Draft Proposed American National Standard

T10 Project 1225D

Revision 3
14-July-97

Information technology - SCSI Controller Commands-2 (SCC-2)

ASC T10 Technical Editor: George O. Penokie
IBM
Dept. 2B7
3605 Highway 52 N.
Rochester, MN 55901
USA

Telephone: 507-253-5208
Facsimile: 507-253-2880
Email: gop@us.ibm.com

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

Printed Tuesday, July 22, 1997 2:28 pm

POINTS OF CONTACT:

T10 Chair
John B. Lohmeyer
Symbios Logic
4420 Arrows West Drive
Colo Spgs, CO 80907-3444
Tel: (719) 533-7560
Fax: (719) 593-7036
Email: john.lohmeyer@symbios.com

T10 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

NCITS Secretariat
NCITS Secretariat
1250 Eye Street, NW Suite 200
Washington, DC 20005
Telephone: 202-737-8888
Facsimile: 202-638-4922
Email: ncits@itic.nw.dc.us

T10 Reflector Internet address for subscription of the T10 reflector: majordomo@symbios.com

Internet address for distribution via T10 reflector: T10@symbios.com

SCSI Bulletin Board 719-533-7950

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 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 command standards and the SCSI-3 Architecture Model Standard. The commands described in this standard facilitate the control and configuration of SCSI storage arrays and thus provide a common command specification for both system integrators and suppliers of SCSI 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

Tables

Figures

Foreword

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

The SCSI Controller Commands-2 (SCC-2) 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 storage array devices;

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

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

This standard was developed by a joint effort between the T10 SCSI Controller Commands working group and the RAID Advisory Board Host Interface working group during 1996-97. The standards approval process started in 1997.

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the NCITS Secretariat, National Committee for Information Technology Standards , 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, NCITS. Committee approval of the standard does not necessarily imply that all committee members voted for approval.

The joint T10 SCSI Controller Commands working group/RAID Advisory Board Host Interface working group, which developed this standard, has the following members:

Dal Allan
Corky Ball
Ed Barnes
Tom Coughlan
Roger Cummings
Rod DeKoning
George Ericson
Edward Fong
John Hartjen
Bill Hutchison
M.K. Jibbe
Paul Massiglia
Brett Quinn
Steve Sicola
Herb Silverman
Bob Snively
Dan Strevey
Al Tayler
Rick Wagner
Ralph Weber
Ruben Yomtoubian

George O. Penokie, Chair

Technical Committee T10 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

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 operating systems. However, details of the coordination are beyond the scope of the SCSI command set.

This standard defines a device model for SCSI storage arrays, commonly known as RAID devices. This standard defines the SCSI commands that may apply to SCSI storage arrays and the SCSI commands that are uniquely for SCSI storage arrays.

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

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

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

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

Figure 1 is intended to show the general relationship between SCSI 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.

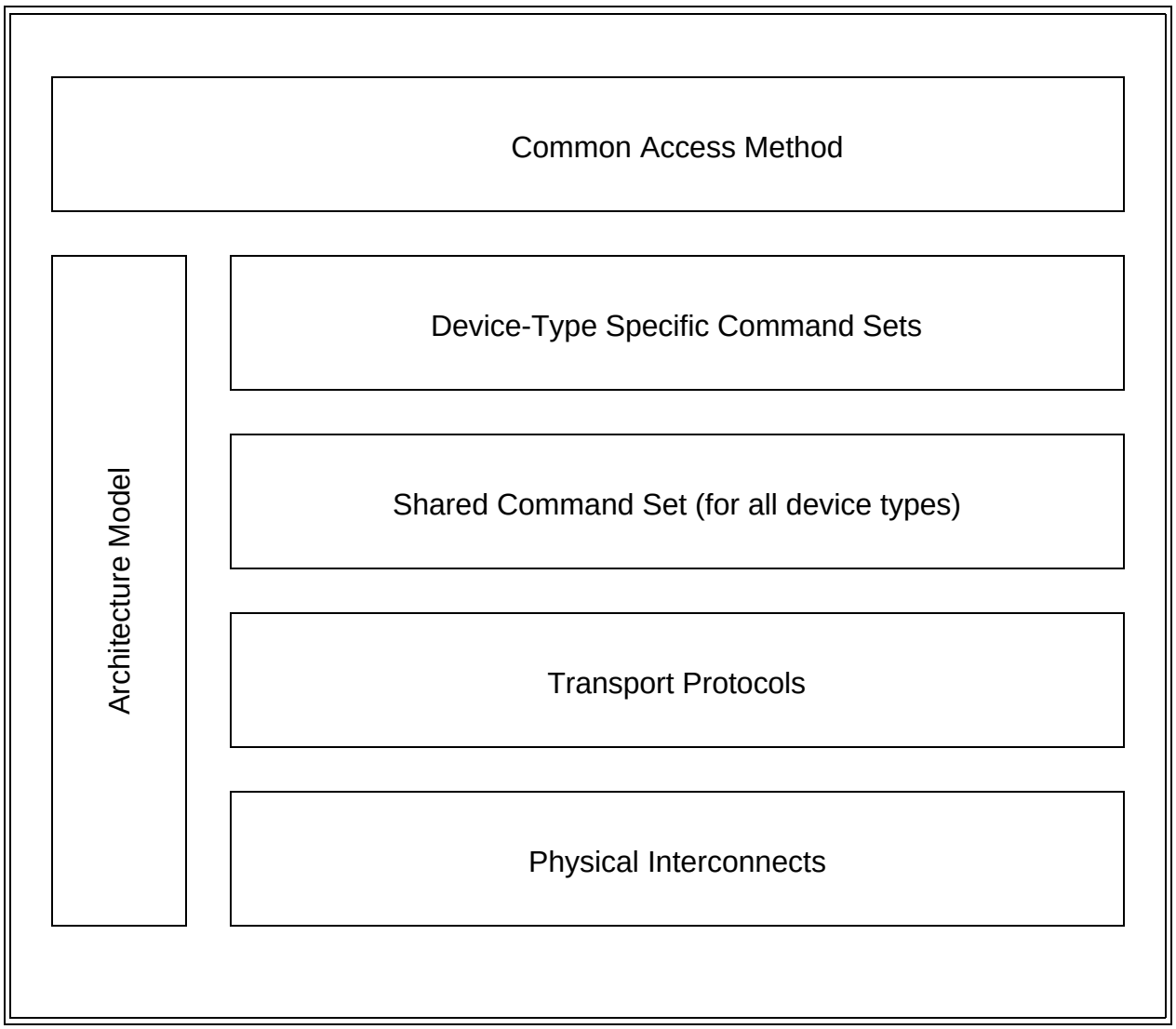

Figure 1 - SCSI document road map

At the time this standard was generated examples of the SCSI general structure included:

Physical Interconnects:

Fibre Channel Arbitrated Loop [X3t11/960D]
Fiber Channel - Physical and Signaling Interface [X3.230-1994]
High Performance Serial Bus [IEEE 1394-1995]
SCSI-3 Parallel Interface [X3.253]
SCSI-3 Fast-20 Parallel Interface [X3.277]
SCSI Parallel Interface - 2 [X3.302]
Serial Storage Architecture Physical Layer 1 [X3.293]
Serial Storage Architecture Physical Layer 2 [X3T10/1146D]

Transport Protocols:

SCSI-3 Interlocked Protocol [X3.292]
Serial Storage Architecture Transport Layer 1 [X3.295]
SCSI-3 Fiber Channel Protocol [X3.269]
SCSI-3 Fiber Channel Protocol - 2 [X3t10/1144D]
SCSI-3 Serial Bus Protocol - 2 [X3T10/1155D]
Serial Storage Architecture SCSI-2 Protocol [X3.294]
Serial Storage Architecture SCSI-3 Protocol [X3T10/1051D]
Serial Storage Architecture Transport Layer 2 X3T10/1147D]

Shared Command Set:

SCSI-3 Primary Commands Standard [X3.301-1997]

Device-Type Specific Commands Sets:

SCSI-3 Block Commands [X3T10/996D]
SCSI-3 Enclosure Services [X3T10/1212D]
SCSI-3 Stream Commands [X3T10/997D]
SCSI-3 Medium Changer Commands [X3T10/999D]
SCSI-3 Controller Commands [X3.276]
SCSI Controller Commands - 2 (this standard)
SCSI-3 Multimedia Command Set [X3T10/1048D]
SCSI-3 Multimedia Command Set - 2 [X3T10/1228D]

Architecture Model:

SCSI-3 Architecture Model [X3.270]
SCSI Architecture Model - 2 [X3T10/1157D]

Common Access Method:

SCSI Common Access Method [X3.232]
SCSI Common Access Method - 3 [X3T10/990D]

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 in this standard refers to a version of SCSI defined since SCSI-2

2 Normative references

The following standards contain provisions which, through reference in the text, constitute provisions of this 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 listed below.

Copies of the following documents can be obtained from ANSI: Approved ANSI standards, approved and draft international and regional standards (ISO, IEC, CEN/CENELEC, ITUT), and approved and draft foreign standards (including BSI, JIS, and DIN). For further information, contact ANSI Customer Service Department at 212-642-4900 (phone), 212-302-1286 (fax) or via the World Wide Web at http://www.ansi.org.

Additional availability contact information is provided below as needed.

2.1 Approved references

SCSI-3 Architecture Model Standard, ANSI X3.270 - 1996
SCSI-3 Primary Commands Standard, ANSI X3.301 - 1997
SCSI-3 Interlocked Protocol Standard, ANSI X3.292 - 1997

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 assignment: The linking of p_extents to redundancy groups or ps_extents to volume sets as defined within this standard (see 5.2.1.2.2 and 5.2.2.11).

3.1.3 association: The linking of SCSI storage array objects (see 5.2.2) in a manner explicitly defined within this standard (see 5.2.2.2).

3.1.4 attachment: The linking of SCSI storage array objects (see 5.2.2) in a vendor specific manner (see 5.2.2.3).

3.1.5 byte: 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. This standard intentionally avoids any definition of how check data is constructed or how it allows the recreation of user data.

3.1.7 check data mapping: The distribution of the check data within a redundancy group (see 5.2.2.12).

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 peripheral device type. See table 21 for a list of component devices.

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

3.1.12 covering: The linking of spare objects to other SCSI storage array objects (see 5.2.2) in a manner explicitly defined within this standard (see 5.2.1.2.2).

3.1.13 device server: An object within the logical unit which executes SCSI tasks according to the rules for task management described in the SCSI-3 Architecture Model Standard.

3.1.14 exchange: Replacing an object with all the characteristics (e.g., p_extents, ps_extents, protected space areas, check data areas, etc.) of another object (see 5.2.2.5).

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

3.1.16 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.

3.1.17 invalid: An illegal or unsupported field or code.

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

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

3.1.20 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.

3.1.21 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.

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

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

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

3.1.25 optional: The referenced item is not required to claim compliance with this standard. If an optional item is Implemented it shall be as defined in this standard.

3.1.26 page: Regular parameter structures used in several commands that are identified with a value known as a page code.

3.1.27 p_extent: All or part of the host addressable space within a single peripheral device of a SCSI storage array (see 5.2.2.10).

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

3.1.29 protected space: The portion of a redundancy group that does not contain check data (see 5.2.2.12).

3.1.30 ps_extent: All or part of a redundancy groups protected space contained within a single peripheral device (see 5.2.2.11).

3.1.31 rebuild operation: Re-creation and saving of all the protected space contents and any check data within a p_extent using check data and protected space contents from the remaining p_extents within the redundancy group (see 5.2.3.2).

3.1.32 recalculate operation: Re-creation of check data from protected space contents (see 5.2.1.2.2).

3.1.33 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 (see 5.2.2.12).

3.1.34 regenerate operation: Re-creation of inaccessible protected space contents from accessible check data and protected space contents (see 5.2.3.4).

3.1.35 reserved: Bits, fields, and codes that are set aside for future standardization.

3.1.36 SCSI storage array: A peripheral device that processes SCSI command descriptor blocks and performs the services of a SACL. A single SCSI storage array may contain multiple SACLs.

3.1.37 SCSI storage array logical unit number (LUN_Z): The logical unit number that an application client uses to communicate with, configure, and determine information about an SCSI storage array and the logical units attached to it (see 5.2.1.1). The LUN_Z value shall be zero.

3.1.38 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 device server and a command.

3.1.39 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 an entire set.

3.1.40 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 (see 5.2.2.13).

3.1.41 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 (see 5.1).

3.1.42 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.

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

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

3.1.45 unit: A standard basic quantity in bits, bytes, words, logical blocks, etc., specified by the GRANULARITY OF UNITS field (see table 76).

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

3.1.47 user data mapping: The distribution of user data within a volume set (see 5.2.2.14).

3.1.48 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.

3.1.49 verify operation: Re-creation of check data from protected space contents and the comparison of the recreated check data with the current check data (see 5.2.3.5).

3.1.50 volume set: One or more ps_extents grouped into a single LUN_V (see 5.2.2.14).

3.1.51 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
LUN_Z	SCSI storage array 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 Keywords

3.3.1 expected: A keyword used to describe the behavior of the hardware or software in the design models assumed by this standard. Other hardware and software design models may also be implemented.

3.3.2 invalid: A keyword used to describe an illegal or unsupported bit, byte, word, field or code value. Receipt of an invalid bit, byte, word, field or code value shall be reported as error.

3.3.3 mandatory: A keyword indicating an item that is required to be implemented as defined in this standard.

3.3.4 may: A keyword that indicated flexibility of choice with no implied preference.

3.3.5 obsolete: A keyword indicating that an item was defined in prior SCSI standards but has been removed from this standard.

3.3.6 optional: A keyword that describes features that are not required to be implemented by this standard. However, if any optional feature defined by this standards is implemented, then it shall be implemented as defined in this standard.

3.3.7 reserved: A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard. Recipients may check reserved bits, bytes, words or fields for zero values and report errors if non-zero values are received. Receipt of reserved code values in defined fields shall be reported as error.

3.3.8 shall: A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard.

3.3.9 should: A keyword indicating flexibility of choice with a strongly preferred alternative; equivalent to the phrase "it is strongly recommended".

3.4 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 storage arrays and the SCSI commands that apply to SCSI 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 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 storage array devices. The model assumes all the SCSI peripheral devices controlled within a SCSI 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, giving possible corrective actions.

5.1.2 Protocol conversion layer

A system is typically composed of many protocol conversion layers (such as the one shown in 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.

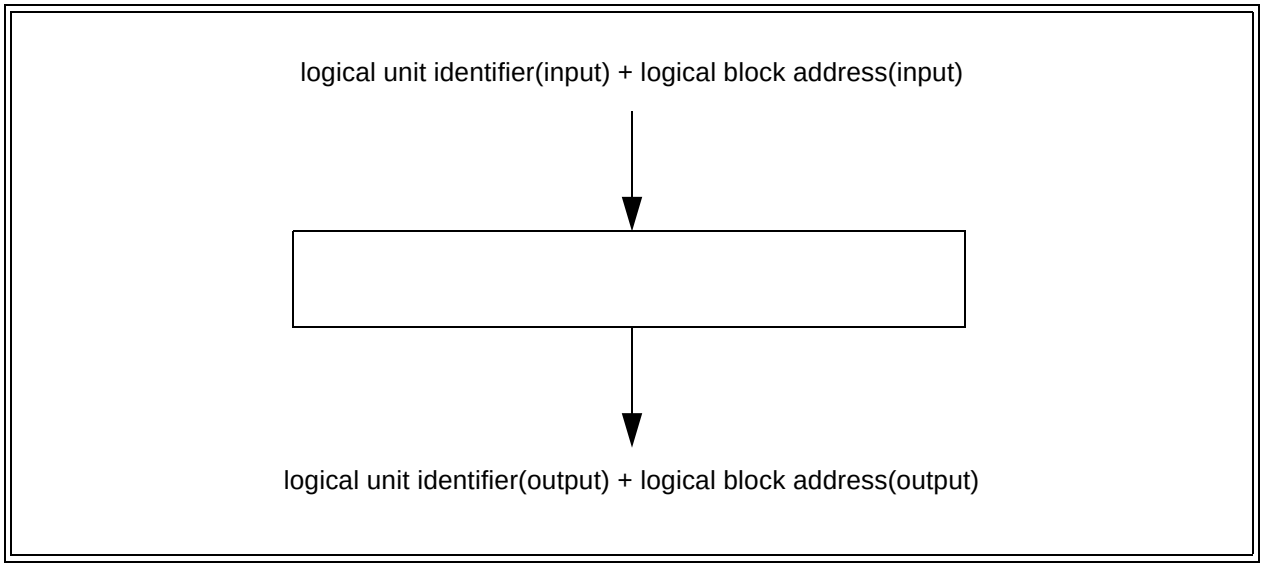

Figure 2 - Protocol conversion layer

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, they 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.

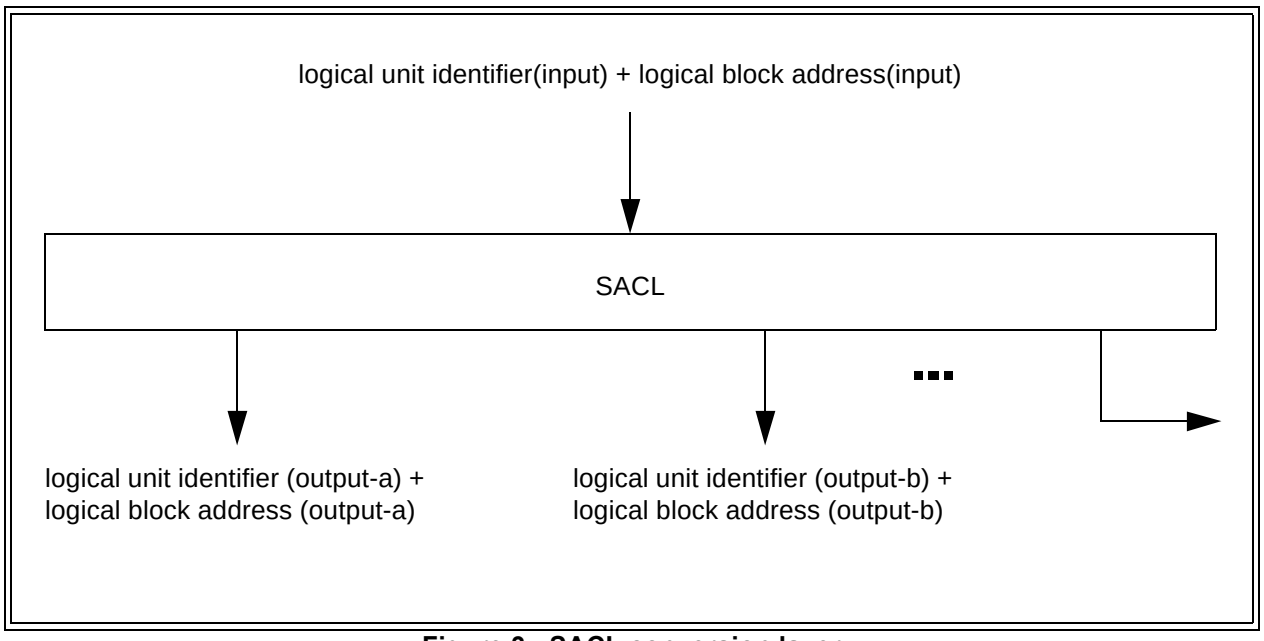

Figure 3 - SACL conversion layer

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.

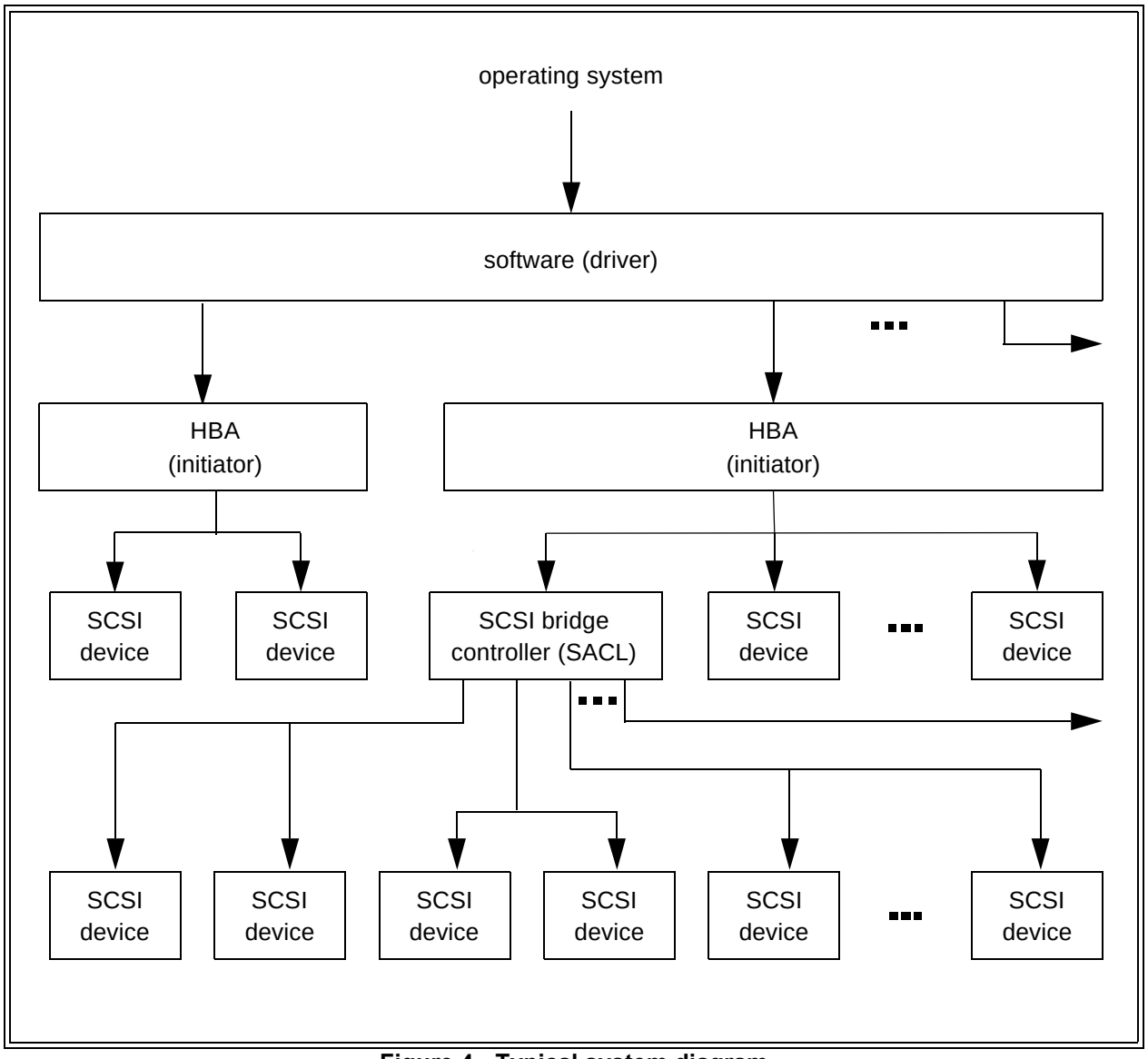

Figure 4 - Typical system diagram

5.1.5 Branch of generic layers

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

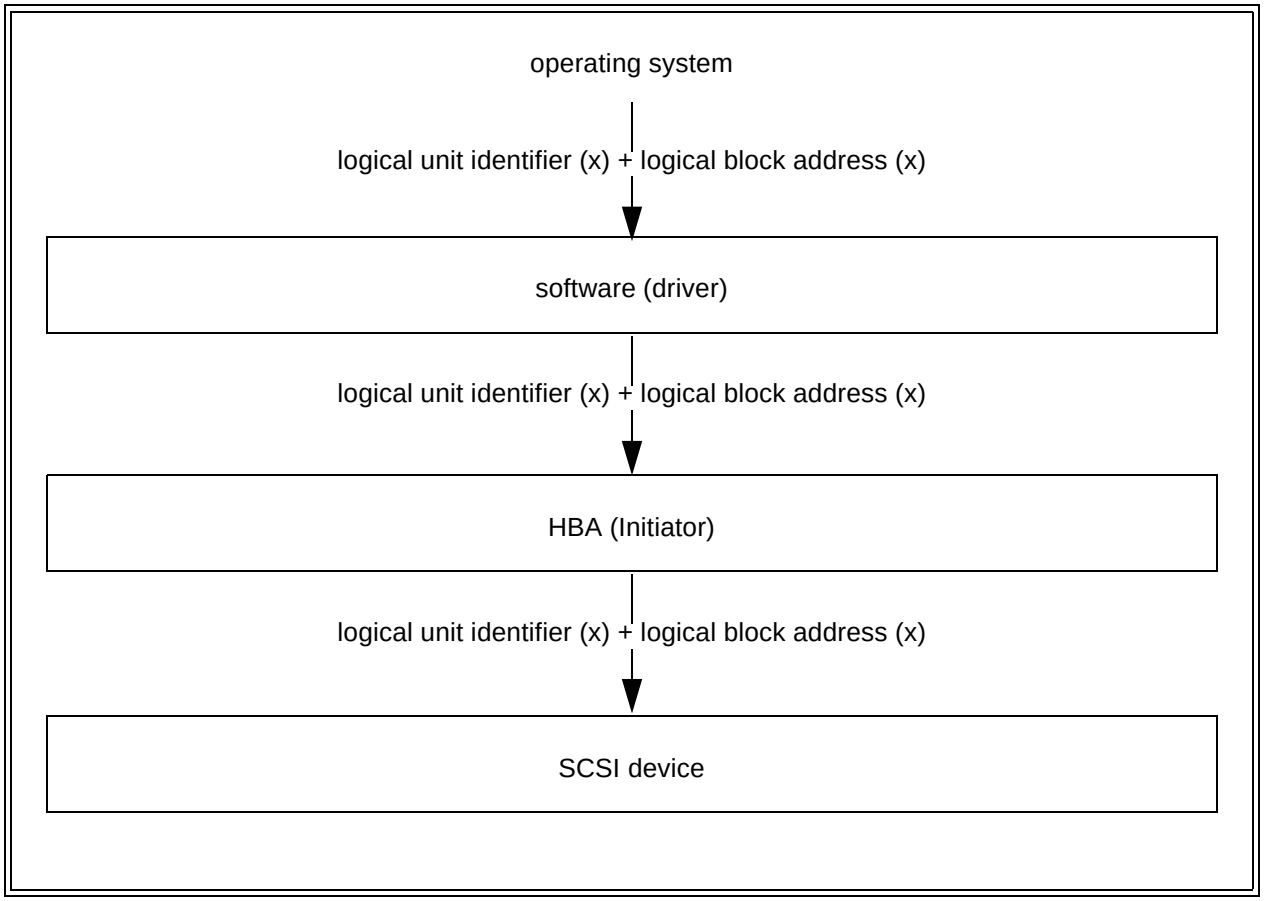

Figure 5 - Branch of generic layers

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 one or more output logical unit identifier(y)(s) and one or more output logical block address(y)(s). All other layers pass the logical unit identifier and logical block address through.

