

Western Digital Ultrastar Data60 Daisy-Chain Configuration

Table of Contents

Introduction	4
Service Mount Tables.....	5
Decoder-DB-Mount/Directory-Names Table	5
NetworkHybrid-DB-Mount/Directory-Names Table.....	6
Archiver-DB-Mount/Directory-Names Table	7
LogDecoder-DB-Mount/Directory-Names Table	7
LogHybrid-DB-Mount/Directory-Names Table	9
How to Start Configuring Ultrastar60 Storage.....	10
Configuring Daisy-Chained Decoder Storage (First & Second).....	12
Creating Ultrastar60 Block Devices (First & Second)	12
Create the Partitions and Mount Points for the New Block Devices (First & Second)	15
Updating the Decoder Database Configuration (First & Second)	22
Configuring Daisy-Chained Decoder Storage (Third & Fourth)	24
Creating Ultrastar60 Block Devices (Third & Fourth).....	25
Create the Partitions and Mount Points for the New Block Devices (Third & Fourth).....	27
Updating the Decoder Database Configuration (Third & Fourth).....	34
Configuring Daisy-Chained Log Decoder Storage (First & Second).....	36
Creating Ultrastar60 Block Devices (First & Second)	36
Create the Partitions and Mount Points for the New Block Devices (First & Second)	39
Updating the Log Decoder Database Configuration (First & Second)	46
Configuring Daisy-Chained Log Decoder Storage (Third & Fourth)	48
Creating Ultrastar60 Block Devices (Third & Fourth).....	49
Create the Partitions and Mount Points for the New Block Devices (Third and Fourth)	51
Updating the Decoder Database Configuration (Third & Fourth).....	58
Configuring Daisy-Chained Network Hybrid Storage (First & Second)	60
Creating Ultrastar60 Block Devices (First & Second)	60
Create the Partitions and Mount Points for the New Block Devices (First and Second).....	63
Updating the Network Hybrid Database Configuration (First & Second).....	68
Configuring Daisy-Chained Log Hybrid Storage (First & Second).....	70
Creating Ultrastar60 Block Devices (First & Second)	70
Create the Partitions and Mount Points for the New Block Devices (First & Second)	73

Updating the Log Hybrid Database Configuration (First & Second)	78
Configure Daisy-Chained Archiver Storage (First & Second)	80
Creating Ultrastar60 Block Devices (First & Second)	80
Create the Partitions and Mount Points for the New Block Devices (First & Second)	83
Update the Archiver Database/Collection Configuration (First & Second)	88
Configure Daisy-Chained Archiver Storage (Third & Fourth)	91
Creating Ultrastar60 Block Devices	91
Create the Partitions and Mount Points for the New Block Devices (Third & Fourth)	94
Update the Archiver Database/Collection Configuration (Third & Fourth)	99
Appendix	102
Identifying the Physical IDs, LUN IDs, and Connectors	102
Relationship between UUID/PARTLABEL and Mount Points	103
Updating Decoder Database Configuration Screenshots	103
Updating Log Decoder Database Configuration Screenshots	105
Updating Network Hybrid Database Configuration Screenshots	106
Updating Log Hybrid Database Configuration Screenshots	107

Introduction

This guide is to provide instructions for how to setup the Western Digital Ultrastar Data60 storage enclosures in a daisy-chain configuration. The following items are pre-requisites and important information to know before performing the steps in this guide. In this document, the Western Digital Ultrastar Data60 storage unit will be referred to in the following ways interchangeably: Ultrastar Data60, WD Ultrastar, and Ultrastar60.

When daisy-chaining Ultrastar60 storage units, four can be connected to Series 7 Core appliances, depending on the orchestrated NW category (see chart below). A maximum of two Ultrastar60s can be daisy-chained when connecting the storage units to Series 7 Hybrid appliances.

NW Category	Maximum Ultrastar60 Supported	Total Number of Disks (HDD)
Decoder	4	4 * 60 = 240
LogDecoder	4	4 * 60 = 240
Concentrator	Not Supported for Daisy-Chaining	1 * 60 = 60
Archiver	4	4 * 60 = 240
Network Hybrid	2	2 * 60 = 120
Log Hybrid	2	2 * 60 = 120

This document assumes that the Ultrastar60s are installed and connected following the [Western Digital Ultrastar60 User Guide](#) (Section **4.11.2 One Host Cable Configurations, Diagram Two Enclosures: One Host with a Single HBA**)

The daisy-chain configurations in this document represent two to four Ultrastar60s with 60 drives each. The drives will be configured into two block devices, each containing 30 physical drives per unit. When utilizing daisy-chaining, the partition labels and mount points MUST adhere to the specific naming convention as detailed in the [Service-Mount Tables](#) section of this document.

Always use the cables that come with the Ultrastar60 hardware (Amphenol cables). Otherwise, the Ultrastar60's will be in an unsupported configuration and will not work as expected. If any cables are missing, please contact your NetWitness Salesperson for assistance.

Daisy-chaining the Ultrastar60s does not support the use of multiple Host Bus Adapter (HBA) controllers. Only one HBA controller can be connected to the Ultrastar60s. The Series 7 appliances should show the HBA as Controller ID 1.

It is required to have the Series 7 appliance correctly orchestrated based off the NW Categories (e.g. Decoder, Archiver, Log Decoder, etc.) before configuring the Ultrastar60 storage. If the storage is configured first, the orchestration process may not configure the appliance correctly. This can lead to a Support case to fix the issue, or in some cases a full reimage of the appliance will be needed.

It is important to note that the Operating System controls the naming and UUIDs of block devices. The block device names and UUIDs within this guide may vary from what is seen within a customer's environment. This is to be expected and will require the user to replace these names/UUIDs in the commands within this document with the ones from the customer's environment.

Service Mount Tables

Decoder-DB-Mount/Directory-Names Table

This section provides the block device number, size, and mount points for each Ultrastar60 that can be connected to a Network Decoder.

First	Block Device #	PART LABEL	Mount Point
	0	decoroot	/var/netwitness/decoder
		decondex	/var/netwitness/decoder/index
		decosessiondb	/var/netwitness/decoder/sessiondb
		decometadb	/var/netwitness/decoder/metadb
		decopacketdb	/var/netwitness/decoder/packetdb
	1	decoroot	/var/netwitness/decoder/decoder0
		decondex	/var/netwitness/decoder/decoder0/index0
		decosessiondb	/var/netwitness/decoder/decoder0/sessiondb0
		decometadb	/var/netwitness/decoder/decoder0/metadb0
		decopacketdb	/var/netwitness/decoder/decoder0/packetdb0

Second	Block Device #	PART LABEL	Mount Point
	2	decoroot	/var/netwitness/decoder/decoder1
		decondex	/var/netwitness/decoder/decoder1/index1
		decosessiondb	/var/netwitness/decoder/decoder1/sessiondb1
		decometadb	/var/netwitness/decoder/decoder1/metadb1
		decopacketdb	/var/netwitness/decoder/decoder1/packetdb1
	3	decoroot	/var/netwitness/decoder/decoder2
		decondex	/var/netwitness/decoder/decoder2/index2
		decosessiondb	/var/netwitness/decoder/decoder2/sessiondb2
		decometadb	/var/netwitness/decoder/decoder2/metadb2
		decopacketdb	/var/netwitness/decoder/decoder2/packetdb2

Third	Block Device #	PART LABEL	Mount Point
	4	decoroot	/var/netwitness/decoder/decoder3
		decondex	/var/netwitness/decoder/decoder3/index3
		decosessiondb	/var/netwitness/decoder/decoder3/sessiondb3
		decometadb	/var/netwitness/decoder/decoder3/metadb3
		decopacketdb	/var/netwitness/decoder/decoder3/packetdb3
	5	decoroot	/var/netwitness/decoder/decoder4
		decondex	/var/netwitness/decoder/decoder4/index4
		decosessiondb	/var/netwitness/decoder/decoder4/sessiondb4
		decometadb	/var/netwitness/decoder/decoder4/metadb4
		decopacketdb	/var/netwitness/decoder/decoder4/packetdb4

Fourth	Block Device #	PART LABEL	Mount Point
	6	decoroot	/var/netwitness/decoder/decoder5
		deconex	/var/netwitness/decoder/decoder5/index5
		decosessiondb	/var/netwitness/decoder/decoder5/sessiondb5
		decometadb	/var/netwitness/decoder/decoder5/metadb5
		decopacketdb	/var/netwitness/decoder/decoder5/packetdb5
	7	decoroot	/var/netwitness/decoder/decoder6
		deconex	/var/netwitness/decoder/decoder6/index6
		decosessiondb	/var/netwitness/decoder/decoder6/sessiondb6
		decometadb	/var/netwitness/decoder/decoder6/metadb6
		decopacketdb	/var/netwitness/decoder/decoder6/packetdb6

NetworkHybrid-DB-Mount/Directory-Names Table

This section provides the block device number, size, and mount points for each Ultrastar60 that can be connected to a Network Hybrid.

First	Block Device #	Size (TB)	PART LABEL	Mount Point
	0	442	decopacket	/var/netwitness/decoder/packetdb0
			concessiondb	/var/netwitness/concentrator/sessiondb0
			concmetadb	/var/netwitness/concentrator/metadb0
	1	442	decopacket	/var/netwitness/decoder/packetdb1
			concessiondb	/var/netwitness/concentrator/sessiondb1
			concmetadb	/var/netwitness/concentrator/metadb1

Second	Block Device #	Size (TB)	PART LABEL	Mount Point
	2	442	decopacket	/var/netwitness/decoder/packetdb2
			concessiondb	/var/netwitness/concentrator/sessiondb2
			concmetadb	/var/netwitness/concentrator/metadb2
	3	442	decopacket	/var/netwitness/decoder/packetdb3
			concessiondb	/var/netwitness/concentrator/sessiondb3
			concmetadb	/var/netwitness/concentrator/metadb3

Archiver-DB-Mount/Directory-Names Table

This section provides the block device number, size, and mount points for each Ultrastar60 that can be connected to an Archiver.

First	Block Device #	Size (TB)	PART LABEL	Mount Point
	0	442	database	/var/netwitness/archiver/database0
	1	442	database	/var/netwitness/archiver/database1

Second	Block Device #	Size (TB)	PART LABEL	Mount Point
	2	442	database	/var/netwitness/archiver/database2
	3	442	database	/var/netwitness/archiver/database3

Third	Block Device #	Size (TB)	PART LABEL	Mount Point
	4	442	database	/var/netwitness/archiver/database4
	5	442	database	/var/netwitness/archiver/database5

Fourth	Block Device #	Size (TB)	PART LABEL	Mount Point
	6	442	database	/var/netwitness/archiver/database6
	7	442	database	/var/netwitness/archiver/database7

LogDecoder-DB-Mount/Directory-Names Table

This section provides the block device number, size, and mount points for each Ultrastar60 that can be connected to a Log Decoder.

