

5.2 Status

SAM-2

The status codes are specified in table 12. Status shall be sent from the logical unit to the application client whenever a command ends with a service response of TASK COMPLETE OR LINKED COMMAND COMPLETE. The receipt of any status, except INTERMEDIATE or INTERMEDIATE-CONDITION MET, shall indicate that the associated task has ended.

Table 12 — Status codes

Status Code Status

0h	GOOD
2h	CHECK CONDITION
4h	CONDITION MET
8h	BUSY
10h	INTERMEDIATE
14h	INTERMEDIATE-CONDITION MET
18h	RESERVATION CONFLICT
22h	Obsolete
28h	TASK SET FULL
30h	ACA ACTIVE
40h	TASK ABORTED
All other codes	Reserved

.....
.....

ACA ACTIVE. This status shall be returned when an auto contingent allegiance exists within a task set and an initiator issues a command for that task set when at least one of the following is true:

- a) There is a task with the ACA attribute in the task set;
- b) The initiator issuing the command did not cause the ACA condition;
- c) The task created to execute the command did not have the ACA attribute and the NACA bit was set to one in the CDB CONTROL byte of the faulting command (see 5.6.1).

The initiator may reissue the command after the ACA condition has been cleared.

TASK ABORTED. This status shall be returned when a task is aborted by another initiator and the Control Mode page bit TAS is one (see 5.6.x).

5.4 Task and command lifetimes

This clause specifies the events delimiting the beginning and end of a task or pending SCSI command from the viewpoint of the device server and application client. The device server shall create a task upon receiving an SCSI Command Received indication unless the command represents a continuation of a linked command as described in clause 5.

The task shall exist until:

- a) The device server sends a protocol service response for the task of TASK COMPLETE;
- b) The task is aborted as described in 5.5.

Insert new clause following 5.4 with primary content taken from the current clause 7.4. Also remove 7.4 and change the one reference to 7.4 (in 7.3) from:

task abort: One of the events described in 7.4 has occurred.

to:

task abort: A task has been aborted as described in 5.5.

5.5 Aborting tasks

5.5.1 What causes a task to be aborted

A task is aborted when an event or initiator action causes termination of the task prior to its normal successful completion.

The following events cause a task or several tasks to be aborted:

- a) The return of an Execute Command service response of SERVICE DELIVERY OR TARGET FAILURE as described in clause 5;
- b) A power on condition; or
- c) Protocol specific events.

The action of an initiator may abort task(s) created by the initiator itself or task(s) created by another initiator or both its own tasks and other initiator(s) task(s).

The following initiator actions affect only the task(s) created by the initiator that takes the action:

- a) Completion of an ABORT TASK task management function directed to the specified task;
- b) Completion of an ABORT TASK SET task management function under the conditions specified in 6.2;
- c) An ACA or CA condition was cleared and the QERR field was set to 11b in the control mode page (see the SPC-2 standard); or
- d) An ACA condition was cleared and the task had the ACA attribute.

The following initiator actions affect the task(s) created by the initiator that takes the action and/or task(s) created by another initiator:

- a) Completion of a CLEAR TASK SET task management function referencing the task set containing the specified task;
- b) An ACA or CA condition was cleared and the QERR field was set to 01b in the control mode page (see the SPC-2 standard); and

- c) Completion of a PERSISTENT RESERVE OUT command with a PREEMPT AND CLEAR service action directed to the initiator that created the task;
- d) A logical unit reset (see 5.6.7); or
- e) A target reset (see 5.6.6).

5.5.2 When an initiator aborts its own tasks

When an initiator acts to cause its own task(s) to be aborted, no notification that the task(s) have been aborted shall be returned to the initiator other than the completion response for the command or task management function action that caused the task(s) to be aborted and notification(s) associated with related effects of the action (e.g., a target reset unit attention condition).

5.5.3 When an initiator aborts another initiator's tasks

When an initiator acts to cause the task(s) of another initiator to be aborted, the other initiator shall be notified that the task(s) have been aborted. The method of notifying the other initiator shall depend on the setting of the TAS bit in the Control mode page that applies to the other initiator.

If the TAS bit is zero, the method of notification shall be a unit attention condition. The additional sense code set for the unit attention condition depends on the action that caused the task(s) to be aborted.

If the TAS bit is one, the method of notification shall be the termination of each aborted task with a TASK ABORTED status. When the TAS bit is one, the COMMANDS CLEARED BY ANOTHER INITIATOR unit attention condition shall not be established, however, the establishment of any other applicable unit attention condition shall not be affected.