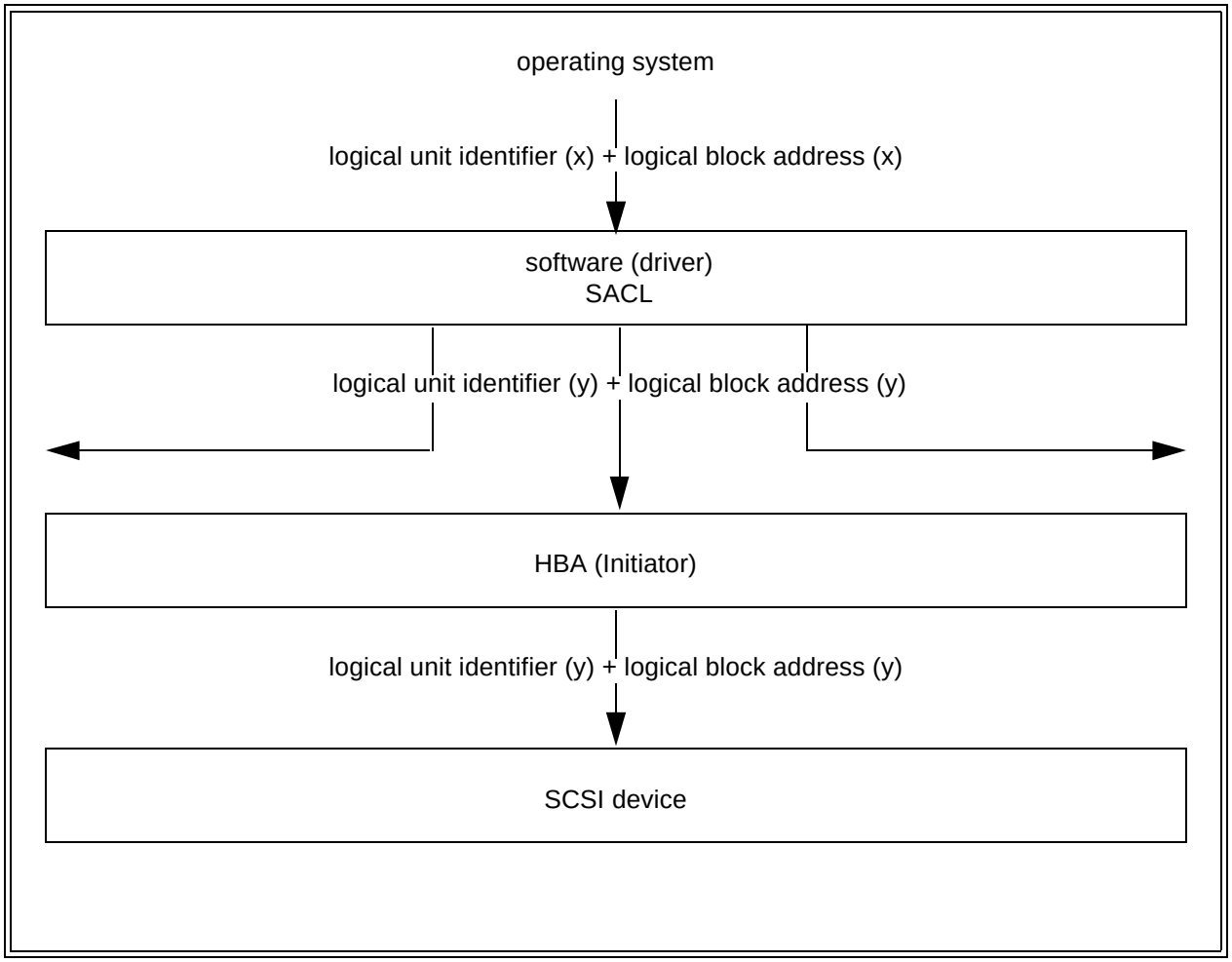

Figure 6 - Software SACL with a branch of SCSI disks

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 one or more output logical unit identifier(y)(s) and one or more output logical block address(y)(s). All other layers pass the logical unit identifier and logical block address through.

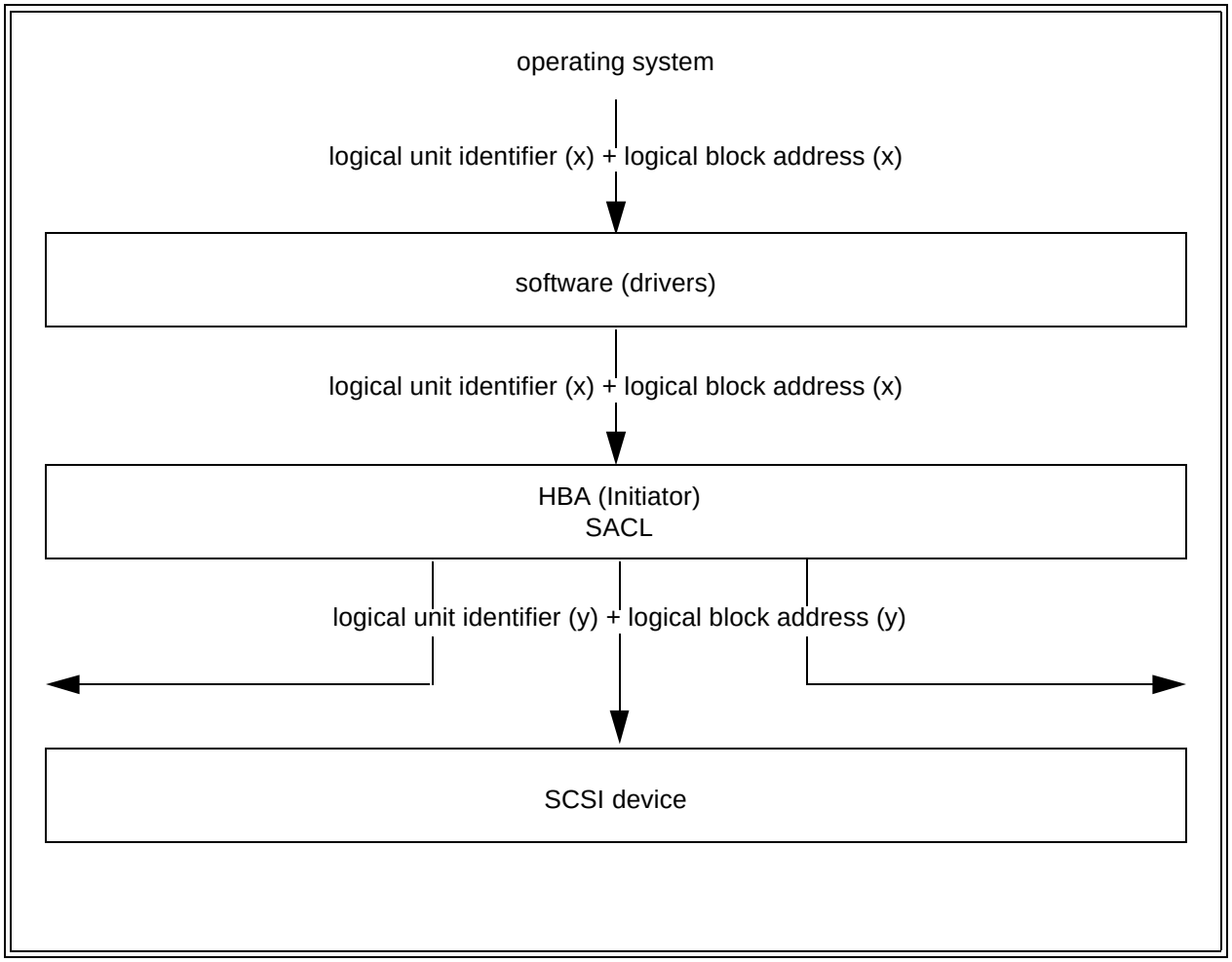

Figure 7 - Branch with HBA SACL

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 one or more output logical unit identifier(y)(s) and one or more output logical block address(y)(s). All other layers pass the logical unit identifier and logical block address through.

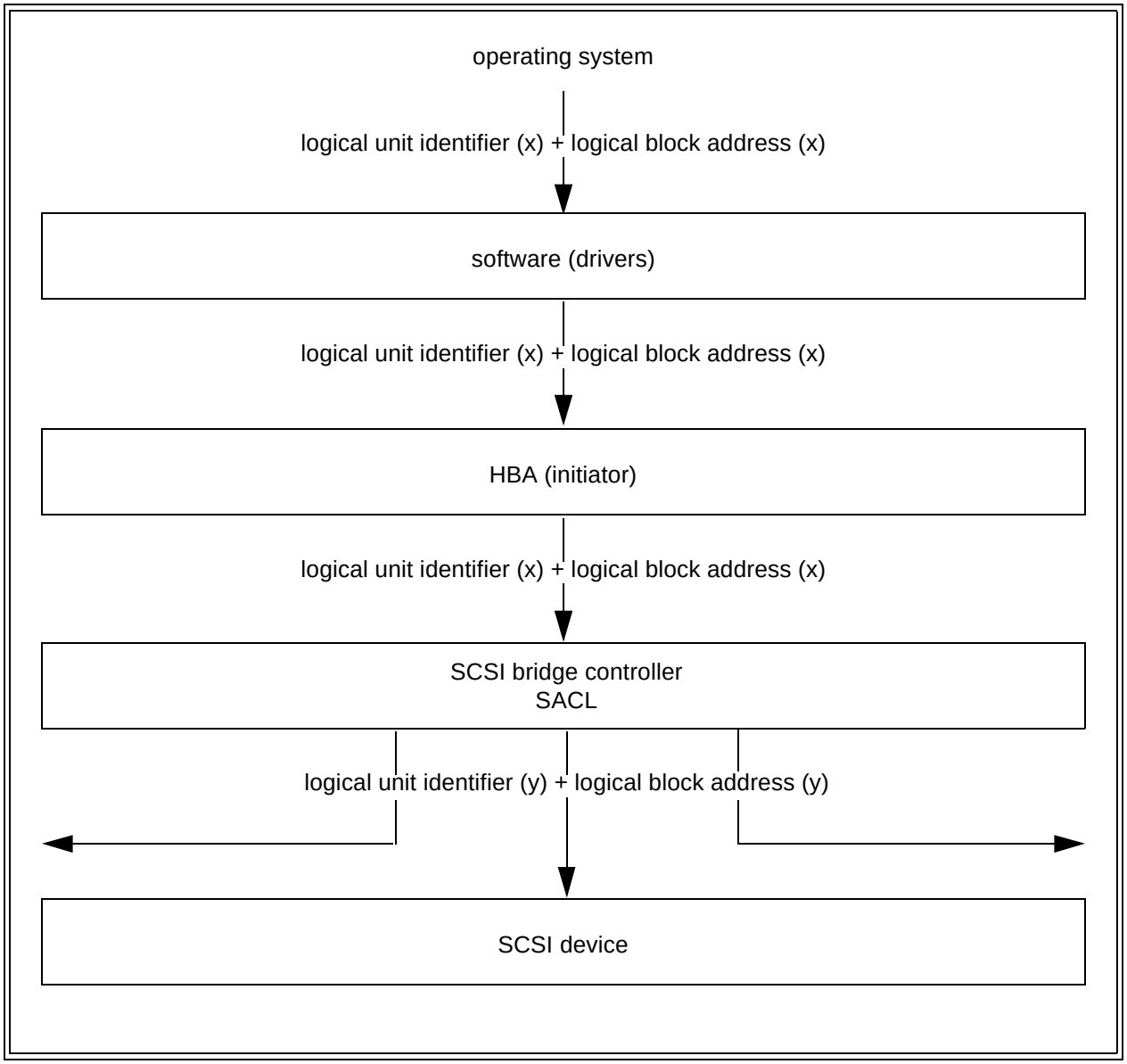

Figure 8 - Branch with bridge controller SACL

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 one or more output logical unit identifier(y)(s) and one or more output logical block address(y)(s). The bridge controller functions then convert the input logical unit identifier(y) and the input logical block address(y) to one or more output logical unit identifier(z)(s) and one or more output logical block address(z)(s). All other layers pass the logical unit identifier and logical block address through.

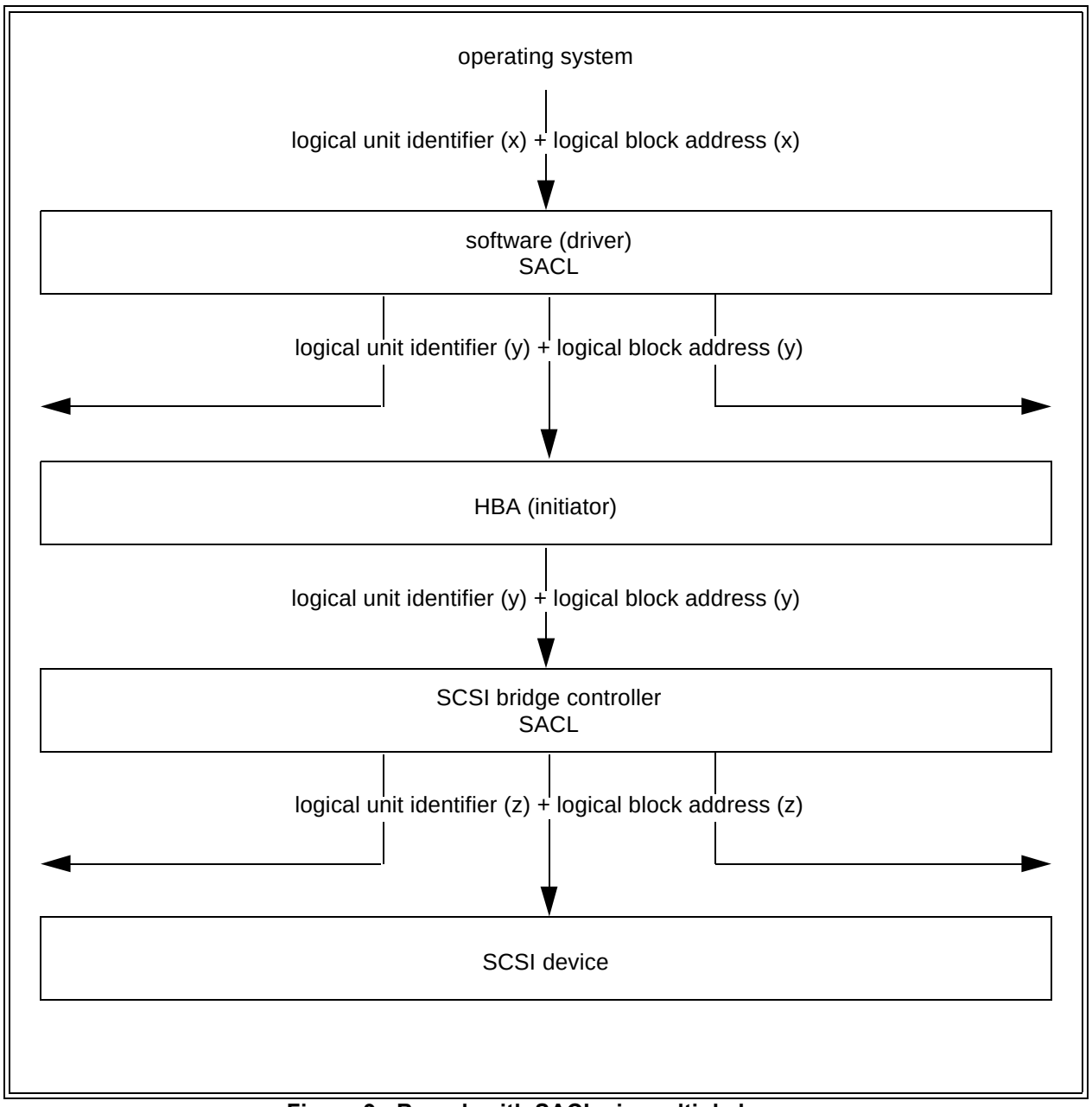

Figure 9 - Branch with SACLs in multiple layers

5.2 Model for SCSI storage arrays

The SCSI storage array model defines:

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

5.2.1 SCSI storage array addressing

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

Table 1 - Addressing methods within a SCSI storage array

Method of addressing	Application client addressing	Suffix	Space addressed
Component device	indirect	_C	none
Logical Unit	direct	_P	physical
Peripheral device	direct	_P	physical
Redundancy group	indirect	_R	protected
SCSI storage array	direct	_Z	none
Spare	indirect	_S	none
Volume set	direct	_V	user data

The application client shall access redundancy groups, spares, and components by issuing commands to the LUN_Z of an SCSI storage array. The application client may directly address peripheral devices and volume sets by using eight byte LUN fields (see 5.2.1.2.1). For SCSI 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 6.11.1.1) to allow direct addressing of peripheral devices and volume sets that conform to the SCSI storage array addressing model.

All peripheral device addresses, except LUN_Z, 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 storage array LUN_Z address

All SCSI storage arrays shall accept LUN_Z as a valid address. For SCSI storage arrays LUN_Z shall be the logical unit that an application client addresses to configure an SCSI storage array and to determine information about the target and the logical units contained within the target. INQUIRY commands sent to LUN_Z shall return a device type of array controller device.

The peripheral device address method shall be used when addressing the LUN_Z of an SCSI storage array (see 5.2.1.2.3).

5.2.1.2 Direct addressing

5.2.1.2.1 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.

SCSI storage array devices shall use the eight byte LUN structure described in the following paragraphs.

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 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
N/A	zero fill	6 - 7

The eight byte LUN structure requirements as viewed from the application client are shown in table 3.

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 address methods defined for the ADDRESS METHOD field. The ADDRESS METHOD field only defines address methods for objects that are directly addressable by an application client.

Table 5 - ADDRESS METHOD

Codes	Description	Clause
10b	Logical unit addressing method	5.2.1.2.2
00b	Peripheral device addressing method	5.2.1.2.3
01b	Volume set addressing method	5.2.1.2.4
11b	Reserved	

5.2.1.2.2 Logical unit address method

All SCSI commands are allowed when using the logical unit address method is selected, however logical units and SCSI storage arrays are only required to support mandatory SCSI commands. SCSI 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 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 NUMBER			LUN				

The TARGET field, BUS NUMBER field, and LUN field address the logical unit to which the received command shall be relayed. The command shall be relayed to the logical unit (LUN field value) within the target

(TARGET field value) located on the SCSI bus (BUS NUMBER field value).

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).

5.2.1.2.3 Peripheral device address method

All SCSI commands are allowed when the peripheral address method is selected, however peripheral devices and SCSI storage arrays are only required to support mandatory SCSI commands. SCSI 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 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 IDENTIFIER					
n	TARGET/LUN							

The BUS IDENTIFIER field identifies the bus or path that the SCSI storage array shall use to relay the received command. The BUS IDENTIFIER field may use the same value encoding as the BUS NUMBER field (see 5.2.1.2.2). However, bus identifier zero shall indicate that the command is to be relayed to a logical unit within the current level of the SCSI storage array.

The TARGET/LUN field indicates the address of the peripheral device to which the SCSI storage array shall relay the received command. If the BUS IDENTIFIER field is not zero the TARGET/LUN field indicates the address of the target on the bus indicated by the BUS IDENTIFIER field to which the received command shall be relayed. The received command to shall be relayed to LUN zero.

A BUS IDENTIFIER field of zero represents a logical interconnection of SACL logical units. This representation of the SACLs logical units may be used for logical units when the SACL either does not use hierarchical addressing when assigning LUNs to objects or the SCAL has objects that need LUNs and are not attached to actual buses (e.g, fans, cache, controllers, etc.).

A BUS IDENTIFIER field greater than zero represent physical interconnects that connect a group of SCSI devices. Each of the buses shall be assigned a number from 1 to n by the SCAL. The bus identifiers shall be used in the BUS IDENTIFIER field by the SCAL when assigning addresses to peripheral devices attached to those buses.

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).

The SCSI storage array device located within the current level shall be addressed by a BUS IDENTIFIER field and a TARGET/LUN field of all zeros, also known as LUN_Z (see 5.2.1.1).

5.2.1.2.4 Volume set address method

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

NOTE 5 - The volume set might not be under the control of the addressed SCSI storage array. It is allowed to be in an SCSI 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 peripheral device type.(e.g., direct access device, streaming device, etc.)

When the volume set addressing method is selected the SCSI 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_V or LUN							(LSB)

The LUN_V or LUN field indicates the address of the volume set to which the SCSI storage array shall direct the received command.

5.2.1.3 Indirect addressing

Indirect addresses appear only in command descriptor blocks or parameter lists. Directly addressed devices may also be addressed indirectly. However, component devices, redundancy groups, and spares may be addressed only indirectly.

5.2.1.3.1 Component device address method

An application client shall only address a component device by including the LUN_C address in the command information sent to the LUN_Z (see 5.2.1.1) of the SCSI storage array that controls or will control the component device.

The format of a LUN_C or LUN field within a command descriptor block or a parameter list when addressing a component device is defined in table 9.

Table 9 - Component device address

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

5.2.1.3.2 Logical unit address method

An application client indirectly addresses a logical unit by including the logical unit's address in a command information sent to the LUN_Z (see 5.2.1.1) of the SCSI storage array.

The format of a LUN_P or LUN field within a command descriptor block or a parameter list when addressing a logical unit is defined in table 6.

5.2.1.3.3 Peripheral device address method

An application client indirectly addresses a peripheral device by including the peripheral device's address in the command information sent to the LUN_Z (see 5.2.1.1) of the SCSI storage array.

The format of a LUN_P or LUN field within a command descriptor block or a parameter list when addressing a peripheral device is defined in table 7.

5.2.1.3.4 Redundancy group address method

An application client shall only address a redundancy group by including the LUN_R address in the command information sent to the LUN_Z (see 5.2.1.1) of the SCSI storage array that controls or will control the redundancy group.

The format of a LUN_R or LUN field within a command descriptor block or a parameter list when addressing a redundancy group is defined in table 10.

Table 10 - Redundancy group address

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

5.2.1.3.5 Spare address method

An application client shall only address a spare by including the LUN_S address in the command information sent to the LUN_Z (see 5.2.1.1) of the SCSI storage array that controls or will control the spare.

The format of a LUN_S or LUN field within a command descriptor block or a parameter list when addressing a spare is defined in table 11.

Table 11 - Spare address

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

5.2.1.3.6 Volume set address method

An application client indirectly addresses a volume set by including the volumes set's address in the command information sent to the LUN_Z (see 5.2.1.1) of the SCSI storage array.

The format of a LUN_V or LUN field within a command descriptor block or a parameter list when addressing a volume set is defined in table 8.

5.2.2 SCSI storage array objects

An SCSI 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 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 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 storage array. The ADD PERIPHERAL DEVICE/COMPONENT DEVICE service action (see 6.4.1.1) 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 storage arrays. The CREATE/MODIFY REDUNDANCY GROUP service action (see 6.6.1.3) and CREATE/MODIFY VOLUME SET service action (see 6.8.1.6) create associations and the delete service actions (see 6.6.1.4 and 6.8.1.8) remove associations. The rules for objects that are associated are explicitly defined throughout this clause and within the clauses describing the create/modify service actions.

An SCSI 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 (6.4.1.2) 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 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

An object or group of objects may be covered by another object as listed below:

a) a component device or redundancy group may be covered by a component device,
b) a p_extent or redundancy group may be covered be a p_extent, or
c) a peripheral device or redundancy group may be covered by a peripheral device.

The service actions of the SPARE(OUT) command (see 6.10.1) establish 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 (e.g., p_extents, ps_extents, protected space areas, check data areas, etc.) 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 storage array without any spare requests being issued from an application client. The SCSI 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 functionally 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

The exchange service actions (see 6.4.1.4 and 6.4.1.5) are used to exchange objects. 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. When an old peripheral device,

p_extent, or component device is exchanged, the replacement 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 replacement device. The characteristics of the old peripheral device are vendor specific.

A p_extent shall only be exchanged with a p_extent of equal size. An old peripheral device shall only be exchanged with a replacement peripheral device when the replacement peripheral device is of equal or greater size than the old peripheral device.

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 storage array availability.

5.2.2.7 Removing objects

Objects that have been removed from a SCSI storage array shall not be addressable by an application client, until the object is re-added or recreated. The remove service actions (see 6.4.1.7, 6.6.1.4 and 6.10.1.3) 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 6.4.1.7, 6.6.1.4, and 6.10.1.3.

When the object is removed all attachments, associations, and coverings 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 3.1.28 for the definition of peripheral devices.

5.2.2.10 P_extent

See 3.1.27 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 both 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 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 6.3.1.1) 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 3.1.30 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 (6.6.1.3.) 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 shall be unassigned ps_extents if the addressable logical block is:

a) not defined as containing check data,
b) within an operating peripheral device connected within a SCSI storage array, and
c) has not been configured into a volume set.

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 6.5.1.3) 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 3.1.33 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.

