected LBA_V range within the selected volume set.
11b	Reserved

If any part of the selected volume set(s) has an underlaying redundancy group protection with a redundancy type of no redundancy the VERIFY VOLUME SET CHECK DATA service action shall terminate with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code to INVALID FIELD IN CDB.

A continuous verification bit (CONTVER) of zero indicates the check data shall be verified only once. A CONTVER bit of one indicates the check data shall be continuously verified. Verification shall continue until a VERIFY VOLUME SET CHECK DATA service action is requested with the CONTVER bit set to zero and the VERIFY RANGE field set to 00h or the VERIFY RANGE field set to 01h or 10h and the LUN_V field set to the address of the volume set that is equal to the LUN_V from the original VERIFY VOLUME SET CHECK DATA service action.

NOTE 24 - If continuous verification is selected the verification executes as a background operation within the target in a vendor specific manner. Continuously may be defined as only verifying check data that underlay the LBA_V range of any write commands that occur within the range of LBA_V(s) that overlay a redundancy group that has continuous verification enabled.

Any verification failures occurring before the VERIFY VOLUME SET CHECK DATA service action has completed shall cause the target to terminate the command with a CHECK CONDITION status. The sense key shall be set to MEDIUM ERROR, and the additional sense code shall be set to MISCOMPARE DURING VERIFY OPERATION.

Any verification failures occurring after the VERIFY VOLUME SET CHECK DATA service action has completed the target shall generate a unit attention condition for all initiators. When reporting the unit attention condition the additional sense code shall be set to MISCOMPARE DURING VERIFY OPERATION.

The VERIFY VOLUME SET CHECK DATA parameter list (see table 98) contains the range of LBA_V(s) to be verified.

The relationship between the VERIFY VOLUME SET CHECK DATA service action and the VERIFY CHECK DATA service action when both service actions have requested the same area be verified is vendor specific. This standard only requires the requested area be verified.

Table 98 - VERIFY VOLUME SET CHECK DATA parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				START LBA_V				
2								
3								(LSB)
4	(MSB)							
5				NUMBER OF LBA_V(S)				
6								
7								(LSB)

The START LBA_V field specifies the LBA_V the target shall use to begin the verification.

The NUMBER OF LBA_V(S) field specifies the number of consecutive LBA_V(s) the target shall use for the verification.

6.7 SPARE (IN) command

6.7.1 SPARE (IN) command service actions

The service actions for the SPARE (IN) command are listed in table 99.

Table 99 - Service actions for SPARE (IN) command

Service name	Service actions	Type	Subclause
REPORT P_EXTENT SPARE	00h	M	6.7.1.1
REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE	01h	M	6.7.1.2
RESERVED	02h-17h		
VENDOR SPECIFIC	18h-1Fh		
Key: M = Service action implementation is mandatory. O = Service action implementation is optional.			

6.7.1.1 REPORT P_EXTENT SPARE service action

The REPORT P_EXTENT SPARE service action (see table 100) requests that information regrading p_extent spares within the target be sent to the application client.

Table 100 - REPORT P_EXTENT SPARE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BCh)							
1	RESERVED			SERVICE ACTION (00h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_S				
5								(LSB)
6	(MSB)							
7				ALLOCATION LENGTH				
8								
9								(LSB)
10	RESERVED							RPTSEL
11	CONTROL							

The LUN_S field specifies the address of the p_extent spare that shall be reported per table 101. If the requested logical unit has not been configured the command shall be terminated with a CHECK

CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED.

A report selected bit (RPTSEL) of zero indicates the target shall report all the p_extent spare(s) within the target. The LUN_S field shall be ignored when the RPTSEL bit is zero. A RPTSEL bit of one indicates the target shall report only on the p_extent spare indicated in the LUN_S field.

The REPORT P_EXTENT SPARE parameter list (see table 101) contains a four-byte header that contains the length in bytes of the parameter list and a list of REPORT P_EXTENT SPARE DESCRIPTORS.

Table 101 - REPORT P_EXTENT SPARE parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				REPORT P_EXTENT SPARE LIST LENGTH (n-3)				
2								
3								(LSB)
	REPORT P_EXTENT SPARE DESCRIPTORS(S) (if any)							
4				REPORT P_EXTENT SPARE DESCRIPTOR (First) (Length x)				
x+3								
	. . .							
n-y+1				REPORT P_EXTENT SPARE DESCRIPTOR (Last) (Length y)				
n								

The REPORT P_EXTENT SPARE LIST LENGTH field specifies the length in bytes of the following REPORT P_EXTENT SPARE DESCRIPTOR(s).

The REPORT P_EXTENT SPARE DESCRIPTOR is defined in table 102.

Table 102 - Format of REPORT P_EXTENT SPARE DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)			LUN_S				
1								(LSB)
2	RESERVED							
3	RESERVED							
4				P_EXTENT DESCRIPTOR				
15								
16	RESERVED							
17	RESERVED	STATE OF THE SPARE						
18	(MSB)			COVERED LIST LENGTH (n-19)				
19								(LSB)
20	RESERVED							
21	RESERVED							
22	(MSB)			COVERED LUN_R LIST LENGTH (m-23)				
23								(LSB)
	COVERED LUN_R(S) (if any)							
24	RESERVED							
25	RESERVED							
26				COVERED LUN_R 0				
27								
	. . .							
m-3	RESERVED							
m-2	RESERVED							
m-1				COVERED LUN_R x				
m								
	COVERED P_EXTENT DESCRIPTOR(S) (if any)							
m+1				COVERED P_EXTENT DESCRIPTOR 0				
m+12								
	. . .							
n-11				COVERED P_EXTENT DESCRIPTOR y				
n								

The LUN_S field specifies the address of the p_extent spare that covers the listed P_EXTENT SPARE DESCRIPTOR(s).