First	Block Device #	PART LABEL	Mount Point
	0	logdecoroot	/var/netwitness/logdecoder
		logdeconex	/var/netwitness/logdecoder/index
		logdecosessiondb	/var/netwitness/logdecoder/sessiondb
		logdecometadb	/var/netwitness/logdecoder/metadb
		logdecopacketdb	/var/netwitness/logdecoder/packetdb
	1	logdecoroot	/var/netwitness/logdecoder/logdecoder0
		logdeconex	/var/netwitness/logdecoder/logdecoder0/index0
		logdecosessiondb	/var/netwitness/logdecoder/logdecoder0/sessiondb0
		logdecometadb	/var/netwitness/logdecoder/logdecoder0/metadb0
		logdecopacketdb	/var/netwitness/logdecoder/logdecoder0/packetdb0

Second	Block Device #	PART LABEL	Mount Point
	2	logdecoroot	/var/netwitness/logdecoder/logdecoder1
		logdeconex	/var/netwitness/logdecoder/logdecoder1/index1
		logdecosessiondb	/var/netwitness/logdecoder/logdecoder1/sessiondb1
		logdecometadb	/var/netwitness/logdecoder/logdecoder1/metadb1
		logdecopacketdb	/var/netwitness/logdecoder/logdecoder1/packetdb1
	3	logdecoroot	/var/netwitness/logdecoder/logdecoder2
		logdeconex	/var/netwitness/logdecoder/logdecoder2/index2
		logdecosessiondb	/var/netwitness/logdecoder/logdecoder2/sessiondb2
		logdecometadb	/var/netwitness/logdecoder/logdecoder2/metadb2
		logdecopacketdb	/var/netwitness/logdecoder/logdecoder2/packetdb2

Third	Block Device #	PART LABEL	Mount Point
	4	logdecoroot	/var/netwitness/logdecoder/logdecoder3
		logdeconex	/var/netwitness/logdecoder/logdecoder3/index3
		logdecosessiondb	/var/netwitness/logdecoder/logdecoder3/sessiondb3
		logdecometadb	/var/netwitness/logdecoder/logdecoder3/metadb3
		logdecopacketdb	/var/netwitness/logdecoder/logdecoder3/packetdb3
	5	logdecoroot	/var/netwitness/logdecoder/logdecoder4
		logdeconex	/var/netwitness/logdecoder/logdecoder4/index4
		logdecosessiondb	/var/netwitness/logdecoder/logdecoder4/sessiondb4
		logdecometadb	/var/netwitness/logdecoder/logdecoder4/metadb4
		logdecopacketdb	/var/netwitness/logdecoder/logdecoder4/packetdb4

Fourth	Block Device #	PART LABEL	Mount Point
	6	logdecoroot	/var/netwitness/logdecoder/logdecoder5
		logdeconex	/var/netwitness/logdecoder/logdecoder5/index5
		logdecosessiondb	/var/netwitness/logdecoder/logdecoder5/sessiondb5
		logdecometadb	/var/netwitness/logdecoder/logdecoder5/metadb5
		logdecopacketdb	/var/netwitness/logdecoder/logdecoder5/packetdb5
	7	logdecoroot	/var/netwitness/logdecoder/logdecoder6
		logdeconex	/var/netwitness/logdecoder/logdecoder6/index6
		logdecosessiondb	/var/netwitness/logdecoder/logdecoder6/sessiondb6
		logdecometadb	/var/netwitness/logdecoder/logdecoder6/metadb6
		logdecopacketdb	/var/netwitness/logdecoder/logdecoder6/packetdb6

LogHybrid-DB-Mount/Directory-Names Table

This section provides the block device number, size, and mount points for each Ultrastar60 that can be connected to a Network Hybrid.

First	Block Device #	Size (TB)	PART LABEL	First Ultrastar60 Mounts
	0	442	logdecopacket	/var/netwitness/logdecoder/packetdb0
			concessiondb	/var/netwitness/concentrator/sessiondb0
			concmetadb	/var/netwitness/concentrator/metadb0
	1	442	logdecopacket	/var/netwitness/logdecoder/packetdb1
			concessiondb	/var/netwitness/concentrator/sessiondb1
			concmetadb	/var/netwitness/concentrator/metadb1

Second	Block Device #	Size (TB)	PART LABEL	Second Ultrastar60 Mounts
	2	442	logdecopacket	/var/netwitness/logdecoder/packetdb2
			concessiondb	/var/netwitness/concentrator/sessiondb2
			concmetadb	/var/netwitness/concentrator/metadb2
	3	442	logdecopacket	/var/netwitness/logdecoder/packetdb3
			concessiondb	/var/netwitness/concentrator/sessiondb3
			concmetadb	/var/netwitness/concentrator/metadb3

How to Start Configuring Ultrastar60 Storage

To begin setting up the Ultrastar60 storage, certain pieces of information are required:

- Enclosure IDs
- Connector IDs
- Physical IDs
- LUN IDs

The following steps will provide the commands needed to capture this information for later use within this document.

Important: Ensure that the NetWitness service (Decoder, Log Decoder, Concentrator, Archiver, etc.) has already been orchestrated before starting any storage configuration.

The Series 7 appliances only support a single Host Bus Adapter (HBA) when daisy-chaining Ultrastar60 storage. As such, the Controller ID should be 1. The Controller ID is not dependent on the PCIe slot of the appliance and should not change.

To confirm the identity of the HBA controller ID, use the following command:

```
arcconf list|egrep 'Controller [1-9]+:'|awk '{print $2}'|cut -d : -f1
```

1. Identify the Enclosures and associated Connectors. To do this, run the following command to confirm what is connected to the appliance.

```
arcconf list 1|egrep 'Enclosure [1-9]+, Slot [0-9]+\ (Connector' | awk '{print $14,$15,$18}'|uniq
```

First Storage Unit	[Enclosure 1, 0 CN0]
Second Storage Unit	[Enclosure 1, 1 CN1]

2. Identify the Physical IDs and LUN IDs of the drives in Enclosure1 for all connected storage units (for example, Connectors CN0 and CN1). Run the following command to confirm all drives are being seen correctly. If there are any missing drives, stop and contact NetWitness Support for assistance.

```
arcconf list 1|egrep 'Enclosure 1, Slot [0-9]+\ (Connector'|awk '{print $1, $2, $14,$15,$16,$17,$18}'
```

Please see the Appendix, under [Identifying the Physical IDs, LUN IDs and Connectors](#) for example output.

3. Run the following command to filter the output to just the LUN IDs on Connector 0 (CN0/First Storage Unit). It will be important to capture the Physical ID/LUN IDs from this output. This information will be required later when creating the block devices.

```
arcconf list 1|egrep 'Enclosure 1, Slot [0-9]+\ (Connector 0'|awk '{print $1, $2, $14,$15,$16,$17,$18}'
```

```
Physical 0,0 [Enclosure 1, Slot 1 Connector 0:CN0 ]
Physical 0,1 [Enclosure 1, Slot 1 Connector 0:CN0 ]
Physical 0,2 [Enclosure 1, Slot 1 Connector 0:CN0 ]
Physical 0,3 [Enclosure 1, Slot 1 Connector 0:CN0 ]
Physical 0,4 [Enclosure 1, Slot 1 Connector 0:CN0 ]
Physical 0,5 [Enclosure 1, Slot 1 Connector 0:CN0 ]
Physical 0,6 [Enclosure 1, Slot 1 Connector 0:CN0 ]
```

Generally, the output for the first storage unit (CN0) should have a Connector ID of 0, a Physical ID of 0, and LUN IDs ranging from 0 to 59 (i.e. 0,0 0,1 0,2 ... 0,57 0,58 0,59).

If this is not what was output from the above command, stop and request assistance from NetWitness Support.

4. Run the following command to filter the output to just the LUN IDs on Connector 1 (CN1/Second Storage Unit). Just like in the previous step it will be important to capture this output for block device creation.

```
arcconf list 1|egrep 'Enclosure 1, Slot [0-9]+\ (Connector 1'|awk '{print $1, $2, $14,$15,$16,$17,$18}'
```

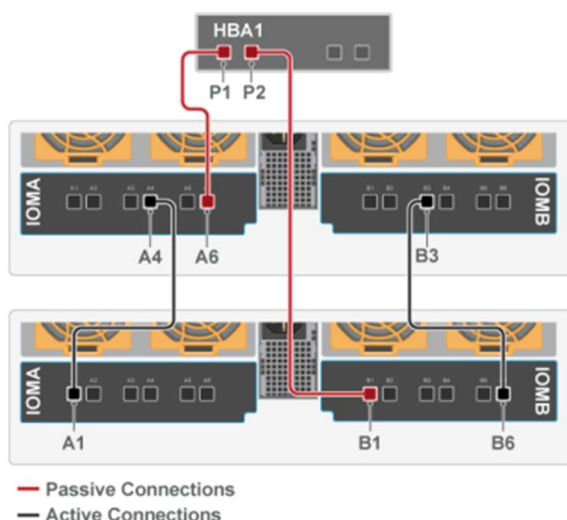
```
Physical 0,60 [Enclosure 1, Slot 1 Connector 1:CN1 ]
Physical 0,61 [Enclosure 1, Slot 1 Connector 1:CN1 ]
Physical 0,62 [Enclosure 1, Slot 1 Connector 1:CN1 ]
Physical 0,63 [Enclosure 1, Slot 1 Connector 1:CN1 ]
Physical 0,64 [Enclosure 1, Slot 1 Connector 1:CN1 ]
Physical 0,65 [Enclosure 1, Slot 1 Connector 1:CN1 ]
Physical 0,66 [Enclosure 1, Slot 1 Connector 1:CN1 ]
```

Generally, the output for the second storage unit (CN1) should have a Connector ID of 1, a Physical ID of 0, and LUN IDs ranging from 60 to 119 (e.g., 0,60 0,61 0,62 ... 0,117 0,118 0,119).

If this is not what was output from the above command, stop and request assistance from NetWitness Support.

Configuring Daisy-Chained Decoder Storage (First & Second)

Two Enclosures: One Host with a Single HBA



Source Device	Destination Device
Host1: P1	Enclosure1: IOMA, A6
Host1: P2	Enclosure2: IOMB, B1
Enclosure1: IOMA, A4	Enclosure2: IOMA, A1
Enclosure1: IOMB, B3	Enclosure2: IOMB, B6

This section provides the steps needed to attach the first and second Ultrastar60 storage unit to the Decoder. Please keep in mind that setting up the first and second storage units will be pretty much identical. NetWitness currently supports two, three, and four storage units in a daisy-chain configuration. The instructions in this section will show how to configure the first and second units at the same time.

Please review the [How to Start Configuring Ultrastar60 Storage](#) section before proceeding as the output for Enclosure, Connector, Physical ID, and LUN ID will be required. If setting up more than two Ultrastar60 storage units, make sure that only the first and second units are currently connected. This will help make sure that the correct units are being configured when using the following steps.

Creating Ultrastar60 Block Devices (First & Second)

1. Perform a block device listing on the appliance. This command looks for any devices that are of size 442 TB. This will catch any configured Ultrastar60 storage. Since all connected units are unconfigured, no block devices should be returned.

```
lsblk | grep '442T'
```

2. Retrieve the Physical ID and LUN ID output for Connector 0 (CN0/First Storage Unit) and Connector 1 (CN1/Second Storage Unit), per the [How to Start Configuring Ultrastar60 Storage](#) section.

Important: The `arconf` utility requires that the LUN IDs captured earlier must be entered in reverse order (0,59 0,58 0,57 ... 0,2 0,1 0,0), instead of normal order (0,0 0,1 0,2 ... 0,57 0,58 0,59). Pay particular attention when entering the commands of this section due to this reversal.

- Break up the 120 LUNS into four groups of 30, remembering to use reverse order.