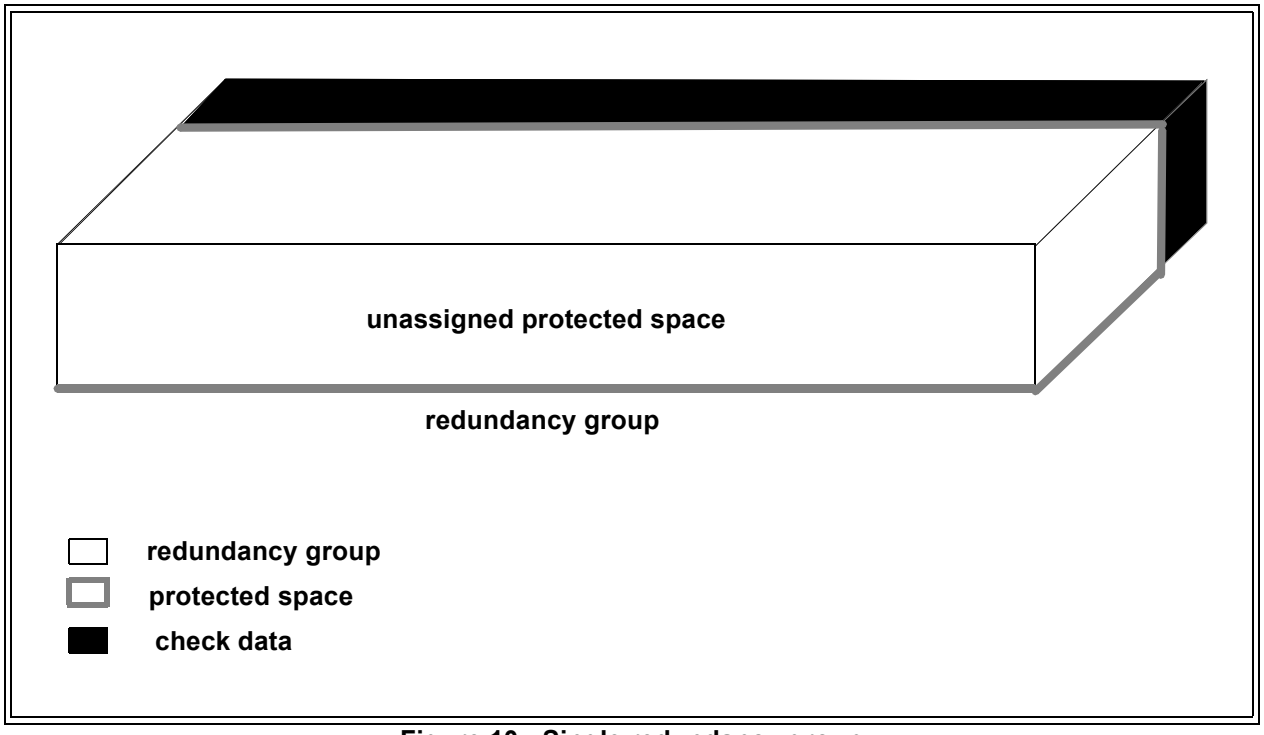

Figure 10 - Single redundancy group

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

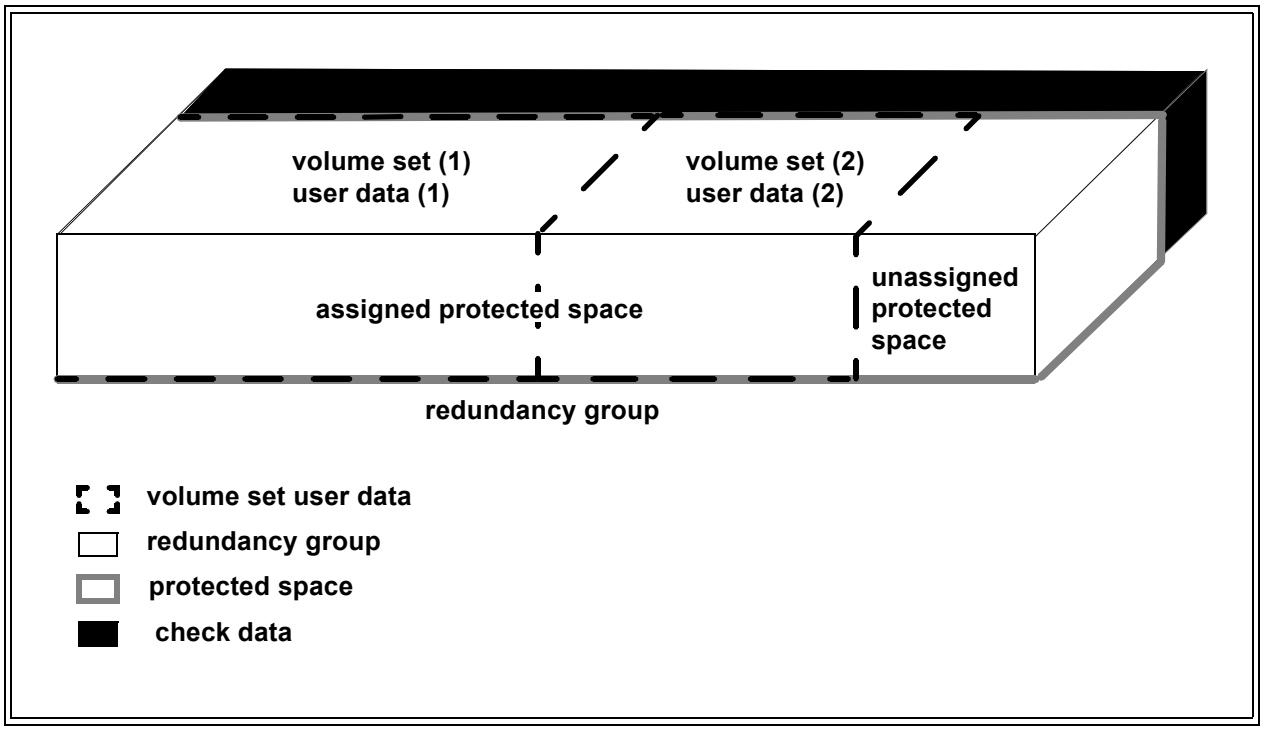

Figure 11 - Multiple volume sets associated with a single redundancy group

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 number of consecutive units to skip from the beginning of the p_extent before the first unit of check data is mapped 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 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.

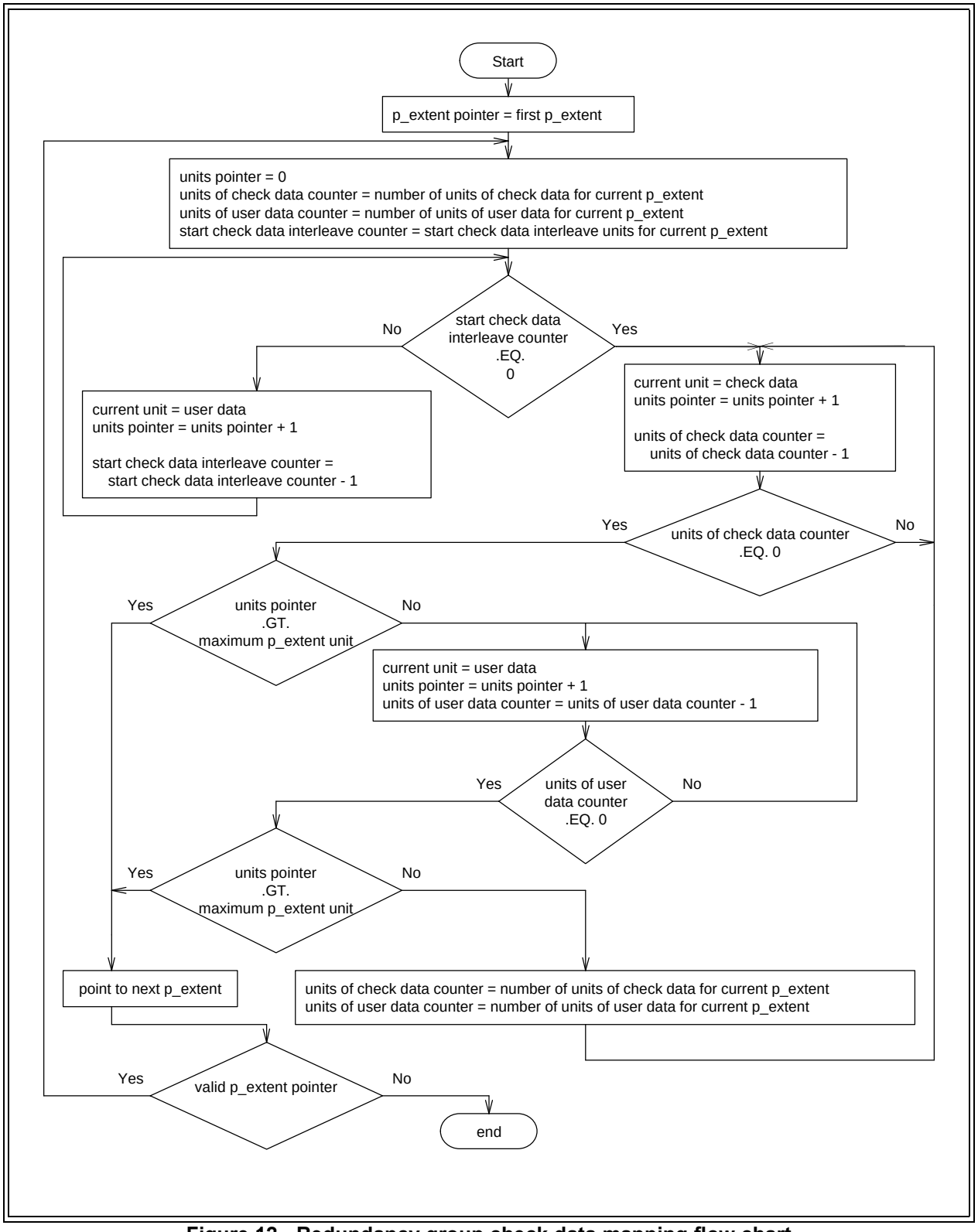

Figure 12 - Redundancy group check data mapping flow chart

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) S redundancy;
f) P+S redundancy;
g) Vendor specific redundancy.

5.2.2.12.1 No redundancy method of check data mapping

In a redundancy group in which 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 6.6.1.3) 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 in which 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 (6.6.1.3). If this method is requested then the following fields in the CREATE/MODIFY REDUNDANCY GROUP service action (see 6.6.1.3) 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 in which 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 6.6.1.3) 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 S redundancy method of check data mapping

In a redundancy group in which the application client requests S redundancy the user data is protected by spares located within the check data areas. In this redundancy method check data areas contain no information until the user data area covered by the check data area fails, at which time the covered user data is automatically exchanged (see 5.2.2.5) into the check data area. 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 6.6.1.3) 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.

5.2.2.12.5 P+S redundancy method of check data mapping

The P+S redundancy method is a combination of the XOR redundancy method and the S redundancy method. This combination provides the user data with protection from loss through the XOR check data areas and the ability to rebuild lost user data and protected space into the spare check data areas. After the rebuild is complete the user data is again fully protected from loss through the XOR 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 6.6.1.3) 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.

In the P+S redundancy method SACL uses the NUMBER OF UNITS OF CHECK DATA field twice when mapping a p_extent. As a result the order of usage of the NUMBER OF UNITS OF CHECK DATA field and the NUMBER OF UNITS OF USER DATA field within a p_extent shall be either:

a) START CHECK DATA INTERLEAVE UNIT;
b) NUMBER OF UNITS OF CHECK DATA (XOR);
c) NUMBER OF UNITS OF CHECK DATA (spare);
d) NUMBER OF UNITS OF USER DATA;
e) repeat from b until end of p_extent.

or:

a) NUMBER OF UNITS OF CHECK DATA (spare);
b) NUMBER OF UNITS OF USER DATA;
c) NUMBER OF UNITS OF CHECK DATA (XOR);
d) NUMBER OF UNITS OF CHECK DATA (spare);
e) NUMBER OF UNITS OF USER DATA;
f) repeat from c until end of p_extent.

See the CREATE/MODIFY REDUNDANCY GROUP service action (6.6.1.3) for more information on the selection of the P+S redundancy method mapping.

5.2.2.12.6 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. The usage of the GRANULARITY OF UNITS, START CHECK DATA INTERLEAVE UNIT, NUMBER OF UNITS OF CHECK DATA, and NUMBER OF UNITS OF USER DATA fields is vendor specific, but should follow the concepts defined in this standard.

5.2.2.13 Spares

See 3.1.40 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). A spare may be configured to cover multiple redundancy groups, peripheral devices, or 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 configured to 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 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 3.1.50 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 6.5.1.3) 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.

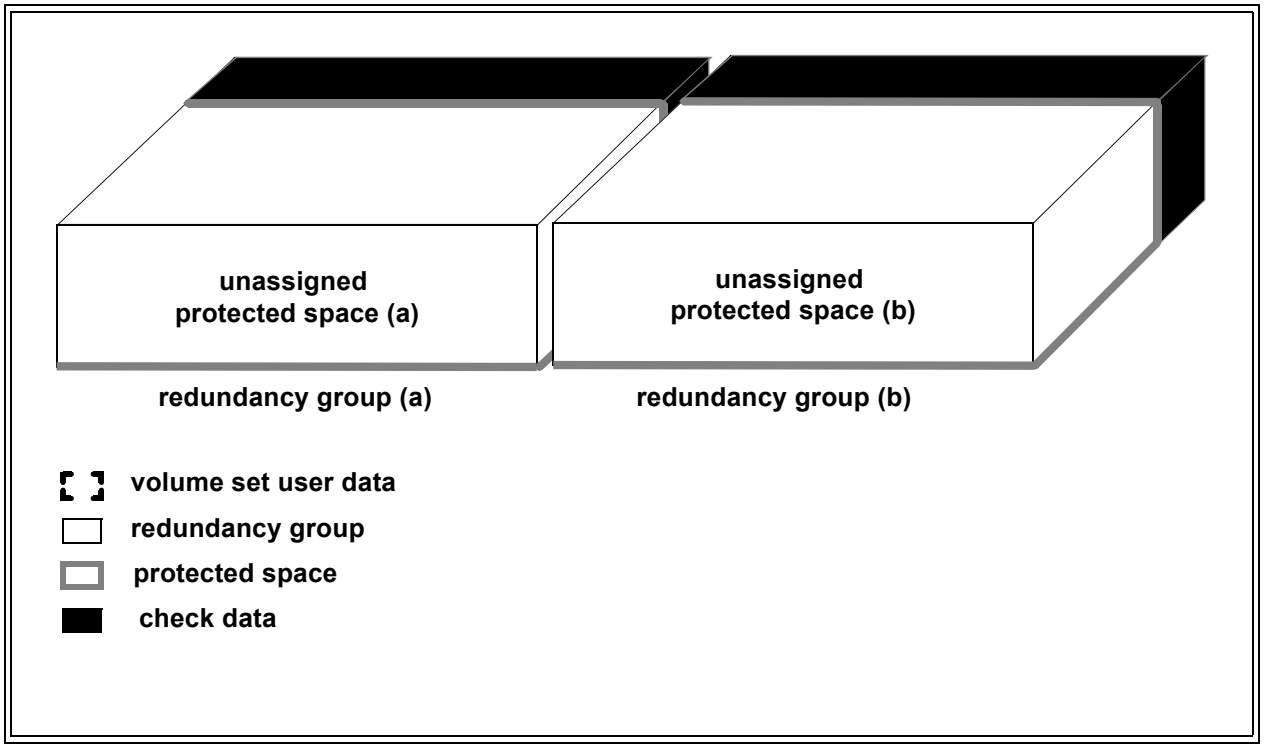

Figure 13 - Multiple redundancy groups

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.

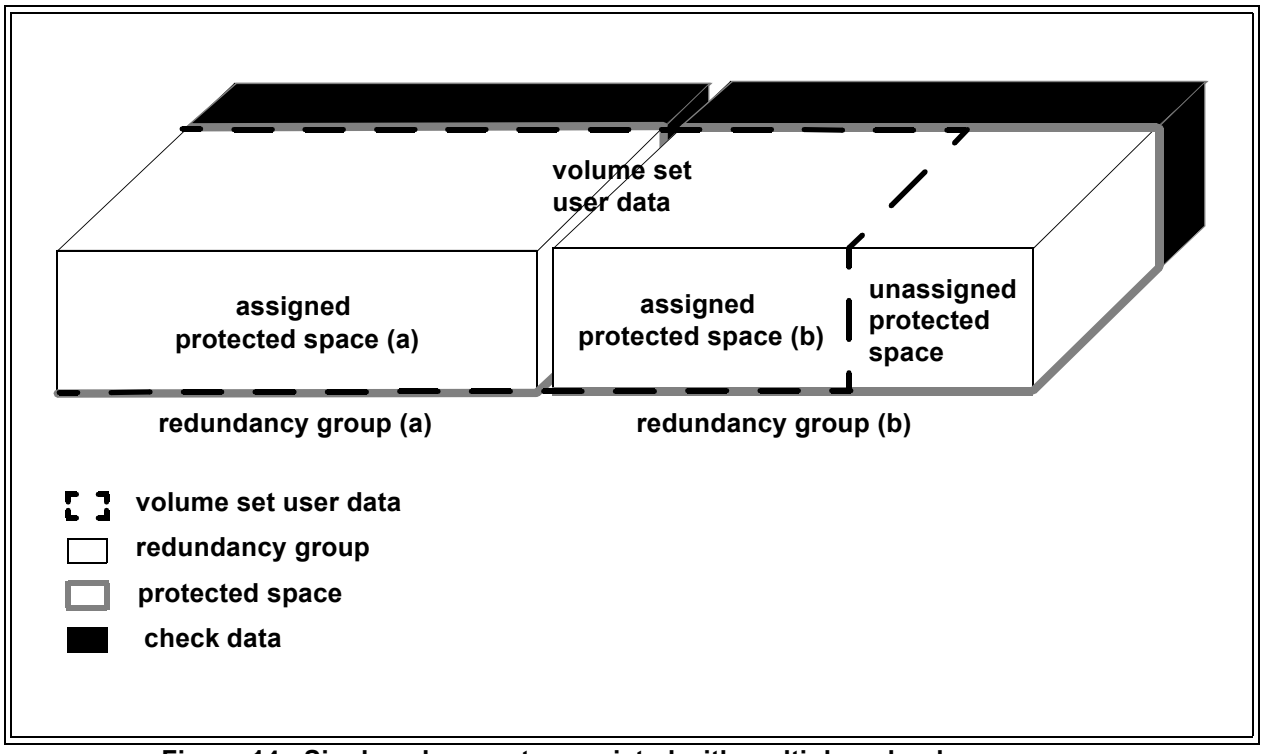

Figure 14 - Single volume set associated with multiple redundancy groups

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. The units are as defined by the GRANULARITY OF UNITS field used to create the 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 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 17 shows the most general implementation of the three parameters used to map the user data.

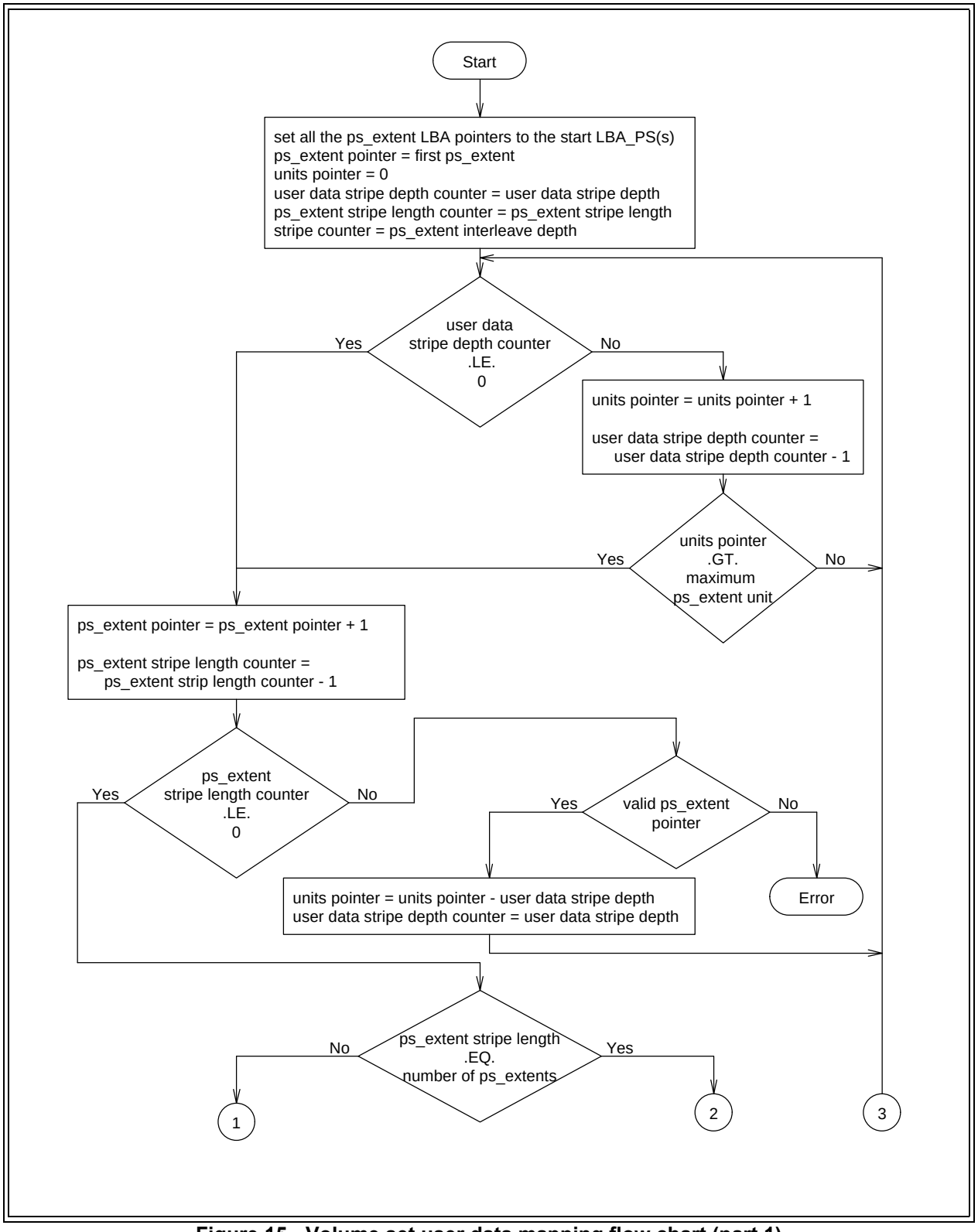

Figure 15 - Volume set user data mapping flow chart (part 1)

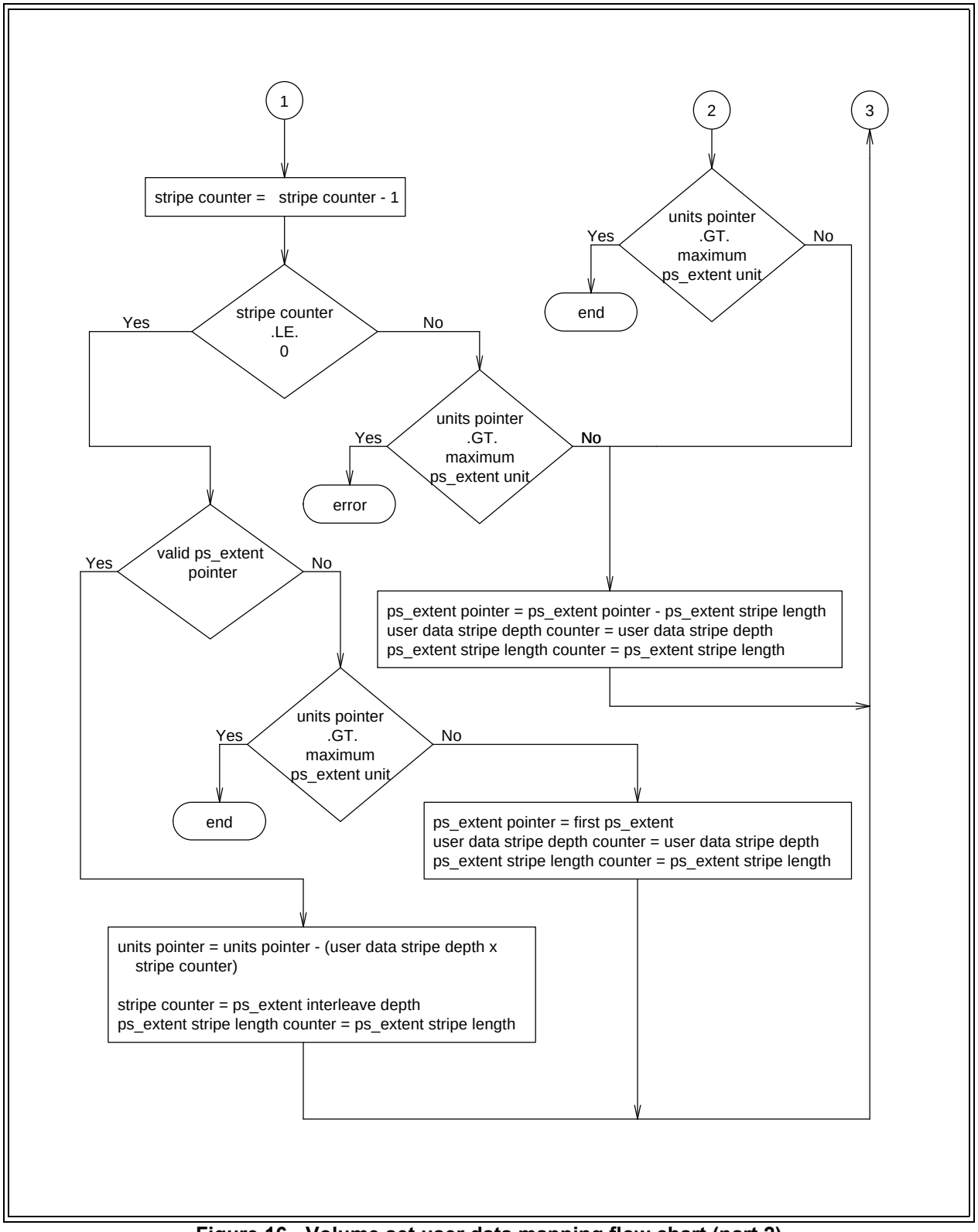

Figure 16 - Volume set user data mapping flow chart (part 2)

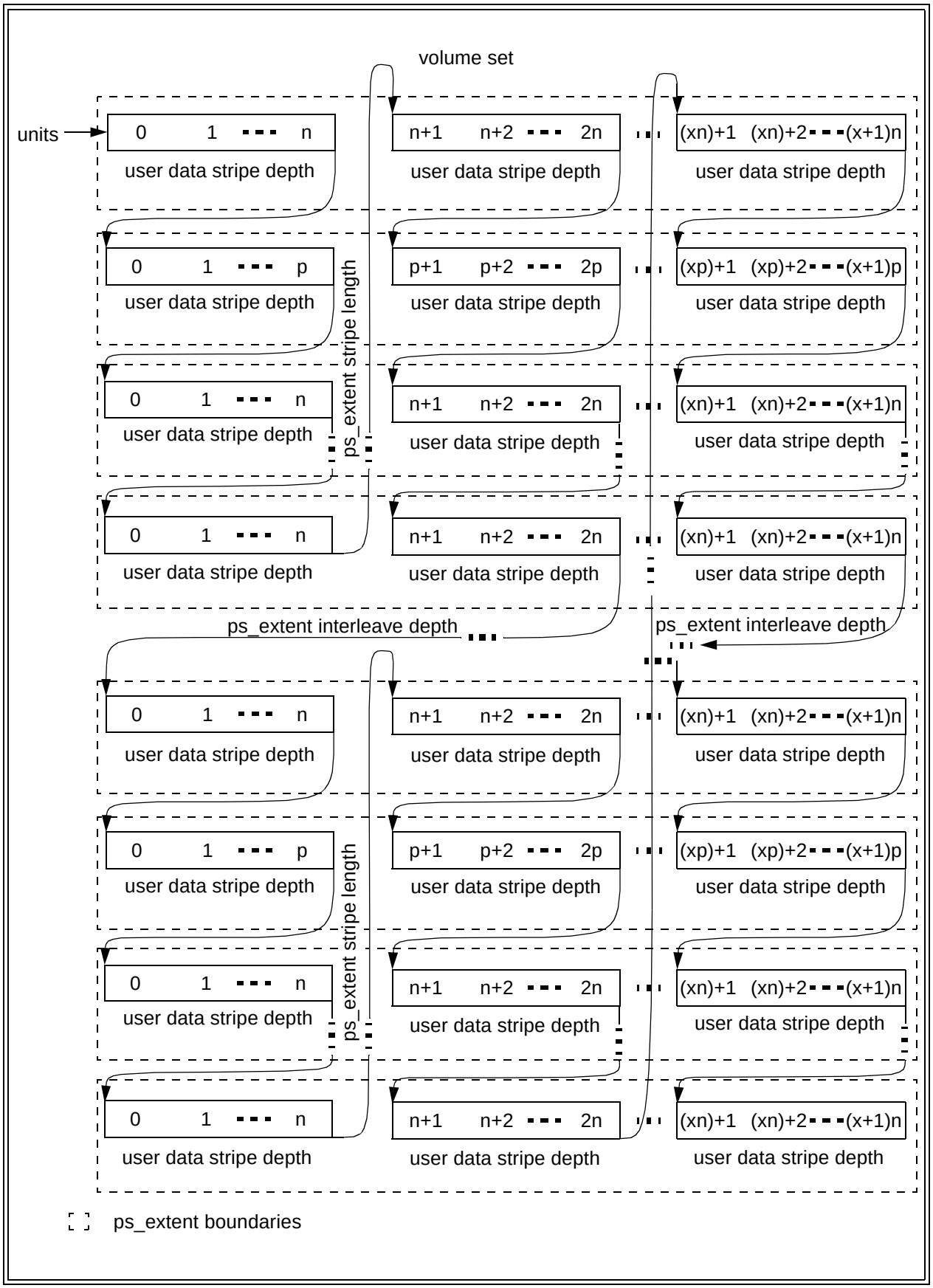

Figure 17 - Volume set user data mapping flow

5.2.3 SCSI storage array operations

5.2.3.1 Deassign LUN_V operation

The deassign LUN_V operation removes the LUN_V value from a configured volume set. The SACL shall identify all volume sets that are allowed to be deassigned using an identification as defined in the vital product data device identification page (83h) (see SCSI-3 Primary Commands Standard).