The P_EXTENT DESCRIPTOR contains the position and range of the p_extent spare addressed in the LUN_S field. See table 14 for a description of the P_EXTENT DESCRIPTOR.

The STATE OF THE SPARE field is defined in table 41. If the STATE OF THE SPARE field contains the state 'spare in use' then only report the redundancy group or p_extent being covered by the p_extent spare indicated in the LUN_S field of the REPORT P_EXTENT SPARE DESCRIPTOR.

The COVERED LIST LENGTH field specifies the length in bytes of the following COVERED P_EXTENT DESCRIPTOR(s) and COVERED LUN_R fields.

The COVERED LUN_R LIST LENGTH field specifies the length in bytes of the following COVERED LUN_R FIELD(s).

The COVERED LUN_R field contains the address of a redundancy group covered by the p_extent spare addressed in the LUN_S field.

The COVERED P_EXTENT DESCRIPTOR contains the position and range of a p_extent covered by the p_extent spare addressed in the LUN_S field. See table 14 for a description of the COVERED P_EXTENT DESCRIPTOR.

6.7.1.2 REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action

The REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action (see table 103) requests that information regarding peripheral device spares or component device spares within the target be sent to the application client.

Table 103 - REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BCh)							
1	RESERVED			SERVICE ACTION (01h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_S				
5								(LSB)
6	(MSB)			ALLOCATION LENGTH				
7								
8								
9								(LSB)
10	RESERVED						PORCSEL	RPTSEL
11	CONTROL							

The LUN_S field specifies the address of the peripheral device spare or the component device spare that shall be reported per table 104. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the

additional sense code set to LOGICAL UNIT NOT CONFIGURED.

A report selected bit (RPTSEL) of zero indicates the target shall report all the peripheral device spare(s) or component device spare(s) within the target. The LUN_S field shall be ignored when the RPTSEL bit is zero. A RPTSEL bit of one indicates the target shall report only on the peripheral device spare or the component device spare indicated in the LUN_S field.

The report peripheral device spare or component device spare selection bit (PORCSEL) of zero indicates the target shall report on all the peripheral device spares within the target. A PORCSEL bit of one indicates the target shall report on all the component device spares within the target.

The REPORT COMPONENT DEVICE/PERIPHERAL DEVICE SPARE parameter list (see table 104) contains a four-byte header that contains the length in bytes of the parameter list and a list of REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE DESCRIPTORS.

Table 104 - REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1		REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE LIST LENGTH (n-3)						
2								
3								(LSB)
	REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE DESCRIPTORS(S) (if any)							
4		REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE DESCRIPTOR (First) (Length x)						
x+3								
	. . .							
n-y+1		REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE DESCRIPTOR (Last) (Length y)						
n								

The REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE LIST LENGTH field specifies the length in bytes of the following REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE DESCRIPTOR(s).

The REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE DESCRIPTOR is defined in table 105.

Table 105 - Format of REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				LUN_S				(LSB)
2				RESERVED				
3				RESERVED				
4								
5				LUN_P or LUN_C				
6				RESERVED				
7	RESERVED				STATE OF THE SPARE			
8				RESERVED				
9				RESERVED				
10	(MSB)							
11				COVERED LOGICAL UNIT LIST LENGTH (n-11)				(LSB)
				COVERED LOGICAL UNIT DESCRIPTOR(S) (if any)				
12								
15				COVERED LOGICAL UNIT DESCRIPTOR 0				
				. . .				
n-3								
n				COVERED LOGICAL UNIT DESCRIPTOR y				

The LUN_S field specifies the address of the peripheral device spare or component device spare that covers the listed PERIPHERAL DEVICE/COMPONENT DEVICE SPARE DESCRIPTOR(s).

The LUN_P or LUN_C field contains the address of the logical unit that defines the peripheral device spare or the component device spare addressed in the LUN_S field.

The STATE OF THE SPARE field is defined in table 41. If the STATE OF THE SPARE field contains the state 'spare in use' then only report the logical unit being covered by the peripheral device spare or component device spare indicated in the LUN_S field of the REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE DESCRIPTOR.

The COVERED LOGICAL UNIT LIST LENGTH field specifies the length in bytes of the following COVERED LOGICAL UNIT DESCRIPTORS.

The COVERED LOGICAL UNIT DESCRIPTOR (see table 106) contains the list of logical units covered by the

peripheral device or component device spare addressed in the LUN_S field.

Table 106 - Data format of COVERED LOGICAL UNIT DESCRIPTOR

Bit / Byte	7	6	5	4	3	2	1	0
0	RESERVED							
1	RESERVED				LOGICAL UNIT TYPE			
2	(MSB)			LUN				
3								(LSB)

The LOGICAL UNIT TYPE field (see table 107) indicates the type of logical unit addressed in the LUN field.

