# SCSI-3 Fault Tolerant Controller Configurations

utilizing SCC & New Event Codes

Edited by

Steve Sicola

High Availability Study Group

Document No.: Rev 2.0

October 18, 1995

## Overview

The Fault Tolerant Controller Profile contains the background information about fault tolerant controller configurations, and the basic features and assumptions about fault tolerant controller configuations. The problems with respect to industry standard interopability for fault tolerant configurations are described, which leads to slight changes deemed necessary of SCC and SCSI-3 in general. The modifications to the SCC specification as well as the addition of two new ASC/ASCq's to solve these interoperability problems follows, after which a functional description is presented for the use of SCC and SCSI-3 with fault tolerant controller configurations

## Fault Tolerant Controller Configurations

### Architectural Concepts

Fault Tolerant Controller Configurations are defined as any two or more controllers sharing access paths to a set of devices. Controllers in a configuration can be in an active LUN service simultaneously, called active-active, or in an active-standby with respect to LUN service. An active-active configuration allows any host to use either controller for a LUN access (load balancing) and fault tolerance. An active-standby configuration allow for only fault tolerance.

Concurrent access by more than one controller to the same LUN is allowed, but typically not supported by industry operating systems because of software interlock issues. Typical operating systems access a LUN behind a single controller until and if that controller fails, except for potential load balance optimizations where the operating system can move service from one controller another for performance reasons. Some controllers may articficially enforce this ownership model by only responding with a ready LUN on the controller that is 'preferred' to handle the device unless or if a controller failure occurs.

Fault tolerant controller configurations provide a redundant path to LUNs and devices in the event of failure. Failover is defined as the event in which a surviving controller takes over service responsibilities for a failed partner in the fault tolerant controller configuration. Failback is the event in which a controller returns to the fault tolerant configuration after re-initializtion or replacement after which that controller can accept service requests for LUN accesses.

Failure detection is achieved on of two ways. The first method of detection is host-based, where the hosts detect the failure. Hosts can detect the failure in one of number of methods, from command timeouts to periodic polling of each controller in the configuration. The time frame for host based controller failure detection is based upon the setting of specific command timeouts or periodic polling intervals. These times are typically vendor unique and can range from seconds to minutes.

The second failure detection mechanism lies with the controller configuration itself. The controllers have the opportunity for detection based upon the controller design or by a similar, lower level periodic polling or 'heartbeat' between controllers in the configuration.

### The Problem

The problem with today's implementations of fault tolerant controller configurations is that the configuration of and reporting on the configurations is done differently by every controller vendor. Furthermore, the failure detections from hosts and controllers is handled differently by host operating systems and controllers as well. These problems pose serious issues for interopability in the open systems environment.

In order to achieve interoperability in the open systems environment, standardization of the creation of and reporting on fault tolerant controller configurations is required. The issue of configuration simplicity as well configuration check simplicity is desired for open system operating system driver development. Furthermore, the failure detection mechanism should be standardized to be used optionally for quicker failover/failback in highly available system configurations.

The standard for creation of and reporing on fault tolerant controller configurations is within the scope of SCC. The standard for failure detection mechansim is simply two new ASC/ASCq's in SCSI-3 that can be used to identify the failover/failback event.

### SCC & SCSI-3

The SCC specification defines a model comprised of a single controller (SACL) with one or more controller components (among others). Fault Tolerant Controller configurations that are in some cases in industry are non-compliant with the SCC model. These configurations can become compliant based upon assumptions about the configuration. In fact, only one assumption actually changes how most controllers in the industry would achieve compliance. The others are basic assumptions about controller configurations that are not specified by SCC, but are present nonetheless in every fault tolerant controller configuration.

The assumptions about fault tolerant controller configurations under the SCC model are:

1. Two or more controllers sharing access paths to storage devices. The SACL within each controller in a fault tolerant configuration is logically presented as a single SACL with multiple ports. Specifically, the two controllers must present the exact same configuration (for LUNs and configured containers/devices) when SCC 'Report' commands are presented to either controller. This is key assumption for compliance. The controllers must report in a standard way from any controller in the configuration. In SCC, that method is through the Controller Base Address (LUN0) on each controller.
2. Controllers in a fault tolerant configuration communicate with each other to relate changes in state or configuration. The mechansim by which controllers communicate with each other while in a fault tolerant configuration is outside the scope of this document. Most controllers today communicate directly or indirectly about changes in state or configuration. The communication may take place directly via a communication path between each controller in which the controllers actively communication changes. The communication may take place indirectly through stored changes on attached devices. This assumption allow assumption (1) validity.
4. Controllers will include those with single or multiple host interfaces, and single or multiple shared device interfaces.
5. Controller may be pre-configured or configured from the attached host as a fault tolerant configuration. Configurations are verified during controller initialization as well as after initial configuration.
6. Any/all surviving controller within the configuration can assume the service of storage from the failed controller.

In order for controllers to be in a fault tolerant configuration, these assumptions are logical conclusions so that hosts can easily configure and identify the fault tolerant controller configurations. However, there are several commands in the current SCC specification that do not specify how the creation of and reporting on fault tolerant controller configurations is achieved.