After a successful deassign LUN_V operation the SACL is required to maintain all the configuration characteristics of the deassigned volume set. The SACL is also required to treat any attempt to address a volume set using the deassigned LUN_V as an attempt to address an unconfigured LUN_V. The application client or the SACL is free to use the deassigned LUN_V in the configuration of a new volume set.

After a volume set is deassigned, using the DEASSIGN LUN_V service action (6.8.1.7), the application client uses the REPORT UNASSIGNED VOLUME SETS service action (6.7.1.3) to determine the identification of any deassigned volume set(s). The application client may use the identification to assign a LUN_V to a specific deassigned volume set using the ASSIGN LUN_V service action (6.8.1.1).

5.2.3.2 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 rebuilt protected space contents or any rebuilt check data shall be written to the p_extent being rebuilt.

5.2.3.3 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.

5.2.3.4 Regenerate operation

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

5.2.3.5 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 6.8.1.10 and 6.6.1.8 for exception handling information.

5.2.4 SCSI storage array states

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

The state of a SCSI storage array (LUN_Z), 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 (6.3.1.8).

The REPORT STATES service action reports all available state information for the selected logical unit(s) (e.g., If a selected 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 other report service actions where only one state shall be returned per report service action. The priority of reporting among multiple

states is vendor specific.

For all state changes the target shall generate a unit attention condition for all initiators, except when the state change occurred due to a service action, in which case the unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to STATE CHANGE HAS OCCURRED.

5.2.5 SCSI storage array configuration options

A SCSI storage array may be implemented to manage configurations using one or more of the following methods. It is only required that a SCSI storage array implement one of the configuration methods. The service actions listed below only represent the minimum service actions that are required to implement the various configuration methods.

5.2.5.1 Simple configuration method

The simple configuration method requires the least amount of service actions to configure the storage array. The service actions parameters provide an application client with a minimum amount of control over the redundancy group space mapping and the volume set user data mapping.

The service actions that are required if the simple configuration method is supported are:

a) REPORT STATES (6.3.1.8),
b) REPORT SUPPORTED CONFIGURATION METHOD (6.3.1.9),
c) REPORT UNCONFIGURED CAPACITY (6.3.1.10),
d) REPORT STORAGE ARRAY CONFIGURATION (6.7.1.2), and
e) CREATE/MODIFY STORAGE ARRAY CONFIGURATION (6.8.1.4).

There are no other service actions needed for controlling configurations when the simple configuration method is being used. When available, the application client may choose to use the general configuration method's spare configuration service actions, instead of the equivalent simple configuration method service actions.

The above service actions provide the application client control over the following configuration properties:

a) type of protection,
b) amount of user data spreading,
c) volume set's logical unit number,
d) volume set's capacity,
e) rebuild and recalculate priority, and
f) data handling characteristics (i.e., normal user data transfer size and sequentially of reads and writes).

5.2.5.2 Basic configuration method

The basic configuration method requires an application client to separately configure redundancy groups and volume sets where as the simple configuration method combines those operations into one service action. The service actions have parameters that provide an application client with a minimum amount of control over the volume set user data mapping and the redundancy group mapping. The service actions that are required if the basic configuration method is supported are:

a) REPORT STATES (6.3.1.8),
b) REPORT SUPPORTED CONFIGURATION METHOD (6.3.1.9),
c) REPORT UNCONFIGURED CAPACITY (6.3.1.10),
d) REPORT BASIC REDUNDANCY GROUP (6.5.1.1),
e) REPORT BASIC VOLUME SET (6.7.1.1),

f) CREAT/MODIFY BASIC REDUNDANCY GROUP (6.6.1.2), and
g) CREATE/MODIFY VOLUME SET (6.8.1.4).

There are no other service action needed for controlling volume set configurations when the basic configuration method is being used. When available, the application client may choose to use the general configuration method's redundancy group configuration and/or spare configuration service actions, instead of the equivalent basic configuration method service actions.

The above service actions provide the application client control over the following configuration properties:

a) type of protection,
b) amount of check data spreading,
c) redundancy group's logical unit number,
d) size of the redundancy group,
e) amount of user data spreading,
f) volume set's logical unit number,
g) volume set's capacity, and
h) list of redundancy groups to associate with volume set.

5.2.5.3 General configuration method

The general configuration method gives an application client control over all the details of the redundancy group, volume set, and spare configuration. But the number of service actions and parameters is large and the application client must have a detailed understanding of the SCSI storage array to configure it using the general configuration method.

The service actions that are required if the general configuration method is supported are:

a) REPORT STATES (6.3.1.8),
b) REPORT SUPPORTED CONFIGURATION METHOD (6.3.1.9),
c) REPORT ASSIGNED/UNASSIGNED P_EXTENT (6.3.1.1),
d) REPORT COMPONENT DEVICE (6.3.1.2),
e) REPORT COMPONENT DEVICE ATTACHMENTS (6.3.1.3),
f) REPORT PERIPHERAL DEVICE (6.3.1.5),
g) REPORT PERIPHERAL DEVICE ASSOCIATIONS (6.3.1.6),
h) REPORT UNASSIGNED REDUNDANCY GROUP SPACE (6.5.1.3),
i) REPORT REDUNDANCY GROUPS (6.5.1.2),
j) REPORT VOLUME SETS (6.7.1.4),
k) CREATE/MODIFY REDUNDANCY GROUP (6.6.1.3), and
l) CREATE/MODIFY VOLUME SET (6.8.1.6).

At a minimum, the following additional service actions are recommended if the general configuration method is supported:

a) REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE (6.9.1.2), and
b) CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT SPARE (6.10.1.2).

The above service actions provide the application client control over the following configuration properties:

a) type of protection,
b) granularity of units,
c) redundancy group's logical unit number,
d) size of each p_extent,
e) location of each p_extent,
f) number of p_extents,
g) start location of check data in each p_extent,

h) check data stripe size in each p_extent,
i) user data stripe size in each p_extent,
j) volume set's logical unit number,
k) user data's ps_extent stripe length,
l) user data's ps_extent interleave depth,
m) size of each ps_extent,
n) location of each ps_extent,
o) number of ps_extents,
p) list of redundancy group to associate with each ps_extent,
q) user data stripe depth in each ps_extent,
r) spare's logical unit number,
s) p_extent, peripheral device, or component device to be make into a spare, and
t) list of objects to be covered by the spare (e.g., redundancy groups, p_extents, component devices, etc.).

5.2.6 SCSI 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 storage array indicate a task failed. Some are informational, in that, they provide the application client information on the condition of the SCSI storage array. (e.g., the SCSI storage array is predicting a failure of a logical unit, something is broken but the SCSI 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 storage array devices

6.1 Op codes for SCSI storage array commands

The operation codes for commands that apply only to SCSI storage array devices are listed in table 12.

Table 12 - 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	6.3
MAINTENANCE (OUT)	A4h	O	6.4
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	B,G	6.5
REDUNDANCY GROUP (OUT)	BBh	BC,GC	6.6
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	O	6.9
SPARE (OUT)	BDh	O	6.10
START STOP UNIT	1Bh	O	SPC
TEST UNIT READY	00h	M	SPC
VOLUME SET (IN)	BEh	M	6.7
VOLUME SET (OUT)	BFh	M	6.8
WRITE BUFFER	3Bh	O	SPC

Key: B = Command implementation is mandatory if basic configuration method is implemented.
BC = Command implementation is mandatory if basic configuration method is implemented with configuration support.
G = Command implementation is mandatory if general configuration method is implemented.
GC = Command implementation is mandatory if general configuration method is implemented with configuration support.
M = Command implementation is mandatory.
O = Command implementation is optional.
S = Command implementation is mandatory if simple configuration method is implemented.
SPC = SCSI-3 Primary Commands Standard

6.2 Glossary of SCSI storage array service actions

6.2.1 MAINTENANCE (IN) command service actions

6.2.1.1 REPORT ASSIGNED/UNASSIGNED P_EXTENT (6.3.1.1): Reports a list of assigned or unassigned p_extent space(s).

6.2.1.2 REPORT COMPONENT DEVICE (6.3.1.2): Reports a list of components. For each component in the list the following information is reported: the logical unit number, the type of the component, and the state of the component.

6.2.1.3 REPORT COMPONENT DEVICE ATTACHMENTS (6.3.1.3): Reports a list of objects attached to the selected component(s).

6.2.1.4 REPORT DEVICE IDENTIFICATION (6.3.1.4): Reports the vital product data's device identification.

6.2.1.5 REPORT PERIPHERAL DEVICE (6.3.1.5): Reports a list of peripheral devices. For each peripheral device in the list the following information is reported: the logical unit number, the type of the peripheral device, and the state of the peripheral device.

6.2.1.6 REPORT PERIPHERAL DEVICE ASSOCIATIONS (6.3.1.6): Reports a list of objects associated with the selected peripheral device(s).

6.2.1.7 REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER (6.3.1.7): Reports an ASCII string that represents information about the device (e.g., physical location, etc.).

6.2.1.8 REPORT STATES (6.3.1.8): Reports the current state of the selected logical unit(s).

6.2.1.9 REPORT SUPPORTED CONFIGURATION METHOD(6.3.1.9): Reports the configuration methods supported by the target.

6.2.1.10 REPORT UNCONFIGURED CAPACITY (6.3.1.10): Reports the free p_extent and ps_extent space, in blocks, available for configuration.

6.2.1.11 MAINTENANCE (OUT) command service actions

6.2.1.12 ADD PERIPHERAL DEVICE/COMPONENT DEVICE (6.4.1.1): Makes a device available for configuration.

6.2.1.13 ATTACH TO COMPONENT DEVICE (6.4.1.2) : Creates an attachment between a component device and one or more objects.

6.2.1.14 BREAK PERIPHERAL DEVICE/COMPONENT DEVICE (6.4.1.3): Place a device into the broken state.

6.2.1.15 EXCHANGE P_EXTENT (6.4.1.4): Replace one p_extent with another p_extent.

6.2.1.16 EXCHANGE PERIPHERAL DEVICE/COMPONENT DEVICE (6.4.1.5): Replace one device with another device.

6.2.1.17 INSTRUCT COMPONENT DEVICE (6.4.1.6): Sends a command directly to a component device.

6.2.1.18 REMOVE PERIPHERAL DEVICE/COMPONENT DEVICE (6.4.1.7): Makes a device unavailable for configuration.

6.2.1.19 SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER (6.4.1.8): Assigns an ASCII string to a device.

6.2.1.20 REDUNDANCY GROUP (IN) command service actions

6.2.1.21 REPORT BASIC REDUNDANCY GROUP (6.5.1.1): Reports information about a redundancy group using the basic configuration method.

6.2.1.22 REPORT REDUNDANCY GROUPS (6.5.1.2): Reports information about a redundancy group using the general configuration method.

6.2.1.23 REPORT UNASSIGNED REDUNDANCY GROUP SPACE (6.5.1.3): Report the amount of redundancy group free space in ps_extents.

6.2.1.24 REDUNDANCY GROUP (OUT) command service actions

6.2.1.25 CONTROL GENERATION OF CHECK DATA (6.6.1.1): Turns on/off the protection within a redundancy group.

6.2.1.26 CREATE/MODIFY BASIC REDUNDANCY GROUP (6.6.1.2): Create or modify a redundancy group using the basic configuration method.

6.2.1.27 CREATE/MODIFY REDUNDANCY GROUP (6.6.1.3): Create or modify a redundancy group using the general configuration method.

6.2.1.28 DELETE REDUNDANCY GROUP (6.6.1.4): Remove the redundancy group from the configuration.

6.2.1.29 REBUILD P_EXTENT (6.6.1.5): Rebuild a p_extent using information contained within a redundancy group.

6.2.1.30 REBUILD PERIPHERAL DEVICE (6.6.1.6): Rebuild a peripheral device using information contained within a redundancy group.

6.2.1.31 RECALCULATE CHECK DATA (6.6.1.7): Recalculate the check data within the redundancy group.

6.2.1.32 VERIFY CHECK DATA (6.6.1.8): Verify the protected space and check data is consistent within a redundancy group.

6.2.1.33 SPARE (IN) command service actions

6.2.1.34 REPORT P_EXTENT SPARE (6.9.1.1): Report information on p_extent spares.

6.2.1.35 REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE (6.9.1.2): Report information on device spaces.

6.2.1.36 SPARE (OUT) command service actions

6.2.1.37 CREATE/MODIFY P_EXTENT SPARE (6.10.1.1): Create or modify a p_extent spare.

6.2.1.38 CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE (6.10.1.2): Create or modify a device spare.

6.2.1.39 DELETE SPARE (6.10.1.3): Remove a spare from the configuration.

6.2.1.40 VOLUME SET (IN) command service actions

6.2.1.41 REPORT BASIC VOLUME SET (6.7.1.1): Reports information about a volume set using the basic configuration format.

6.2.1.42 REPORT STORAGE ARRAY CONFIGURATION (6.7.1.2): Reports information about a volume set using the simple configuration method.

6.2.1.43 REPORT UNASSIGNED VOLUME SETS (6.7.1.3): Reports configurated volume sets that do not have a LUN_V assigned.

6.2.1.44 REPORT VOLUME SETS (6.7.1.4): Reports information about a volume set using the general configuration method.

6.2.1.45 VOLUME SET (OUT) command service actions

6.2.1.46 ASSIGN LUN_V (6.8.1.1): Assigns a LUN_V to a volume set that is already configured.

6.2.1.47 CONTROL GENERATION OF CHECK DATA (6.8.1.2): Turns on/off the protection of any redundancy group(s) associated with a volume set.

6.2.1.48 CONTROL WRITE OPERATIONS (6.8.1.3): Enable/disable write operations to a volume set.

6.2.1.49 CREATE/MODIFY BASIC VOLUME SET (6.8.1.4): Create or modify a volume set using the basic configuration method.

6.2.1.50 CREATE/MODIFY STORAGE ARRAY CONFIGURATION (6.8.1.5): Create or modify a volume set using the simple configuration method.

6.2.1.51 CREATE/MODIFY VOLUME SET (6.8.1.6): Create or modify a volume set using the general configuration method.

6.2.1.52 DEASSIGN LUN_V (6.8.1.7): Remove the LUN_V from a volume set.

6.2.1.53 DELETE VOLUME SET (6.8.1.8): Remove the volume set from the configuration.

6.2.1.54 RECALCULATE VOLUME SET CHECK DATA (6.8.1.9): Recalculate the check data of any redundancy group(s) associated with a volume set.

6.2.1.55 VERIFIY VOLUME SET CHECK DATA (6.8.1.10): Verify the consistency of the protected space and check data of any redundancy group(s) associated with a volume set.

6.3 MAINTENANCE(IN) command

6.3.1 MAINTENANCE(IN) command service actions

The service actions for the MAINTENANCE(IN) command are listed in table 13.

Table 13 - Service actions for MAINTENANCE(IN) command

Service name	Service action	Type	Subclause
REPORT ASSIGNED/UNASSIGNED P_EXTENT	00h	G	6.3.1.1
REPORT COMPONENT DEVICE	01h	G	6.3.1.2
REPORT COMPONENT DEVICE ATTACHMENTS	02h	G	6.3.1.3
REPORT DEVICE IDENTIFICATION	07h	O	6.3.1.4
REPORT PERIPHERAL DEVICE	03h	G	6.3.1.5
REPORT PERIPHERAL DEVICE ASSOCIATIONS	04h	G	6.3.1.6
REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER	05h	O	6.3.1.7
REPORT STATES	06h	M	6.3.1.8
REPORT SUPPORTED CONFIGURATION METHOD	09h	M	6.3.1.9
REPORT UNCONFIGURED CAPACITY	08H	B,S	6.3.1.10
Reserved	0Ah-17h		
Vendor specific	18h-1Fh		

Key: B = Service action implementation is mandatory if basic configuration method is implemented.
G = Service action implementation is mandatory if general configuration method is implemented.
M = Service action implementation is mandatory.
O = Service action implementation is optional.
S = Service action implementation is mandatory if simple configuration method is implemented.

6.3.1.1 REPORT ASSIGNED/UNASSIGNED P_EXTENT service action

The REPORT ASSIGNED/UNASSIGNED P_EXTENT service action (see table 14) 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 14 - 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							

If the RPTSEL bit is one, the LUN_P field specifies the address of the peripheral device from which the p_extent(s) shall be reported per table 15. 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 15) 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 15 - 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 16.

Table 16 - 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 17.

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 44.

Table 17 - 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.

6.3.1.2 REPORT COMPONENT DEVICE service action

The REPORT COMPONENT DEVICE service action (see table 18) requests that information regarding component device(s) within the target be sent to the application client.

Table 18 - 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							

If the RPTSEL bit is one, the LUN_C field specifies the address of the component device that shall be reported per table 19. 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 19) contains a four-byte header that contains the length in bytes of the parameter list and a list of COMPONENT DEVICE DESCRIPTORS.

Table 19 - 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 20.

Table 20 - 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 21 for a list of component device types.

The component device types table (see table 21) contains a list of non-data type devices the target may address as component devices.

TABLE 21 - 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 46.

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

6.3.1.3 REPORT COMPONENT DEVICE ATTACHMENTS service

The REPORT COMPONENT DEVICE ATTACHMENTS service action (see table 22) requests that information regarding logical units that are attached to component device(s) be sent to the application client.

Table 22 - 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)							
7				ALLOCATION LENGTH				
8								
9								(LSB)
10	RESERVED							RPTSEL
11	CONTROL							

If the RPTSEL bit is one, the LUN_C field specifies the address of the component device for which the target shall report information per table 23. 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 23) contains a four-byte header that contains the length in bytes of the parameter list and a list of COMPONENT DEVICE ATTACHMENT DESCRIPTORs.

Table 23 - REPORT COMPONENT DEVICE ATTACHMENTS parameter list

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

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 24.

Table 24 - Format of COMPONENT DEVICE ATTACHMENT DESCRIPTOR

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

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 25 for the format of the LOGICAL UNIT DESCRIPTOR.

Table 25 - 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)							
3				LUN				(LSB)

The LOGICAL UNIT TYPE field (see table 26) indicates the type of logical unit addressed in the LUN field.

Table 26 - 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	LUN_Z
8h-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.

6.3.1.4 REPORT DEVICE IDENTIFICATION service action

The REPORT DEVICE IDENTIFICATION service action

The REPORT DEVICE IDENTIFICATION service action (see table 37) requests the vital product data's device identification page (83h) (see SCSI-3 Primary Commands Standard) for the selected logical unit be sent to the application client.

Table 27 - REPORT DEVICE IDENTIFICATION service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	RESERVED			SERVICE ACTION (07h)				
2	RESERVED							
3	RESERVED				LOGICAL UNIT TYPE			
4	(MSB)	LUN						
5								(LSB)
6	(MSB)	ALLOCATION LENGTH						
7								
8								
9								(LSB)
10	RESERVED							
11	CONTROL							

The LOGICAL UNIT TYPE field contains the type of logical unit designated by the LUN field. See table 26 for the list of logical unit types. If the requested logical unit unknown 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 the device identification as defined in the vital product data's device identification page (see SCSI-3 Primary Commands Standard).

The parameter list shall be identical to the vital product data's device identification page as defined in the SCSI-3 Primary Commands Standard.

6.3.1.5 REPORT PERIPHERAL DEVICE service action

The REPORT PERIPHERAL DEVICE service action (see table 28) requests that information regarding peripheral device(s) within the target be sent to the application client.

Table 28 - 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)			ALLOCATION LENGTH				
7								
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 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.

The SELECT REPORT field contains the information on which peripheral device(s) the target shall report. See table 29 for the defined states.

TABLE 29 - 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 30) contains a four-byte header that contains the length in bytes of the parameter list and a list of PERIPHERAL DEVICE DESCRIPTORS.

Table 30 - 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 31.

Table 31 - 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 44.

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

6.3.1.6 REPORT PERIPHERAL DEVICE ASSOCIATIONS service action

The REPORT PERIPHERAL DEVICE ASSOCIATIONS service action (see table 32) requests that information regarding logical units that are associated with peripheral device(s) be sent to the application client.

Table 32 - 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							

If the RPTSEL bit is one, 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 34. 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 34) shall be reported for each object associated with the peripheral device listed in the LUN_P field (see table 34). A RPTMBUS bit of one indicates all LOGICAL UNIT DESCRIPTOR(s) (see table 34) shall be reported for each object associated with the peripheral device listed in the LUN_P field (see table 34).

The REPORT PERIPHERAL DEVICE ASSOCIATIONS parameter list (see table 33) contains a four-byte header that contains the length in bytes of the parameter list and a list of PERIPHERAL DEVICE ASSOCIATION DESCRIPTORS.

Table 33 - 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)				
x+3				(Length x)				
	. . .							
n-y+1				PERIPHERAL DEVICE ASSOCIATIONS DESCRIPTOR (Last)				
n				(Length y)				

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 34.

Table 34 - 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 25 for the format of the LOGICAL UNIT DESCRIPTOR.

6.3.1.7 REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action

The REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action (see table 35) requests the position of the selected logical unit within the target be sent to the application client.

Table 35 - 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 36. 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 36) contains a four-byte field that contains the length in bytes of the parameter list and the position of the selected logical unit within the target.

Table 36 - REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER parameter list

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

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

The 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.

6.3.1.8 REPORT STATES service action

The REPORT STATES service action (see table 37) requests that state information about the selected logical unit(s) within the target be sent to the application client.

Table 37 - 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 26 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 39.

The REPORT STATES field contains the information on which logical unit(s) the target shall report state information. See table 38 for the defined states.

Table 38 - 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 39) contains a four-byte header that indicates the length in bytes of the parameter list plus a list of LOGICAL UNIT STATE DESCRIPTORS.

Table 39 - 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 40.

Table 40 - 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 21 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 26 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 41, table 42, table 43, table 44, table 45, and table 46 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 replaceable 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 replaceable 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 replaceable is vendor specific.

Table 41 defines the STATE OF THE LOGICAL UNIT field contents when the LOGICAL UNIT TYPE field indicates LUN_Z.

Table 41 - LUN_Z 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 (i.e., the addressed SCSI storage array) 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 LUN_Z 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 REPORT STATUS service action with the REPORT STATES field set to 00b.

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

Table 42 defines the STATE OF THE LOGICAL UNIT field contents when the LOGICAL UNIT TYPE field indicates volume set. Other reporting commands also use the state codes defined in table 42.

.

Table 42 - 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 user data has been lost.
0Fh	Dynamic reconfiguration in progress	The addressed volume set is being reconfigured. In this state all user data is still protected.
03h	Exposed	Within the addressed volume set user data is not protected. In this state all user data is still valid, however, a failure will cause a loss of user data or a loss of user data availability.
0Eh	Fractionally exposed	Within the addressed volume set part of the user data is not protected. In this state all user data is still valid, however, a failure may cause a loss of user data or a loss of user data availability.
04h	Partially exposed	Within the addressed volume set one or more logical unit(s) have failed. In this state all user 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 user data is protected.
0Ch	Protection disabled	Within the addressed volume set the generation of check user data has been disabled. In this state all user data is still valid, however, a failure will cause a loss of user data or a loss of user 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 user 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 or more underlaying redundancy groups.
0Bh	Spare in use	Within the addressed volume set a spare is being used. In this state all user data is still protected.
0Dh	Verify in progress	Within the addressed volume set user data is being verified. NOTE 17 - The verify operation may involve one or more underlaying redundancy groups.
10h-3Fh	Reserved	
40h-7Fh	Vendor specific	

Table 43 defines the STATE OF THE LOGICAL UNIT field contents when the LOGICAL UNIT TYPE field indicates redundancy group. Other reporting commands also use the state codes defined in table 43.

Table 43 - Redundancy group states

Codes	States	Description
00h	Available	The addressed redundancy group is configured.
0Ch	Dynamic reconfiguration in progress	The addressed redundancy group is being reconfigured. In this state the protected space is still protected.
01h	Exposed	Within the addressed redundancy group user data is not protected. In this state all user data is still valid, however, a failure causes a loss of user data or a loss of user data availability.
02h	Invalidated protected space	Within the addressed redundancy group user 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 user data has been disabled. In this state all user data is still valid, however, a failure causes a loss of user data or a loss of user 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 user data is being verified.
0Dh-3Fh	Reserved	
40h-7Fh	Vendor specific	

Table 44 defines the STATE OF THE LOGICAL UNIT field contents when the LOGICAL UNIT TYPE field indicates peripheral device or p_extent. Other reporting commands also use the state codes defined in table 44.

Table 44 - 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 45 defines the STATE OF THE LOGICAL UNIT field contents when the LOGICAL UNIT TYPE field indicates spare. Other reporting commands also use the state codes defined in table 45.

Table 45 - 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 been exchanged with a failed object.
06h-3Fh	Reserved	
40h-7Fh	Vendor specific	

Table 46 defines the STATE OF THE LOGICAL UNIT field contents when the LOGICAL UNIT TYPE field indicates component device. Other reporting commands also use the state codes defined in table 46.

Table 46 - Component device states

Codes	States	Description
00h	Available	The addressed component device is 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	

6.3.1.9 REPORT SUPPORTED CONFIGURATION METHOD service action

The REPORT SUPPORTED CONFIGURATION METHOD service action (see table 52) requests that information regarding the configuration method(s) the target supports and wether an application client may change configurations using those methods. An application client may use this information to determine the minimum set of service actions supported by the target.

Table 47 - REPORT SUPPORTED CONFIGURATION METHOD service action

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	RESERVED			SERVICE ACTION (09h)				
2	RESERVED							
3	RESERVED							
4	RESERVED							
5	RESERVED							
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
6	(MSB)							
7				ALLOCATION LENGTH				
8								
9								(LSB)
10	RESERVED							
11	CONTROL							

The REPORT SUPPORTED CONFIGURATION METHOD parameter list (see table 53) contains the supported configuration methods. The application client may use this information to determine the minimum set of service actions supported by the target.

Table 48 - REPORT SUPPORTED CONFIGURATIN METHOD parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	RESERVED			BASIC	RESERVED			SIMPLE
1	RESERVED							GENERAL
2	RESERVED							
3	RESERVED							

The SIMPLE field (see table 49) indicates if the target supports the simple configuration method.

Table 49 - SIMPLE

Codes	Description
00b	Target does not support the simple configuration method.
01b	Target supports, at a minimum, the simple configuration method's mandatory reporting service actions (see 5.2.5.1).
10b	Reserved
11b	Target supports, at a minimum, the simple configuration method's mandatory reporting and mandatory configuration service actions (see 5.2.5.1).