Storage	Groups	LUN IDs
First	Group 1	0,29 0,28 0,27 ... 0,2 0,1 0,0
	Group 2	0,59 0,58 0,57 ... 0,32 0,31 0,30
Storage	Groups	LUN IDs
Second	Group 3	0,89 0,88 0,87 ... 0,62 0,61 0,60
	Group 4	0,119 0,118 0,117 ... 0,92 0,91 0,90

- Create the first RAID 50 Block Device 0 using the Group 1 LUNs, from the first storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,29 0,28 0,27 0,26 0,25 0,24 0,23 0,22 0,21 0,20 0,19 0,18 0,17 0,16 0,15
0,14 0,13 0,12 0,11 0,10 0,9 0,8 0,7 0,6 0,5 0,4 0,3 0,2 0,1 0,0 noprompt
```

- Create the second RAID 50 Block Device 1 using Group 2 LUNs, from the first storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,59 0,58 0,57 0,56 0,55 0,54 0,53 0,52 0,51 0,50 0,49 0,48 0,47 0,46 0,45
0,44 0,43 0,42 0,41 0,40 0,39 0,38 0,37 0,36 0,35 0,34 0,33 0,32 0,31 0,30
noprompt
```

- Create the third RAID 50 Block Device 2 using Group 3 LUNs, from the second storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,89 0,88 0,87 0,86 0,85 0,84 0,83 0,82 0,81 0,80 0,79 0,78 0,77 0,76 0,75
0,74 0,73 0,72 0,71 0,70 0,69 0,68 0,67 0,66 0,65 0,64 0,63 0,62 0,61 0,60
noprompt
```

- Create the fourth RAID 50 Block Device 3 using Group 4 LUNs, from the second storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,119 0,118 0,117 0,116 0,115 0,114 0,113 0,112 0,111 0,110 0,109 0,108
0,107 0,106 0,105 0,104 0,103 0,102 0,101 0,100 0,99 0,98 0,97 0,96 0,95
0,94 0,93 0,92 0,91 0,90 noprompt
```

8. Run the **lsblk** command to find the newly created Ultrastar60 block devices. The command should return four devices of 442 TB in size, similar to the output below.

```
lsblk | grep '442T'
```

First Storage	sdc	8:32	0	442T	0	disk
	sdd	8:32	0	442T	0	disk
Second Storage	sde	8:32	0	442T	0	disk
	sdf	8:32	0	442T	0	disk

Sometimes the block devices will not display immediately, this is normal. A rescan of the bus can be performed using the following command. Once it is complete, rerun the **lsblk** command above.

Create the Partitions and Mount Points for the New Block Devices (First & Second)

This section will show how to create new partitions and set up mount points using the newly created block devices on the first and second Ultrastar60 storage units.

Please use the block devices that were discovered while running the **lsblk** command in the previous section. The devices discovered should take the place of `/dev/sdc` and `/dev/sdd`, for the first storage unit. The second set of devices should take the place of `/dev/sde` and `/dev/sdf`, for the second storage unit in the following examples. The [Decoder-DB-Mount/Directory-Names Table](#) will be utilized for this section and should be referenced as needed.

If the appliance has been rebooted since the block device creation was performed, rerun the **lsblk** command again as the block device names may have changed. The operating system is responsible for device naming and can change the names between reboots.

The block device of `/dev/sdc`, `/dev/sdd`, `/dev/sde`, and `/dev/sdf` will be used in examples throughout this section.* These are only place holders for the block devices within the customer's appliance.

If there are any questions, please stop and contact NetWitness Customer Support for assistance.

1. Set the disk label type to gpt for the new block devices (`/dev/sdc`, `/dev/sdd`, `/dev/sde`, `/dev/sdf`)*.

Command:

```
parted -s <block device> mklabel gpt
```

Example:

Storage Unit	Command
First	<code>parted -s /dev/sdc mklabel gpt</code>
	<code>parted -s /dev/sdd mklabel gpt</code>
Storage Unit	Command
Second	<code>parted -s /dev/sde mklabel gpt</code>
	<code>parted -s /dev/sdf mklabel gpt</code>

2. Create the required Decoder partitions utilizing the **PART LABEL** column from the [Decoder-DB-Mount/Directory-Names Table](#). Run all the commands below for each block device (`/dev/sdc`, `/dev/sdd`, `/dev/sde`, `/dev/sdf`)* to create the required partitions.

Command:

```
parted --align minimal <block device> mkpart <part label> xfs start_size  
end_size
```

Example:

First Storage Unit	parted --align minimal /dev/sdc mkpart decoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdc mkpart decondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdc mkpart decosessiondb xfs 50GiB 1106GiB
	parted --align minimal /dev/sdc mkpart decometadb xfs 1106GiB 18514GiB
	parted --align minimal /dev/sdc mkpart decopacketdb xfs 18514GiB 100%
	parted --align minimal /dev/sdd mkpart decoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdd mkpart decondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdd mkpart decosessiondb xfs 50GiB 1106GiB
	parted --align minimal /dev/sdd mkpart decometadb xfs 1106GiB 18514GiB
	parted --align minimal /dev/sdd mkpart decopacketdb xfs 18514GiB 100%
Second Storage Unit	parted --align minimal /dev/sde mkpart decoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sde mkpart decondex xfs 20GiB 50GiB
	parted --align minimal /dev/sde mkpart decosessiondb xfs 50GiB 1106GiB
	parted --align minimal /dev/sde mkpart decometadb xfs 1106GiB 18514GiB
	parted --align minimal /dev/sde mkpart decopacketdb xfs 18514GiB 100%
	parted --align minimal /dev/sdf mkpart decoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdf mkpart decondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdf mkpart decosessiondb xfs 50GiB 1106GiB
	parted --align minimal /dev/sdf mkpart decometadb xfs 1106GiB 18514GiB
	parted --align minimal /dev/sdf mkpart decopacketdb xfs 18514GiB 100%

- Confirm that all partitions have been created correctly for all block devices (/dev/sdc, /dev/sdd, /dev/sde, /dev/sdf)* by running the parted print command for all block devices. The output will be the same for all block devices.

Command:

parted <block device> print

Example:

Storage Unit	Command
First	parted /dev/sdc print
	parted /dev/sdd print
Storage Unit	Command
Second	parted /dev/sde print
	parted /dev/sdf print

All block device outputs should look identical to the output below.

```

Model: Adaptec LOGICAL VOLUME (scsi)
Disk /dev/sdc: 486TB
Sector Size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File System  Name        Flags
  1      3146kB  21.5GB  21.5GB                decoroot
  2      21.5GB  53.7GB  32.2GB                decondex
  3      53.7GB  4948GB  4894GB                decosessiondb
  4      4948GB  18.7TB  13.7TB                decometadb
  5      18.7TB  486TB   467TB                decopacketdb

```

4. Create the file systems on all the newly created Decoder partitions. The numbers under the **Number** column from the previous step's print outputs should be appended to the block device name as shown in the commands below.

Command:

```
mkfs.xfs -f <block device><partition number>
```

Example:

Storage Unit	Block Device	Command
First	<i>/dev/sdc*</i>	mkfs.xfs -f <i>/dev/sdc1</i>
		mkfs.xfs -f <i>/dev/sdc2</i>
		mkfs.xfs -f <i>/dev/sdc3</i>
		mkfs.xfs -f <i>/dev/sdc4</i>
		mkfs.xfs -f <i>/dev/sdc5</i>
	<i>/dev/sdd*</i>	mkfs.xfs -f <i>/dev/sdd1</i>
		mkfs.xfs -f <i>/dev/sdd2</i>
		mkfs.xfs -f <i>/dev/sdd3</i>
		mkfs.xfs -f <i>/dev/sdd4</i>
		mkfs.xfs -f <i>/dev/sdd5</i>

Storage Unit	Block Device	Command
Second	<i>/dev/sde*</i>	mkfs.xfs -f <i>/dev/sde1</i>
		mkfs.xfs -f <i>/dev/sde2</i>
		mkfs.xfs -f <i>/dev/sde3</i>
		mkfs.xfs -f <i>/dev/sde4</i>
		mkfs.xfs -f <i>/dev/sde5</i>
	<i>/dev/sdf*</i>	mkfs.xfs -f <i>/dev/sdf1</i>
		mkfs.xfs -f <i>/dev/sdf2</i>
		mkfs.xfs -f <i>/dev/sdf3</i>
		mkfs.xfs -f <i>/dev/sdf4</i>
		mkfs.xfs -f <i>/dev/sdf5</i>

5. Manually mount the decoroot partition from the first block device (*/dev/sdc*)*. This partition will act as the container for all other mounted storage partitions.

Command:

```
mount <partition> <mounting location>
```

Example:

```
mount /dev/sdc1 /var/netwitness/decoder
```

6. Create the directories where the remaining decoroot partitions will be mounted.

Command:

```
mkdir /var/netwitness/decoder/decoder<#>
```

Example:

Storage Unit	Command
First	mkdir /var/netwitness/decoder/decoder0
Second	mkdir /var/netwitness/decoder/decoder1
	mkdir /var/netwitness/decoder/decoder2

7. Mount all remaining decoroot partitions to the appropriate folder locations (/dev/sdd1, /dev/sde1, /dev/sdf1)*.

Command:

```
mount <partition> <mounting location>
```

Example:

Storage Unit	Command
First	mount /dev/sdd1 /var/netwitness/decoder/decoder0
Second	mount /dev/sde1 /var/netwitness/decoder/decoder1
	mount /dev/sdf1 /var/netwitness/decoder/decoder2

- With all decoroot partitions mounted, create all other mount point folders.

Command:

```
mkdir /var/netwitness/decoder/<db>
mkdir /var/netwitness/decoder/decoder<#>/<db>
```

Example:

Storage Unit	Command
First	<pre>mkdir /var/netwitness/decoder/index mkdir /var/netwitness/decoder/sessiondb mkdir /var/netwitness/decoder/metadb mkdir /var/netwitness/decoder/packetdb</pre>
	<pre>mkdir /var/netwitness/decoder/decoder0/index0 mkdir /var/netwitness/decoder/decoder0/sessiondb0 mkdir /var/netwitness/decoder/decoder0/metadb0 mkdir /var/netwitness/decoder/decoder0/packetdb0</pre>
Storage Unit	Command
Second	<pre>mkdir /var/netwitness/decoder/decoder1/index1 mkdir /var/netwitness/decoder/decoder1/sessiondb1 mkdir /var/netwitness/decoder/decoder1/metadb1 mkdir /var/netwitness/decoder/decoder1/packetdb1</pre>
	<pre>mkdir /var/netwitness/decoder/decoder2/index2 mkdir /var/netwitness/decoder/decoder2/sessiondb2 mkdir /var/netwitness/decoder/decoder2/metadb2 mkdir /var/netwitness/decoder/decoder2/packetdb2</pre>

- Run the **blkid** command to list all the block device partitions on the first block device (*/dev/sdc*)*. Capture the output from the command as the UUIDs listed will be needed in later steps (see screenshot for example output). It is important to note which UUIDs are associated with which PARTLABELs from the output. These PARTLABELs directly correspond with the PART LABEL column from the [Decoder-DB-Mount/Directory-Names Table](#) section of this document.

```
blkid | egrep '/dev/sdc' | awk '{print $1, $2, $5}'
```

```
/dev/sdc1: UUID="1af45678-12c4-f23a-12d4-cf234537a90e2" PARTLABEL="decoroot"
/dev/sdc2: UUID="2bc45786-1c42-a2af-4d32-23ff4537a912a" PARTLABEL="decondex"
/dev/sdc3: UUID="3de45878-2c41-113a-ff21-cf23ff37a11a2" PARTLABEL="decosessiondb"
/dev/sdc4: UUID="4fa46678-c4ef-3890-00cf-d123e537770e2" PARTLABEL="decometadb"
/dev/sdc5: UUID="5ee43678-3321-efc1-652c-00924537ab76d" PARTLABEL="decopacketdb"
```