When a device server is completing one or more tasks from an initiator with the TASK ABORTED status it should complete all of those tasks before new tasks from that initiator are entered into the task set.

5.6.7 Logical Unit reset

A logical unit reset is a response to a LOGICAL UNIT RESET task management request (see 6.5), or some other logical unit reset event, such as a target hard reset (see 5.6.6). The definition of such events may be device-specific or dependent on the protocol and interconnect. Each appropriate SCSI standard shall specify the conditions under which a logical unit reset shall be executed.

To execute a logical unit reset the logical unit shall:

- a) Abort all tasks in its task set(s) as described in 5.5.
- b) Clear an auto contingent allegiance ($NACA=1$, see 5.1.2) or contingent allegiance ($NACA=0$) condition, if one is present;
- c) Release all reservations established using the reserve/release management method (persistent reservations shall not be affected);
- d) Return the device's operating mode to the appropriate initial conditions, similar to those conditions that would be found following device power-on. The MODE SELECT parameters (see the SPC-2 standard) shall be restored to their last saved values if saved values have been established. MODE SELECT parameters for which no saved values have been established shall be returned to their default values;.
- e) Set a unit attention condition (see 5.6.5); and
- f) Initiate a logical unit reset for all dependent logical units (see 4.10.4).

In addition to the above, the logical unit shall execute any additional functions required by the applicable standards.

6 Task Management Functions

...

CLEAR TASK SET (Logical Unit Identifier ||) - Abort all tasks in the specified task set [as described in 6.4](#). This function shall be supported by all logical units, except in the following cases, when support for this function is optional:

- a) The logical unit does not support tagged tasks (see 4.9); or
- b) The logical unit supports the basic task management model (see 7.2).

LOGICAL UNIT RESET (Logical Unit Identifier ||) - Perform a logical unit reset as described in 5.6.7 by [aborting](#) all tasks in the task set(s) and propagating the reset to all dependent logical units (see 3.1.22). Support for this function is mandatory for hierarchical logical units (see 4.10.4) and may be supported by non-hierarchical logical units.

TARGET RESET (Target Identifier ||) - Reset the target device and [abort](#) all tasks in all task sets ([see 5.6.6](#)). All target devices shall support this function.

Argument descriptions:

- Target Identifier:** Target device identifier defined in 4.7.2.
- Logical Unit Identifier:** Logical Unit identifier defined in 4.8.
- Task Address:** Address address defined in 4.9.3.

NOTE 11 The TARGET RESET, CLEAR TASK SET, ABORT TASK and ABORT TASK SET functions provide a means to [abort](#) one or more tasks prior to normal completion.

All SCSI protocol standards shall provide the functionality needed for a task manager to implement all of the task management functions defined above.

6.2 ABORT TASK SET

Function Call:

Service Response = ABORT TASK SET (Logical Unit Identifier ||)

Description:

This function shall be supported by all logical units.

The task manager shall [abort](#) all tasks in the task set which were created by the initiator [as described in 5.5](#).

The task manager shall perform an action equivalent to receiving a series of ABORT TASK requests. All tasks from that initiator in the task set serviced by the logical unit shall be aborted. Tasks from other initiators or in other task sets shall not be [aborted](#). Previously established conditions, including MODE SELECT parameters, reservations, and auto contingent allegiance shall not be changed by the ABORT TASK SET function. A contingent allegiance (NACA=0) shall be cleared by the ABORT TASK SET function.

6.4 CLEAR TASK SET

Function Call:

Service response = CLEAR TASK SET (Logical Unit Identifier ||)

Description:

This function shall be supported by all logical units that support tagged tasks (see 4.9) and may be supported by logical units that do not support tagged tasks.

~~[Deleted paragraph]~~

All tasks in the appropriate task set [as defined by the TST field in the Control mode page \(see SPC-2\)](#) shall be aborted [as described in 5.5](#). The medium may have been altered by partially executed commands. All pending status and sense data for the appropriate task set shall be cleared.

~~[Deleted paragraphs]~~

Previously established conditions, including MODE SELECT parameters, reservations, and auto contingent allegiance ($NACA=1$, see 5.1.2) shall not be changed by the CLEAR TASK SET function. A contingent allegiance ($NACA=0$) shall be cleared by the CLEAR TASK SET function.