The BASIC field (see table 50) indicates if the target supports the basic configuration method.

Table 50 - BASIC

Codes	Description
00h	Target does not support the basic configuration method.
01h	Target supports, at a minimum, the basic configuration method's mandatory reporting service actions (see 5.2.5.2).
10h	Reserved
11h	Target supports, at a minimum, the basic configuration method's mandatory reporting and mandatory configuration service actions (see 5.2.5.2).

The GENERAL field (see table 51) indicates if the target supports the general configuration method.

Table 51 - GENERAL

Codes	Description
00h	Target does not support the general configuration method.
01h	Target supports, at a minimum, the general configuration method's mandatory reporting service actions (see 5.2.5.3).
10h	Reserved
11h	Target supports, at a minimum, the general configuration method's mandatory reporting and mandatory configuration service actions (see 5.2.5.3).

6.3.1.10 REPORT UNCONFIGURED CAPACITY service action

The REPORT UNCONFIGURED CAPACITY service action (see table 52) requests that information regarding the unconfigured capacity within the target be sent to the application client.

Table 52 - REPORT UNCONFIGURED CAPACITY service action

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	RESERVED			SERVICE ACTION (08h)				
2	RESERVED							
3	RESERVED							
4	RESERVED							
5	RESERVED							
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
6	(MSB)							
7				ALLOCATION LENGTH				
8								
9								(LSB)
10	RESERVED							
11	CONTROL							

The REPORT UNCONFIGURED CAPACITY parameter list (see table 53) contains unconfigured capacity information. The application client may use this information when configuring a volume set.

Table 53 - REPORT UNCONFIGURED CAPACITY parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				UNASSIGNED P_EXTENT CAPACITY				
2								
3								(LSB)
4	(MSB)							
5				UNASSIGNED PS_EXTENT CAPACITY				
6								
7								(LSB)
8	RESERVED						MOREPS	MOREP
9	RESERVED							
10	(MSB)							
11				BYTES PER BLOCK				(LSB)

The UNASSIGNED P_EXTENT CAPACITY field contains the largest value the application client may use in the CAPACITY field of the next requested CREATE/MODIFY STORAGE ARRAY CONFIGURATION service action (6.8.1.5) or CREATE/MODIFY BASIC REDUNDANCY GROUP service action (6.6.1.2).

The UNASSIGNED PS_EXTENT CAPACITY field contains the largest value the application client may use in the CAPACITY field of the next requested CREATE/MODIFY BASIC VOLUME SET service action (6.8.1.4).

NOTE 18 - Before each configuration attempt the application client should issue the REPORT UNCONFIGURED CAPACITY service action to determine the maximum available capacity.

A more unassigned p_extent (MOREP) bit of zero indicates the target is reporting, in the UNASSIGNED P_EXTENT CAPACITY field, all the unassigned p_extent (5.2.2.10) blocks within the selected target. A MOREP bit of one indicates the target has more unassigned p_extent capacity than is being reported in the UNASSIGNED P_EXTENT CAPACITY field. The target shall set the MOREP bit to a one when it cannot configure all the unassigned p_extents into a single volume set or redundancy group.

A more unassigned ps_extent (MOREPS) bit of zero indicates the target is reporting, in the UNASSIGNED PS_EXTENT CAPACITY field, all the unassigned ps_extent (5.2.2.11) blocks within the selected target. A MOREPS bit of one indicates the target has more unassigned ps_extent capacity than is being reported in the UNASSIGNED PS_EXTENT CAPACITY field. The target shall set the MOREPS bit to a one when it cannot configure all the unassigned ps_extents into a single volume set.

The BYTES PER BLOCK field contains the size, in bytes, of the logical blocks in the UNASSIGNED P_EXTENT CAPACITY field and the UNASSIGNED PS_EXTENT CAPACITY field. A value of zero in the BYTES PER BLOCK field shall indicate the number of bytes per logical block is 512.

6.4 MAINTENANCE (OUT) commands

6.4.1 MAINTENANCE (OUT) command service actions

The service actions for the MAINTENANCE(OUT) command are listed in table 54.

Table 54 - Service actions for MAINTENANCE (OUT) command

Service name	Service action	Type	Subclause
ADD PERIPHERAL DEVICE/COMPONENT DEVICE	00h	O	6.4.1.1
ATTACH TO COMPONENT DEVICE	01h	O	6.4.1.2
BREAK PERIPHERAL DEVICE/COMPONENT DEVICE	07h	O	6.4.1.3
EXCHANGE P_EXTENT	02h	O	6.4.1.4
EXCHANGE PERIPHERAL DEVICE/COMPONENT DEVICE	03h	O	6.4.1.5
INSTRUCT COMPONENT DEVICE	04h	O	6.4.1.6
REMOVE PERIPHERAL DEVICE/COMPONENT DEVICE	05h	O	6.4.1.7
SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER	06h	O	6.4.1.8
Reserved	08h-17h		
Vendor specific	18h-1Fh		
Key: O = Service action implementation is optional.			

6.4.1.1 ADD PERIPHERAL DEVICE/COMPONENT DEVICE service action

The ADD PERIPHERAL DEVICE/COMPONENT DEVICE service action (see table 55) 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 55 - 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 21 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 21 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.

6.4.1.2 ATTACH TO COMPONENT DEVICE service action

The ATTACH TO COMPONENT DEVICE service action (see table 56) 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. On successful completion of this service action a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense

code shall be set to COMPONENT DEVICE ATTACHED.

NOTE 19 - 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 20 - 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 56 - 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)							
7				LIST LENGTH				
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 57). If the requested component device 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 57 - 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 25 for the format of the LOGICAL UNIT DESCRIPTOR. 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.

6.4.1.3 BREAK PERIPHERIAL DEVICE/COMPONENT DEVICE service action

The BREAK PERIPHERAL DEVICE/COMPONENT DEVICE service action (see table 58) requests a peripheral device or a component device be placed into a broken state.

Table 58 - 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 21 for a list of valid component device types.

6.4.1.4 EXCHANGE P_EXTENT service action

The EXCHANGE P_EXTENT service action (see table 59) requests the target replace a p_extent with another p_extent. If the exchange operation fails to complete successfully and the IMMED bit is zero 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.

NOTE 21 - If the IMMED bit is one and the exchange operation fails then a deferred error is reported.

Table 59 - 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 and parameter list have been validated.

The EXCHANGE P_EXTENT parameters list (see table 60) contains the addresses of the logical units that shall be exchanged.

Table 60 - 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 17 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 17 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 shall not be an error for the old p_extent and new p_extent to address the same p_extent.

6.4.1.5 EXCHANGE PERIPHERAL DEVICE/COMPONENT DEVICE service action

The EXCHANGE PERIPHERAL DEVICE/COMPONENT DEVICE service action (see table 61) requests that the target replace one peripheral device or component device with another peripheral device or component device.

Table 61 - 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.

6.4.1.6 INSTRUCT COMPONENT DEVICE service action

The INSTRUCT COMPONENT DEVICE service action (see table 62) requests the target to take the requested action on the addressed component device.

Table 62 - 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)							
7				LIST LENGTH				
8								
9								(LSB)
10	RESERVED							
11	CONTROL							

The COMPONENT DEVICE INSTRUCTION field contains the action to be taken. See table 63 for a list of actions.

Table 63 - 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 64) contains vendor specific actions to be applied to the component device addressed in the INSTRUCT COMPONENT DEVICE service action command descriptor block.

Table 64 - INSTRUCT COMPONENT DEVICE parameter list

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

6.4.1.7 REMOVE PERIPHERAL DEVICE/COMPONENT DEVICE service action

The REMOVE PERIPHERAL DEVICE/COMPONENT device service action (see table 65) requests a peripheral device or a component device be removed from the target.

Table 65 - 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 being removed 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.

6.4.1.8 SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action

The SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action (see table 66) requests the target set the identifier in the addressed logical unit with the value received in the SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER parameter list. On successful completion of this service action a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to DEVICE IDENTIFIER CHANGED.

Table 66 - 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)							
7				LIST LENGTH				
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 received by the IDENTIFIER field contents 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 67) contains the location identifier of the addressed logical unit.

Table 67 - SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER parameter list

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

The 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.

6.5 REDUNDANCY GROUP (IN) command

6.5.1 REDUNDANCY GROUP (IN) command service actions

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

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

Service name	Service action	Type	Subclause
REPORT BASIC REDUNDANCY GROUP	02h	B	6.5.1.1
REPORT REDUNDANCY GROUPS	00h	G	6.5.1.2
REPORT UNASSIGNED REDUNDANCY GROUP SPACE	01h	G	6.5.1.3
RESERVED	03h-17h		
VENDOR SPECIFIC	18h-1Fh		
Key: B = Service action implementation is mandatory if basic configuration method is implemented. G = Service action implementation is mandatory if general configuration method is implemented.			

6.5.1.1 REPORT BASIC REDUNDANCY GROUP service action

The REPORT BASIC REDUNDANCY GROUP service action (see table 69) requests that information regarding the selected redundancy group be sent to the application client. This service action differs from the REPORT REDUNDANCY GROUP service action (see 6.6.1.3) in that it does not report detailed information on the mapping of protected space and check data.

.

Table 69 - REPORT BASIC REDUNDANCY GROUP service action

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

The LUN_R field specifies the address of the redundancy group for which information shall be reported per table 70. 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.

The REPORT BASIC REDUNDANCY GROUP parameter list is defined in table 70.

Table 70 - REPORT BASIC REDUNDANCY **parameter list**

Bit Byte	7	6	5	4	3	2	1	0
0	RESERVED							
1	REDUNDANCY GROUP METHOD							
2	BUSPROC	RESERVED		EQSPRD	RESERVED			
3	RESERVED	STATE OF THE REDUNDANCY GROUP						
4	(MSB)	CAPACITY						
5								
6								
7								(LSB)
8	(MSB)	BYTES PER BLOCK						
9								(LSB)
10	(MSB)	REDUNDANCY GROUP PERIPHERAL DEVICE DESCRIPTOR LIST LENGTH						
11								(LSB)
	REDUNDANCY GROUP PERIPHERAL DEVICE DESCRIPTOR(S)							
12	REDUNDANCY GROUP PERIPHERAL DEVICE DESCRIPTOR (First)							
15								
	. . .							
n-3	REDUNDANCY GROUP PERIPHERAL DEVICE DESCRIPTOR (Last)							
n								

The REDUNDANCY GROUP METHOD field (table 71) indicates the type of protection being used within the addressed redundancy group. For a description of the redundancy group methods see 5.2.2.12.

Table 71 - REDUNDANCY GROUP METHODS

Codes	Description
00h	No redundancy
01h	Copy redundancy
02h	XOR redundancy
03h	P+Q redundancy
04h	P+S redundancy
05h	S redundancy
06h-7Fh	Reserved
80h-FFh	Vendor specific

An equal user data spreading (EQSPRD) bit of zero indicates the target is configured such that the protected space and check data is spread in a nonuniform manner over the peripheral devices associated with the addressed redundancy group. An EQSPRD bit of one indicates the target is configured such that the protected space and check data is spread in a uniform manner over all the peripheral devices associated with the addressed redundancy group.

A bus protection (BUSPROC) bit of zero indicates that the target is configured such that a single bus failure may cause the application client to lose access to user data associated with the addressed redundancy group. A BUSPROC bit of one indicates that the target is configured such that single bus failure does not cause the application client to lose access to any user data associated with the addressed redundancy group.

The REDUNDANCY GROUP STATE field is defined in table 43.

The CAPACITY field indicates the size of the addressed redundancy group in logical blocks.

The BYTES PER BLOCK field indicates the size, in bytes, of the logical blocks in the CAPACITY field.

The REDUNDANCY GROUP PERIPHERAL DEVICE DESCRIPTOR contains a list of peripheral devices associated with the addressed redundancy group. See table 72 for the format of the REDUNDANCY GROUP PERIPHERAL DEVICE DESCRIPTOR field.

TABLE 72 - REDUNDANCY GROUP PERIPHERAL DEVICE DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	LUN_P							
1								
2	WEIGHTING OF PROTECTED SPACE + CHECK DATA							
3	PERCENT OF CHECK DATA							

The LUN_P field indicates the address of a peripheral device associated with the addressed redundancy group.

The WEIGHTING OF USER DATA + CHECK DATA field indicates the portion of the redundancy group's capacity

placed on the selected peripheral device.

The PERCENT OF CHECK DATA field indicates the percentage of the weighted capacity that contains check data on the selected peripheral device.

NOTE 22 - The WEIGHTING OF PROTECTED SPACE + CHECK DATA and the PERCENT OF CHECK DATA fields may contain the same values as the target received from the application client or it may be calculated using equations. One example that may be used by the target to calculate the value of the WEIGHTING OF PROTECTED SPACE + CHECK DATA field is $dw = (vc)/c$ and to calculate to the value of the PERCENT OF CHECK DATA field is $pr = (100 \times cd)/c$ where:

c = capacity of the peripheral device selected in the LUN_P field assigned to the selected redundancy group,
dw = value to be placed in WEIGHTING OF PROTECTED SPACE + CHECK DATA field,
vc = value in the CAPACITY field,
pr = value to be placed in the PERCENT OF CHECK DATA field, and
cd = the number of blocks of check data configured on the selected peripheral device assigned to the selected redundancy group.

6.5.1.2 REPORT REDUNDANCY GROUPS service action

The REPORT REDUNDANCY GROUPS service action (see table 73) requests that information regarding redundancy groups within the target be sent to the application client.

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 73 - 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)			ALLOCATION LENGTH				
7								
8								
9								(LSB)
10	RESERVED							RPTSEL
11	CONTROL							

If the RPTSEL bit is one, the LUN_R field specifies the address of the redundancy group for which information shall be reported per table 74. 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 74) contains a four-byte header that defines the length in bytes of the parameter list and a list of REPORT REDUNDANCY GROUP DESCRIPTORS.

Table 74 - 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 75.

Table 75 - 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 METHOD							
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 METHOD field (see table 71) indicates the type of protection being used within the redundancy group. For a description of the redundancy group methods see 5.2.2.12.

The GRANULARITY OF UNITS field (see table 76) indicates units being used within the redundancy group.

Table 76 - 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 43.

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 77 for the format of the REDUNDANCY GROUP P_EXTENT DESCRIPTOR field.

Table 77 - REDUNDANCY GROUP P_EXTENT DESCRIPTOR

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

See table 17 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.

A start with spare (STRSPARE) bit is only valid for the P+S redundancy method of redundancy (see 5.2.2.12) and shall be ignored for all other methods of redundancy. When the P+S redundancy method is indicated a STRSPARE bit of zero indicates mapping of this p_extent follows the pattern:

a) NUMBER OF UNITS OF CHECK DATA (XOR);
b) NUMBER OF UNITS OF CHECK DATA (spare);
c) NUMBER OF UNITS OF USER DATA;
d) repeat from a until end of p_extent.

When the P+S redundancy method is indicated a STRSPARE bit of one indicates mapping of this p_extent follows the pattern:

a) NUMBER OF UNITS OF CHECK DATA (spare);
B) NUMBER OF UNITS OF USER DATA;
c) NUMBER OF UNITS OF CHECK DATA (XOR);
d) NUMBER OF UNITS OF CHECK DATA (spare);
E) NUMBER OF UNITS OF USER DATA;
f) repeat from c until end of 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.

6.5.1.3 REPORT UNASSIGNED REDUNDANCY GROUP SPACE service action

The REPORT UNASSIGNED REDUNDANCY GROUP SPACE service action (see table 78) 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 78 - 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)			ALLOCATION LENGTH				
7								
8								
9								(LSB)
10	RESERVED							RPTSEL
11	CONTROL							

If the RPTSEL bit is one, 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 79. 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 79) 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 79 - 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 80 for the format of the REPORT UNASSIGNED REDUNDANCY GROUP SPACE DESCRIPTOR.

Table 80 - Format of REPORT UNASSIGNED REDUNDANCY GROUP SPACE DESCRIPTOR

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

See table 71 for a description of the REDUNDANCY GROUP METHOD field.

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

The REDUNDANCY GROUP METHOD field indicates the type of protection being used within the redundancy group that contains the unassigned ps_extents. See table 71 for a list of the types of protection.

An overlaid redundancy group bit (OVERLAY) of zero indicates the redundancy group specified in the LUN_R field does not have any ps_extents that overlap with other redundancy group(s). An overlay bit of one indicates the redundancy group specified in the LUN_R field does have ps_extents that overlap with other redundancy group(s).

For any redundancy group that indicates a it overlays another redundancy group the application client should examine the PS_EXTENT DESCRIPTOR(S) to determine which of the ps_extent(s) are same between the overlaid redundancy groups. The application client shall only use a duplicate ps_extent to configure a single volume set.

The REDUNDANCY GROUP STATE field is defined in table 43.

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 81.

Table 81 - 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 23 - 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.

6.6 REDUNDANCY GROUP (OUT) command

6.6.1 REDUNDANCY GROUP (OUT) command service actions

The service actions for the REDUNDANCY GROUP (OUT) command are listed in table 82.

Table 82 - Service actions for REDUNDANCY GROUP (OUT) command

Service name	Service action	Type	Subclause
CONTROL GENERATION OF CHECK DATA	00h	O	6.6.1.1
CREATE/MODIFY BASIC REDUNDANCY GROUP	07h	BC	6.6.1.2
CREATE/MODIFY REDUNDANCY GROUP	01h	GC	6.6.1.3
DELETE REDUNDANCY GROUP	02h	O	6.6.1.4
REBUILD P_EXTENT	03h	O	6.6.1.5
REBUILD PERIPHERAL DEVICE	04h	O	6.6.1.6
RECALCULATE CHECK DATA	05h	O	6.6.1.7
VERIFY CHECK DATA	06h	O	6.6.1.8
RESERVED	08h-17h		
VENDOR SPECIFIC	18h-1Fh		

Key: BC = Service action implementation is mandatory if basic configuration method is implemented with configuration support.
GC = Service action implementation is mandatory if general configuration method is implemented with configuration support.
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 83) requests that the generation of check data within a redundancy group be enabled or disabled.

Table 83 - 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							

If the ALLRG bit is zero, 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 enabled from the time the redundancy group is created until the time a CONTROL GNERATION OF CHECK DATA service action indicating it's LUN_R is received with a DISCHK bit set to one. 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.

6.6.1.2 CREATE/MODIFY BASIC REDUNDANCY GROUP service action

The CREATE/MODIFY BASIC REDUNDANCY GROUP service action (see table 84) 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). A redundancy group shall only be created or expanded using unassigned p_extents (see 5.2.2.10). If the create operation fails to complete successfully and the IMMED bit is zero 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 and the IMMED bit is zero 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. On successful completion of the create/modify basic redundancy group a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to REDUNDANCY GROUP CREATED OR MODIFIED.

NOTE 24 - If the IMMED bit is one and the create/modify basic redundancy group fails then a deferred error is reported.

Table 84 - CREATE/MODIFY BASIC REDUNDANCY GROUP service action

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BFh)							
1	RESERVED			SERVICE ACTION (07h)				
2	REDUNDANCY GROUP METHOD							
3	BUSPROC	RESERVED		EQSPRD	RESERVED			
4	(MSB)	LUN_R						
5								(LSB)
6	(MSB)	LIST LENGTH						
7								
8								
9								(LSB)
10	CREATE/MODIFY		CONFIGURE		RESERVED			IMMED
11	CONTROL							

The REDUNDANCY GROUP METHOD field indicates the type of protection that shall be used within the redundancy group being created or modified. See table 71 for the format of the REDUNDANCY GROUP METHOD field.

The REDUNDANCY GROUP METHOD field shall only indicate the minimum amount of protection required by the redundancy group being configured or modified. See table 85 for the minimum requirements for each type of redundancy.

Table 85 - Minimum redundancy group protection

Type	Minimum protection within the configured redundancy group
No redundancy	The SACL is not required to protect user data.
Copy redundancy	The SACL shall duplicate all user data at least one time on different peripheral devices.
XOR redundancy	The SACL shall protect user data such that a single peripheral device failure does not cause loss of user data.
P+Q redundancy	The SACL shall protect user data such that the failure of two peripheral devices does not cause loss of user data.
P+S redundancy	The SACL shall protect user data such that the failure of a single peripheral device does not cause loss of user data and after some vendor specific amount of time a second peripheral device failure shall not cause loss of user data. If a second peripheral device fails within the vendor specific amount of time user data may be lost.
S redundancy	The SACL shall have access to at least enough spare space to allow a rebuild of one peripheral device within that space. User data may be lost if the SACL cannot predict or predicts incorrectly a peripheral device failure. User data may also be lost if the SACL does not complete a rebuild before the peripheral device fails.

An equal spreading (EQSPRD) bit of zero indicates the target may spread protected space and check data in a nonuniform manner over the peripheral devices associated with the redundancy group being created or modified. A EQSPRD bit of one indicates the target shall spread protected space and check data in a uniform manner over all the peripheral devices associated with the redundancy group being created or modified.

A bus protection (BUSPROC) bit of zero indicates that the target may be configured such that a single bus failure may cause the application client to lose access to any user data associated with the redundancy group being created or modified. A BUSPROC bit of one indicates that the target shall be configured so that a single bus failure shall not cause the application client to lose access to any user data associated with the redundancy group being created or modified.

The LUN_R field specifies the address of the redundancy group that shall be created or modified.

An immediate (IMMED) bit of zero indicates that status shall be returned after the create/modify basic redundancy group has completed. An IMMED bit of one indicates that the storage array shall return status as soon as the command descriptor block and parameter list have been validated.

The CONFIGURE field is defined in table 86.

TABLE 86 - CONFIGURE

Codes	Description
00b	Any unassigned p_extent(s) within the target that received the CREATE/MODIFY BASIC REDUNDANCY GROUP service action may be used to configure the selected redundancy group to the requested capacity. Any CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTORS (table 89) shall be ignored.
01b	The target shall use the CREATE/MODIFY BASIC REDUNDANCY GROUP parameter list (table 88) to determine the configuration of the redundancy group. The EQSPRD bit shall be ignored.
10b	All unassigned p_extents within the target that received the CREATE/MODIFIY BASIC REDUNDANCY GROUP service action shall be configured into a redundancy group. The CAPACITY field (table 88) and any CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTORS (table 89) shall be ignored.
11b	Reserved

The CREATE/MODIFY field is defined in table 87.

TABLE 87 - CREATE/MODIFY

Codes	Description
00b	The target shall create a redundancy group and shall assign to the created redundancy group the logical unit number contained in the LUN_R field. If the addressed redundancy group already exists within the target the target shall modify the existing redundancy group as requested in the CREATE/MODIFY BASIC REDUNDANCY GROUP service action. The target may preserve the contents of protected space on completion of a modify.
01b	The target shall create a redundancy group, and shall assign to the created redundancy group logical unit numbers per the addressing rules (see 5.2.1). The LUN_R field shall be ignored.
10b	The target shall modify the redundancy group addressed in the LUN_R field. If the addressed redundancy group does not exist the target shall terminate the command 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.The target may preserve the contents of protected space on completion of the modify.
11b	The target shall modify the redundancy group addressed in the LUN_R field. If the addressed redundancy group does not exist the target shall terminate the command 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. The target shall preserve the contents of protected space on completion of the modify.

The CREATE/MODIFY BASIC REDUNDANCY GROUP parameter list (see table 88) contains capacity information and a list of CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTORS that are used to create or modify the addressed redundancy group.

Table 88 - CREATE/MODIFY BASIC REDUNDANCY GROUP parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				CAPACITY				
2								
3								(LSB)
4	(MSB)							
5				BYTES PER BLOCK				(LSB)
6	RESERVED							
7	RESERVED							
	CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTORS(S) (if any)							
8				CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTOR 0				
11								
	. . .							
n-3				CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTOR X				
n								

The CAPACITY field contains the size to configure the redundance group in logical blocks. If the CREATE/MODIFY field is 10b the new size of the redundancy group shall be set to the value received in the CAPACITY field.

Attempts by an application client to modify a redundancy group to a smaller capacity shall result in the target terminating the command with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to PARAMETER VALUE INVALID.

The BYTES PER BLOCK field contains the size, in bytes, of the logical blocks in the CAPACITY field. A value of zero in the BYTES PER BLOCK field shall indicate the number of bytes per logical block is 512.

The CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTOR contains information the target shall use to control the check data mapping within peripheral devices. See table 89 for the format of the CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTOR.

Table 89 - Data format of CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	LUN_P							
1								
2	WEIGHTING OF PROTECTED SPACE + CHECK DATA							
3	PERCENT OF CHECK DATA							

The LUN_P field defines the address of the peripheral device to place user data and check data.

The WEIGHTING OF PROTECTED SPACE + CHECK DATA field contains a value used to calculate the portion of the redundancy group's capacity to place on the selected peripheral device. The target shall determine this capacity by using the equation $c = (dw) \times [(vc) / (\Sigma dw)]$ where:

c = capacity of the peripheral device selected in the LUN_P field,
dw = value of WEIGHTING OF PROTECTED SPACE + CHECK DATA field, and
vc = value of the CAPACITY field.

If the requested capacity (c) will not fit within the unassigned p_extent area(s) on the addressed peripheral device the target shall terminate the command with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to PARAMETER VALUE INVALID.

The PERCENT OF CHECK DATA field contains the percentage of the requested capacity (c) that shall contain check data on the selected peripheral device. If the PERCENT OF CHECK DATA field contains a value greater then 100 the target shall terminate the command with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to PARAMETER VALUE INVALID.

6.6.1.3 CREATE/MODIFY REDUNDANCY GROUP service action

The CREATE/MODIFY REDUNDANCY GROUP service action (see table 90) 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). A redundancy group shall only be created or expanded using unassigned p_extents (see 5.2.2.10). If the create operation fails to complete successfully and the IMMED bit is zero 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 and the IMMED bit is zero 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. On successful completion of this create/modify redundancy group a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to REDUNDANCY GROUP CREATED OR MODIFIED.

NOTE 25 - If the IMMED bit is one and the create/modify redundancy group fails then a deferred error is reported.

Table 90 - 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 METHOD							
3	RESERVED				GRANULARITY OF UNITS			
4	(MSB)			LUN_R				
5								(LSB)
6	(MSB)			LIST LENGTH				
7								
8								
9								(LSB)
10	CREATE/MODIFY		RESERVED					IMMED
11	CONTROL							

The REDUNDANCY GROUP METHOD field indicates the type of protection that shall be used within the redundancy group being created or modified. See table 71 for the format of the REDUNDANCY GROUP METHOD 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 76 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 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 and parameter list have been validated.

The CREATE/MODIFY field is defined in table 91.

TABLE 91 - CREATE/MODIFY

Codes	Description
00b	The target shall create a redundancy group and shall assign to the created redundancy group the logical unit number contained in the LUN_R field. If the addressed redundancy group already exists within the target the target shall modify the existing redundancy group as requested in the CREATE/MODIFIY REDUNDANCY GROUP service action. The target may preserve the contents of protected space on completion of a modify.
01b	The target shall create a redundancy group and shall assign to the created redundancy a logical unit number per the addressing rules (see 5.2.1). The LUN_R field shall be ignored.
10b	The target shall modify the redundancy group addressed in the LUN_R field. If the addressed redundancy group does not exist the target shall terminate the command 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. The target may preserve the contents of protected space on completion of the modify.
11b	The target shall modify the redundancy group addressed in the LUN_R field. If the addressed redundancy group does not exist the target shall terminate the command 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. The target shall preserve the contents of protected space on completion of the modify.