10. Run the **blkid** command again to list all the block device partitions on the second block device (*/dev/sdd*)*. Just like in the previous step capture the same output for later use.

```
blkid | egrep '/dev/sdd' | awk '{print $1, $2, $5}'
```

```
/dev/sdd1: UUID="1af45678-12c4-f23a-12d4-cf234537a90e2" PARTLABEL="decoroot"  
/dev/sdd2: UUID="2bc45786-1c42-a2af-4d32-23ff4389a912a" PARTLABEL="decondex"  
/dev/sdd3: UUID="5ee43678-f23a-113a-ff21-00924537ab76d2" PARTLABEL="decosessiondb"  
/dev/sdd4: UUID="4fa46678-c4ef-3890-00cf-d123e537770e2" PARTLABEL="decometadb"  
/dev/sdd5: UUID="5ee43678-0924-efc1-652c-00924efc1b76d" PARTLABEL="decopacketdb"
```

11. Run the **blkid** command again on the third and fourth block devices (*/dev/sde*, */dev/sdf*)* just like in the previous two steps and capture the UUIDs from the output.

```
blkid | egrep '/dev/sde' | awk '{print $1, $2, $5}'  
blkid | egrep '/dev/sdf' | awk '{print $1, $2, $5}'
```

12. Backup the operating system's *File System Table* file (*/etc/fstab*). Since this file will be edited shortly it is good practice to have a backup if anything goes wrong. This command will make a copy of the **fstab** and place it into the */root* directory, called **fstab.backup**.

```
cp /etc/fstab /root/fstab.backup
```

13. Edit the **fstab** (/etc/fstab) file and add the new UUIDs and mount locations for all block device partitions to the bottom of the file.

The example below shows how to use the UUIDs from the **blkid** outputs and combine them with the associate mount points. The customer should substitute their own UUIDs when adding the entries to the file. Each entry must be on its own line to function. Ten entries will be created for each storage unit attached to the appliance.

If unsure how to edit the fstab file, stop and request assistance from NetWitness Support.

If unsure of which mount points go to which block device partitions, review the outputs from the previous three steps along with the [Decoder-DB-Mount/Directory-Names Table](#).

Example **fstab** Lines

```
UUID=bb10f793-f141-4e74-8021-cd953bdece83 /var/netwitness/decoder xfs noatime,nosuid 1 2
UUID=ef431ad6-0b5c-4ca0-9092-8a635d2b33ea /var/netwitness/decoder/index xfs noatime,nosuid 1 2
UUID=f4ca8ff4-a87b-434b-add8-fe1a918aabf4 /var/netwitness/decoder/sessiondb xfs noatime,nosuid 1 2
UUID=3116ab5c-e813-43e4-a503-200d49e529fc /var/netwitness/decoder/metadb xfs noatime,nosuid 1 2
UUID=872ab261-c592-4416-9bdf-33727e4bb03d /var/netwitness/decoder/packetdb xfs noatime,nosuid 1 2

UUID=8e95d2fc-2dbc-4961-88ed-2ce3c4eb07b4 /var/netwitness/decoder/decoder0 xfs noatime,nosuid 1 2
UUID=94f05b71-5199-4b93-80cc-21fa1216f2eb /var/netwitness/decoder/decoder0/index0 xfs noatime,nosuid 1 2
UUID=8d40cf96-1a73-4fb7-8f59-5997e53415ec /var/netwitness/decoder/decoder0/sessiondb0 xfs noatime,nosuid 1 2
UUID=38661bd8-f690-4ed3-ba4c-0560161df106 /var/netwitness/decoder/decoder0/metadb0 xfs noatime,nosuid 1 2
UUID=b0dd95b5-ee01-49fa-aed8-baf838410750 /var/netwitness/decoder/decoder0/packetdb0 xfs noatime,nosuid 1 2
```

If looking for more information about the relationship between the UUID, PARTLABEL, mount points and the **fstab**, please see the [Relationship between UUID/PARTLABEL and Mount Points](#) section in the Appendix.

14. Reload the system daemon, confirm that the **fstab** is read successfully, and double check the partitions are visible. If any errors are produced, this means the changes done to the **fstab** file are not correct. Either a misspelling has occurred, or one or more device/directory locations are incorrect/missing. Use the backed up **fstab.backup** file if needed.

Do not reboot the device until any mounting issues are corrected as this could cause the system to fail to reboot properly. If any issues arise and you are uncertain about how to continue, stop and reach out to NetWitness Support.

```
systemctl daemon-reload
mount -a
df -hP
```

/dev/sdc1	20G	303M	20G	2%	/var/netwitness/decoder
/dev/sdc2	30G	248M	30G	1%	/var/netwitness/decoder/index
/dev/sdc3	4.5T	32G	4.5T	1%	/var/netwitness/decoder/sessiondb
/dev/sdc4	13T	90G	13T	1%	/var/netwitness/decoder/metadb
/dev/sdc5	131T	931G	130T	1%	/var/netwitness/decoder/packetdb
/dev/sdd1	20G	303M	20G	2%	/var/netwitness/decoder/decoder0
/dev/sdd2	30G	248M	30G	1%	/var/netwitness/decoder/decoder0/index0
/dev/sdd3	4.5T	32G	4.5T	1%	/var/netwitness/decoder/decoder0/sessiondb0
/dev/sdd4	13T	90G	13T	1%	/var/netwitness/decoder/decoder0/metadb0
/dev/sdd5	131T	931G	130T	1%	/var/netwitness/decoder/decoder0/packetdb0

Updating the Decoder Database Configuration (First & Second)

This section will provide step by step instructions for how to adjust the Decoder's storage configuration to properly take advantage of the newly daisy-chained Ultrastar60 storage units. To see screenshots of the NetWitness UI, go to the Appendix under [Updating Decoder Database Configuration Screenshots](#).

1. If the Decoder will be used for 10G capture, follow the **Reconfigure Storage Configuration for 10G Capture** in the [NetWitness Platform Storage Guide](#) before proceeding.

Important: If 10G capture setup is performed after the steps of this section, the values for packet.dir, meta.dir, session.dir, packet.free.space.min, meta.free.space.min, and session.free.space.min will be overwritten. This will cause incorrect space allocation and possible disconnection of storage. Always run 10G capture setup first.

2. Log into the NetWitness UI and go to the Explore View for the Decoder with the Ultrastar60 daisy-chained storage attached.
3. Once in Explore View, navigate to the database folder and select the config folder.

- Once in the config folder, look for the following nodes and replace their values with the ones provided below. The semi-colon in the **Values** column is required and each cell is a single continuous line.

Storage Unit	Node	Value
First & Second	packet.dir	/var/netwitness/decoder/packetdb=423 TB; /var/netwitness/decoder/decoder0/packetdb0=423 TB; /var/netwitness/decoder/decoder1/packetdb1=423 TB; /var/netwitness/decoder/decoder2/packetdb2=423 TB;
	meta.dir	/var/netwitness/decoder/metadb=16.5 TB; /var/netwitness/decoder/decoder0/metadb0=16.5 TB; /var/netwitness/decoder/decoder1/metadb1=16.5 TB; /var/netwitness/decoder/decoder2/metadb2=16.5 TB;
	session.dir	/var/netwitness/decoder/sessiondb=1 TB; /var/netwitness/decoder/decoder0/sessiondb0=1 TB; /var/netwitness/decoder/decoder1/sessiondb1=1 TB; /var/netwitness/decoder/decoder2/sessiondb2=1 TB;
	packet.free.space.min	1 TB
	meta.free.space.min	150 GB
	session.free.space.min	10 GB

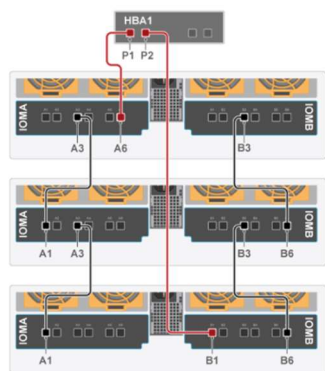
- Next go to the index folder and select its config folder. Update the following value:

Storage Unit	Node	Value
First & Second	index.dir	/var/netwitness/decoder/index=29 GB; /var/netwitness/decoder/decoder0/index0=29 GB; /var/netwitness/decoder/decoder1/index1=29 GB; /var/netwitness/decoder/decoder2/index2=29 GB;

- If no further Ultrastar60 storage units are to be configured, reboot the Decoder server for all changes to take full effect. All configurations are now complete for adding two Ultrastar60 daisy-chained storage units to the Network Decoder.
- If one or more additional Ultrastar60 units are to be added to the Network Decoder, please continue to the [Configuring Daisy-Chained Decoder Storage \(Third & Fourth\)](#) section.

Configuring Daisy-Chained Decoder Storage (Third & Fourth)

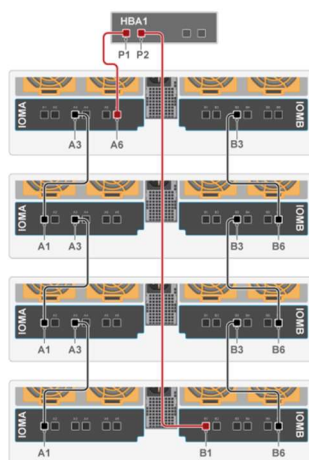
Three Enclosures: One Host with a Single HBA



— Passive Connections
— Active Connections

Source Device	Destination Device
Host1: P1	Enclosure1: IOMA, A6
Host1: P2	Enclosure3: IOMB, B1
Enclosure1: IOMA, A3	Enclosure2: IOMA, A1
Enclosure1: IOMB, B3	Enclosure2: IOMB, B6
Enclosure2: IOMA, A3	Enclosure3: IOMA, A1
Enclosure2: IOMB, B3	Enclosure3: IOMB, B6

Four Enclosures: One Host with a Single HBA



— Passive Connections
— Active Connections

Source Device	Destination Device
Host1: P1	Enclosure1: IOMA, A6
Host1: P2	Enclosure4: IOMB, B1
Enclosure1: IOMA, A3	Enclosure2: IOMA, A1
Enclosure1: IOMB, B3	Enclosure2: IOMB, B6
Enclosure2: IOMA, A3	Enclosure3: IOMA, A1
Enclosure2: IOMB, B3	Enclosure3: IOMB, B6
Enclosure3: IOMA, A3	Enclosure4: IOMA, A1
Enclosure3: IOMB, B3	Enclosure4: IOMB, B6

This section provides the steps needed to attach a third and/or fourth Ultrastar60 storage unit to the Decoder. Please keep in mind that setting up the third and fourth storage units will be identical. NetWitness Decoders currently support two, three, and four storage units in a daisy-chain configuration. The instructions in this section will show how to configure the third and fourth at the same time.

Important: If there is only a third storage unit, do not run any of the fourth storage unit commands.

