
Intel® Entry Storage System SS4200-E Array Recovery

These procedures are provided “as is” with no warranty whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample, strictly for the convenience of Intel customers. Customers use these procedures at their own risk. Intel will provide no support use. Intel provides no guarantee that executing these procedures will result in system array recovery.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

Revision History

Date	Version	Author(s)	Change Summary
12/19/07	1.0		Original document
3/2/2010	1.1		Added SSH and serial port access information for command line access and additional mdadm commands

Table of Contents

1	Introduction.....	4
2	SSH and Serial Port Access for command line operation.....	4
3	File System.....	7
4	Determine the disk drive failure	8
5	First Option: Use the EMC software “Recover Disks” option.....	10
6	Raid Array.....	11
7	Additional command if the previous methods are not successful	Error! Bookmark not defined.

1 Introduction

This document is intended to describe steps that may be taken to attempt to recover user data on an Intel® Entry Storage System SS4200-E that fails to mount the file system or start the raid array.

Note: These procedures may not result in array, or data recovery. The recovery requires SS4200-E Linux operating system root level command line access. If you are unfamiliar with Linux command line use, Intel recommends that you not attempt to use these procedures. A command error at root level access can cause unrecoverable damage to the file system.

2 SSH and Serial Port Access for command line operation

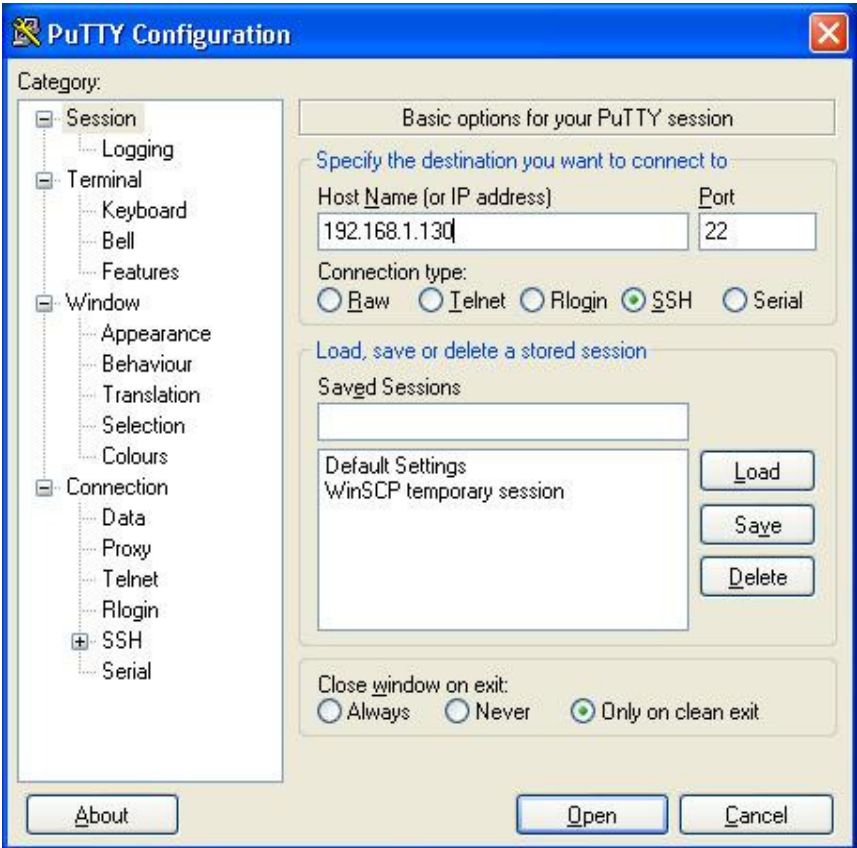
SSH:

Enable SSH access by opening the Intel® Entry Storage System manager and log in if necessary. Then select the “Settings” tab and replace settings.html?t..... in the browser address bar with support.html.