The CREATE/MODIFY REDUNDANCY GROUP parameter list (see table 92) contains a list of CREATE/MODIFY P_EXTENT DESCRIPTORS that shall be combined to create or modify the addressed redundancy group.

Table 92 - 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	CREATE/MODIFY P_EXTENT DESCRIPTOR 0							
27								
	. . .							
n-27	CREATE/MODIFY P_EXTENT DESCRIPTOR X							
n								

The CREATE/MODIFY P_EXTENT DESCRIPTOR contains information the target shall use to control the protected space mapping of the p_extent. See table 93 for the format of the CREATE/MODIFY P_EXTENT DESCRIPTOR.

Table 93 - 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	STRSPARE	(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 17 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 START CHECK DATA INTERLEAVE UNIT field shall be ignored if the NUMBER OF UNITS OF CHECK DATA field equals zero or if the STRSPARE bit is one.

A start with spare (STRSPARE) bit is only valid for the P+S redundancy method of redundancy (see 5.2.2.12) and shall be ignored for all other methods of redundancy. When the P+S redundancy method is selected a STRSPARE bit of zero indicates mapping of this p_extent shall follow the pattern:

a) START CHECK DATA INTERLEAVE UNIT;
b) NUMBER OF UNITS OF CHECK DATA (XOR);
c) NUMBER OF UNITS OF CHECK DATA (spare);
d) NUMBER OF UNITS OF USER DATA;
e) repeat from b until end of p_extent.

When the P+S redundancy method is selected a STRSPARE bit of one indicates mapping of this p_extent shall follow the pattern:

a) NUMBER OF UNITS OF CHECK DATA (spare);
b) NUMBER OF UNITS OF USER DATA;
c) NUMBER OF UNITS OF CHECK DATA (XOR);
d) NUMBER OF UNITS OF CHECK DATA (spare);
e) NUMBER OF UNITS OF USER DATA;
f) repeat from c until end of 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.

6.6.1.4 DELETE REDUNDANCY GROUP service action

The DELETE REDUNDANCY GROUP service action (see table 94) requests that the selected redundancy group be deleted. On successful completion of this service action a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to REDUNDANCY GROUP DELETED.

Table 94 - 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)	LUN_R						
5								(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.

6.6.1.5 REBUILD P_EXTENT service action

The REBUILD P_EXTENT service action (see table 95) requests the rebuild of all or part of one or more redundancy group(s).

Table 95 - 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 and parameter list have been validated.

The REBUILD field specifies the rebuild instructions the target shall use. See table 96 for the format of the REBUILD field.

Table 96 - 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 and the IMMED bit is zero 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 97) contains the p_extent to be rebuilt and a list of redundancy groups to not be used to rebuild the p_extent.

Table 97 - 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)							
15	LUN_R 0							(LSB)
	. . .							
n-3	RESERVED							
n-2	RESERVED							
n-1	(MSB)							
n	LUN_R X							(LSB)

The P_EXTENT DESCRIPTOR defines the boundaries of the rebuild. See table 17 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.

6.6.1.6 REBUILD PERIPHERAL DEVICE service action

The REBUILD PERIPHERAL DEVICE service action (see table 98) requests the rebuild of all or part of one or more redundancy group(s) located on a selected peripheral device.

Table 98 - 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 and parameter data have been validated.

The REBUILD field specifies the rebuild instructions the target shall use. See table 99 for the format of the REBUILD field.

Table 99 - 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 and the IMMED bit is zero 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 100) contains the peripheral device to be rebuilt and a list of redundancy groups to not be used to rebuild the peripheral device.

Table 100 - 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.

6.6.1.7 RECALCULATE CHECK DATA service action

The RECALCULATE CHECK DATA service action (see table 101) requests the target to recalculate check data within a redundancy group. If the recalculate operation fails to complete successfully and the IMMED bit is zero 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.

NOTE 26 - If the IMMED bit is one and the recalculate fails then a deferred error is reported.

Table 101 - 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							

If ALLRG equals zero 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.

6.6.1.8 VERIFY CHECK DATA service action

The VERIFY CHECK DATA service action (see table 102) requests that check data be verified consistent with the protected space within a redundancy group.

Table 102 - 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							

If ALLRG equals zero 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 27 - 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.

6.7 VOLUME SET (IN) command

6.7.1 VOLUME SET (IN) command service actions

The service actions for the VOLUME SET(IN) command are listed in table 103.

Table 103 - Service actions for volume set (in) command

Service name	Service action	Type	Subclause
REPORT BASIC VOLUME SET	03h	B	6.7.1.1
REPORT STORAGE ARRAY CONFIGURATION	02h	S	6.7.1.2
REPORT UNASSIGNED VOLUME SETS	01h	O	6.7.1.3
REPORT VOLUME SETS	00h	G	6.7.1.4
RESERVED	04h-17h		
VENDOR SPECIFIC	18h-1Fh		
Key: B = Service action implementation is mandatory if basic configuration method is implemented. G = Service action implementation is mandatory if general configuration method is implemented. O = Service action implementation is optional. S = Service action implementation is mandatory if simple configuration method is implemented.			

6.7.1.1 REPORT BASIC VOLUME SET service action

The REPORT BASIC VOLUME SET service action (see table 104) requests that information regarding the selected volume set within the target be sent to the application client. This service action differs from the REPORT VOLUME SET service action (see 6.7.1.4) in that it does not report detailed information on the mapping of the user data.

Table 104 - REPORT BASIC VOLUME SET service action

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

The LUN_V field specifies the address of the volume set for which information shall be reported per table 108. 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.

The REPORT BASIC VOLUME SET parameter list is defined in table 105.

Table 105 - REPORT BASIC VOLUME SET parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	RESERVED							
1	RESERVED							
2	RESERVED			EQSPRD	RESERVED			
3	RESERVED	STATE OF THE VOLUME SET						
4	(MSB)	CAPACITY						
5								
6								
7								(LSB)
8	(MSB)	BYTES PER BLOCK						
9								(LSB)
10	(MSB)	REPORT BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR LIST LENGTH						
11								(LSB)
	REPORT BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR(S)							
12	REPORT BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR (First) (Length x)							
(x+11)								
	. . .							
n-(y+1)	REPORT BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR (Last) (Length y)							
n								

An equal user data spreading (EQSPRD) bit of zero indicates the target is configured such that the user data is spread in a nonuniform manner over the peripheral devices associated with the addressed volume set. An EQSPRD bit of one indicates the target is configured such that the user data is spread in a uniform manner over all the peripheral devices associated with the addressed volume set.

The VOLUME SET STATE field is defined in table 42.

The CAPACITY field indicates the size of the user data addressed volume set in logical blocks.

The BYTES PER BLOCK field indicates the size, in bytes, of the logical blocks in the CAPACITY field.

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

The REPORT BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR contains a list of peripheral devices

associated with the addressed volume set. See table 106 for the format of the REPORT BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR field.

TABLE 106 - REPORT BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	LUN_P							
1								
2	RESERVED							
3	WEIGHTING OF USER DATA							
4	RESERVED							
5	RESERVED							
6	(MSB)							
7	ASSOCIATED REDUNDANCY GROUPS LIST LENGTH							(LSB)
	LUN_R(S)							
8	LUN_R (First)							
9								
10	RESERVED							
11	RESERVED							
	. . .							
n-3	LUN_R (Last)							
n-2								
n-1	RESERVED							
n	RESERVED							

The LUN_P field indicates the address of a peripheral device associated with the addressed volume set.

The WEIGHTING OF USER DATA field indicates the portion of the volume set's capacity placed on the selected peripheral device.

NOTE 28 - The WEIGHTING OF USER DATA fields may contain the same values as the target received from the application client or it may be calculated using an equation. One example of an equation that may be used by the target to calculate the value in the WEIGHTING OF USER DATA field is dw = (vc)/c where:

c = capacity of the peripheral device selected in the LUN_P field assigned to the selected volume set,
dw = value to be placed into the WEIGHTING OF USER DATA field, and

vc = value of the CAPACITY field.

The ASSOCIATED REDUNDANCY GROUP LIST LENGTH field specifies the length in bytes of the following list of LUN_R(s).

The LUN_R field specifies the address of the redundancy group(s) associated with the addressed volume set.

6.7.1.2 REPORT STORAGE ARRAY CONFIGURATION service action

The REPORT STORAGE ARRAY CONFIGURATION service action (see table 107) requests that information regarding the selected volume set be sent to the application client. If this service action requests information on a volume set that has more than one associated redundancy group the target shall terminate the command with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

Table 107 - REPORT STORAGE ARRAY CONFIGURATION service action

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

The LUN_V field specifies the address of the volume set for which information shall be reported per table 108. 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.

The REPORT CONFIGURATION parameter list is defined in table 108.

Table 108 - REPORT CONFIGURATION parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	RESERVED							
1	REDUNDANCY GROUP METHOD							
2	BUSPROC	RESERVED		EQSPRD	RESERVED			
3	RESERVED	STATE OF THE VOLUME SET						
4	(MSB)	CAPACITY						
5								
6								
7								(LSB)
8	(MSB)	BYTES PER BLOCK						
9								(LSB)
10	(MSB)	NORMAL USER DATA TRANSFER SIZE						
11								(LSB)
12	RESERVED							
13	REBUILD/RECALCULATE PRIORITY							
14	PERCENTAGE OF SEQUENTIAL READ TRANSFERS							
15	PERCENTAGE OF SEQUENTIAL WRITE TRANSFERS							
16	RESERVED							
17	RESERVED							
18	(MSB)	VOLUME SET PERIPHERAL DEVICE DESCRIPTOR LIST LENGTH						
19								(LSB)
	VOLUME SET PERIPHERAL DEVICE DESCRIPTOR(S)							
20	VOLUME SET PERIPHERAL DEVICE DESCRIPTOR (First)							
23								
	. . .							
n-3	VOLUME SET PERIPHERAL DEVICE DESCRIPTOR (Last)							
n								

The REDUNDANCY GROUP METHOD field (table 71) indicates the type of protection being used within the redundancy group associated with the addressed volume set. For a description of the redundancy group methods see 5.2.2.12.

An equal user data spreading (EQSPRD) bit of zero indicates the volume set is configured such that the user data is spread in a nonuniform manner over the peripheral devices associated within the addressed volume set. An EQSPRD bit of one indicates the volume set is configured such that the user data is spread in a uniform manner over all the peripheral devices associated within the addressed volume set.

A bus protection (BUSPROC) bit of zero indicates that the volume set is configured such that a single bus failure causes the application client to lose access to user data. A BUSPROC bit of one indicates that the volume set is configured such that single bus failure does not cause the application client to lose access to any user data.

The VOLUME SET STATE field is defined in table 42.

The CAPACITY field indicates the size of the addressed volume set in logical blocks.

The BYTES PER BLOCK field indicates the size, in bytes, of the logical blocks in the CAPACITY field and the NORMAL USER DATA TRANSFER SIZE field.

The NORMAL USER DATA TRANSFER SIZE field indicates the number of logical blocks the target expects during a normal user data transfer. A NORMAL USER DATA TRANSFER SIZE field of zero indicates the target has received no direction on the size of user data transfers.

The REBUILD/RECALCULATE PRIORITY field indicates the length of time the target takes to do a rebuild operation or a recalculate operation. See table 109 for a description of the contents of the REBUILD/ RECALCULATE PRIORITY field.

Table 109 - Rebuild/recalculate priority selection

Codes	Description
00h	Shall indicate the target has received no direction on how long it will take to do rebuilds nor recalculates or that the associated redundancy group is configured as no redundancy.
01h	Indicates the target shall suspend rebuild and recalculate operations during all read/write requests from any application client.
02h-FEh	An indication of the length of time the target takes to do a rebuild operation or a recalculate operation. Generally, larger values indicate shorter rebuild and recalculate times.
FFh	Indicates the target shall not accept any read/write requests from an application client until the rebuild or recalculate operation is complete.

The PERCENTAGE OF SEQUENTIAL READ TRANSFERS field indicates the percent of times the target expects reads of sequential logical blocks on consecutive user data read transfers. See table 110 for a description

of the contents of the PERCENTAGE OF SEQUENTIAL READ TRANSFERS field.

TABLE 110 - PERCENTAGE OF SEQUENTIAL READ TRANSFERS

Codes	Description
0	The target has received no direction on the sequentially of user data read transfers.
1-100	The percent of times the target expects reads of sequential logical blocks on consecutive user data read transfers.
101-127	Reserved
128-255	Vendor Specific

The PERCENTAGE OF SEQUENTIAL WRITE TRANSFERS field indicates the percent of times the target expects writes of sequential logical blocks on consecutive user data write transfers. See table 111 for a description of the contents of the PERCENTAGE OF SEQUENTIAL WRITE TRANSFERS field.

TABLE 111 - PERCENTAGE OF SEQUENTIAL WRITE TRANSFERS

Codes	Description
0	The target has received no direction on the sequentially of user data write transfers.
1-100	The percent of times the target expects writes of sequential logical blocks on consecutive user write read transfers.
101-127	Reserved
128-255	Vendor Specific

The VOLUME SET PERIPHERAL DEVICE DESCRIPTOR contains a list of peripheral devices associated with the addressed volume set. See table 112 for the format of the VOLUME SET PERIPHERAL DEVICE DESCRIPTOR field.

TABLE 112 - VOLUME SET PERIPHERAL DEVICE DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	LUN_P							
1								
2	RESERVED							
3	WEIGHTING OF USER DATA							

The LUN_P field indicates the address of a peripheral device associated with the addressed volume set.

The WEIGHTING OF USER DATA field indicates the portion of the volume set's capacity and the redundancy group's placed on the selected peripheral device.

NOTE 29 - The WEIGHTING OF USER DATA fields may contain the same values as the target received from the application client or it may be calculated using an equation. One example of an equation that may be used by the target to calculate the value in the WEIGHTING OF USER DATA field is dw = (vc)/c where:

c = capacity of the peripheral device selected in the LUN_P field assigned to the selected volume set,
dw = value of WEIGHTING OF USER DATA field, and
vc = value of the CAPACITY field.

6.7.1.3 REPORT UNASSIGNED VOLUME SETS service action

The REPORT UNASSIGNED VOLUME SETS service action (see table 113) requests that an identifier for each configured volume set, that does not ALREADY have a lun_v assigned within the target be sent to the application client.

Table 113 - REPORT UNASSIGNED VOLUME SETS service action

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

The REPORT UNASSIGNED VOLUME SETS parameter list (see table 114) contains a four-byte header that contains the length in bytes of the parameter list and a list of REPORT UNASSIGNED VOLUME SET DESCRIPTORS. The list of REPORT UNASSIGNED VOLUME SET DESCRIPTORS shall be a list of all the configured volume sets within the target that do not have a lun_v assigned.

Table 114 - REPORT UNASSIGNED VOLUME SETS parameter list

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

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

The REPORT UNASSIGNED VOLUME SET DESCRIPTOR is defined in table 115.

Table 115 - Format of REPORT UNASSIGNED VOLUME SET DESCRIPTOR

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

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

The IDENTIFY field is defined in the vital product data device identification page (83h) (see SCSI-3 Primary Commands Standard).

6.7.1.4 REPORT VOLUME SETS service action

The REPORT VOLUME SETS service action (see table 116) requests that information regarding volume sets within the target be sent to the application client.

Table 116 - 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)							
7				ALLOCATION LENGTH				
8								
9								(LSB)
10	RESERVED							RPTSEL
11	CONTROL							

If the RPTSEL bit is one, the LUN_V field specifies the address of the volume set for which information shall be reported per table 117. 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 117) contains a four-byte header that contains the length in bytes of the parameter list and a list of REPORT VOLUME SET DESCRIPTORS.

Table 117 - REPORT VOLUME SETS parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1								
2				REPORT VOLUME SET LIST LENGTH (n-3)				
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 118.

Table 118 - Format of REPORT VOLUME SET DESCRIPTOR

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

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 are being used within the volume set being created or modified. See table 76 for the format of the GRANULARITY OF UNITS field.

The VOLUME SET STATE field is defined in table 42.

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 119 for the format of the VOLUME SET PS_EXTENT DESCRIPTOR field.

TABLE 119 - VOLUME SET PS_EXTENT DESCRIPTOR

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

See table 81 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.

A no check data skip bit (NOCHKSKIP) of zero indicates the target did not count any units defined as check data in mapping the user data stripe depth. A NOCHKSKIP bit of one indicates the target counted all units defined as check data in mapping the user data stripe depth.

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.

6.8 VOLUME SET (OUT) command

6.8.1 VOLUME SET (OUT) command service actions

The service actions for the VOLUME SET(OUT) command are listed in table 120.

Table 120 - Service actions for VOLUME SET (OUT) command

Service name	Service action	Type	Subclause
ASSIGN LUN_V	07h	O	6.8.1.1
CONTROL GENERATION OF CHECK DATA	00h	O	6.8.1.2
CONTROL WRITE OPERATIONS	01h	O	6.8.1.3
CREATE/MODIFY BASIC VOLUME SET	09h	BC	6.8.1.4
CREATE/MODIFY STORAGE ARRAY CONFIGURATION	08h	SC	6.8.1.5
CREATE/MODIFY VOLUME SET	02h	GC	6.8.1.6
DEASSIGN LUN_V	06h	O	6.8.1.7
DELETE VOLUME SET	03h	O	6.8.1.8
RECALCULATE VOLUME SET CHECK DATA	04h	O	6.8.1.9
VERIFIY VOLUME SET CHECK DATA	05h	O	6.8.1.10
RESERVED	0Ah-17h		
VENDOR SPECIFIC	18h-1Fh		

Key: BC = Service action implementation is mandatory if basic configuration method is implemented with configuration support.
GC = Service action implementation is mandatory if general configuration method is implemented with configuration support.
O = Service action implementation is optional.
SC = Service action implementation is mandatory if simple configuration method is implemented with configuration support.

6.8.1.1 ASSIGN LUN_V service action

The ASSIGN LUN_V service action requests a lun_v be assigned to an already configured volume set. If the assign 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 ASSIGN FAILURE OCCURED. On successful completion of this service action a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to VOLUME SET ASSIGNED.

Table 121 - ASSIGN LUN_V service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BFh)							
1	RESERVED			SERVICE ACTION (07h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_V				
5								(LSB)
6	(MSB)							
7				LIST LENGTH				
8								
9								(LSB)
10	RESERVED					ENABLE RANGE		RESERVED
11	CONTROL							

The LUN_V field, if used (see table 122), specifies the address the volume set shall be assigned. If the lun_v is already assigned to another volume set within 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 MULTIPLLY ASSIGNED LOGICAL UNIT.

The ENABLE RANGE field is defined in table 122.

Table 122 - ENABLE RANGE

Codes	Description
00b	The target shall assign the logical unit number in LBA_V to the selected unassigned volume set.
01b	Assign a lun_v to the volume set selected in the IDENTIFIER parameter list. The LUN_V field shall be ignored. The target shall assign a logical unit number to the unassigned volume set per the addressing rules (see 5.2.1)
10b	Assign lun_v(s) to all unassigned volume sets within the selected target. The LUN_V field and the IDENTIFIER parameter list shall be ignored. The target shall assign a logical unit number(s) to each unassigned volume set(s) per the addressing rules (see 5.2.1)
11b	Reserved

The IDENTIFIER parameter list (see table 123) contains a a four-byte header that contains the length on bytes of the parameter list and an IDENTIFIER.

Table 123 - IDENTIFIER parameter list

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

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

The IDENTIFIER field is defined in the vital products data device identification page (83h) (see SCSI-3 Primary Commands Standard).

6.8.1.2 CONTROL GENERATION OF CHECK DATA service action

The CONTROL GENERATION OF CHECK DATA service action (see table 124) requests that the generation of check data within the underlaying redundancy group(s) of a volume set be enabled or disabled.

Table 124 - 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							

If the ALLVLU bit is zero, 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.

6.8.1.3 CONTROL WRITE OPERATIONS service action

The CONTROL WRITE OPERATIONS service action (see table 125) 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 125 - 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)							
5	LUN_V							(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). When the DISWR bit is zero, on successful completion of this service action a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit

attention condition the additional sense code shall be set to OPERATOR SELECTED WRITE PERMIT. A DISWR bit of one indicates write operations shall be disabled on the selected volume set(s). When the DISWR bit is one, on successful completion of this service action a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to OPERATOR SELECTED WRITE PROTECT.

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.

6.8.1.4 CREATE/MODIFY BASIC VOLUME SET service action

The CREATE/MODIFY BASIC VOLUME SET service action (see table 126) requests the creation of a new volume set or the modification of an existing volume set. This service action differs from the CREATE/ MODIFY VOLUME SET service action (see 6.8.1.6) in that it does not provide detailed control over the mapping of the user data. A volume set shall only be created or expanded using unassigned ps_extents (see 5.2.2.11). If the create operation fails to complete successfully and the IMMED bit is zero 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 and the IMMED bit is zero 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. On successful completion of this create/modify basic volume set a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to VOLUME SET CREATED OR MODIFIED.

NOTE 30 - If the IMMED bit is one and the create/modify basic volume set fails then a deferred error is reported.

Table 126 - CREATE/MODIFY BASIC VOLUME SET service action

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BFh)							
1	RESERVED			SERVICE ACTION (09h)				
2	RESERVED							
3	RESERVED			EQSPRD	RESERVED			
4	(MSB)	LUN_V						
5								(LSB)
6	(MSB)	LIST LENGTH						
7								
8								
9								(LSB)
10	CREATE/MODIFY		CONFIGURE		RESERVED			IMMED
11	CONTROL							

An equal user data spreading (EQSPRD) bit of zero indicates the target may spread user data in a nonuniform manner over the peripheral devices associated with the volume set being created or modified. A EQSPRD bit of one indicates the target shall spread user data in a uniform manner over all the peripheral devices associated with the volume set being created or modified.

The LUN_V field specifies the address of the volume set that shall be created or modified.

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

The CONFIGURE field is defined in table 127.

TABLE 127 - CONFIGURE

Codes	Description
00b	Any unassigned ps_extent(s) within the target that received the CREATE/MODIFY BASIC VOLUME SET service action may be used to configure the selected volume set to the requested capacity. Any CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTORS (table 138) shall be ignored.
01b	The target shall use the CREATE/MODIFY BASIC VOLUME SET parameter list (table 134) to determine the configuration of the volume set. The EQSPRD bit shall be ignored.
10b	All unassigned p_extents within the target that received the CREATE/MODIFIY BASIC VOLUME SET service action shall be configured into a volume set. The VOLUME SET CAPACITY field (table 134) and any CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTORS (table 138) shall be ignored.
11b	Reserved

The CREATE/MODIFY field is defined in table 128.

TABLE 128 - CREATE/MODIFY

Codes	Description
00b	The target shall create a volume set and shall assign to the created volume set the logical unit number contained in the LUN_V field. If the addressed volume set already exists within the target the target shall modify the existing volume set as requested in the CREATE/MODIFY BASIC VOLUME SET service action. The target may preserve the contents of and access to user data on completion of a modify.
01b	The target shall create a volume set and shall assign to the created volume set logical unit numbers per the addressing rules (xxx). The LUN_V field shall be ignored.
10b	The target shall modify the volume set addressed in the LUN_V field. If the addressed volume set does not exist the target shall terminate the command 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.The target may preserve the contents of and access to user data on completion of the modify.
11b	The target shall modify the volume set addressed in the LUN_V field. If the addressed volume set does not exist the target shall terminate the command 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. The target shall preserve the contents of and access to user data on completion of the modify.

The CREATE/MODIFY BASIC VOLUME SET parameter list (see table 129) contains capacity and a list of BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTORS that are used to create or modify the addressed volume set.

Table 129 - CREATE/MODIFY BASIC VOLUME SET parameter list

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				CAPACITY				
2								
3								(LSB)
4	(MSB)							
5				BYTES PER BLOCK				(LSB)
6	(MSB)							
7				BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR LIST LENGTH				(LSB)
	BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR(S)							
8								
(x+7)				BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR (First) (Length x)				
				. . .				
n-(y+1)								
n				BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR (Last) (Length y)				

The CAPACITY field contains the size to configure the volume set in logical blocks. If the CREATE/MODIFY field is 10b the new size of the volume set being modified shall be set to the value in the CAPACITY field.

NOTE 31 - Attempts by an application client to modify a volume set to a smaller capacity may result in a loss of user data.

The BYTES PER BLOCK field contains the size, in bytes, of the logical blocks in the CAPACITY field. A value of zero in the BYTES PER BLOCK field shall indicate the number of bytes per logical block is 512.

The BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR LIST LENGTH field specifies the length on bytes of the following list of BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR(s).

The BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR contains information the target shall use to control the user data mapping within peripheral devices. See table 130 for the format of the BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR.

Table 130 - Data format of BASIC VOLUME SET PERIPHERAL DEVICE DESCRIPTOR

Bit / Byte	7	6	5	4	3	2	1	0
0	LUN_P							
1								
2	RESERVED							
3	WEIGHTING OF USER DATA							
4	RESERVED							
5	RESERVED							
6	(MSB) ASSOCIATED REDUNDANCY GROUPS LIST LENGTH							
7								(LSB)
	LUN_R(S)							
8	LUN_R (First)							
9								
10	RESERVED							
11	RESERVED							
	. . .							
n-3	LUN_R (Last)							
n-2								
n-1	RESERVED							
n	RESERVED							

The LUN_P field defines the address of the peripheral device to place user data.

The WEIGHTING OF USER DATA field contains a value used to calculate the portion of the volume set's capacity to place on the selected peripheral device. The target shall determine this capacity by using the equation $c = (dw) \times [(vc) / (\Sigma dw)]$ where:

c = capacity of the peripheral device selected in the LUN_P field,
dw = value of WEIGHTING OF USER DATA field, and
vc = value of the CAPACITY field.

If the requested capacity (c) will not fit within the unassigned ps_extent area(s) on the addressed peripheral device the target shall terminate the command with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to PARAMETER VALUE INVALID.

The ASSOCIATED REDUNDANCY GROUPS LIST LENGTH field specifies the length on bytes of the following list of LUN_Rs.

The LUN_R field(s) contains the address of the redundancy group(s) on the selected peripheral device to associate the volume set being created or modified. Distribution of the volume sets user data between multiple redundancy groups is vender specific.

6.8.1.5 CREATE/MODIFY STORAGE ARRAY CONFIGURATION service action

The CREATE/MODIFY STORAGE ARRAY CONFIGURATION service action (see table 131) requests the creation of a new volume set and redundancy group, or the modification of an existing volume set and redundancy group. A storage array configuration shall only be created or expanded using unassigned p_extents (see 5.2.2.10). If the create operation fails to complete successfully and the IMMED bit is zero 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 and the IMMED bit is zero 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. On successful completion of this create/modify storage array configuration a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to VOLUME SET CREATED OR MODIFIED.

NOTE 32 - If the IMMED bit is one and the create/modify storage array configuration fails then a deferred error is reported.