Please review the [How to Start Configuring Ultrastar60 Storage](#) section to get the output for the Enclosure, Connector, Physical ID, and LUN ID of the additional storage. The steps in this section are almost identical to setting up the first and second Ultrastar60 units on the Decoder. As such the steps will be more abbreviated but should align closely with the ones provided in the [Configuring Daisy-Chained Decoder Storage \(First & Second\)](#).

Important: When using the arconf utility to create block devices the LUN IDs of the storage units must be entered in reverse order (0,59 0,58 0,57 ... 0,2 0,1 0,0), instead of normal order (0,0 0,1 0,2 ... 0,57 0,58 0,59). Pay particular attention when entering the commands of this section due to this reversal.

Creating Ultrastar60 Block Devices (Third & Fourth)

1. Perform a block device listing on the appliance. This command looks for any devices that are of size 442 TB. This will catch any configured Ultrastar60 storage. Only the first two Ultrastar60 storage units should appear as they are already configured.

```
lsblk | grep '442T'
```

2. Retrieve the Physical ID and LUN ID output for Connector 0 (CN0/First Storage Unit) and Connector 1 (CN1/Second Storage Unit), per the [How to Start Configuring Ultrastar60 Storage](#) section.
3. Break up the 120 LUNS into four groups of 30, remembering to use reverse order. If only configuring a third storage unit, there will be only 60 total LUNs.

Storage Unit	Groups	LUN IDs
Third	Group 1	0,149 0,148 0,147 ... 0,122 0,121 0,120
	Group 2	0,179 0,178 0,177 ... 0,152 0,151 0,150
Storage Unit	Groups	LUN IDs
Fourth	Group 3	0,209 0,208 0,207 ... 0,182 0,181 0,180
	Group 4	0,239 0,238 0,237 ... 0,212 0,211 0,210

4. Create the first RAID 50 Block Device 0 using the Group 1 LUNs from the **third** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50  
0,149 0,148 0,147 0,146 0,145 0,144 0,143 0,142 0,141 0,140 0,139 0,138  
0,137 0,136 0,135 0,134 0,133 0,132 0,131 0,130 0,129 0,128 0,127 0,126  
0,125 0,124 0,123 0,122 0,121 0,120 noprompt
```

5. Create the second RAID 50 Block Device 1 using Group 2 LUNs from the **third** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50  
0,179 0,178 0,177 0,176 0,175 0,174 0,173 0,172 0,171 0,170 0,169 0,168  
0,167 0,166 0,165 0,164 0,163 0,162 0,161 0,160 0,159 0,158 0,157 0,156  
0,155 0,154 0,153 0,152 0,151 0,150 noprompt
```

6. If there is a **fourth** storage unit, create the third RAID 50 Block Device 2 using Group 3 LUNs.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,209 0,208 0,207 0,206 0,205 0,204 0,203 0,202 0,201 0,200 0,199 0,198
0,197 0,196 0,195 0,194 0,193 0,192 0,191 0,190 0,189 0,188 0,187 0,186
0,185 0,184 0,183 0,182 0,181 0,180 noprompt
```

7. If there is a **fourth** storage unit, create the fourth RAID 50 Block Device 3 using Group 4 LUNs.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,239 0,238 0,237 0,236 0,235 0,234 0,233 0,232 0,231 0,230 0,229 0,228
0,227 0,226 0,225 0,224 0,223 0,222 0,221 0,220 0,219 0,218 0,217 0,216
0,215 0,214 0,213 0,212 0,211 0,210 noprompt
```

8. Run the **lsblk** command to find the newly created Ultrastar60 block devices. The command should return two or four new block devices of 442 TB in size depending on the number of storage units being configured, similar to the output below.

```
lsblk | grep `442T`
```

	sdc	8:32	0	442T	0	disk
	sdd	8:32	0	442T	0	disk
	sde	8:32	0	442T	0	disk
	sdf	8:32	0	442T	0	disk
Third	sdg	8:32	0	442T	0	disk
	sdh	8:32	0	442T	0	disk
Fourth Storage	sdi	8:32	0	442T	0	disk
	sdj	8:32	0	442T	0	disk

Sometimes the block devices will not display immediately, this is normal. A rescan of the bus can be performed using the following command. Once it is complete, rerun the **lsblk** command above.

```
arcconf rescan 1
```

Create the Partitions and Mount Points for the New Block Devices (Third & Fourth)

This section will show how to create new partitions and set up mount points using the newly created block devices on the third and fourth Ultrastar60 storage units.

Please use the block devices that were discovered while running the **lsblk** command in the previous section. The devices discovered should take the place of `/dev/sdg` and `/dev/sdh` for the third storage unit. If there is a fourth storage unit, `/dev/sdi` and `/dev/sdj` should also be replaced in the following examples. The [Decoder-DB-Mount/Directory-Names Table](#) will be utilized for this section and should be referenced as needed.

If the appliance has been rebooted since the block device creation was performed, rerun the **lsblk** command again as the block device names may have changed. The operating system is responsible for device naming and can change the names between reboots.

The block device of `/dev/sdg`, `/dev/sdh`, `/dev/sdi`, and `/dev/sdj` will be used in examples throughout this section.* These are only place holders for the block devices within the customer's appliance.

If there are any questions, please stop and contact NetWitness Customer Support for assistance.

1. Set the disk label type to gpt for the new block devices (`/dev/sdg`, `/dev/sdh`, `/dev/sdi`, `/dev/sdj`)*.

Command:

```
parted -s <block device> mklabel gpt
```

Example:

Storage Unit	Command
Third	<code>parted -s /dev/sdg mklabel gpt</code>
	<code>parted -s /dev/sdh mklabel gpt</code>
Storage Unit	Command
Fourth	<code>parted -s /dev/sdi mklabel gpt</code>
	<code>parted -s /dev/sdj mklabel gpt</code>

2. Create the required Decoder partitions utilizing the **PART LABEL** column from the [Decoder-DB-Mount/Directory-Names Table](#). Run all the commands below for each block device (`/dev/sdc`, `/dev/sdd`, `/dev/sde`, `/dev/sdf`)* to create the required partitions.

Command:

```
parted --align minimal <block device> mkpart <part label> xfs start_size  
end_size
```

Example:

Third Storage Unit	parted --align minimal /dev/sdg mkpart decoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdg mkpart decondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdg mkpart decosessiondb xfs 50GiB 1106GiB
	parted --align minimal /dev/sdg mkpart decometadb xfs 1106GiB 18514GiB
	parted --align minimal /dev/sdg mkpart decopacketdb xfs 18514GiB 100%
	parted --align minimal /dev/sdh mkpart decoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdh mkpart decondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdh mkpart decosessiondb xfs 50GiB 1106GiB
	parted --align minimal /dev/sdh mkpart decometadb xfs 1106GiB 18514GiB
	parted --align minimal /dev/sdh mkpart decopacketdb xfs 18514GiB 100%
Fourth Storage Unit	parted --align minimal /dev/sdi mkpart decoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdi mkpart decondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdi mkpart decosessiondb xfs 50GiB 1106GiB
	parted --align minimal /dev/sdi mkpart decometadb xfs 1106GiB 18514GiB
	parted --align minimal /dev/sdi mkpart decopacketdb xfs 18514GiB 100%
	parted --align minimal /dev/sdj mkpart decoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdj mkpart decondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdj mkpart decosessiondb xfs 50GiB 1106GiB
	parted --align minimal /dev/sdj mkpart decometadb xfs 1106GiB 18514GiB
	parted --align minimal /dev/sdj mkpart decopacketdb xfs 18514GiB 100%

- Confirm that all partitions have been created correctly for all block devices (/dev/sdc, /dev/sdd, /dev/sde, /dev/sdf)* by running the parted print command for all block devices. The output will be the same for all block devices.

Command:

parted <block device> print

Example:

Storage Unit	Command
Third	parted /dev/sdg print
	parted /dev/sdh print
Storage Unit	Command
Fourth	parted /dev/sdi print
	parted /dev/sdj print

All block device outputs should look identical to the output below.

```
Model: Adaptec LOGICAL VOLUME (scsi)
Disk /dev/sdc: 486TB
Sector Size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File System  Name              Flags
  1      3146kB  21.5GB  21.5GB                decoroot
  2      21.5GB  53.7GB  32.2GB                decondex
  3      53.7GB  4948GB  4894GB                decosessiondb
  4      4948GB  18.7TB  13.7TB                decometadb
  5      18.7TB  486TB   467TB                decopacketdb
```

- Create the file systems on all the newly created Decoder partitions. The numbers under the **Number** column from the previous step's print outputs should be appended to the block device name as shown in the commands below.

Command:

```
mkfs.xfs -f <block device><partition Number>
```

Example:

Storage Unit	Block Device	Command
Third	/dev/sdg*	mkfs.xfs -f /dev/sdg1
		mkfs.xfs -f /dev/sdg2
		mkfs.xfs -f /dev/sdg3
		mkfs.xfs -f /dev/sdg4
		mkfs.xfs -f /dev/sdg5
	/dev/sdh*	mkfs.xfs -f /dev/sdh1
		mkfs.xfs -f /dev/sdh2
		mkfs.xfs -f /dev/sdh3
		mkfs.xfs -f /dev/sdh4
		mkfs.xfs -f /dev/sdh5
Storage Unit	Block Device	Command
Fourth	/dev/sdi*	mkfs.xfs -f /dev/sdi1
		mkfs.xfs -f /dev/sdi2
		mkfs.xfs -f /dev/sdi3
		mkfs.xfs -f /dev/sdi4
		mkfs.xfs -f /dev/sdi5
	/dev/sdj*	mkfs.xfs -f /dev/sdj1
		mkfs.xfs -f /dev/sdj2
		mkfs.xfs -f /dev/sdj3
		mkfs.xfs -f /dev/sdj4
		mkfs.xfs -f /dev/sdj5

- Confirm that the first decoroot is mounted to /var/netwitness/decoder before continuing. This partition acts as the container for all other mounted storage partitions and must be available.