Table 107 - LOGICAL UNIT TYPES

Codes	Descriptions
0h	Physical logical unit (peripheral device)
1h	Reserved
2h-3h	Reserved
4h	Component logical unit (component device)
5h	Redundancy group
6h	Reserved
7h-Bh	Reserved
Ch-Fh	Vendor specific

The LUN field contains the logical unit number of the logical unit indicated by the LOGICAL UNIT TYPE field.

6.8 SPARE (OUT) command

6.8.1 SPARE (OUT) command service actions

The service actions for the SPARE (IN) command are listed in table 108.

Table 108 - Service actions for spare (out) command

Service name	Service actions	Type	Subclause
CREATE/MODIFY P_EXTENT SPARE	00h	O	6.8.1.1
CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE	01h	O	6.8.1.2
DELETE SPARE	02h	O	6.8.1.3
RESERVED	03h-17h		
VENDOR SPECIFIC	18h-1Fh		
Key: M = Service action implementation is mandatory. O = Service action implementation is optional.			

6.8.1.1 CREATE/MODIFY P_EXTENT SPARE service action

The CREATE/MODIFY P_EXTENT SPARE service action (see table 109) requests the target to create a p_extent spare or modify an existing p_extent spare. If the create operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to CREATION OF LOGICAL UNIT FAILED. If the modification operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to MODIFICATION OF LOGICAL UNIT FAILED.

Table 109 - CREATE/MODIFY P_EXTENT SPARE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BDh)							
1	RESERVED			SERVICE ACTION (00h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_S				
5								(LSB)
6	(MSB)			LIST LENGTH				
7								
8								
9								(LSB)
10	RESERVED	SETLUN	RESERVED					IMMED
11	CONTROL							

The LUN_S field specifies the address of the p_extent spare that shall be created or modified. If the requested logical unit is not configurable the command shall be terminated with a CHECK CONDITION

status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

An immediate (IMMED) bit of zero indicates that status shall be returned after the create/modify p_extent spare operation has completed. An IMMED bit value of one indicates that the target shall return status as soon as the command descriptor block has been validated, and the entire CREATE/MODIFIY P_EXTENT SPARE parameters list has been transferred.

A set LUN (SETLUN) bit of zero indicates the target shall assign to the p_extent spare being created the logical unit number contained in the LUN_S field. A SETLUN bit of one indicates that the target shall assign a logical unit number to the p_extent spare. The LUN_S field shall be ignored when the SETLUN bit is set to one.

The CREATE/MODIFY P_EXTENT SPARE parameter list (see table 110) contains a list of COVERED P_EXTENT DESCRIPTORS that shall be used to create or modify the addressed p_extent spare.

Table 110 - CREATE/MODIFY P_EXTENT SPARE parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	SPARE P_EXTENT DESCRIPTOR							
11								
12	(MSB)	COVERED LIST LENGTH (n-13)						
13								(LSB)
14	(MSB)	COVERED LUN_R LIST LENGTH (m-15)						
15								(LSB)
	COVERED LUN_R(S) (if any)							
16	RESERVED							
17	RESERVED							
18	COVERED LUN_R 0							
19								
	. . .							
m-3	RESERVED							
m-2	RESERVED							
m-1	COVERED LUN_R x							
m								
	COVERED P_EXTENT DESCRIPTOR(S) (if any)							
m+1	COVERED P_EXTENT DESCRIPTOR 0							
m+12								
	. . .							
n-11	COVERED P_EXTENT DESCRIPTOR y							
n								

The SPARE P_EXTENT DESCRIPTOR contains the position and range of the p_extent spare that shall be created or modified. See table 14 for a description of the SPARE P_EXTENT DESCRIPTOR.

The COVERED LIST LENGTH field specifies the length in bytes of the following COVERED LUN_R fields and COVERED P_EXTENT DESCRIPTOR(s).

The COVERED LUN_R LIST LENGTH field specifies the length in bytes of the following COVERED LUN_R FIELD(s).

The COVERED LUN_R field contains the address of a redundancy group that shall be covered by the p_extent spare addressed in the LUN_S field.

The COVERED P_EXTENT DESCRIPTOR contains the position and range of a p_extent that shall be covered by the p_extent spare addressed in the LUN_S field. See table 14 for a description of the COVERED P_EXTENT DESCRIPTOR.

6.8.1.2 CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action

The CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action (see table 111) requests the target to create a peripheral device spare or a component device spare or modify an existing peripheral device spare or a component device spare. If the create operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to CREATION OF LOGICAL UNIT FAILED. If the modification operation fails to complete successfully the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to HARDWARE ERROR, and the additional sense code set to MODIFICATION OF LOGICAL UNIT FAILED.

Table 111 - CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BDh)							
1	RESERVED			SERVICE ACTION (01h)				
2	(MSB)							
3	LUN_P or LUN_C							(LSB)
4	(MSB)							
5	LUN_S							(LSB)
6	(MSB)							
7								
8	LIST LENGTH							
9								(LSB)
10	RESERVED	SETLUN	RESERVED				PORCSEL	IMMED
11	CONTROL							

The LUN_P or LUN_C field contains the address of the logical unit that defines the peripheral device spare or the component device spare addressed in the LUN_S field. If the requested logical unit is not configurable the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

The LUN_S field specifies the address of the peripheral device spare or the component device spare that shall be created or modified.