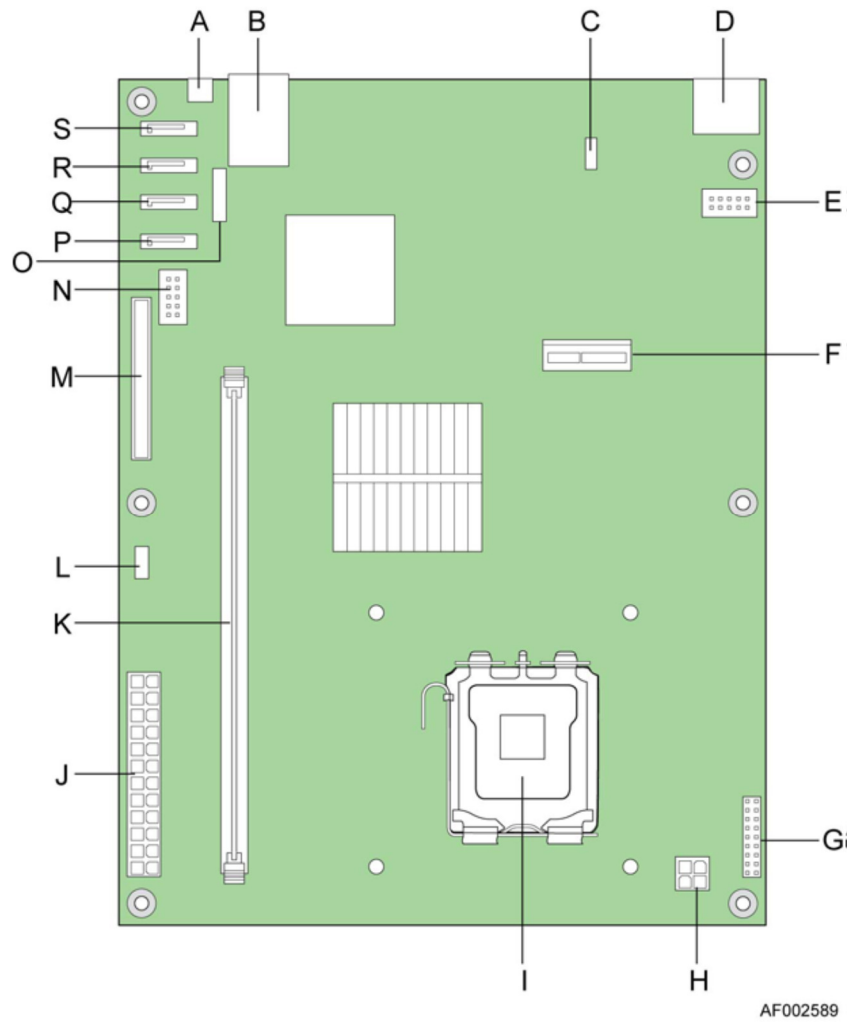
Select the support .html page “Support Access” link and then select “Allow remote access for support (SSH and SFTP)”



Use the freeware application PuTTY to access the SS4200-E via the network. Keep the defaults and enter the hostname or system IP address and press “Open”:



Serial Port:



AF002589

System Board Components

A	Reset / Recovery Button	K	DIMM slot
B	LAN Ethernet Port / USB 2.0 Ports	L	Left Chassis Fan Connector
C	Right Chassis Fan Connector	M	IDE DOM Connector
D	eSATA Ports	N	USB Header
E	RS-232 Debug Port	O	CMOS Battery
F	PCI-e Debug Port	P	SATA4 Connector
G	Front Panel Connector	Q	SATA3 Connector
H	CPU Power Connector	R	SATA2 Connector
I	CPU	S	SATA1 Connector
J	Main Power Connector		

Connector E is the SS4200-E RS-232 Serial Port Connector. Acquire a DH-10 to DB-9 serial cable. Remove the SS4200-E cover, and attach the DH-10 serial cable connector to the SS4200-E RS-232 Debug Port. Connect the DB-9 serial cable connector to the serial/COM port of your computer.