5.5.3.6.4 Preempting an existing persistent reservation with the PREEMPT AND ABORT service action

The initiator's request for and the device server's responses to a PERSISTENT RESERVE OUT command PREEMPT AND ABORT service action are identical to the PREEMPT service action (see 5.5.3.6.3) except for the following additions. If no reservation conflict occurred, the device server shall perform the following uninterrupted series of actions:

- a) Perform the uninterrupted series of actions described for the PREEMPT service action (see 5.5.3.6.3);
- b) All tasks from preempted initiators (called preempted tasks) shall be terminated and initiator notification shall be provided as specified by the TAS bit in the Control mode page that applies to the preempted initiator as follows:
 - a) If the TAS bit is set to zero then all preempted tasks shall be terminated as if an ABORT TASK SET task management function had been performed by each preempted initiator; or
 - b) If the TAS bit is set to one then all preempted tasks from initiators other than the initiator that sent the PREEMPT AND ABORT service action shall be terminated with a TASK ABORTED status (see SAM-2). The preempted tasks from the initiator that sent the PREEMPT AND ABORT service action (if any) shall be terminated as if an ABORT TASK SET task management function had been performed by that initiator.

If a terminated task is a COPY or EXTENDED COPY command, all commands and data transfers generated by the command shall be terminated before the ABORT CLEAR TASK SET task management function is considered completed. After the ABORT CLEAR TASK SET function has completed, all new tasks are subject to the persistent reservation restrictions established by the preempting initiator;

- c) The device server shall clear any ACA or CA condition associated with an initiator being preempted and shall clear any tasks with an ACA attribute from that initiator. If $\tau_{ST}=000b$ (see 8.3.6) and ACA or CA conditions exist for initiators other than the initiator being preempted, the PERSISTENT RESERVE OUT command shall be terminated prior to processing with a status of ACA ACTIVE if $NACA=1$ (see SAM-2) or BUSY if $NACA=0$. If $\tau_{ST}=001b$, then ACA or CA conditions for initiators other than the initiator being preempted shall not prevent the execution of the PERSISTENT RESERVE OUT command; and
- d) For SCSI devices that implement the PREVENT ALLOW MEDIUM REMOVAL command, the device server shall perform an action equivalent to the execution of a PREVENT ALLOW MEDIUM REMOVAL command with the PREVENT field equal to zero for the initiator or initiators being preempted (see 7.15).

The actions described in the preceding list shall be performed for all initiators that are registered with the SERVICE ACTION RESERVATION KEY value, without regard for whether the preempted initiator(s) hold the reservation.

Any asynchronous event reporting operations in progress are not affected by the PREEMPT AND ABORT service action.

Byte 4, bit 7 was reserved.

Table 161 — Control mode page

Bit Byte	7	6	5	4	3	2	1	0	
0	PS	Reserved	PAGE CODE (0Ah)						
1	PAGE LENGTH (0Ah)								
2	TAS	TST	Reserved			GLTSD	RLEC		
3	QUEUE ALGORITHM MODIFIER			Reserved	QERR		DQUE		
4	Reserved	RAC	Reserved		SWP	RAERP	UAAERP	EAERP	
5	Reserved				AUTOLOAD MODE				
6	(MSB)		READY AER HOLDOFF PERIOD						(LSB)
7	(MSB)		BUSY TIMEOUT PERIOD						(LSB)
8	(MSB)		EXTENDED SELF-TEST COMPLETION TIME						(LSB)
9	(MSB)		EXTENDED SELF-TEST COMPLETION TIME						(LSB)
10	(MSB)		EXTENDED SELF-TEST COMPLETION TIME						(LSB)
11	(MSB)		EXTENDED SELF-TEST COMPLETION TIME						(LSB)

...

The queue error management (QERR) field specifies how the device server shall handle blocked tasks when another task receives a CHECK CONDITION status (see table 161). The task set type (see the TST field definition above) defines which tasks are blocked. If $\tau_{ST}=000b$, then all tasks from all initiators are blocked. If $\tau_{ST}=001b$, then only tasks from the initiator that receives the CHECK CONDITION status are blocked.

Table 164 — Queue error management (QERR) field

Value	Definition
00b	Blocked tasks in the task set shall resume after an ACA or CA condition is cleared (see SAM-2).
01b	All the blocked tasks in the task set shall be aborted when the CHECK CONDITION status is sent. If the TAS bit is zero , a unit attention condition (see SAM-2) shall be generated for each initiator that had blocked tasks aborted except for the initiator to which the CHECK CONDITION status was sent. The device server shall set the additional sense code to COMMANDS CLEARED BY ANOTHER INITIATOR. If the TAS bit is one all tasks blocked for initiators other than the initiator for which the CHECK CONDITION status was sent shall be completed with a TASK ABORTED status and no unit attention shall be generated.
10b	Reserved
11b	Blocked tasks in the task set belonging to the initiator to which a CHECK CONDITION status is sent shall be aborted when the status is sent.