An immediate (IMMED) bit of zero indicates that status shall be returned after the create/modify peripheral device/component device spare operation has completed. An IMMED bit value of one indicates that the target shall return status as soon as the command descriptor block has been validated, and the entire CREATE/MODIFIY PERIPHERAL DEVICE/COMPONENT DEVICE spare parameters list has been transferred.

The peripheral device spare or component device spare selection bit (PORCSEL) of zero indicates the target shall create or modify a peripheral device spare. When PORCSEL is zero the LUN_P or LUN_C field shall contain a lun_p value. A PORCSEL bit of one indicates the target shall create or modify a component device spare. When PORCSEL is one the LUN_P or LUN_C field shall contain a lun_c value.

A set LUN (SETLUN) bit of zero indicates the target shall assign to the peripheral device spare or component device spare being created the logical unit number contained in the LUN_S field. A SETLUN bit of one indicates that the target shall assign a logical unit number to the peripheral device spare or component device spare. The LUN_S field shall be ignored when the SETLUN bit is set to one.

The CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE parameter list (see table 112) contains a list of COVERED LOGICAL UNIT DESCRIPTORS that shall be used to create or modify the addressed peripheral device spare or component device spare.

Table 112 - CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE parameter list

Bit Byte	7	6	5	4	3	2	1	0
	COVERED LOGICAL UNIT(S) (if any)							
0	COVERED LOGICAL UNIT DESCRIPTOR 0							
3								
	. . .							
n-3	COVERED LOGICAL UNIT DESCRIPTOR X							
n								

The COVERED LOGICAL UNIT DESCRIPTOR contains the address of a logical unit that shall be covered by the peripheral device spare or component device spare being created or modified. See table 106 for a description of the COVERED LOGICAL UNIT DESCRIPTOR.

6.8.1.3 DELETE SPARE service action

The DELETE SPARE service action (see table 113) requests the target delete the addressed spare.

Table 113 - DELETE SPARE service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BDh)							
1	RESERVED			SERVICE ACTION (02h)				
2	RESERVED							
3	RESERVED							
4	(MSB)	LUN_S						
5								(LSB)
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
10	RESERVED							
11	CONTROL							

The LUN_S field specifies the address of the spare that shall be deleted. If the requested logical unit has not been configured the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED. If the addressed spare has covered one of the redundancy groups, peripheral devices, component devices, or p_extents the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to REMOVE OF LOGICAL UNIT FAILED.

6.9 Parameters for direct-access devices

6.9.1 Mode parameters

This subclause defines the descriptors and pages for mode parameters used with SCSI-3 storage array devices.

The mode parameter list, including the mode parameter header and mode block descriptor are described in the SCSI-3 Primary Commands Standard.

The MEDIUM TYPE field is contained in the mode parameter header (see SCSI-3 Primary Commands Standard). For SCSI-3 storage array devices the MEDIUM TYPE field is reserved.

The DEVICE SPECIFIC PARAMETER field is contained in the mode parameter header (see SCSI-3 Primary Commands Standard). For SCSI-3 storage array devices the DEVICE SPECIFIC PARAMETER field is reserved.

The DENSITY CODE field is contained in the mode parameter block descriptor (see SCSI-3 Primary Commands Standard). For SCSI-3 storage array devices the DENSITY CODE field is reserved.

The mode page codes for SCSI-3 storage array devices are shown in table 114.

Table 114 - Mode page codes

Page Code	Description	Subclause
0Ah	Control mode page	SPC
02h	Disconnect-reconnect page	SPC
1Bh	LUN mapping page	6.9.1.1
09h	Peripheral device page	SPC
0Dh	Power condition page	SPC
Key: SPC = SCSI-3 Primary Commands Standard.		

6.9.1.1 LUN mapping page

The LUN mapping page (see table 115) is only required for protocols that do not support 8 byte LUN addressing (see 5.2.1). The LUN mapping page contains a list of LUN mappings that may be used to address peripheral devices and volume sets within a target that conforms to the SCSI-3 Interlocked Protocol Standard. When the LUN mappings are being used LUNs 1-31 of the IDENTIFY message (See A.1) shall each point to a specific LUN mapping within the LUN mapping page. The LUN mapping shall be used by the target to determine to which bus/target/LUN to send the command attached to the IDENTIFY message.

Table 115 - LUN mapping page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	RESERVED	PAGE CODE (1Bh)					
1	PAGE LENGTH (FAh)							
2	RESERVED							
3	RESERVED							ACTIVE
4	(MSB)	LUN 1 MAPPING						
11								(LSB)
	. . .							
244	(MSB)	LUN 31 MAPPING						
251								(LSB)

The parameters savable (PS) bit is only used with the MODE SENSE command. This bit is reserved with the MODE SELECT command. A PS bit of one indicates that the target is capable of saving the page in a non-volatile vendor-specific location.

A LUN mapping active bit (ACTIVE) of zero indicates the LUN XX MAPPING fields shall not be used to address any peripheral devices or volume sets. An ACTIVE bit of one indicates a LUN XX MAPPING field shall be used to determine to which bus/target/LUN to send the command. LUN field addresses 1-31 of the IDENTIFY message (see A.1) shall reference a specific LUN XX MAPPING field within the LUN mapping page when the ACTIVE bit is one. (e.g., A LUN field value of 3 would cause the target to use the LUN 3 MAPPING field to determine the bus/target/LUN per addressing rules (see 5.2.1).

