5.2 Status

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 (see 7.4) and the Control Mode page bit TAS is set to true.

5.2.1 Status precedence

If more than one condition applies to a completed task, the report of a BUSY, RESERVATION CONFLICT, ACA….

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 7.4.

b) A power on condition occurs;

c) The logical unit executes a logical unit reset operation as described in 5.6.7;

d) The task manager executes an ABORT TASK referencing the specified task; or

e) The task manager executes an ABORT TASK SET or a CLEAR TASK SET task management function directed to the task set containing the specified task.

An SCSI command is pending when the associated SCSI Command Received indication is passed to the device server. The command ends on the occurrence of one of the conditions described above or when the device server sends a service response for the task of LINKED COMMAND COMPLETE.

The application client assumes that the task exists from the time the **Send SCSI Command** protocol service request is invoked until it receives one of the following target responses:

a) A service response of TASK COMPLETE for that task;

b) Notification of a unit attention condition with one of the following additional sense codes:

a) COMMANDS CLEARED BY ANOTHER INITIATOR (if in reference to the task set containing the task);

b) POWER ON;

c) RESET; or

d) TARGET RESET.

c) A service response of SERVICE DELIVERY OR TARGET FAILURE for the command. In this case, system implementations shall guarantee that the task associated with the failed command has ended;

Note xx: The following three responses indicate the referenced task(s) no longer exist only if the Control Mode page bit TAS is set to false.

d) A service response of FUNCTION COMPLETE following an ABORT TASK task management request directed to the specified task;

e) A service response of FUNCTION COMPLETE following an ABORT TASK SET or a CLEAR TASK SET task management function directed to the task set containing the specified task; or

f) A service response of FUNCTION COMPLETE in response to a TARGET RESET.

5.6.x Aborting Tasks (between 5.6.5 & 5.6.6)

The behavior of a task that is aborted is dependent upon the setting of the TAS bit in the Control Mode page. If the TAS bit it false then no further responses from the task are sent to the initiator. If the TAS bit it true then the task shall be terminated with a TASK ABORTED status.

When a device server is processing a task abort event (see 7.4) and the TAS bit in the Control Mode page is true, the following rules apply:

♦ There is no requirement on the ordering between the FUNCTION COMPLETE response to the task management function and the TASK ABORTED status of the tasks except as defined by each SCSI protocol.

♦ All tasks for an initiator that are aborted shall be terminated with a TASK ABORTED status before processing any tasks that arrived at the device server after the task abort event.

LOGICAL UNIT RESET (Logical Unit Identifier ||) - Perform a logical unit reset as described in 5.6.7 by ~~terminating~~ 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 ~~terminate~~ abort all tasks in all task sets. 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 ~~terminate~~ 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.

6.1 ABORT TASK

Function call:

Service Response = ABORT TASK (Task Address ||)

Description:

This function shall be supported by a logical unit that supports tagged tasks and may be supported by a logical unit that does not support tagged tasks.

The task manager shall abort the specified task if it exists. Previously established conditions, including MODE SELECT parameters, reservations, and auto contingent allegiance shall not be changed by the ABORT TASK function.

be If the logical unit supports this function, a response of FUNCTION COMPLETE shall indicate that the task ~~was~~ will be aborted or was not in the task set. ~~In either case,~~ If the Control Mode Page bit TAS is set to false, the target shall guarantee that no further responses from the task are sent to the initiator. If the Control Mode Page bit TAS is set to true, the target shall complete the task with a status of TASK ABORTED.

Note: this requires a change to FCP ABORT TASK – namely if the Task Aborted Status bit is true, the target shall not use the LS bit in the BA_ACC to the ABTS, rather send the status of TASK ABORTED

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 ~~terminate~~ abort all tasks in the task set which were created by the initiator.

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 ~~terminated~~ 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.

If the TST field equals 000b in the Control mode page (see SPC-2), the target shall perform an action equivalent to receiving a series of ABORT TASK requests from each initiator. If the TST field equals 001b the target shall perform an action equivalent to receiving a series of ABORT TASK requests from only the requesting initiator.

All tasks in the appropriate task set shall be aborted. The medium may have been altered by partially executed commands. All pending status and sense data for the appropriate task set shall be cleared.

If the Control Mode Page bit TAS is set to false, n~~No~~ status shall be sent for any task~~.~~ ~~A~~, and a unit attention condition shall be generated for all other initiators with aborted tasks (if any). When reporting the unit attention condition the additional sense code shall be set to COMMANDS CLEARED BY ANOTHER INITIATOR. If the Control Mode Page bit TAS is set to true, no unit attention conditions shall be generated because of aborted tasks.

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.

7.4 Task Abort events

A Task Abort event is one of the following:

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) Completion of a CLEAR TASK SET task management function referencing the task set containing the specified task;

d) Completion of a PERSISTENT RESERVE with a Preempt and Clear action directed to the specified task;

e) An ACA or CA condition was cleared and the Q_{ERR} field was set to 01b or 11b in the control mode page (see the SPC-2 standard);

f) An ACA condition was cleared and the task had the ACA attribute;

g) A logical unit reset (see 5.6.7);

h) The return of an **Execute Command** service response of SERVICE DELIVERY OR TARGET FAILURE as described in clause 5; or

i) A power on condition.

h) and i) don't fit my definition of
"Aborted" real well. Suggestions??

Table 158 — Control mode page

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

Byte 4, bit 7 was reserved.

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 TST=000b, then all tasks from all initiators are blocked. If TST=001b, then only tasks from the initiator that receives the CHECK CONDITION status are blocked.

Table 161 — 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.
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 aborted tasks shall be terminated with a TASK ABORTED status.

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.