A disable queuing (DQUE) bit of zero specifies that tagged queuing shall be enabled if the device server supports tagged queuing. A DQUE bit of one specifies that tagged queuing shall be disabled. Any queued commands received by the device server shall be aborted. The method used to abort queued commands is protocol-specific.

A task aborted status (TAS) bit of zero specifies that aborted tasks shall be terminated by the device server without any response to the initiator. A TAS bit of one specifies that tasks aborted by the actions of another initiator shall be terminated with a TASK ABORTED status (see SAM-2).

The report a check (RAC) bit provides control of reporting long busy conditions or CHECK CONDITION status. A RAC bit of one specifies that a CHECK CONDITION status should be reported rather than a long busy condition (e.g., longer than the BUSY TIMEOUT PERIOD). A RAC bit of zero specifies that long busy conditions (e.g., busy condition during auto contingent allegiance) may be reported.

Table 4 - Clearing Effects of SCSI Initiator Actions

FCP-2 SCSI Target Object	FCP-2 SCSI Initiator Action Clears Object									
	Target Power Cycle	Reset LIP(y,x) ³	LOGO ¹³ , PLOGI	ABTS w/Last Seq.	PRLI ^{8, 13} , PRLO	TPRLO ⁵	SCSI Target Reset	Clear Task Set ⁴	Abort Task Set ⁴	SCSI Logical Unit Reset ^{4, 9}
Open Tasks (FCP exchanges) Aborted All tasks for all SCSI Initiator ports with open tasks All tasks, only for port initiating action Only specified task for port initiating action	Y - -	Y - -	N Y N	N N Y	N Y N	Y - -	Y ² - -	Y ² - -	N Y ² N	Y ² - -
FCP exchange information For all SCSI Initiators Only for SCSI Initiator port initiating action	Y -	Y -	N Y	N N	N Y	Y -	<u>N</u> -	N -	N N	<u>N</u> -

14

NOTES:

1. If the NL_Port has an AL_PA different that its hard address and the NL_Port experiences a power cycle or recognizes LIP(AL_PD,AL_PS), the NL_Port shall relinquish its current AL_PA and attempt to acquire its hard ad-dress.
2. Tasks are cleared internally within the SCSI Target, but open FCP sequences shall be individually aborted by the SCSI Initiator via the ABTS_LS protocol that also has the effect of aborting the associated FCP exchange.
3. This is also known as LIP(AL_PD,AL_PS). If the destination recognizes a selective hard reset LIP where the AL_PD matches the AL_PA of the receiving NL_Port, the receiving NL_Port shall perform the behavior described in this column.
4. For multiple-LUN SCSI Targets, Clear Task Set, Abort Task Set, and SCSI Logical Unit Reset effect only the ad-dressed LUN, not the entire SCSI Target.
5. Actions listed shall be performed when the Global bit = '1'b. If the Global bit = '0'b, then the actions listed under PRLI/PRLO should be performed for the designated SCSI Initiator.
6. This includes explicit and implicit LOGO/PLOGI.
7. After clearing the preexisting UA the Target shall set initial UA condition(s) to its initial power-on value.
8. The Target shall clear the object only if "establish image pair" = 1.
9. A definition of the Logical Unit Reset Task Management function is contained in the SAM-2 document.
10. Refer to the SPC document for a definition of Persistent Reserve In/Out usage.
11. When the most recent APTPL value received by the device server is zero.
12. Mode pages may be common and shared among all initiators, or they may be unshared and unique for each initiator. See table 5 for description of proper management of mode pages.
13. LOGO and PRLO may be either implicit or explicit. Implicit fabric logout is one possible reason for implicit LOGO and PRLO as specified in FC-FS and FC-FLA.
14. If the Control Mode Page bit TAS is set to one then tasks associated with the initiators that did not send the task management function are terminated with a TASK ABORTED status.

TARGET RESET, when set to 1, performs a reset to the SCSI device as defined in SAM-2. TARGET RESET aborts all tasks for all initiators and resets all internal states of the target to their initial power on and default values as established by PRLI. A unit attention condition is created for all initiators. The FCP login state of the affected image pairs is not changed by the TARGET RESET. The TARGET RESET bit is mandatory for FCP-2.

For tasks that are aborted the method of notification to the affected initiator is dependent upon the TAS bit in the Control Mode Page that applies to the initiator (see SAM-2).

If a task is being aborted with the TASK ABORTED status and the target does not have sequence initiative for that task it may either delay the sending of the status until sequence initiative arrives or allow the task to be terminated during the ambiguous exchange handling described immediately below.