### SCC & SCSI-3 Changes

The Attach to Component Device will require modifications to the following SCC commands:

Attach to Component Device
Report Component Device Attachments

The Attach to Component Device command (when addressing the controller device) is proposed to have the following format, which is exactly the same format today, with the exception of denoting the LUN_C field specifically:

Table 1 - 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=0 | | | | | | | |
| 5 | LUN_C=0 | | | | | | | (LSB) |
| 6 | (MSB) LIST_LENGTH = | | | | | | | |
| 7 | Number of Controllers to | | | | | | | |
| 8 | Create an Attachment With *8 | | | | | | | |
| 9 | | | | | | | | (LSB) |
| 10 | RESERVED | | | | | | | |
| 11 | CONTROL | | | | | | | |

When an ATTACH TO COMPONENT DEVICE service action is recieved with LUN_C=0, this designates an action to attach controller components together in a fault tolerant controller configuration, sharing access paths to configured devices within this fault tolerant configuraton. All configuration state is mutual between attached controllers.

The LUN_C field of 0 specifies that the controllers named in the parameter list shall be attached together into the controller configuration.

When a new controller is to be added to a configuration, or when a controller is to be deleted from a configuration, the ATTACH COMPONENT DEVICE service action shall be employed with a new list of controllers. Any controller that has failed, is still part of the configuration until replaced, thereafter requiring another ATTACH COMPONENT DEVICE service action, some type of automatic replacement by the controller, or a manual intervention to one controller in the configuration to update the configuration.

Table 2 - ATTACH COMPONENT DEVICE (LUN_C=0) parameter List

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | Controller Component World Wide Name | | | | | | | |
| 0 | Controller Component World Wide Name 1 | | | | | | | |
| 7 | | | | | | | | |
| ... | | | | | | | | |
| n-7 | Controller Component Name x = n/8 | | | | | | | |
| n | | | | | | | | |
| | | | | | | | | |

The parameter list contains a set of Controller Component World Wide names, each 8 bytes in length

The result of this command, if successful, will be the controllers specified being attached. A name of the resulting configuration will be determined by the controller receiving the ATTACH TO COMPONENT DEVICE COMMAND. This name shall be reported when the REPORT COMPONENT DEVICE ATTACHMENTS service action is invoked.

The reporting of controller attachments and therefore the fault tolerant configuration is achieved with the REPORT COMPONENT DEVICE ATTACHMENTS service action. The command received by a controller with the LUN_C field of 0h will report specific information about the attachments that this controller has in effect with other controllers, the name of this attachment, and the potential controllers that could be attached. This implies the controllers can communicate, otherwise the attachment would not be possible.

The REPORT COMPONENT DEVICE ATTACHMENTS service action requires the following changes:

Table 2 - REPORT COMPONENT DEVICE ATTACHMENTS (LUN_C=0) Service Action

| Bit<br>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=0 | | | | | | | |
| 5 | | | | | | | | (LSB) |
| 6 | (MSB) | | | | | | | |
| 7 | Allocation Length | | | | | | | |
| 8 | | | | | | | | |
| 9 | | | | | | | | (LSB) |
| 10 | RESERVED | | | | | | | RPTSEL |
| 11 | CONTROL | | | | | | | |

When the LUN_C field is zero, then the rest of the data in the Report Component Attachment command are ignored. The controller being queried will report:

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | Controller Component Name | | | | | | | |
| 0 | (MSB) Controller Attachment Name (CAN) | | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | Number of Controllers in Attachment (x) | | | | | | | |
| 9 | (MSB) Controller 1 World Wide Name | | | | | | | |
| 17 | | | | | | | | (LSB) |
| ... | | | | | | | | |
| z=17+<br>x*8 | Eligible Controller Attachment List Length(y) | | | | | | | |
| y+Z | (MSB) Eligible Controller 1 World Wide Name | | | | | | | |
| y+Z+7 | | | | | | | | (LSB) |
| ... | | | | | | | | |
| y+z*8 | | | | | | | | |

The parameter list contains the name of the attachment as well as theworld wide names of the controllers currently part of the attachment as well as those eligible for attachment into this configuration.

The unique identification of a Controller Configuration is required. It does not need to be a World-wide type name, rather a ‘system-wide’ unique name that can cover the configuration across the hosts to which it is attached. This name will survive any controller failures and replacements, so as to not rely upon the serial number of the controller or any packaging specific addresses. Hosts will ‘know’ about the Controller configuration by name, simplifying any mapping of access paths to devices and fault tolerance in general. The model used for the Controller Attachment is that of the FibreChannel model, utilizing an 8 bit Vendor Unique field as the most significant byte and the last 7 bytes assigned by the controller during configuration.

### SCSI-3 Changes

The event codes would be used with exceptions and be returned following the conventions of the Exceptions Mode page. These events would be reported by one ore more of the controllers in the fault tolerant controller configuration. The choice of which controller in a multi-controller fault tolerant configuration is outside the scope of this profile because the mechanisms to allow choice are here with the use of the new event codes, coupled with some of the persistent reserve concepts and other SCSI-3 facilities.