Table 131 - CREATE/MODIFY STORAGE ARRAY CONFIGURATION service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BFh)							
1	RESERVED			SERVICE ACTION (08h)				
2	REDUNDANCY GROUP METHOD							
3	BUSPROC	RESERVED		EQSPRD	RESERVED			
4	(MSB)	LUN_V						
5								(LSB)
6	(MSB)	LIST LENGTH						
7								
8								
9								(LSB)
10	CREATE/MODIFY		CONFIGURE		RESERVED			IMMED
11	CONTROL							

The REDUNDANCY GROUP METHOD field indicates the type of protection that shall be used within the redundancy group being created or modified. See table 71 for the format of the REDUNDANCY GROUP

METHOD field.

The REDUNDANCY GROUP METHOD field shall only indicate the minimum amount of protection required by the redundancy group being configured or modified. See table 85 for the minimum requirements for each type of redundancy.

An equal user data spreading (EQSPRD) bit of zero indicates the target may spread user data in a nonuniform manner over the peripheral devices associated with the volume set being created or modified. A EQSPRD bit of one indicates the target shall spread user data in a uniform manner over all the peripheral devices associated with the volume set being created or modified.

A bus protection (BUSPROC) bit of zero indicates that the target may be configured such that a single bus failure may cause the application client to lose access to user data within the volume set being created or modified. A BUSPROC bit of one indicates that the target shall be configured so a single bus failure shall not cause the application client to lose access to any user data within the volume set being created or modified.

The LUN_V field specifies the address of the volume set that shall be created or modified.

An immediate (IMMED) bit of zero indicates that status shall be returned after the create/modify storage array operation has completed. An IMMED bit of one indicates that the storage array shall return status as soon as the command descriptor block has been validated, and the entire CREATE/MODIFIY STORAGE ARRAY CONFIGURATION parameters list has been transferred.

The CONFIGURE field is defined in table 132.

TABLE 132 - CONFIGURE

Codes	Description
00b	Any unassigned p_extent(s) within the target that received the CREATE/MODIFY STORAGE ARRAY CONFIGURATION service action may be used to configure the selected volume set and redundancy group to the requested capacity. Any CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTORS (table 138) shall be ignored.
01b	The target shall use the CREATE/MODIFY STORAGE ARRAY CONFIGURATION parameter list (table 134) to determine the configuration of the volume set and redundancy group. The EQSPRD bit shall be ignored.
10b	All unassigned p_extents within the target that received the CREATE/MODIFIY STORAGE ARRAY CONFIGURATION service action shall be configured into a volume set and a redundancy group. The VOLUME SET CAPACITY field (table 134) and any CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTORS (table 138) shall be ignored.
11b	Reserved

The CREATE/MODIFY field is defined in table 133.

TABLE 133 - CREATE/MODIFY

Codes	Description
00b	The target shall create a volume set and a redundancy group and shall assign to the created volume set the logical unit number contained in the LUN_V field. The target shall assign to the created redundancy group a logical unit number per the addressing rules (see 5.2.1). If the addressed volume set already exists within the target the target shall modify the existing volume set and its associated redundancy group as requested in the CREATE/MODIFY STORAGE ARRAY CONFIGURATION service action. The target may preserve the contents of and access to user data on completion of a modify.
01b	The target shall create a volume set and a redundancy group, and shall assign to the created volume set and redundancy group logical unit numbers per the addressing rules (see 5.2.1). The LUN_V field shall be ignored.
10b	The target shall modify the volume set addressed in the LUN_V field and its associated redundancy group. If the addressed volume set does not exist the target shall terminate the command 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.The target may preserve the contents of and access to user data on completion of the modify.
11b	The target shall modify the volume set addressed in the LUN_V field and its associated redundancy group. If the addressed volume set does not exist the target shall terminate the command 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. The target shall preserve the contents of and access to user data on completion of the modify.

The CREATE/MODIFY STORAGE ARRAY CONFIGURATION parameter list (see table 134) contains user data mapping information and a list of CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTORS that are used to create or modify the addressed volume set and its associated redundancy group.

Table 134 - CREATE/MODIFY STORAGE ARRAY CONFIGURATION parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				CAPACITY				
2								
3								(LSB)
4	(MSB)							
5				BYTES PER BLOCK				(LSB)
6	(MSB)							
7				NORMAL USER DATA TRANSFER SIZE				(LSB)
8	RESERVED							
9	REBUILD/RECALCULATE PRIORITY							
10	PERCENTAGE OF SEQUENTIAL READ TRANSFERS							
11	PERCENTAGE OF SEQUENTIAL WRITE TRANSFERS							
	CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTORS(S) (if any)							
12								
15				CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTOR 0				
	. . .							
n-3								
n				CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTOR X				

The CAPACITY field contains the size to configure the volume set and the redundance group in logical blocks. If the CREATE/MODIFY field is 10b the new size of the volume set being modified shall be set to the value in the CAPACITY field and the new size of the redundancy group shall be set to the value in the CAPACITY field.

NOTE 33 - The actual capacity, in blocks, of the configured volume set depends on the selected type of protection (e.g., copy redundancy would result in an actual capacity half the value in the CAPACITY field). The application client should use the REPORT STORAGE ARRAY CONFIGURATION service action (6.7.1.2) to determine the final capacity of a newly configured volume set.

NOTE 34 - Attempts by an application client to modify a volume set to a smaller capacity may result in a loss of user data.

The BYTES PER BLOCK field contains the size, in bytes, of the logical blocks in the CAPACITY field and the NORMAL USER DATA TRANSFER SIZE field. A value of zero in the BYTES PER BLOCK field shall indicate the

number of bytes per logical block is 512.

The NORMAL USER DATA TRANSFER SIZE field contains the number of logical blocks the application client normally requests transferred on each user data transfer. The target shall treat the NORMAL USER DATA TRANSFER SIZE field as an advisory parameter. A NORMAL USER DATA TRANSFER SIZE field of zero indicates the application client is providing no direction on the size of user data transfers.

The REBUILD/RECALCULATE PRIORITY field contains the length of time the target should take to do a rebuild operation or a recalculate operation. The target shall treat the REBUILD/RECALCULATE PRIORITY field as an advisory parameter. If the REDUNDANCY GROUP METHOD field contains a zero (i.e., no redundancy) then the target shall ignore the REBUILD/RECALCULATE PRIORITY field. See table 135 for a description of the contents of the REBUILD/RECALCULATE PRIORITY field.

Table 135 - Rebuild/recalculate priority selection

Codes	Description
00h	The application client is providing no direction on the length of time for rebuilds or recalculates
01h	The target shall suspend rebuild and recalculate operations during all read/write requests from any application client.
02h-FEh	An indication of the length of time the target should take to do a rebuild operation or a recalculate operation. Generally, larger values indicate shorter rebuild and recalculate times. NOTE 35 - The effect of different rebuild/recalculate times is to increase and decrease the performance of a target. Lower values increase performance but at a cost of being exposed to data loss for a longer time. Higher values decrease performance but keep the exposure to data loss at a minimum.
FFh	The target shall not accept any read/write requests from an application client until the rebuild or recalculate operation is complete. All attempts to read or write shall be l 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 READY, REBUILD IN PROGRESS or LOGICAL UNIT NOT READY, RECALCULATION IN PROGRESS.

The PERCENTAGE OF SEQUENTIAL READ TRANSFERS field contains the percent of times an application client is expected to do reads of sequential logical blocks on consecutive user data read transfers. The target shall treat the PERCENTAGE OF SEQUENTIAL READ TRANSFERS field as an advisory parameter. See table 136 for a description of the contents of the PERCENTAGE OF SEQUENTIAL READ TRANSFERS field.

TABLE 136 - PERCENTAGE OF SEQUENTIAL READ TRANSFERS

Codes	Description
0	The application client is providing no direction on the sequentially of user data read transfers.
1-100	The percent of times an application client is expected to do reads of sequential logical blocks on consecutive user data read transfers.
101-127	Reserved
128-255	Vendor Specific

The PERCENTAGE OF SEQUENTIAL WRITE TRANSFERS field contains the percent of times an application client is expected to do writes of sequential logical blocks on consecutive user data write transfers. The target

shall treat the PERCENTAGE OF SEQUENTIAL WRITE TRANSFERS field as an advisory parameter. See table 137 for a description of the contents of the PERCENTAGE OF SEQUENTIAL WRITE TRANSFERS field.

TABLE 137 - PERCENTAGE OF SEQUENTIAL WRITE TRANSFERS

Codes	Description
0	The application client is providing no direction on the sequentially of user data write transfers.
1-100	The percent of times an application client is expected to do writes of sequential logical blocks on consecutive user write read transfers.
101-127	Reserved
128-255	Vendor Specific

The CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTOR contains information the target shall use to control the user data mapping within peripheral devices. See table 138 for the format of the CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTOR.

Table 138 - Data format of CREATE/MODIFY PERIPHERAL DEVICE DESCRIPTOR

Bit Byte	7	6	5	4	3	2	1	0
0	LUN_P							
1								
2	RESERVED							
3	WEIGHTING OF USER DATA							

The LUN_P field defines the address of the peripheral device to place user data.

The WEIGHTING OF USER DATA field contains a value used to calculate the portion of the volume set's and redundancy group's capacity to place on the selected peripheral device. The target shall determine this capacity by using the equation $c = (dw) \times [(vc) / (\Sigma dw)]$ where:

c = capacity of the peripheral device selected in the LUN_P field,
dw = value of WEIGHTING OF USER DATA field, and
vc = value of the CAPACITY field.

If the requested capacity (c) will not fit within the unassigned p_extent area(s) on the addressed peripheral device the target shall terminate the command with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to PARAMETER VALUE INVALID.

6.8.1.6 CREATE/MODIFY VOLUME SET service action

The CREATE/MODIFY VOLUME SET service action (see table 139) requests the creation of a new volume set or the modification of an existing volume set. If the create operation fails to complete successfully and the IMMED bit is zero 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. On successful completion of this service action a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to VOLUME SET CREATED OR MODIFIED.

NOTE 36 - If the IMMED bit is one and the create/modify volume set fails then a deferred error is reported.

Table 139 - 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)							
7				LIST LENGTH				
8								
9								(LSB)
10	CREATE/MODIFY			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 76 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 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.

The CREATE/MODIFY field is defined in table 140.

TABLE 140 - CREATE/MODIFY

Codes	Description
00b	The target shall create a volume set and shall assign to the created volume set the logical unit number contained in the LUN_V field. 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. The target may preserve the contents of and access to user data on completion of a modify.
01b	The target shall create a volume set and shall assign to the created volume set a logical unit number per the addressing rules (see 5.2.1). The LUN_V field shall be ignored.
10b	The target shall modify the volume set addressed in the LUN_V field. If the addressed volume set does not exist the target shall terminate the command 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. The target may preserve the contents of and access to user data on completion of the modify.
11b	The target shall modify the volume set addressed in the LUN_V field. If the addressed volume set does not exist the target shall terminate the command 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. The target shall preserve the contents of and access to user data on completion of the modify.

The CREATE/MODIFY VOLUME SET parameter list (see table 141) 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 141 - 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 142 for the format of the CREATE/MODIFY PS_EXTENT DESCRIPTOR.

Table 142 - 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			NOCHKSKIP	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 81 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 37 - When the INCDEC bit is one the START LBA_PS field contains the largest logical block in the p_extent.

A no check data skip bit (NOCHKSKIP) of zero indicates the target shall not count any units defined as check data to determine when the user data stripe depth is reached. For an example of user data mapping with a NOCHLSKIP bit of zero see figure C.4. A NOCHKSKIP bit of one indicates the target shall count all units defined as check data to determine when the user data stripe depth is reached. For an example of user data mapping with a NOCHLSKIP bit of one see figure C.3.

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.

6.8.1.7 DEASSIGN LUN_V service action

The DEASSIGN LUN_V service action (see table 143) requests that a lun_v be deassigned. After a

successful completion of this service action the selected lun_v(s) are free to be used in the creation of new volume set(s). The target shall maintain the deassigned volume set(s) configuration and identifier. Any service actions, except ASSIGN LUN_V service action or CREATE/MODIFY VOLUME SET service action, that attempt to use a deassigned lun_v that has not been configured into a new volume set shall be terminated with a CHECK CONDITON status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT CONFIGURED. On successful completion of this service action a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to VOLUME SET DEASSIGNED.

Table 143 - DEASSIGN LUN_V service action

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BFh)							
1	RESERVED			SERVICE ACTION (06h)				
2	RESERVED							
3	RESERVED							
4	(MSB)			LUN_V				
5								(LSB)
6	RESERVED							
7	RESERVED							
8	RESERVED							
9	RESERVED							
10	RESERVED						ALLVLU	RESERVED
11	CONTROL							

If the ALLVLU bit is zero, the LUN_V field specifies the lun_v that shall be deassigned. 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 only the selected lun_v shall be deassigned. An ALLVLU bit of one indicates all lun_v(s) within the target shall be deassigned. The LUN_V field shall be ignored if the ALLVLU bit is one.

6.8.1.8 DELETE VOLUME SET service action

The DELETE VOLUME SET service action (see table 144) 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. On successful completion of this service action a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to VOLUME SET DELETED.

Table 144 - 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.

6.8.1.9 RECALCULATE VOLUME SET CHECK DATA service action

The RECALCULATE VOLUME SET CHECK DATA service action (see table 145) 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 and the IMMED bit is zero 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.

NOTE 38 - If the IMMED bit is one and the recalculate volume set check data fails then a deferred error is reported.

Table 145 - 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)							
7				LIST LENGTH				
8								
9								(LSB)
10	RESERVED						ALLVLU	IMMED
11	CONTROL							

If the ALLVLU bit is zero, 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 146) contains the range of LBA_Vs to be recalculated.

Table 146 - 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.

6.8.1.10 VERIFY VOLUME SET CHECK DATA service action

The VERIFY VOLUME SET CHECK DATA service action (see table 147) 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 147 - 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)			LIST LENGTH				
7								
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 148.

Table 148 - 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 39 - 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 149) 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 149 - 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.

6.9 SPARE (IN) command

6.9.1 SPARE (IN) command service actions

The service actions for the SPARE (IN) command are listed in table 150.

Table 150 - Service actions for SPARE (IN) command

Service name	Service action	Type	Subclause
REPORT P_EXTENT SPARE	00h	O	6.9.1.1
REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE	01h	O	6.9.1.2
RESERVED	02h-17h		
VENDOR SPECIFIC	18h-1Fh		
Key: O = Service action implementation is optional.			

6.9.1.1 REPORT P_EXTENT SPARE service action

The REPORT P_EXTENT SPARE service action (see table 151) requests that information regrading p_extent spares within the target be sent to the application client.

Table 151 - 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							

If the RPTSEL bit is one, the LUN_S field specifies the address of the p_extent spare that shall be reported per table 152. 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 152) 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 152 - 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 153.

Table 153 - 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							COVERALL
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 17 for a description of the P_EXTENT DESCRIPTOR.

A cover all (COVERALL) bit of zero indicates the objects the p_extent spare covers are listed in the COVERED LUN_R and COVERED P_EXTENT DESCRIPTOR field(s). A COVERALL bit of one indicates all p_extents within the SCSI-3 storage array that received the REPORT P_EXTENT SPARE service action are covered by the p_extent spare addressed in the LUN_S field. When the COVERALL bit is set to one the target shall return a COVERED LIST LENGTH of zero.

The STATE OF THE SPARE field is defined in table 45. If the STATE OF THE SPARE field contains the state 'spare in use' then the device server shall 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 17 for a description of the COVERED P_EXTENT DESCRIPTOR.

6.9.1.2 REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action

The REPORT PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action (see table 154) requests that information regarding peripheral device spares or component device spares within the target be sent to the application client.

Table 154 - 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							

If the RPTSEL bit is one, the LUN_S field specifies the address of the peripheral device spare or the component device spare that shall be reported per table 155. 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 155) 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 155 - 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 156.

Table 156 - 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/LUN_C				
6	RESERVED							COVERALL
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/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.

A cover all (COVERALL) bit of zero indicates the objects the peripheral device spare or component device spare covers are listed in the COVERED LOGICAL UNIT DESCRIPTOR field(s). A COVERALL bit of one indicates all peripheral devices or component devices within the SCSI-3 storage array that received the REPORT PERIPHERIAL DEVICE/COMPONENT DEVICE SPARE service action are covered by the peripheral device spare or component device spare addressed in the LUN_S field. When the COVERALL bit is set to one the target shall return a COVERED LOGICAL UNIT LIST LENGTH of zero.

The STATE OF THE SPARE field is defined in table 45. If the STATE OF THE SPARE field contains the state 'spare in use' then the device server shall 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 157) contains the list of logical units covered by the peripheral device or component device spare addressed in the LUN_S field.

Table 157 - 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 158) indicates the type of logical unit addressed in the LUN field.

Table 158 - 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.

6.10 SPARE (OUT) command

6.10.1 SPARE (OUT) command service actions

The service actions for the SPARE (IN) command are listed in table 159.

Table 159 - Service actions for spare (out) command

Service name	Service action	Type	Subclause
CREATE/MODIFY P_EXTENT SPARE	00h	O	6.10.1.1
CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE	01h	O	6.10.1.2
DELETE SPARE	02h	O	6.10.1.3
RESERVED	03h-17h		
VENDOR SPECIFIC	18h-1Fh		
Key: O = Service action implementation is optional.			

6.10.1.1 CREATE/MODIFY P_EXTENT SPARE service action

The CREATE/MODIFY P_EXTENT SPARE service action (see table 160) requests the target to create a p_extent spare or modify an existing p_extent spare. If the create operation fails to complete successfully and the IMMED bit is zero 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 and the IMMED bit is zero 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. On successful completion of this create/modify p_extent spare a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to SPARE CREATED OR MODIFIED.

NOTE 40 - If the IMMED bit is one and the create/modify p_extent spare fails then a deferred error is reported.

Table 160 - 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	CREATE/MODIFY		COVER		RESERVED			IMMED
11	CONTROL							

The LUN_S field specifies the address of the p_extent spare that shall be created or modified. If the application client requests a spare be modified and the SETLUN bit is set to one then 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 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.

The COVER field is defined in table 161.

TABLE 161 - COVER

Codes	Description
00b	The target shall use the CREATE/MODIFY P_EXTENT SPARE parameter list (table 163) to determine the p_extents and/or redundancy groups to be covered.
01b	All p_extents that are equal in size to the SPARE P_EXTENT DESCRIPTOR (table 163) within the SCSI-3 storage array that received the CREATE/MODIFY P_EXTENT SPARE service action shall be covered by the spare p_extent being created or modified except those listed in the CREATE/MODIFY P_EXTENT SPARE parameter list (table 163). In addition, any p_extents associated with redundancy group(s) listed in the CREATE/MODIFY P_EXTENT SPARE parameter list (table 163) shall not be covered by the spare p_extent being created or modified.
10b	All p_extents that are equal in size to the SPARE P_EXTENT DESCRIPTOR (table 163) within the SCSI-3 storage array that received the CREATE/MODIFY P_EXTENT SPARE service action that are not already covered by a spare shall be covered by the spare p_extent being created or modified. All fields, except the SPARE P_EXTENT DESCRIPTOR, within the CREATE/MODIFY P_EXTENT SPARE parameter list (table 163) shall be ignored.
11b	All p_extents that are equal in size to the SPARE P_EXTENT DESCRIPTOR (table 163) within the SCSI-3 storage array that received the CREATE/MODIFY P_EXTENT SPARE service action shall be covered by the spare p_extent being created or modified. All fields, except the SPARE P_EXTENT DESCRIPTOR, within the CREATE/MODIFY P_EXTENT SPARE parameter list (table 163) shall be ignored.

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. When the application client requests a spare p_extent be created the LUN_S field shall be ignored if the SETLUN bit is set to one.

The CREATE/MODIFY field is defined in table 166.

TABLE 162 - CREATE/MODIFY

Codes	Description
00b	The target shall create a p_extent spare and shall assign it the logical unit number contained in the LUN_S field. If the addressed p_extent spare already exists within the target the target shall modify the existing p_extent spare as requested in the CREATE/MODIFIY PERIPHERIAL DEVICE P_EXTENT SPARE service action.
01b	The target shall create a p_extent spare and shall assign it a logical unit number per the addressing rules (see 5.2.1). The LUN_S field shall be ignored.
10b	The target shall modify the p_extent spare addressed in the LUN_S field. If the addressed p_extent spare does not exist the target shall terminate the command 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.
11b	Reserved

The CREATE/MODIFY P_EXTENT SPARE parameter list (see table 163) contains a list of COVERED P_EXTENT DESCRIPTORS that shall be used to create or modify the addressed p_extent spare.

Table 163 - 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)							
13	COVERED LIST LENGTH (n-13)							(LSB)
14	(MSB)							
15	COVERED LUN_R LIST LENGTH (m-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 17 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 17 for a description of the COVERED P_EXTENT DESCRIPTOR.

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

The CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action (see table 164) 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 and the IMMED bit is zero 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 and the IMMED bit is zero 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. On successful completion of this create/modify peripheral device/component device spare a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code shall be set to SPARE CREATED OR MODIFIED.

NOTE 41 - If the IMMED bit is one and the exchange operation fails then a deferred error is reported.

Table 164 - 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)			LUN_P /LUN_C				
3								(LSB)
4	(MSB)			LUN_S				
5								(LSB)
6	(MSB)			LIST LENGTH				
7								
8								
9								(LSB)
10	CREATE/MODIFY		COVER		RESERVED		PORCSEL	IMMED
11	CONTROL							

The LUN_P/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. 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 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.

The COVER field is defined in table 165.

TABLE 165 - COVER

Codes	Description
00b	The target shall use the CREATE/MODIFY PERIPHERAL DEVICE/ COMPONENT DEVICE SPARE parameter list (table 167) to determine the peripheral device, component device and/or redundancy groups to be covered.
01b	All peripheral device(s) that have a capacity equal to or less than the peripheral device addressed in the LUN_P/LUN_C field (table 164) or component device(s) that have the same component device type as the component device addressed in LUN_P/LUN_C field (table 164) within the SCSI-3 storage array that received the CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action shall be covered by the spare peripheral device or spare component device being created or modified except those listed in the CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE parameter list (table 167). In addition, any peripheral device(s) or component device(s) associated with redundancy group(s) listed in the CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE parameter list (table 167) shall not be covered by the spare peripheral device or spare component device being created or modified.
10b	All peripheral device(s) that have a capacity equal to or less than the peripheral device addressed in the LUN_P/LUN_C field (table 164) or component device(s) that have the same component device type as the component device addressed in LUN_P/LUN_C field (table 164) within the SCSI-3 storage array that received the CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action that are not already covered by a spare peripheral device or spare component device shall be covered by the spare peripheral device or spare component device being created or modified. All fields within the CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE parameter list (table 167) shall be ignored.
11b	All peripheral device(s) that have a capacity equal to or less than the peripheral device addressed in the LUN_P/LUN_C field (table 164) or component device(s) that have the same component device type as the component device addressed in LUN_P/LUN_C field (table 164) within the SCSI-3 storage array that received the CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE service action shall be covered by the spare peripheral device or spare component device being created or modified. All fields within the CREATE/ MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE parameter list (table 167) shall be ignored.

The CREATE/MODIFY field is defined in table 166.

TABLE 166 - CREATE/MODIFY

Codes	Description
00b	The target shall create a peripheral device spare or component device spare and shall assign it the logical unit number contained in the LUN_S field. If the addressed peripheral device spare or component device spare already exists within the target the target shall modify the existing peripheral device spare or component device spare as requested in the CREATE/MODIFIY PERIPHERIAL DEVICE SPARE /COMPONENT DEVICE SPARE service action.
01b	The target shall create a peripheral device spare or component device spare and shall assign it a logical unit number per the addressing rules (see 5.2.1). The LUN_S field shall be ignored.
10b	The target shall modify the peripheral device spare or component device spare addressed in the LUN_S field. If the addressed peripheral device spare or component device spare does not exist the target shall terminate the command 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.
11b	Reserved

The CREATE/MODIFY PERIPHERAL DEVICE/COMPONENT DEVICE SPARE parameter list (see table 167) 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 167 - 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 157 for a description of the COVERED LOGICAL UNIT DESCRIPTOR.

6.10.1.3 DELETE SPARE service action

The DELETE SPARE service action (see table 168) requests the target delete the addressed spare. On successful completion of this service action a unit attention shall be generated for all initiators except the one that issued the service action. When reporting the unit attention condition the additional sense code

shall be set to SPARE DELETED.

Table 168 - 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.

6.11 Parameters for direct-access devices

6.11.1 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 169.

Table 169 - Mode page codes

Page Code	Description	Subclause
0Ah	Control mode page	SPC
02h	Disconnect-reconnect page	SPC
1Bh	LUN mapping page	6.11.1.1
09h	Peripheral device page	SPC
0Dh	Power condition page	SPC
Key: SPC = SCSI-3 Primary Commands Standard.		

6.11.1.1 LUN mapping page

The LUN mapping page (see table 170) 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 170 - 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 6.11.1.1) 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 42 - *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 43 - *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 44 - 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 5.2.1.2.1) 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 5.2.1.2.1). 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 45 - 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 5.2.1.2.1). 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 46 - 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 47 - 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 P+S redundancy mapping

Figure C.1 contains an example of how P+S redundancy check data would be interleaved with user data.