- Use standard serial communications tool, e.g. HyperTerminal or ProComm Plus
- Communication parameters:
 - 115200 Bit per second
 - 8 Data bits
 - No Parity
 - One (1) Stop bit
 - Hardware Flow control
- Log into operating system as User ID: root; Password: soho+<Web UI admin password>
 - For example, if Web UI admin password is “password”, serial port password will be sohopassword.
- **Note: You are logged in as root and changes to key system files can result in a non-working system**

These two methods will take you to a terminal window with command prompt.

3 File System

The SS4200-E storage system uses the Ext3 file system in data=ordered journaling mode. This will protect against most file system corruption issues. Corruption is still possible due to hardware errors, and software bugs.

On every mount of the file system, the Storage System software does a file system check. This will replay the data in the ext3 logs, correcting any file system issues that may have been caused by an improper shutdown. If this fails a full file system check is performed. The full check will make the file system mountable if possible. It may also recover some user data that would otherwise be lost, and place it in the lost+found directory. The lost+found directory is at /mnt/soho_storage/lost+found. If the full file system check doesn't make the file system mountable, it is likely the user data is permanently lost. However it may be desirable to attempt the file system check manually:

e2fsck -f /dev/evms/md0vol1

If this finishes successfully attempt to mount the file system:

mount /dev/evms/md0vol1 /mnt/soho_storage -t ext3

If this succeeds the file system is now mounted. It's possible that some user files were lost in the file system corruptions and recovered by the full file system check. They would be placed in /mnt/soho_storage/lost+found. If that is the case the data can be made available to the user by copying it to the public folder:

```
cp -r /mnt/soho_storage/lost+found /mnt/soho_storage/samba/shares/public
```

The files in lost+found may not be complete, or may be corrupted, but the user will now have access to some or all of their data. The Storage System device should now be rebooted to restart all services.

4 Determine the disk drive failure

In a four drive configurations the disks are designated as /dev/sda1, /dev/sdb1, /dev/sdc1 and /dev/sdd1. To see the status of a disk, run the following command:

```
mdadm --examine /dev/sdX1 (where X is the specific disk you want to see: a, b, c or d)
```

A typical response from a rebuilding system is:

```
/dev/sda1:
```

```
Magic : a92b4efc
```

```
Version : 00.90.00
```

```
UUID : deb59104:76824796:23cdfde1:5cc09783 <- UUID significance explained below
```

```
Creation Time : Wed Feb 10 15:04:08 2010
```

```
Raid Level : raid5
```

```
Used Dev Size : 976759936 (931.51 GiB 1000.20 GB)
```

```
Array Size : 2930279808 (2794.53 GiB 3000.61 GB)
```

```
Raid Devices : 4
```

```
Total Devices : 4
```

```
Preferred Minor : 0
```

```
Update Time : Tue Mar 2 12:15:45 2010
```

```
State : clean
```

```
Active Devices : 3
```

```
Working Devices : 4
```

```
Failed Devices : 1
```

```
Spare Devices : 1
```

```
Checksum : 502b1d58 - correct
```

```
Events : 0.112
```

```
Layout : left-asymmetric
```

```
Chunk Size : 32K
```

```
Number Major Minor RaidDevice State
```



```

this  0  254   0   0  active sync  /dev/evms/.nodes/sda1
      0  0  254   0   0  active sync  /dev/evms/.nodes/sda1
      1  1  254   1   1  active sync  /dev/evms/.nodes/sdb1
      2  2  254   2   2  active sync  /dev/evms/.nodes/sdc1
      3  3   0   0   3  faulty removed
      4  4  254   3   4  spare  /dev/evms/.nodes/sdd1

```

Here we see that there are Active Devices : 3, Working Devices : 4, Failed Devices : 1 and Spare Devices: 1. We also see that drive 3 has been determined "faulty removed" and drive 4 has been added as a spare. The SS4200-E will automatically add a new disk as a spare and start the rebuild. When the rebuild completes, the faulty removed disk will be replaced with the spare.