Command:

```
df -h
```

```
/dev/sdc1      20G  303M  20G   2% /var/netwitness/decoder
/dev/sdc2      30G  248M  30G   1% /var/netwitness/decoder/index
/dev/sdc3      4.5T   32G  4.5T   1% /var/netwitness/decoder/sessiondb
/dev/sdc4      13T   90G   13T   1% /var/netwitness/decoder/metadb
/dev/sdc5     131T  931G  130T   1% /var/netwitness/decoder/packetdb
/dev/sdd1      20G  177M   20G   1% /var/netwitness/decoder/decoder0
/dev/sdd2      30G  248M  30G   1% /var/netwitness/decoder/decoder0/index0
/dev/sdd3      4.5T   32G  4.5T   1% /var/netwitness/decoder/decoder0/sessiondb0
/dev/sdd4      13T   90G   13T   1% /var/netwitness/decoder/decoder0/metadb0
/dev/sdd5     131T  931G  130T   1% /var/netwitness/decoder/decoder0/packetdb0
/dev/sde1      20G  177M   20G   1% /var/netwitness/decoder/decoder1
/dev/sde2      30G  248M  30G   1% /var/netwitness/decoder/decoder1/index1
/dev/sde3      4.5T   32G  4.5T   1% /var/netwitness/decoder/decoder1/sessiondb1
/dev/sde4      13T   90G   13T   1% /var/netwitness/decoder/decoder1/metadb1
/dev/sde5     426T   3.0T  423T   1% /var/netwitness/decoder/decoder1/packetdb1
/dev/sdf1      20G  177M   20G   1% /var/netwitness/decoder/decoder2
/dev/sdf2      30G  248M  30G   1% /var/netwitness/decoder/decoder2/index2
/dev/sdf3      4.5T   32G  4.5T   1% /var/netwitness/decoder/decoder2/sessiondb2
/dev/sdf4      13T   90G   13T   1% /var/netwitness/decoder/decoder2/metadb2
/dev/sdf5     426T   3.0T  423T   1% /var/netwitness/decoder/decoder2/packetdb2
```

6. Create the directories' mount points for the new decoroot partitions.

Command:

```
mkdir /var/netwitness/decoder/decoder<#>
```

Example:

Storage Unit	Command
Third	mkdir /var/netwitness/decoder/decoder3
	mkdir /var/netwitness/decoder/decoder4
Storage Unit	Command
Fourth	mkdir /var/netwitness/decoder/decoder5
	mkdir /var/netwitness/decoder/decoder6

7. Mount the decoroot partitions to the appropriate folder locations (/dev/sdg1, /dev/sdh1, /dev/sdi1, /dev/sdj)*.

Command:

```
mount <partition> <mounting location>
```

Example:

Storage Unit	Command
Third	mount /dev/sdg1 /var/netwitness/decoder/decoder3
	mount /dev/sdh1 /var/netwitness/decoder/decoder4
Storage Unit	Command
Fourth	mount /dev/sdi1 /var/netwitness/decoder/decoder5
	mount /dev/sdj1 /var/netwitness/decoder/decoder6

8. With all the new decoroot partitions mounted, create all other mount point folders.

Command:

```
mkdir /var/netwitness/decoder/decoder<#>/<db>
```

Example:

Storage Unit	Command
Third	mkdir /var/netwitness/decoder/decoder3/index3
	mkdir /var/netwitness/decoder/decoder3/sessiondb3
	mkdir /var/netwitness/decoder/decoder3/metadb3
	mkdir /var/netwitness/decoder/decoder3/packetdb3
	mkdir /var/netwitness/decoder/decoder4/index4
	mkdir /var/netwitness/decoder/decoder4/sessiondb4
	mkdir /var/netwitness/decoder/decoder4/metadb4
	mkdir /var/netwitness/decoder/decoder4/packetdb4

Storage Unit	Command
Fourth	mkdir /var/netwitness/decoder/decoder5/index5
	mkdir /var/netwitness/decoder/decoder5/sessiondb5
	mkdir /var/netwitness/decoder/decoder5/metadb5
	mkdir /var/netwitness/decoder/decoder5/packetdb5
	mkdir /var/netwitness/decoder/decoder6/index6
	mkdir /var/netwitness/decoder/decoder6/sessiondb6
	mkdir /var/netwitness/decoder/decoder6/metadb6
	mkdir /var/netwitness/decoder/decoder6/packetdb6

9. Run the **blkid** command to list all the block device partitions on each of the block devices (/dev/sdg, /dev/sdh, /dev/sdi, /dev/sdj)*. Capture the output from all the commands as the UUIDs listed will be needed in later steps (see screenshot for example output). It is important to note which UUIDs are associated with which PARTLABELs from the output. These PARTLABELs directly correspond with the PART LABEL column in the [Decoder-DB-Mount/Directory-Names Table](#) section of this document.

Storage Unit	Command
Third	blkid egrep '/dev/sdg' awk '{print \$1, \$2, \$5}'
	blkid egrep '/dev/sdh' awk '{print \$1, \$2, \$5}'

Storage Unit	Command
Fourth	blkid egrep '/dev/sdi' awk '{print \$1, \$2, \$5}'
	blkid egrep '/dev/sdj' awk '{print \$1, \$2, \$5}'

```
/dev/sdc1: UUID="1af45678-12c4-f23a-12d4-cf234537a90e2" PARTLABEL="decoroot"
/dev/sdc2: UUID="2bc45786-1c42-a2af-4d32-23ff4537a912a" PARTLABEL="deconindex"
/dev/sdc3: UUID="3de45878-2c41-113a-ff21-cf23ff37a11a2" PARTLABEL="decosessiondb"
/dev/sdc4: UUID="4fa46678-c4ef-3890-00cf-d123e537770e2" PARTLABEL="decometadb"
/dev/sdc5: UUID="5ee43678-3321-efc1-652c-00924537ab76d" PARTLABEL="decopacketdb"
```

10. Backup the operating system's *File System Table* file (*/etc/fstab*). Since this file will be edited shortly, it is good practice to have a backup if anything goes wrong. This command will make a copy of the **fstab** and place it into the */root* directory, called **fstab.backup**.

```
cp /etc/fstab /root/fstab.backup
```

11. Edit the **fstab** (*/etc/fstab*) file and add the new UUIDs and mount locations for all block device partitions to the bottom of the file.

The example below shows how to use the UUIDs from the **blkid** outputs and combine them with the associate mount points. The customer should substitute their own UUIDs when adding the entries to the file. Each entry must be on its own line to function. Ten entries will be created for each storage unit attached to the appliance.

If unsure how to edit the **fstab** file, stop and request assistance from NetWitness Support.

If unsure of which mount points go to which block device partitions, review the **lsblk** outputs along with the [Decoder-DB-Mount/Directory-Names Table](#) for the appropriate storage unit.

Example **fstab** Lines

```
UUID= bb10f793-f141-4e74-8021-cd953bdece83 /var/netwitness/decoder/decoder1 xfs noatime,nosuid 1 2
UUID=ef431ad6-0b5c-4ca0-9092-8a635d2b33ea /var/netwitness/decoder/decoder1/index1 xfs noatime,nosuid 1 2
UUID=f4ca8ff4-a87b-434b-add8-fe1a918aabf4 /var/netwitness/decoder/decoder1/sessiondb1 xfs noatime,nosuid 1 2
UUID=3116ab5c-e813-43e4-a503-200d49e529fc /var/netwitness/decoder/decoder1/metadb1 xfs noatime,nosuid 1 2
UUID=872ab261-c592-4416-9bdf-33727e4bb03d /var/netwitness/decoder/decoder1/packetdb1 xfs noatime,nosuid 1 2

UUID= 8e95d2fc-2dbc-4961-88ed-2ce3c4eb07b4 /var/netwitness/decoder/decoder2 xfs noatime,nosuid 1 2
UUID=94f05b71-5199-4b93-80cc-21fa1216f2eb /var/netwitness/decoder/decoder2/index2 xfs noatime,nosuid 1 2
UUID=8d40cf96-1a73-4fb7-8f59-5997e53415ec /var/netwitness/decoder/decoder2/sessiondb2 xfs noatime,nosuid 1 2
UUID=38661bd8-f690-4ed3-ba4c-0560161df106 /var/netwitness/decoder/decoder2/metadb2 xfs noatime,nosuid 1 2
UUID=b0dd95b5-ee01-49fa-aed8-baf838410750 /var/netwitness/decoder/decoder2/packetdb2 xfs noatime,nosuid 1 2
```

If looking for more information about the relationship between the UUID, PARTLABEL, mount points and the **fstab**, please see the [Relationship between UUID/PARTLABEL and Mount Points](#) section in the Appendix.

12. Reload the system daemon, confirm that the **fstab** is read successfully, and double check the partitions are visible. If any errors are produced, this means the changes done to the **fstab** file are not correct. Either a misspelling has occurred, or one or more device/directory locations are incorrect/missing. Use the backed up **fstab.backup** file if needed.

Do not reboot the device until any mounting issues are corrected as this could cause the system to fail to reboot properly. If an issue occurs and you are unsure how to proceed, stop and contact NetWitness Support for assistance.

```
systemctl daemon-reload
mount -a
df -hP
```

/dev/sdc1	20G	303M	20G	2%	/var/netwitness/decoder
/dev/sdc2	30G	248M	30G	1%	/var/netwitness/decoder/index
/dev/sdc3	4.5T	32G	4.5T	1%	/var/netwitness/decoder/sessiondb
/dev/sdc4	13T	90G	13T	1%	/var/netwitness/decoder/metadb
/dev/sdc5	131T	931G	130T	1%	/var/netwitness/decoder/packetdb
/dev/sdd1	20G	177M	20G	1%	/var/netwitness/decoder/decoder0
/dev/sdd2	30G	248M	30G	1%	/var/netwitness/decoder/decoder0/index0
/dev/sdd3	4.5T	32G	4.5T	1%	/var/netwitness/decoder/decoder0/sessiondb0
/dev/sdd4	13T	90G	13T	1%	/var/netwitness/decoder/decoder0/metadb0
/dev/sdd5	131T	931G	130T	1%	/var/netwitness/decoder/decoder0/packetdb0
/dev/sde1	20G	177M	20G	1%	/var/netwitness/decoder/decoder1
/dev/sde2	30G	248M	30G	1%	/var/netwitness/decoder/decoder1/index1
/dev/sde3	4.5T	32G	4.5T	1%	/var/netwitness/decoder/decoder1/sessiondb1
/dev/sde4	13T	90G	13T	1%	/var/netwitness/decoder/decoder1/metadb1
/dev/sde5	426T	3.0T	423T	1%	/var/netwitness/decoder/decoder1/packetdb1
/dev/sdf1	20G	177M	20G	1%	/var/netwitness/decoder/decoder2
/dev/sdf2	30G	248M	30G	1%	/var/netwitness/decoder/decoder2/index2
/dev/sdf3	4.5T	32G	4.5T	1%	/var/netwitness/decoder/decoder2/sessiondb2
/dev/sdf4	13T	90G	13T	1%	/var/netwitness/decoder/decoder2/metadb2
/dev/sdf5	426T	3.0T	423T	1%	/var/netwitness/decoder/decoder2/packetdb2
/dev/sdg1	20G	177M	20G	1%	/var/netwitness/decoder/decoder3
/dev/sdg2	30G	248M	30G	1%	/var/netwitness/decoder/decoder3/index3
/dev/sdg3	4.5T	32G	4.5T	1%	/var/netwitness/decoder/decoder3/sessiondb3
/dev/sdg4	13T	90G	13T	1%	/var/netwitness/decoder/decoder3/metadb3
/dev/sdg5	131T	931G	130T	1%	/var/netwitness/decoder/decoder3/packetdb3
/dev/sdh1	20G	177M	20G	1%	/var/netwitness/decoder/decoder4
/dev/sdh2	30G	248M	30G	1%	/var/netwitness/decoder/decoder4/index4
/dev/sdh3	4.5T	32G	4.5T	1%	/var/netwitness/decoder/decoder4/sessiondb4
/dev/sdh4	13T	90G	13T	1%	/var/netwitness/decoder/decoder4/metadb4
/dev/sdh5	131T	931G	130T	1%	/var/netwitness/decoder/decoder4/packetdb4
/dev/sdi1	20G	177M	20G	1%	/var/netwitness/decoder/decoder5
/dev/sdi2	30G	248M	30G	1%	/var/netwitness/decoder/decoder5/index5
/dev/sdi3	4.5T	32G	4.5T	1%	/var/netwitness/decoder/decoder5/sessiondb5
/dev/sdi4	13T	90G	13T	1%	/var/netwitness/decoder/decoder5/metadb5
/dev/sdi5	426T	3.0T	423T	1%	/var/netwitness/decoder/decoder5/packetdb5
/dev/sdj1	20G	177M	20G	1%	/var/netwitness/decoder/decoder6
/dev/sdj2	30G	248M	30G	1%	/var/netwitness/decoder/decoder6/index6
/dev/sdj3	4.5T	32G	4.5T	1%	/var/netwitness/decoder/decoder6/sessiondb6
/dev/sdj4	13T	90G	13T	1%	/var/netwitness/decoder/decoder6/metadb6
/dev/sdj5	426T	3.0T	423T	1%	/var/netwitness/decoder/decoder6/packetdb6

Updating the Decoder Database Configuration (Third & Fourth)

This section will provide step by step instructions for how to adjust the Decoder's storage configuration to properly take advantage of the newly daisy-chained Ultrastar60 storage units. To see screenshots of the NetWitness UI, go to the Appendix under [Updating Decoder Database Configuration Screenshots](#).