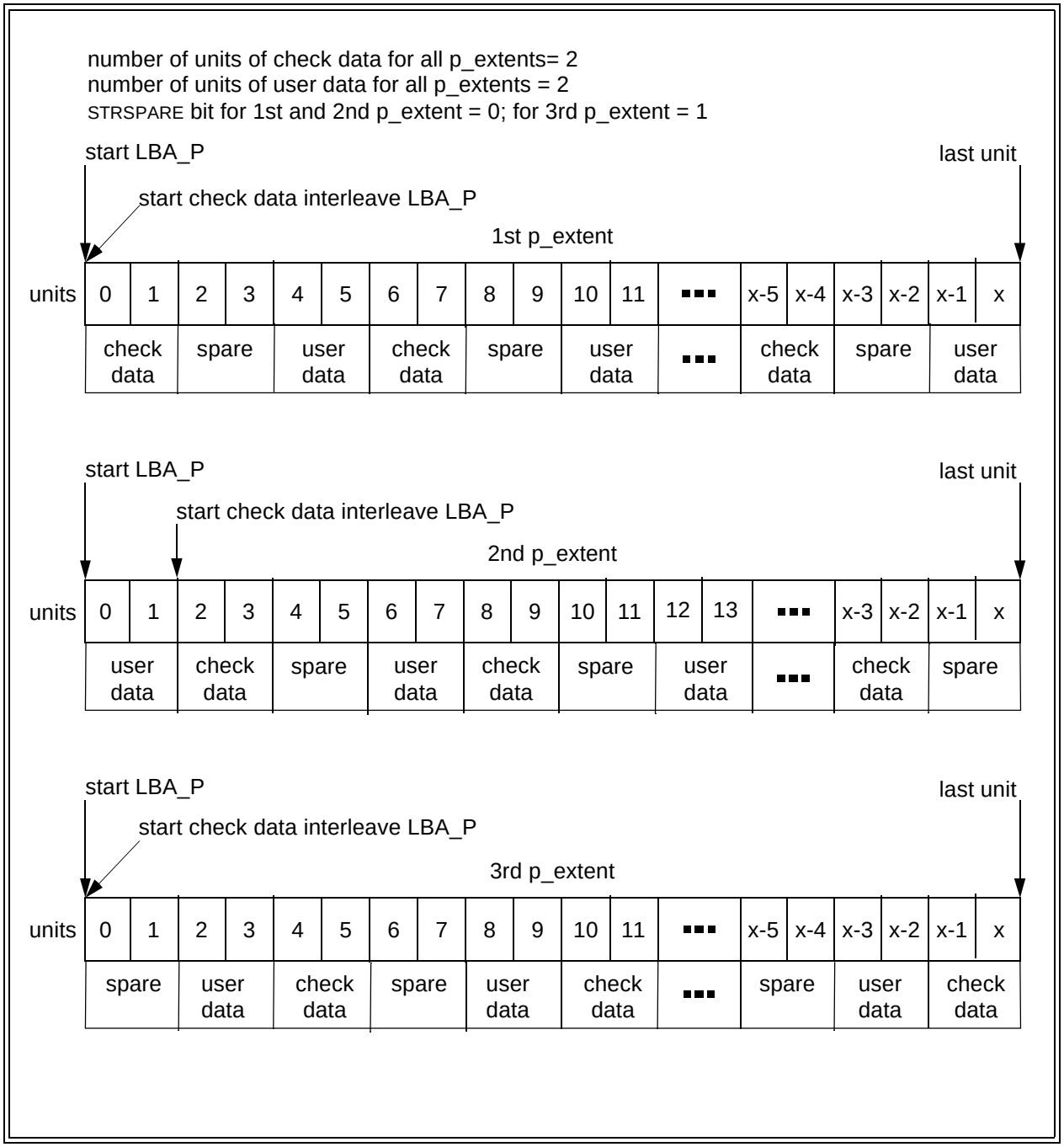

Figure C.1 - P+S redundancy mapping example

C.2 Example XOR redundancy mapping

Figure C.2 contains an example of how XOR redundancy check data would be interleaved with user data in a RAID 5 mapping.

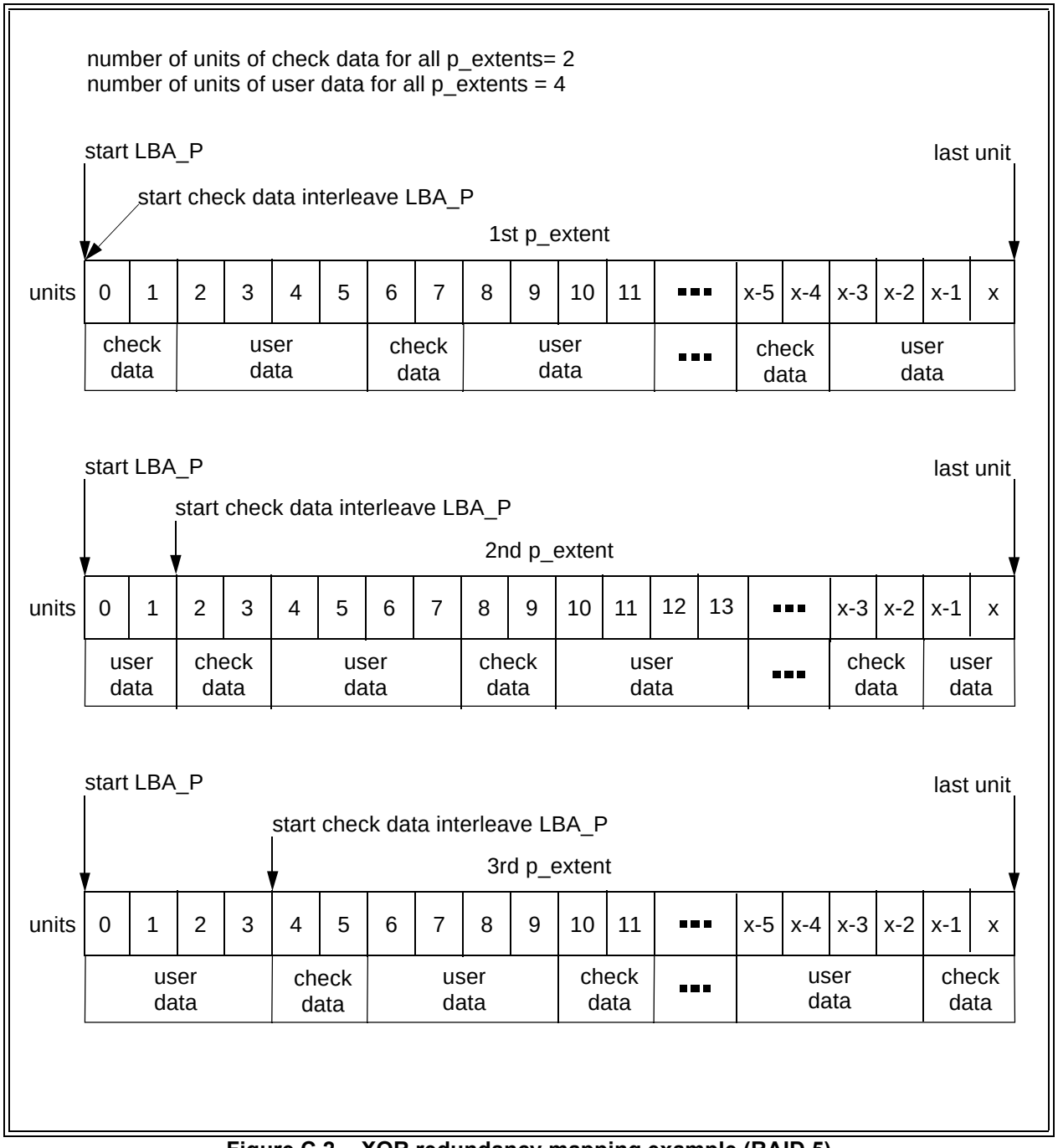

Figure C.2 - XOR redundancy mapping example (RAID 5)

C.3 User data mapping examples

C.3.1 Examples of user data mapped in a RAID 5 configuration

Figure C.3 contains an example of how user data would be mapped in a RAID 5 configuration where the

user data mapping counts check data.

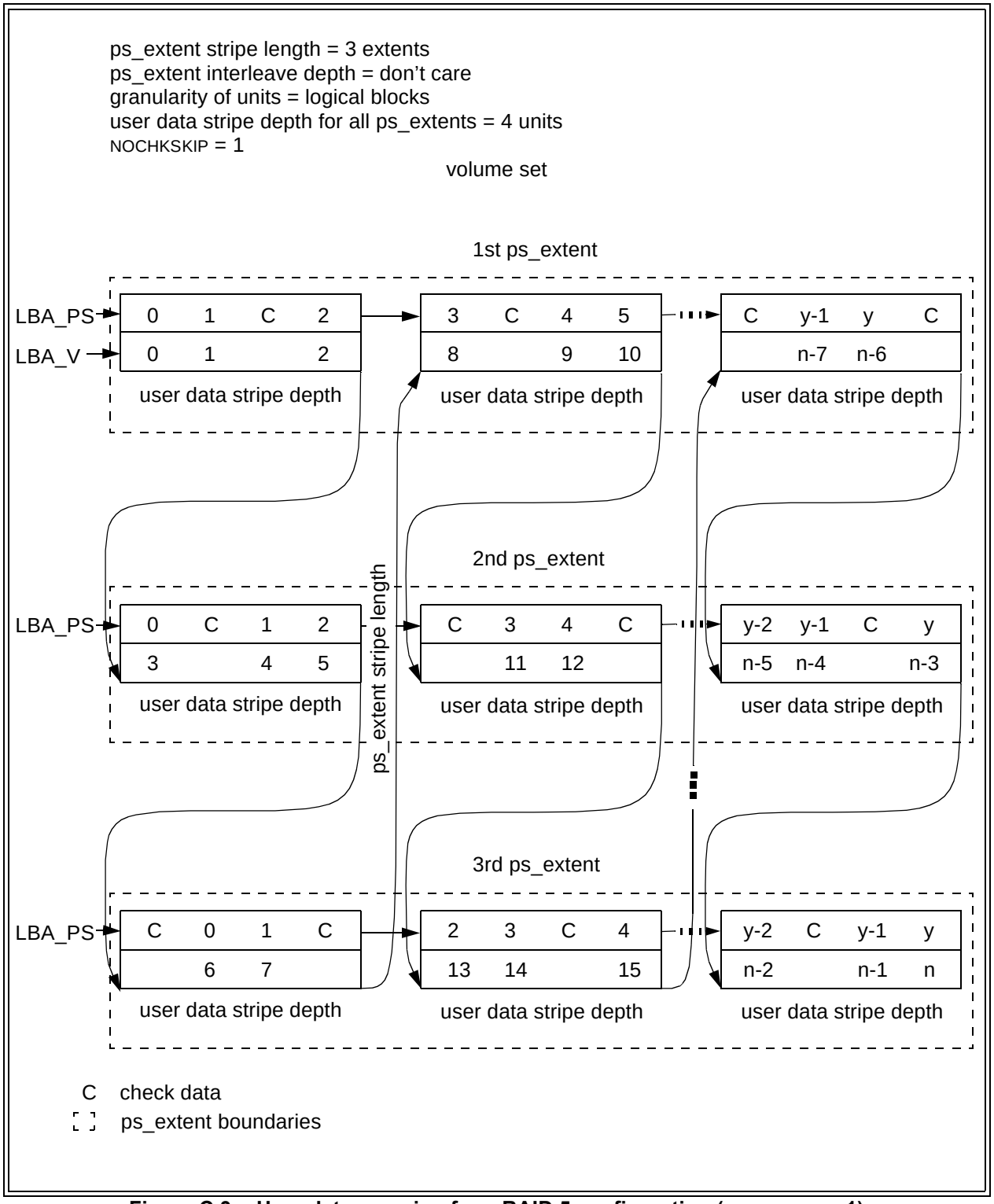

Figure C.3 - User data mapping for a RAID 5 configuration (NOCHKSKIP = 1)

Figure C.4 contains an example of how user data would be mapped in a RAID 5 configuration where the

user data mapping does not counts check data.

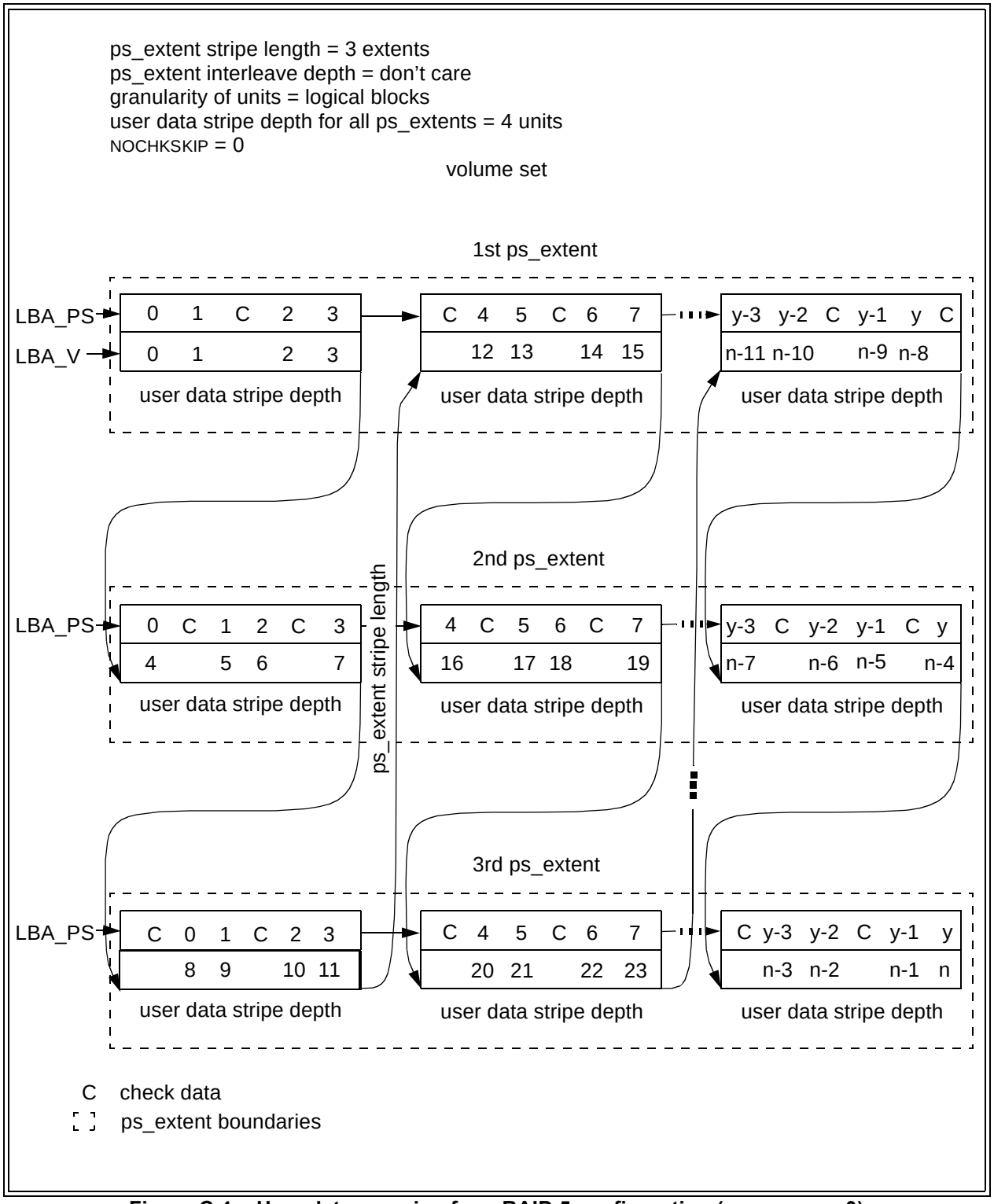

Figure C.4 - User data mapping for a RAID 5 configuration (NOCHKSKIP = 0)

C.3.2 Example of user data mapping in a RAID 3 configuration

Figure C.5 contains an example of how user data would be mapped in a RAID 3 configuration.

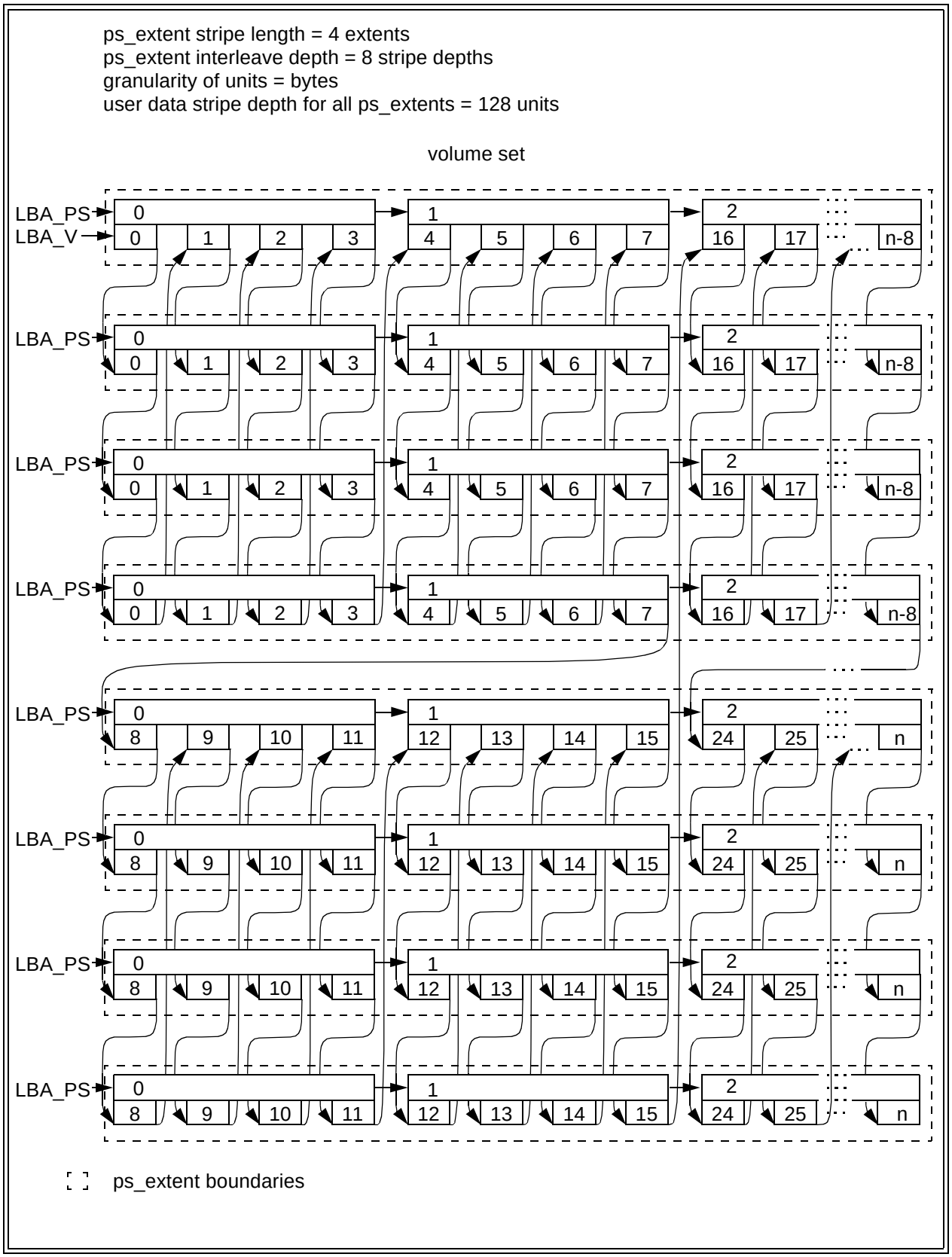

Figure C.5 - User data mapping for a RAID 3 configuration

Annex D (informative)

D.0 Example of a SCSI storage array configuration using a CREATE/MODIFY ARRAY CONFIGURATION service action

This annex is only intended to be an example of how a storage array could use the parameters received in a CREATE/MODIFY STORAGE ARRAY CONFIGURATION service action. Each storage array will have different factors that will effect the way it uses these parameters to configure volume sets and redundancy groups. These factors would include things like:

a) the number of peripheral devices within the storage array,
b) the capacity of the peripheral devices,
c) the ratio of link speed vs. media speed,
d) the ratio of access time vs. link speed, etc.

On receipt of a CREATE/MODIFY STORAGE ARRAY CONFIGURATION service action the target will examine the parameters and configure a volume set and a redundancy group using those parameters.

The contents of the NORMAL USER DATA TRANSFER SIZE field and the PERCENTAGE OF SEQUENTIAL TRANSFERS field in combination are used by the target to determine the user data stripe depth mapping.

See table D.1 for an example of user data stripe depths the target in this example selects based on the contents of the NORMAL USER DATA TRANSFER SIZE field and PERCENTAGE OF SEQUENTIAL READ TRANSFERS field. In this example the PERCENTAGE OF SEQUENTIAL WRITE TRANSFERS field is ignored.

Table D.1 - User data stripe depth mapping selection

		PERCENTAGE OF SEQUENTIAL READ TRANSFERS			
		1% to 25%	25% to 50% or 0%	51% to 75%	greater than 75%
NORMAL USER DATA TRANSFER SIZE in logical blocks	1 to 128	1	2		3
	129 to 512 or 0	2		4	
	greater than 512	3	4		5

Notes:
1) user data stripe depth = 4 x normal user data transfer size
2) user data stripe depth = 2 x normal user data transfer size
3) user data stripe depth = normal user data transfer size
4) user data stripe depth = normal user data transfer size / (number of disk drives/2)
5) user data stripe depth = normal user data transfer size / number of disk drives

The contents of the REBUILD/RECALCULATE PRIORITY field is used by the target to determine the how long the a rebuild or recalculate operation will take to complete.

See table D.2 for an example of how the target in this example selects the rebuild and recalculate priorities using the rebuild/recalculate priority.

.

Table D.2 - Rebuild priority selection

Codes	Description
00h	Rebuild or recalculate at least one stripe for every read or write request from an application client. (default)
01h	Suspend rebuild and recalculate operations during all read/write requests from any application clients.
02h-1Fh	Rebuild or recalculate at least 1/8th of a stripe for every read or write request from an application client.
20h-3Fh	Rebuild or recalculate at least 1/4th of a stripe for every read or write request from an application client.
40h-5Fh	Rebuild or recalculate at least 1/2th of a stripe for every read or write request from an application client.
60h-7Fh	Rebuild or recalculate at least one stripe for every read or write request from an application client.
80h-9Fh	Rebuild or recalculate at least two stripes for every read or write request from an application client.
A0h-BFh	Rebuild or recalculate at least four stripes for every read or write request from an application client.
C0h-DFh	Rebuild or recalculate at least eight stripes for every read or write request from an application client.
E0h-FEh	Rebuild or recalculate at least 16 stripes for every read or write request from an application client.
FFh	Do not accept any read/write requests from an application client until the rebuild or recalculate operation is complete.

Note:
If the redundancy group is configured as copy redundancy or S redundancy the target in this example will rebuild in 1 MByte stripes.

The contents of the NORMAL USER DATA TRANSFER SIZE field and the PERCENTAGE OF SEQUENTIAL READ TRANSFERS field in combination are used by the target to determine the amount of read ahead information to transfer from the disk drives.

See table D.3 for an example of amount of read ahead the target in this example selects based on the contents of the NORMAL USER DATA TRANSFER SIZE field and PERCENTAGE OF SEQUENTIAL READ TRANSFERS field.

Table D.3 - Read ahead selection

		PERCENTAGE OF SEQUENTIAL READ TRANSFERS			
		1% to 25%	25% to 50% or 0%	51% to 75%	greater than 75%
NORMAL USER DATA TRANSFER SIZE in logical blocks	1 to 128	1	2		3
	129 to 512 or 0	1	4		2
	greater than 512	1	5		6

Notes:
1) number of logical blocks to read ahead = no read ahead
2) number of logical blocks to read ahead = 2 x normal user data transfer size
3) number of logical blocks to read ahead = 4 x normal user data transfer size
4) number of logical blocks to read ahead = normal user data transfer size
5) number of logical blocks to read ahead = normal user data transfer size / 2
6) number of logical blocks to read ahead = normal user data transfer size up to max cache size

D.0.1 Example

A SCSI storage array is connected to an application client that runs applications that normally transfer 4 Kbytes of data at a time with sequential user data read requests occurring 60% of the time. In this example the application client issues a CREATE/MODIFY STORAGE ARRAY CONFIGURATION service action to the SCSI storage array with the NORMAL USER DATA TRANSFER SIZE field set to 8 blocks (each block is 512 bytes) and the PERCENTAGE OF SEQUENTIAL READ TRANSFERS field set to 60.

The SCSI storage array will create a volume set with a user data mapping as shown in figure D.1:

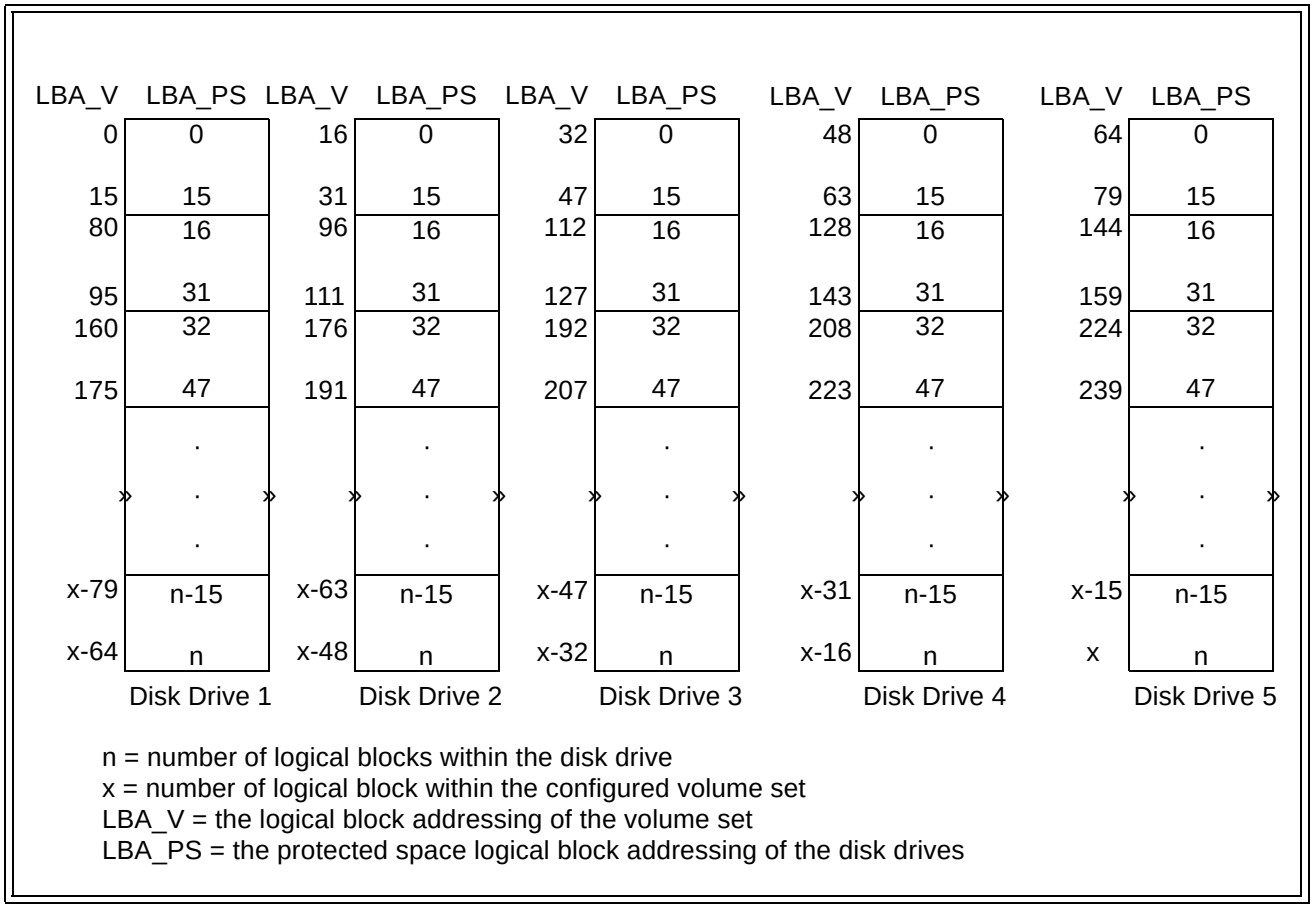

Figure D.1 - Mapping of user data in volume set example

If, for example, the application client sends a read request for a 4 Kbyte transfer starting with LBA_V 112 then the storage subsystem would read LBA_PSes 16 through 23 from disk drive 3 and transfer that information to the application client. The SCSI storage array would also read ahead LBA_PSes 24 through 31 from disk drive 3 and LBA_PSes 16 through 23 from disk drive 4. The read ahead information is placed into the SCSI storage arrays' cache in anticipation that the next read from the application client will request that information.