The TARGET RESET is transmitted by the initiator (exchange originator) using a new exchange. The initiator and target clear all resources that can be cleared unambiguously. Any open exchanges that are in an ambiguous state as defined in the next paragraph shall be terminated using a recovery abort by whichever port detects the ambiguous state. The ports may issue additional recovery abort operations if they are unable to determine in a simple manner whether the state of an FCP I/O operation is ambiguous.

For a target or initiator FCP_Port, an exchange is in an ambiguous state if the FCP_Port has sequence initiative and there exists an unacknowledged frame for the sequence or if the FCP_Port has transferred sequence initiative but the transfer of the initiative has not been confirmed. For a target FCP_Port, an exchange is also in an ambiguous state if the exchange exists between the target FCP_Port and an initiator other than the initiator FCP_Port that performed the TARGET RESET.

[deleted paragraph]

LOGICAL UNIT RESET, when set to 1, performs a reset to the logical unit as defined in SAM-2. LOGICAL UNIT RESET resets the logical unit and all dependent logical units as described below. The LOGICAL UNIT RESET bit is mandatory for FCP-2.

To execute a LOGICAL UNIT RESET the logical unit shall:

- 1) Abort all tasks in its task set(s). For tasks that are aborted the method of notification to the affected initiator is dependent upon the TAS bit in the Control Mode Page that applies to the initiator (see SAM-2). If a task is being aborted with the TASK ABORTED status and the target does not have sequence initiative for that task it may either delay the sending of the status until sequence initiative arrives or allow the task to be terminated during the ambiguous exchange handling described immediately below;
- 2) Clear an auto contingent allegiance (NACA = 1) or contingent allegiance (NACA = 0) condition, if one is present;
- 3) Release all SCSI non-persistent device reservations;

Return the device's operating mode to the appropriate initial conditions, similar to those conditions that would be found following device power-on. The MODE SELECT parameters (see

the SPC-2 standard) shall be restored to their last saved values if saved values have been established. MODE SELECT parameters for which no saved values have been established shall be returned to their default values;

- 4) Set a unit attention condition; and
- 5) Initiate a logical unit reset for all dependent logical units (see 4.11).

The FCP login state of the affected image pairs is not changed by the LOGICAL UNIT RESET.

The LOGICAL UNIT RESET is transmitted by the initiator (exchange originator) using a new exchange. Any open exchanges that are in an ambiguous state shall be terminated by whichever port detects the ambiguous state using a recovery abort. The ports may issue additional recovery abort operations if they are unable to determine in a simple manner whether the state of an FCP I/O operation is ambiguous.

For a target or initiator FCP_Port, an exchange is in an ambiguous state if the FCP_Port has sequence initiative and there exists an unacknowledged frame for the sequence or if the FCP_Port has transferred sequence initiative but the transfer of the initiative has not been confirmed. For a target FCP_Port, an exchange is also in an ambiguous state if the exchange exists between the target FCP_Port and an initiator other than the initiator FCP_Port that performed the TARGET RESET.

~~[deleted paragraph]~~

CLEAR TASK SET causes all tasks from all initiators in the specified task set to be aborted as defined in SAM-2. For tasks that are aborted the method of notification to the affected initiator is dependent upon the TAS bit in the Control Mode Page that applies to the initiator. That notification may include a unit attention condition for initiators that did not send the CLEAR TASK SET task management function (see SAM-2). If a task is being aborted with the TASK ABORTED status and the target does not have sequence initiative for that task it may either delay the sending of the status until sequence initiative arrives or allow the task to be terminated during the ambiguous exchange handling described immediately below. The CLEAR TASK SET bit is mandatory for FCP-2.

The CLEAR TASK SET is transmitted by the initiator (exchange originator) using a new exchange. The initiator and target clear any resources that can be cleared unambiguously. Any open exchanges that are in an ambiguous state shall be terminated by whichever port detects the ambiguous state using a recovery abort. The ports may issue additional recovery abort operations if they are unable to determine in a simple manner whether the state of an FCP I/O operation is ambiguous.

For a target or initiator FCP_Port, an exchange is in an ambiguous state if the FCP_Port has sequence initiative and there exists an unacknowledged frame for the sequence or if the FCP_Port has transferred sequence initiative but the transfer of the initiative has not been confirmed. For a target FCP_Port, an exchange is also in an ambiguous state if the exchange exists between the target FCP_Port and an initiator other than the initiator FCP_Port that performed the CLEAR TASK SET.

~~[deleted paragraph]~~