1. If the Decoder will be used for 10G capture, follow the **Reconfigure Storage Configuration for 10G Capture** in the [NetWitness Platform Storage Guide](#) before proceeding.

Important: If 10G capture setup is performed after any Ultrastar60's have been setup, the values for packet.dir, meta.dir, session.dir, packet.free.space.min, meta.free.space.min, and session.space.free.min will be overwritten. This will cause incorrect space allocation and possible disconnection of storage. Always run 10G capture setup first.

2. Log into the NetWitness UI and go to the Explore View for the Decoder with the Ultrastar60 daisy-chained storage attached.
3. Once in Explore View, navigate to the database folder and select the config folder.
4. Once in the config folder, look for the following nodes. The first and second Ultrastar60 partition entries should already be in these nodes. Add the entries in the **Value** column to the end of the existing entry. If only the Third storage is present, use only those entries. If the Third and Fourth storage are present, combine the **Value** column entries in the table and then add them to the node. Make sure that none of the semi-colons are missing as they are required.

Storage Unit	Node	Value
Third & Fourth	packet.free.space.min	1 TB
	meta.free.space.min	150 GB
	session.free.space.min	10 GB

Storage Unit	Node	Value
Third	packet.dir	/var/netwitness/decoder/decoder3/packetdb3=423 TB; /var/netwitness/decoder/decoder4/packetdb4=423 TB;
	meta.dir	/var/netwitness/decoder/decoder3/metadb3=16.5 TB; /var/netwitness/decoder/decoder4/metadb4=16.5 TB;
	session.dir	/var/netwitness/decoder/decoder3/sessiondb3=1 TB; /var/netwitness/decoder/decoder4/sessiondb4=1 TB;

Storage Unit	Node	Value
Fourth	packet.dir	/var/netwitness/decoder/decoder5/packetdb5=423 TB; /var/netwitness/decoder/decoder6/packetdb6=423 TB;
	meta.dir	/var/netwitness/decoder/decoder5/metadb5=16.5 TB; /var/netwitness/decoder/decoder6/metadb6=16.5 TB;
	session.dir	/var/netwitness/decoder/decoder5/sessiondb5=1 TB; /var/netwitness/decoder/decoder6/sessiondb6=1 TB;

5. Next go to the index folder and select its config folder. The first and second Ultrastar60 partition entries should already be in this node. Add the entries in the **Value** column to the end of the existing entry. If only the Third storage is present, use only that entry. If the Third and Fourth storage are present, combine the **Value** column entries in the table and then add them to the node. Make sure that none of the semi-colons are missing as they are required.

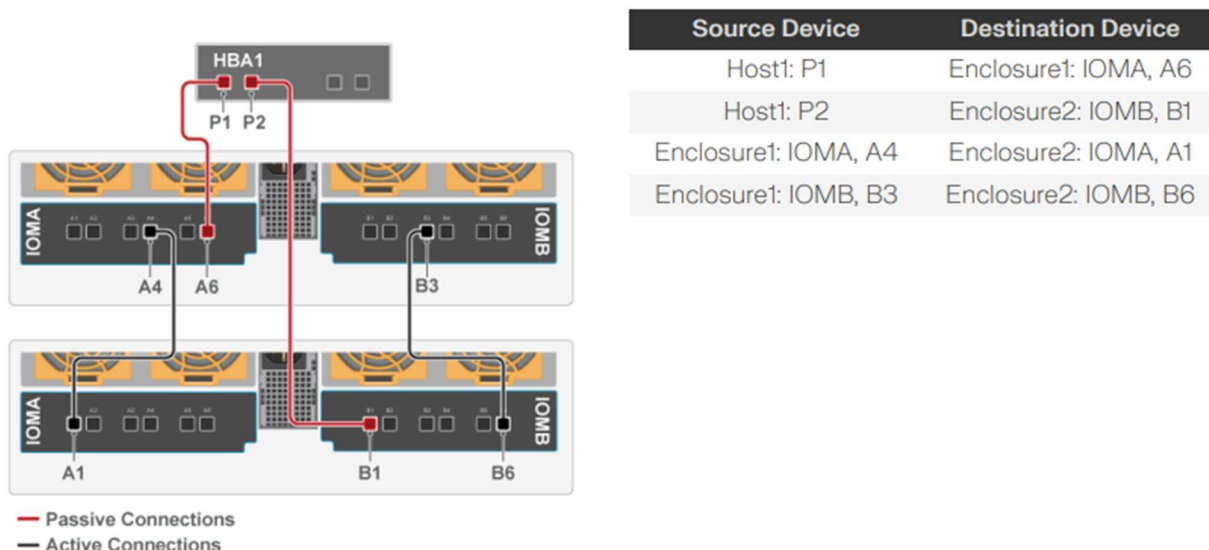
Storage Unit	Node	Value
Third	index.dir	/var/netwitness/decoder/decoder3/index3=29 GB; /var/netwitness/decoder/decoder4/index4=29 GB;

Storage Unit	Node	Value
Fourth	index.dir	/var/netwitness/decoder/decoder5/index5=29 GB; /var/netwitness/decoder/decoder6/index6=29 GB;

6. Once all the configuration has been completed reboot the Decoder server for all changes to take full effect.
7. All steps have now been completed for daisy-chaining storage to the Network Decoder.

Configuring Daisy-Chained Log Decoder Storage (First & Second)

Two Enclosures: One Host with a Single HBA



This section will go through the steps needed to set up and connect two Ultrastar60 storage units to a Log Decoder. Please review the [How to Start Configuring Ultrastar60 Storage](#) section before proceeding as the output for Enclosure, Connector, Physical ID, and LUN ID will be required. Make sure that only the two Ultrastar60 storage units are currently connected, as this will help with setup later.

Important: The `arcconf` utility requires that the LUN IDs captured earlier must be entered in reverse order (0,59 0,58 0,57 ... 0,2 0,1 0,0), instead of normal order (0,0 0,1 0,2 ... 0,57 0,58 0,59). Pay particular attention when entering the commands of this section due to this reversal.

Creating Ultrastar60 Block Devices (First & Second)

1. Perform a block device listing on the appliance. This command looks for any devices that are of size 442 TB. This will catch any configured Ultrastar60 storage. Since all connected units are unconfigured, no block devices should be returned.

```
lsblk | grep '442T'
```

2. Retrieve the Physical ID and LUN ID output for Connector 0 (CN0/First Storage Unit) and Connector 1 (CN1/Second Storage Unit), per the [How to Start Configuring Ultrastar60 Storage](#) section.

- Break up the 120 LUNS into four groups of 30, remembering to use reverse order.

Storage	Groups	LUN IDs
First	Group 1	0,29 0,28 0,27 ... 0,2 0,1 0,0
	Group 2	0,59 0,58 0,57 ... 0,32 0,31 0,30
Storage	Groups	LUN IDs
Second	Group 3	0,89 0,88 0,87 ... 0,62 0,61 0,60
	Group 4	0,119 0,118 0,117 ... 0,92 0,91 0,90

- Create the first RAID 50 Block Device 0 using the Group 1 LUNs, from the first storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,29 0,28 0,27 0,26 0,25 0,24 0,23 0,22 0,21 0,20 0,19 0,18 0,17 0,16 0,15
0,14 0,13 0,12 0,11 0,10 0,9 0,8 0,7 0,6 0,5 0,4 0,3 0,2 0,1 0,0 noprompt
```

- Create the second RAID 50 Block Device 1 using Group 2 LUNs, from the first storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,59 0,58 0,57 0,56 0,55 0,54 0,53 0,52 0,51 0,50 0,49 0,48 0,47 0,46 0,45
0,44 0,43 0,42 0,41 0,40 0,39 0,38 0,37 0,36 0,35 0,34 0,33 0,32 0,31 0,30
noprompt
```

- Create the third RAID 50 Block Device 2 using Group 3 LUNs, from the second storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,89 0,88 0,87 0,86 0,85 0,84 0,83 0,82 0,81 0,80 0,79 0,78 0,77 0,76 0,75
0,74 0,73 0,72 0,71 0,70 0,69 0,68 0,67 0,66 0,65 0,64 0,63 0,62 0,61 0,60
noprompt
```

- Create the fourth RAID 50 Block Device 3 using Group 4 LUNs, from the second storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,119 0,118 0,117 0,116 0,115 0,114 0,113 0,112 0,111 0,110 0,109 0,108
0,107 0,106 0,105 0,104 0,103 0,102 0,101 0,100 0,99 0,98 0,97 0,96 0,95
0,94 0,93 0,92 0,91 0,90 noprompt
```

8. Run the **lsblk** command to find the newly created Ultrastar60 block devices. The command should return four devices of 442 TB in size, similar to the output below.

```
lsblk | grep '442T'
```

First Storage	sdc	8:32	0	442T	0	disk
	sdd	8:32	0	442T	0	disk
Second Storage	sde	8:32	0	442T	0	disk
	sdf	8:32	0	442T	0	disk

Sometimes the block devices will not display immediately. This is normal. A rescan of the bus can be performed using the below command. Once it is complete, rerun the **lsblk** command above.

```
arcconf rescan 1
```

Create the Partitions and Mount Points for the New Block Devices (First & Second)

This section will show how to create new partitions and set up mount points using the newly created block devices on the first and second Ultrastar60 storage units.

Please use the block devices that were discovered while running the **lsblk** command in the previous section. The devices discovered should take the place of `/dev/sdc` and `/dev/sdd`, for the first storage unit. The second set of devices should take the place of `/dev/sde` and `/dev/sdf` for the second storage unit in the following examples. The [LogDecoder-DB-Mount/Directory-Names Table](#) will be utilized for this section and should be referenced as needed.

If the appliance has been rebooted since the block device creation was performed, rerun the **lsblk** command again as the block device names may have changed. The operating system is responsible for device naming and can change the names between reboots.

The block device of `/dev/sdc`, `/dev/sdd`, `/dev/sde`, and `/dev/sdf` will be used in examples throughout this section.* These are only place holders for the block devices within the customer's appliance.

If there are any questions, please stop and contact NetWitness Customer Support for assistance.

1. Set the disk label type to gpt for the new block devices (`/dev/sdc`, `/dev/sdd`, `/dev/sde`, `/dev/sdf`)*.

Command:

```
parted -s <block device> mklabel gpt
```

Example:

Storage Unit	Command
First	<code>parted -s /dev/sdc mklabel gpt</code>
	<code>parted -s /dev/sdd mklabel gpt</code>
Storage Unit	Command
Second	<code>parted -s /dev/sde mklabel gpt</code>
	<code>parted -s /dev/sdf mklabel gpt</code>

2. Create the required Log Decoder partitions utilizing the **PART LABEL** column from the [LogDecoder-DB-Mount/Directory-Names Table](#). Run all the commands below for each block device (`/dev/sdc`, `/dev/sdd`, `/dev/sde`, `/dev/sdf`)* to create the required partitions.