The LUN XX MAPPING fields specify the bus/target/LUN of a peripheral device or volume set. See table 3 for a definition of the LUN XX MAPPING field. A value of zeros in the LUN XX MAPPING field shall indicate an undefined bus/target/LUN.

Any request from an application client to change a LUN XX MAPPING field shall be delayed until all tasks using the LUN XX MAPPING field to be changed have completed. The MODE SELECT command shall not complete until all the LUN XX MAPPING fields being changed have been changed.

Any attempt by an application client to address an undefined bus/target/LUN shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to LOGICAL UNIT NOT SUPPORTED.

Annex A

(normative)

A.0 SCSI-3 storage array IDENTIFY message format

This annex defines a modification from the description in the SCSI-3 Interlocked Protocol Standard of the IDENTIFY message LUN field. In addition to the requirements and definitions listed in this annex, all requirements and definitions of the IDENTIFY message contained within the SCSI-3 Interlocked Protocol Standard shall be followed by all SCSI-3 storage arrays that conform to the SCSI-3 Interlocked Protocol Standard.

A.1 IDENTIFY message

The IDENTIFY message (see table A.1) shall be used between SCSI devices that conform to the SCSI-3 Interlocked Protocol Standard.

Table A.1 - IDENTIFY message format

Bit / Byte	7	6	5	4	3	2	1	0
0	IDENTIFY	DISCPRIV	VOLSEL	LUN				

See the SCSI-3 Interlocked Protocol Standard for the definition of the IDENTIFY bit.

See the SCSI-3 Interlocked Protocol Standard for the definition of the DISCPRIV bit.

A volume select bit (VOLSEL) bit of zero indicates that the target shall use the LUN field as either a pointer to a LUN XX MAPPING field of the LUN mapping mode page (see 6.9.1.1) or as the address of a peripheral device within the target. If the ACTIVE bit of the LUN mapping mode page is set to zero or there is no LUN mapping mode page then the LUN field indicates the address of a peripheral device. If the ACTIVE bit of the LUN mapping mode page is set to one then the LUN field is a pointer to a LUN XX MAPPING field of the LUN mapping mode page.

A VOLSEL bit of one indicates that the target shall use the LUN field as the address of a volume set. If the VOLSEL bit is set to one the LUN mapping mode page shall not be used.

The logical unit number (LUN) field indicates the address of a peripheral device, the address of a volume set or the pointer to a LUN XX MAPPING field. The response to an invalid value in the LUN field is described in SCSI-3 Architecture Model Standard.

Annex B

(informative)

B.0 SCSI-3 storage array addressing examples

This annex contains several examples addressing an SCSI-3 storage array.

B.1 Addressing Examples for the 8-byte LUN structure

Several addressing examples follow. The conventions used within these examples are:

Layer 1 M:Ts:Ps:Ls or M:P:T or M:L or u
Layer 2 M:Ts:Ps:Ls or M:P:T or M:L or u
Layer 3 M:Ts:Ps:Ls or M:P:T or M:L or u
Layer 4 M:Ts:Ps:Ls or M:P:T or M:L or u

Where:*M* is the address method (2 bit field)
P is the bus number (6 bit field)
Ps is the bus number (3 bit field)
T is the target (8 bit field)
Ts is the target (6 bit field)
L is the logical unit number (14 bit field)
Ls is the logical unit number (5 bit field)
u is unused and set to zero (16 bit field)

NOTE 25 - *P* and *Ps* is a value that starts at one, since the zero value is reserved for the SCSI-3 storage array and devices that have no external path.

NOTE 26 - *T* and *Ts* is a value that starts at zero and is limited to one less than the number of attachable SCSI devices, since the path initiator's address is also included in that address space.

B.1.1 Example 1:

Addressing the first layer SCSI-3 storage array (for all control, creation, management functions and for identify)

Layer 1 0:0:0
Layer 2 u
Layer 3 u
Layer 4 u

Addresses will appear on the first level paths as required by the function.

B.1.2 Example 2:

Addressing a fan at address 7 within the first layer SCSI-3 storage array (a component device not physically on an identifiable SCSI path.)

Layer 1 0:0:0
Layer 2 u
Layer 3 u
Layer 4 u

The address on the component device is within the CDB of the command.

B.1.3 Example 3:

Addressing a local peripheral device

The address of the second peripheral device on the third path would be:

Layer 1 0:3:1
Layer 2 u
Layer 3 u
Layer 4 u

The second level path would use path 3 to access target 1. The LUN value is 0.

B.1.4 Example 4:

Addressing any volume set controlled by the first SCSI-3 storage array (including volume sets constructed from ps_extents defined by lower SCSI-3 storage arrays)

Layer 1 1:L
Layer 2 u
Layer 3 u
Layer 4 u

Addresses will appear on the first level paths as required by the function.

B.1.5 Example 5:

Addressing an SCSI-3 storage array at the second layer of the hierarchy.

> NOTE 27 - The second layer SCSI-3 storage array is also a peripheral device on the first layer path. In this example the SCSI-3 storage array is on the fourth path and is the third target address.

Layer 1 0:4:2
Layer 2 0:0:0
Layer 3 u
Layer 4 u

A LUN address of the following form would be emitted on the fourth path to target address 2 (following the shift the address rules per 5.2.1.2) to get at the SCSI-3 storage array.