The specific ASC/ASCq event codes are:

FAILOVER - tbd code

FAILBACK - tbd code

The use of these codes is optional by the controller. Controllers that utilize these ASC/ASCq’s and host operating system drivers that recognize these event codes can react to the failover or failback of a controller in a configuration in a proactive, performance oriented way, rather than in the command error recovery path after a command timeout has occurred on the failing controller.

### Benefits of Change to Industry

The benefits of the changes to SCC and the new ASC/ASCq’s are:

1. In a multi-host, multi-controller environment where the controllers may or may not share access to storage, these changes will provide for easy configuration checks by Operating System Drivers during initialization or during normal operations. The fact that SCC supports this function will further ease in interoperability in the open system environment, with varying operating systems running on various host systems in open networks.
2. In any highly available system environment, the use of (1) above and the new ASC/ASCq's will provide for much faster failover (recovery from controller failure by a partner controller sharing access to storage) than would normally be provided by timeouts. The ASC/ASCq combination provides for quick notification of failure from a surviving partner controller .
3. In a multi-host, multi-interconnect environment, the changes to SCC will afford host a much easier, standard method for identifying opportunities for load balancing storage service across controllers in a redundant controller configuration that provides percieved multi-port, single controller support with fault tolerance. These changes are consistent and complimentary to the proposed Persistent Reserve changes (Snively).

## Functional Description

Utilizing SCC, the Attach Component Device command may be used by host to create an attachment between controllers to create or add to a fault tolerant controller configuration. The configuration may also be setup by other means available to the controllers (external user interfaces) in which case the use of Create Attachment command is unnecessary.

In order for host computers to recognize fault tolerant controller configurations, the Report Component Device Attachment command must be used to interrogate controllers to determine that multiple paths do exist to the same storage devices.

The controllers must also share the use of LUN0 on every controller in the same fault tolerant controller configuration. LUN0 on each controller will report the same configuration. This keeps the subsystem consistent. Furthermore, any host or external user interface configurations entered on one controller MUST also be relayed immediately to all other controllers in the same fault tolerant controller configuration.

The controller attachment is named for use in systems where controllers may come and go from the fault tolerant configuration due to reconfigurations, failures, and upgrades. The naming must cover the needs of the overall system the configuration is attached to. The naming must handle a change in membership, either from an addition to the configuration, a deletion from the configuration, or from a replacement in configuration (after failure). The name must be unique with a system installation, but not necessarily world-wide unique. The name must essentially be a controller configuration 'handle' that can be used by any host operation system to key off of in order to handle multiple paths to the same devices or LUNs.

A normally functioning fault tolerant controller configuration consisting of two or more controller devices acting as one SACL, will react to one of the partner controllers failure in the following way:

1. One or more controllers will detect the failure of a partner controller.
2. If the ASC/ASCqs are used, then the detecting controller(s) will respond with them after the completion of the their current command from an attached host. The surviving controllers will also be alerted to this fact by means of their direct or indirect communication path between controllers.
3. The controllers will decide which controller will take over the failed controller's LUN service, or may wait for host that utilize reserve and release features to move the LUN service to one or more of the surviving controller devices.
4. The failed controller device will still show up in the report component device attachment, but of course will not respond to any host requests.

A functioning fault tolerant controller configuration that has a partner controller either restarted or replaced after failure, will react to this event in the following way:

1. The controller(s) in the configuration may automatically actively incorporate the restarted/replaced controller device in the fault tolerant configuration.
2. The controller(s) in the configuation may be directed to incorporate the restarted/replaced controller device in the fault tolerant configuration by a local interface or via SCC over the host interconnect.
3. The restarted/replaced controller will then be ready for LUN service after verification of the configuration to LUNs and configured containers/devices by direct or indirect communication with the other controller(s) in the fault tolerant controller configuration.
4. The hosts may be notified by one or more controller devices that the previously failed member of the configuration has returned using the optional ASC/ASCq mechanism. After this load balancing may occur from other controllers to this newly returned controller.

The slight changes to SCC have allowed for a standard sequence of operations to occur for all hosts depending on the actual configuration of the controllers as well as their model for error detection, failover, and failback. The LUN access model is also noted above by nature of the different bullets pertaining to how LUNs are accessed and balanced for load purposes across hosts. The section below attempts to describe the LUN access models supported with these configurations.

**LUN Access Model**

The access to LUNs may be achieved in a number of ways, both outside the scope of the SCC model. The SCC model implies directly that the attached controllers must share the configuration. The SCC model does not state anything about the method in which LUNs are accessed from hosts through both controllers simultaneously, whether Reserve/Release functions are used and supported within the hosts, or whether the controller artificially constrain access to LUNs to one controller unless a failure occurs. All of these access models work under SCC. One note about artificial constraints is that the LUN currently visible but not controlled by a controller should return a status of LUN present/Not Ready.

Devices that have not been attached by a controller may show up to all controllers sharing access paths to devices. The controllers may not all be in the same configuration, but have access to the device. Any command that attaches the device to a particular control unit will therefore attach it to that controller configuration, thereby removing its visibility from other controllers in other controller configurations that just happen to share the access paths to the devices.