Note: Check if all working disks have identical UUID after issuing this command on all disks.

mdadm –detail /dev/md0 will provide a cleaner output while providing much of the same information:

```

/dev/md0:
Version : 00.90.03
Creation Time : Wed Feb 10 15:04:08 2010
Raid Level : raid5
Array Size : 2930279808 (2794.53 GiB 3000.61 GB)
Used Dev Size : 976759936 (931.51 GiB 1000.20 GB)
Raid Devices : 4
Total Devices : 4
Preferred Minor : 0
Persistence : Superblock is persistent

Update Time : Tue Mar  2 12:15:45 2010
State : clean, degraded, recovering
Active Devices : 3
Working Devices : 4
Failed Devices : 0
Spare Devices : 1

Layout : left-asymmetric
Chunk Size : 32K

```

Rebuild Status : 31% complete

Preferred Minor : 0

Persistence : Superblock is persistent

Update Time : Tue Mar 2 12:15:45 2010

State : clean, degraded, recovering

Active Devices : 3

Working Devices : 4

Failed Devices : 0

Spare Devices : 1

Layout : left-asymmetric

Chunk Size : 32K

Rebuild Status : 32% complete

UUID : deb59104:76824796:23cdfde1:5cc09783

Events : 0.112

Number	Major	Minor	RaidDevice	State	
0	254	0	0	active sync	/dev/evms/.nodes/sda1
1	254	1	1	active sync	/dev/evms/.nodes/sdb1
2	254	2	2	active sync	/dev/evms/.nodes/sdc1
4	254	3	3	spare rebuilding	/dev/evms/.nodes/sdd1

Here we don't see the faulty removed drive 3, just the rebuilding spare.

The --examine disk or --detail volume commands should show disks status and a failed disk if present.

5 First Option: Use the EMC software “Recover Disks” option

Boot with a failed drive removed. If that fails to make the data available, reinsert the failed drive and run the “Recover Disks” operation from the support page (assuming you can boot to the UI).

If you cannot boot to the UI, you can run an equivalent Recover Disks function from the command line.

echo "recover 1" > /tmp/dm.fifo.in where "1" in the command line after "recover" corresponds to the drive in the array (0-3) that is foreign or faulted using the Determine the disk drive failure method above.

The command will not delete or destroy data. It will only attempt to forcefully restart the array without the faulted drive, and then re-add the drive. If the array cannot be restarted the recovery will fail. If the faulted drive cannot be added the recovery will succeed but the array will be degraded until the drive is replaced.

6 Raid Array

If the existing raid array can be started and used by the SS4200-E storage system, it will be. However, if the array is corrupted by multiple drive failures it will not be usable and user data is lost. It may be possible to recover the array if 3 of the 4 original drives are available and functioning. If the previous existing array was parity (raid 5) the following command can be used to attempt to recover the array:

```
mdadm --assemble --force /dev/md0 /dev/sda1 /dev/sdb1 /dev/sdc1 /dev/sdd1
```

In the case where 1 out of 4 disks has a different UUID, and others are the same, try following command:

```
mdadm --assemble --run --force /dev/md0 /dev/sda1 /dev/sdb1 /dev/sdd1 (assume here sdc1 has the different UUID and sda1, sdb1, sdd1 are same, This command will force rebuild the md0 without the different UUID) Or if the previous array was mirror (raid 10):
```

```
mdadm --assemble --force /dev/md0 /dev/sda1 /dev/sdb1
```

```
mdadm --assemble --force /dev/md1 /dev/sdc1 /dev/sdd1
```

If this is successful the array will begin recovery. Use **cat /proc/mdstat** to monitor recovery of the array. If it does not succeed because one or more of the drives is bad, the hardware is bad, or any other reason, the user data is irretrievably lost. If it succeeds the user data may still be recovered. Reboot the device. The file system including the user data is still on the array and if not too badly corrupted will be available. The Storage System device will find the file system and check it. If it can be mounted it will be. If not you may try a manual check as described in the previous section.

It's possible the file system check succeeded and created a lost+found area. If so you want to copy the data as described in the previous section.

If the procedures outlined in this document do not restore access to your data, the issue is beyond recovery. You will need to restore data from a tested backup:

<http://www.intel.com/support/motherboards/server/sb/CS-025645.htm>

Additional mdadm resource:

<http://linuxmanpages.com/man8/mdadm.8.php>