Layer 1 0:0:0
Layer 2 u
Layer 3 u
Layer 4 u

B.1.6 Example 6:

Addressing a peripheral device of the above second level SCSI-3 storage array (the second peripheral device on the third path of that second level SCSI-3 storage array).

Layer 1 0:4:2
Layer 2 0:3:1
Layer 3 u
Layer 4 u

The LUN address emitted on the fourth path to target address 2 (following the shift the address rules per 5.2.1.2). This peripheral device could be used as a component of a volume set defined by the first level

SCSI-3 storage array or the second level SCSI-3 storage array.

Layer 1 0:3:1
Layer 2 u
Layer 3 u
Layer 4 u

The LUN address emitted on the third path of the second level SCSI-3 storage array to target address 1 would be 0.

B.1.7 Example 7:

Addressing any volume set of the above second level SCSI-3 storage array.

NOTE 28 - The second level SCSI-3 storage array's entry path is being addressed directly. This volume set could be used as a component of a volume set defined by the first layer SCSI-3 storage array.

Layer 1 0:4:2
Layer 2 1:L
Layer 3 u
Layer 4 u

Address emitted on the fourth path to target address 2 (following the shift the address rules per 5.2.1.2). This is a standard volume set address.

Layer 1 1:L
Layer 2 u
Layer 3 u
Layer 4 u

Addresses will appear on the second level paths as required by the function.

B.1.8 Example 8:

Addressing a peripheral device that has LUN's behind a standard target. As an example, LUN 4 behind the second target on the third path of the first level SCSI-3 storage array.

Layer 1 2:1:3:4
Layer 2 u
Layer 3 u
Layer 4 u

B.1.9 Example 9:

NOTE 29 - Fourth layer devices must be single LUN devices.

As a peripheral device addressing example, if the first layer was P=3,T=7; the second layer was P=4, T=6; the third layer was P=1, T=5; and the fourth layer was P=7, T=2 then the address would be:

Layer 1 0:3:7
Layer 2 0:4:6
Layer 3 0:1:5
Layer 4 0:7:2

The LUN issued on the first layer path 3, target 7 would be:

Layer 1 0:4:6

Layer 2 0:1:5
Layer 3 0:7:2
Layer 4 u

The LUN issued on the second layer path 4, target 6 would be:

Layer 1 0:1:5
Layer 2 0:7:2
Layer 3 u
Layer 4 u

The LUN issued on the third layer path 1, target 5 would be:

Layer 1 0:7:2
Layer 2 u
Layer 3 u
Layer 4 u

The LUN issued on the fourth layer path 7, target 2 would be 0.

B.2 Addressing Examples for the 6-bit LUN structure

The following examples assume the LUN mapping page is not active.

Several addressing examples follow. The conventions used within these examples are:

Layer 1 V:L

Where: *V* is the Volume Select (1 bit field)
L is the Logical Unit Number (5 bit field)

NOTE 30 - Information on logical units at layers below layer 1 is not available to the application client when the 6-bit LUN structure is used. Logical units below layer 1 may only be addressed directly using a vendor specific translation of the 6-bit LUN.

B.2.10 Example 1:

Addressing the first layer SCSI-3 storage array (for all control, creation, management functions and for identify)

Layer 1 0:0

Addresses will appear on the first level paths as required by the function.

B.2.11 Example 2:

Addressing a fan at address 7 within the SCSI-3 storage array.

Layer 1 0:7

No first level path will be used.

B.2.12 Example 3:

Addressing any peripheral device.

The address of a drive in the SCSI-3 storage array:

Layer 1 0:L

Addresses will appear on the first level paths as required by the function.

B.2.13 Example 4:

Addressing any volume set:

Layer 1 1:L

Addresses will appear on the first level paths as required by the function.

Annex C

(informative)

C.0 Examples of check data and user data mappings

C.1 Example XOR redundancy mapping

Figure C.1 contains an example of how XOR redundancy check data would be interleaved with user data.

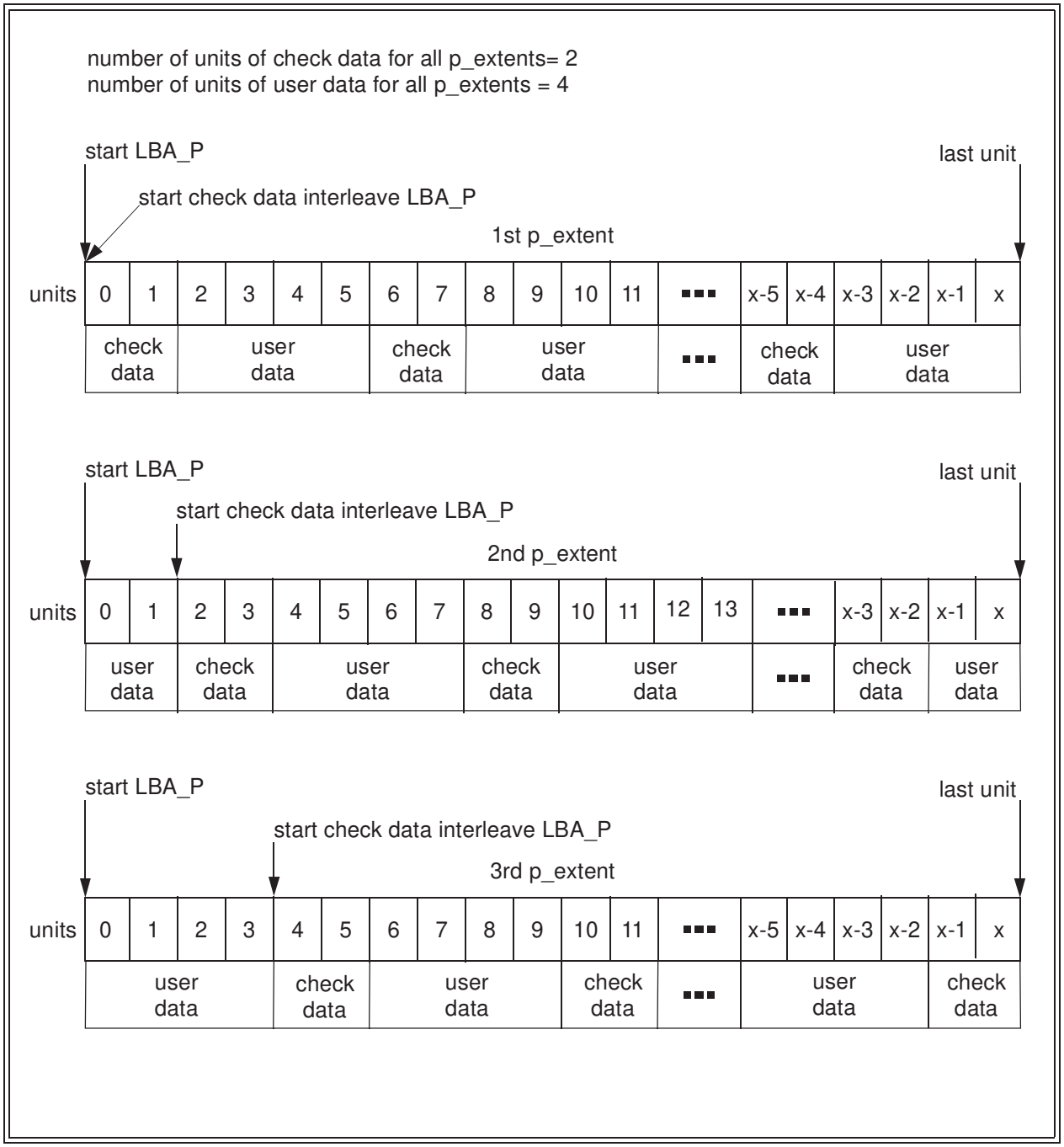

Figure C.1 - XOR redundancy mapping example

C.2 User data mapping examples

C.2.1 Example of user data mapped in a RAID 5 configuration

Figure C.2 contains an example of how user data would be mapped in a RAID 5 configuration.

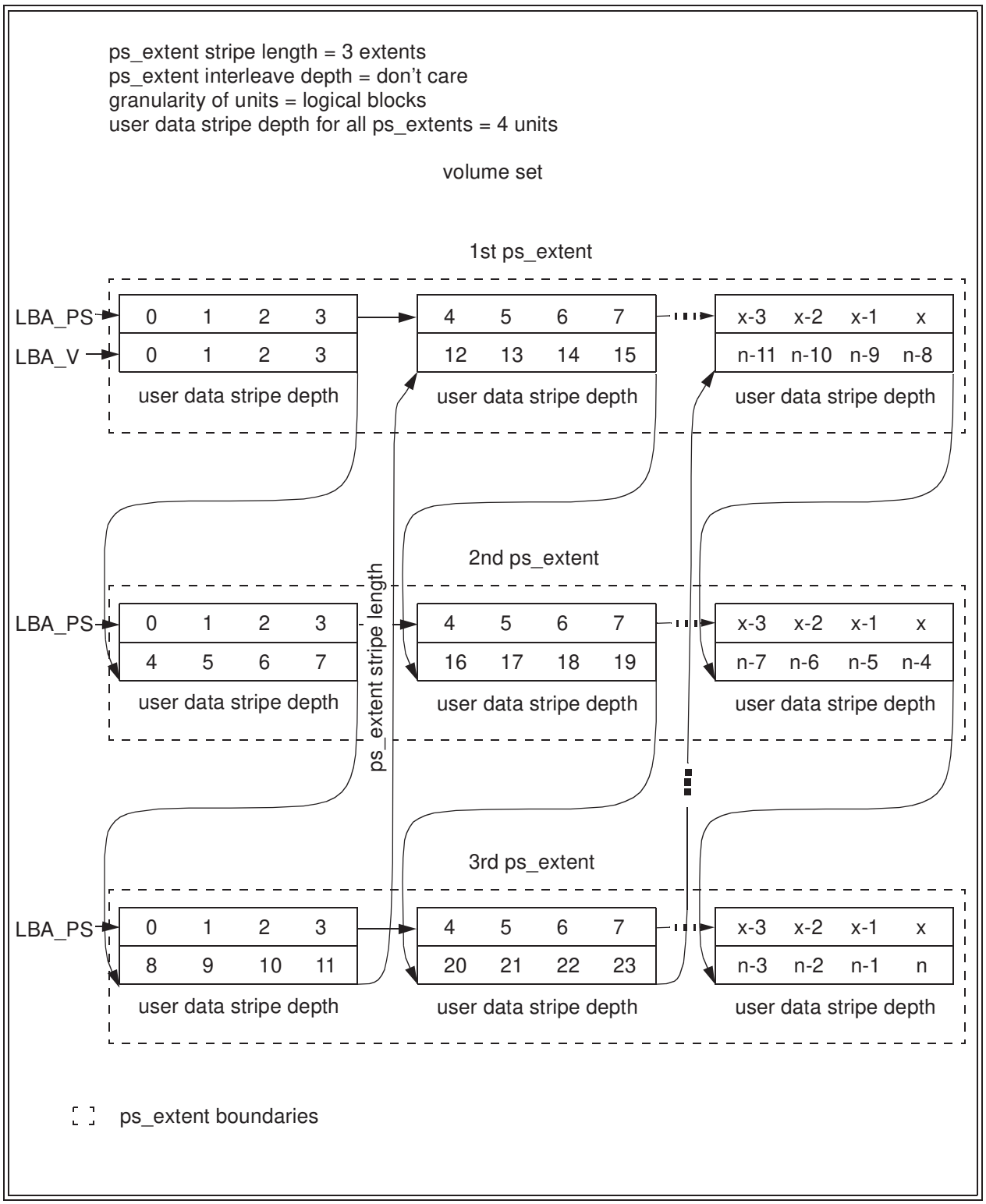

Figure C.2 - User data mapping for a RAID 5 configuration

C.2.2 Example of user data mapping in a RAID 3 configuration

Figure C.3 contains an example of how user data would be mapped in a RAID 3 configuration.

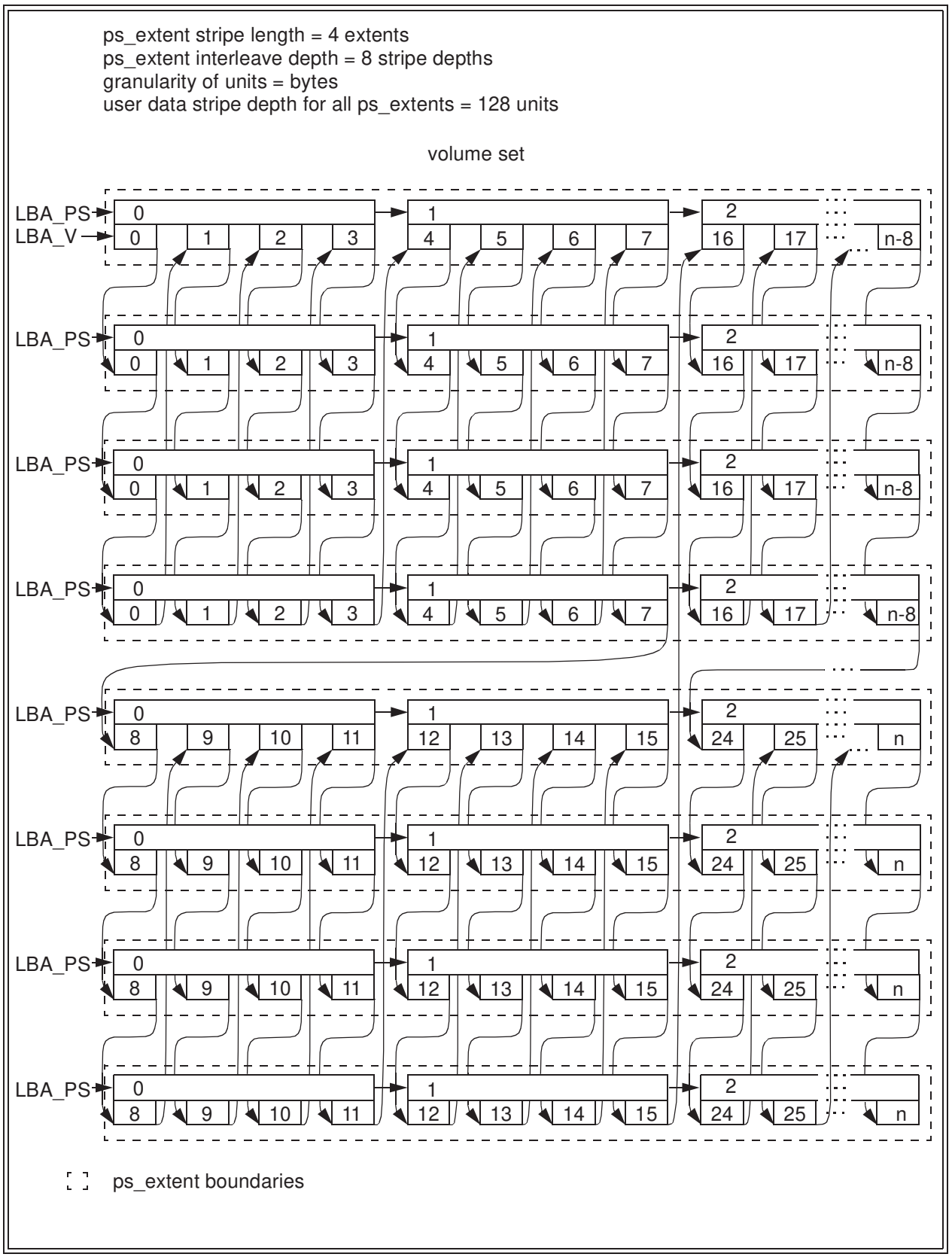

Figure C.3 - User data mapping for a RAID 3 configuration