Command:

```
parted --align minimal <block device> mkpart <part label> xfs start_size  
end_size
```

Example:

First Storage Unit	parted --align minimal /dev/sdc mkpart logdecoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdc mkpart logdecondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdc mkpart logdecosessiondb xfs 50GiB 1.5TiB
	parted --align minimal /dev/sdc mkpart logdecometadb xfs 1.5TiB 50TiB
	parted --align minimal /dev/sdc mkpart logdecopacketdb xfs 50TiB 100%
	parted --align minimal /dev/sdd mkpart logdecoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdd mkpart logdecondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdd mkpart logdecosessiondb xfs 50GiB 1.5TiB
	parted --align minimal /dev/sdd mkpart logdecometadb xfs 1.5TiB 50TiB
	parted --align minimal /dev/sdd mkpart logdecopacketdb xfs 50TiB 100%
Second Storage Unit	parted --align minimal /dev/sde mkpart logdecoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sde mkpart logdecondex xfs 20GiB 50GiB
	parted --align minimal /dev/sde mkpart logdecosessiondb xfs 50GiB 1.5TiB
	parted --align minimal /dev/sde mkpart logdecometadb xfs 1.5TiB 50TiB
	parted --align minimal /dev/sde mkpart logdecopacketdb xfs 50TiB 100%
	parted --align minimal /dev/sdf mkpart logdecoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdf mkpart logdecondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdf mkpart logdecosessiondb xfs 50GiB 1.5TiB
	parted --align minimal /dev/sdf mkpart logdecometadb xfs 1.5TiB 50TiB
	parted --align minimal /dev/sdf mkpart logdecopacketdb xfs 50TiB 100%

- Confirm that all partitions have been created correctly for all block devices (/dev/sdc, /dev/sdd, /dev/sde, /dev/sdf)* by running the parted print command for all block devices. The output will be the same for all block devices.

Command:

parted <block device> print

Example:

Storage Unit	Command
First	parted /dev/sdc print
	parted /dev/sdd print
Storage Unit	Command
Second	parted /dev/sde print
	parted /dev/sdf print

All block device outputs should look identical to the output below.

```
Model: Adaptec LOGICAL VOLUME (scsi)
Disk /dev/sdc: 486TB
Sector Size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File System  Name                Flags
  1      3146kB  21.5GB  21.5GB                logdecoroot
  2      21.5GB  53.7GB  32.2GB                logdecondex
  3      53.7GB  4948GB  4894GB                logdecosessiondb
  4      4948GB  18.7TB  13.7TB                logdecometadb
  5      18.7TB  486TB   467TB                logdecopacketdb
```

4. Create the file systems on all the newly created Log Decoder partitions. The numbers under the Number column from the previous step's print outputs should be appended to the block device name as shown in the below commands.

Command:

```
mkfs.xfs -f <block device><partition Number>
```

Example:

Storage Unit	Block Device	Command
First	<i>/dev/sdc*</i>	mkfs.xfs -f <i>/dev/sdc1</i>
		mkfs.xfs -f <i>/dev/sdc2</i>
		mkfs.xfs -f <i>/dev/sdc3</i>
		mkfs.xfs -f <i>/dev/sdc4</i>
		mkfs.xfs -f <i>/dev/sdc5</i>
	<i>/dev/sdd*</i>	mkfs.xfs -f <i>/dev/sdd1</i>
		mkfs.xfs -f <i>/dev/sdd2</i>
		mkfs.xfs -f <i>/dev/sdd3</i>
		mkfs.xfs -f <i>/dev/sdd4</i>
		mkfs.xfs -f <i>/dev/sdd5</i>
Storage Unit	Block Device	Command
Second	<i>/dev/sde*</i>	mkfs.xfs -f <i>/dev/sde1</i>
		mkfs.xfs -f <i>/dev/sde2</i>
		mkfs.xfs -f <i>/dev/sde3</i>
		mkfs.xfs -f <i>/dev/sde4</i>
		mkfs.xfs -f <i>/dev/sde5</i>
	<i>/dev/sdf*</i>	mkfs.xfs -f <i>/dev/sdf1</i>
		mkfs.xfs -f <i>/dev/sdf2</i>
		mkfs.xfs -f <i>/dev/sdf3</i>
		mkfs.xfs -f <i>/dev/sdf4</i>
		mkfs.xfs -f <i>/dev/sdf5</i>

5. Manually mount the logdecoroot partition from the first block device (*/dev/sdc*)*. This partition will act as the container for all other mounted storage partitions.

Command:

```
mount <partition> <mounting location>
```

Example:

```
mount /dev/sdc1 /var/netwitness/logdecoder
```

6. Create the directories where the remaining logdecoroot partitions will be mounted.

Command:

```
mkdir /var/netwitness/logdecoder/logdecoder<#>
```

Example:

Storage Unit	Command
First	mkdir /var/netwitness/logdecoder/logdecoder0

Storage Unit	Command
Second	mkdir /var/netwitness/logdecoder/logdecoder1
	mkdir /var/netwitness/logdecoder/logdecoder2

7. Mount all remaining decoroot partitions to the appropriate folder locations (/dev/sdd1, /dev/sde1, /dev/sdf1)*.

Command:

```
mount <partition> <mounting location>
```

Example:

Storage Unit	Command
First	mount /dev/sdd1 /var/netwitness/logdecoder/logdecoder0

Storage Unit	Command
Second	mount /dev/sde1 /var/netwitness/logdecoder/logdecoder1
	mount /dev/sdf1 /var/netwitness/logdecoder/logdecoder2

- With all logdecoroot partitions mounted, create all other mount point folders.

Command:

```
mkdir /var/netwitness/decoder/<db>
mkdir /var/netwitness/decoder/decoder<#>/<db>
```

Example:

Storage Unit	Command
First	mkdir /var/netwitness/logdecoder/ index
	mkdir /var/netwitness/logdecoder/ sessiondb
	mkdir /var/netwitness/logdecoder/ metadb
	mkdir /var/netwitness/logdecoder/ packetdb
	mkdir /var/netwitness/logdecoder/logdecoder0/ index0
	mkdir /var/netwitness/logdecoder/logdecoder0/ sessiondb0
	mkdir /var/netwitness/logdecoder/logdecoder0/ metadb0
	mkdir /var/netwitness/logdecoder/logdecoder0/ packetdb0
Storage Unit	Command
Second	mkdir /var/netwitness/logdecoder/logdecoder1/ index1
	mkdir /var/netwitness/logdecoder/logdecoder1/ sessiondb1
	mkdir /var/netwitness/logdecoder/logdecoder1/ metadb1
	mkdir /var/netwitness/logdecoder/logdecoder1/ packetdb1
	mkdir /var/netwitness/logdecoder/logdecoder2/ index2
	mkdir /var/netwitness/logdecoder/logdecoder2/ sessiondb2
	mkdir /var/netwitness/logdecoder/logdecoder2/ metadb2
	mkdir /var/netwitness/logdecoder/logdecoder2/ packetdb2

- Run the **blkid** command to list all the block device partitions on the first block device (/dev/sdc)*. Capture the output from the command as the UUIDs listed will be needed in later steps (see screenshot for example output). It is important to note which UUIDs are associated with which PARTLABELs from the output. These PARTLABELs directly correspond with the PART LABEL column from the [LogDecoder-DB-Mount/Directory-Names Table](#) section of this document.

```
blkid | egrep '/dev/sdc' | awk '{print $1, $2, $5}'
```

```
/dev/sdc1: UUID="1a770e78-3890-f23a-12d4-cf234537a90e2" PARTLABEL="logdecoroot"
/dev/sdc2: UUID="2bc45786-1c42-a2af-4d32-009245f23a76d2" PARTLABEL="logdeconex"
/dev/sdc3: UUID="5ee4ff28-f23a-113a-0924-009245456b76d2" PARTLABEL="logdecosessiondb"
/dev/sdc4: UUID="4fa46678-c4ef-3890-00cf-d1e4367770e2" PARTLABEL="logdecometadb"
/dev/sdc5: UUID="5ee43678-a2af-efc1-652c-00924efc1b76d" PARTLABEL="logdecopacketdb"
```

10. Run the **blkid** command again to list all the block device partitions on the second block device (*/dev/sdd*)*. Just like in the previous step, capture the same output for later use.

```
blkid | egrep '/dev/sdd' | awk '{print $1, $2, $5}'
```

```
/dev/sdd1:  UUID="1af45678-12c4-f23a-12d4-cf234537a90e2"  PARTLABEL="logdecoroot"  
/dev/sdd2:  UUID="2bc45786-1c42-a2af-4d32-23ff4389a912a"  PARTLABEL="logdecondex"  
/dev/sdd3:  UUID="5ee43678-f23a-113a-ff21-00924537ab76d2"  PARTLABEL="logdecosessiondb"  
/dev/sdd4:  UUID="4fa46678-c4ef-3890-00cf-d123e537770e2"  PARTLABEL="logdecometadb"  
/dev/sdd5:  UUID="5ee43678-0924-efc1-652c-00924efc1b76d"  PARTLABEL="logdecopacketdb"
```

11. Run the **blkid** command again on the third and fourth block devices (*/dev/sde*, */dev/sdf*)* just like in the previous two steps, and capture the UUIDs from the output.

```
blkid | egrep '/dev/sde' | awk '{print $1, $2, $5}'  
blkid | egrep '/dev/sdf' | awk '{print $1, $2, $5}'
```

12. Backup the operating system's *File System Table* file (*/etc/fstab*). Since this file will be edited shortly, it is good practice to have a backup if anything goes wrong. This command will make a copy of the **fstab** and place it into the */root* directory, called **fstab.backup**.

```
cp /etc/fstab /root/fstab.backup
```

13. Edit the **fstab** (/etc/fstab) file and add the new UUIDs and mount locations for all block device partitions to the bottom of the file.

The example below shows how to use the UUIDs from the **blkid** outputs and combine them with the associate mount points. The customer should substitute their own UUIDs when adding the entries to the file. Each entry must be on its own line to function. Ten entries will be created for each storage unit attached to the appliance.

If unsure how to edit the fstab file, stop and request assistance from NetWitness Support.

If unsure of which mount points go to which block device partitions, review the outputs from the previous three steps along with the [LogDecoder-DB-Mount/Directory-Names Table](#).

Example of **fstab** Lines

```
UUID= bb10f793-f141-4e74-8021-cd953bdece83 /var/netwitness/logdecoder xfs noatime,nosuid 1 2
UUID=ef431ad6-0b5c-4ca0-9092-8a635d2b33ea /var/netwitness/logdecoder/index xfs noatime,nosuid 1 2
UUID=f4ca8ff4-a87b-434b-add8-fe1a918aabf4 /var/netwitness/logdecoder/sessiondb xfs noatime,nosuid 1 2
UUID=3116ab5c-e813-43e4-a503-200d49e529fc /var/netwitness/logdecoder/metadb xfs noatime,nosuid 1 2
UUID=872ab261-c592-4416-9bdf-33727e4bb03d /var/netwitness/logdecoder/packetdb xfs noatime,nosuid 1 2

UUID= 8e95d2fc-2dbc-4961-88ed-2ce3c4eb07b4 /var/netwitness/logdecoder/logdecoder0 xfs noatime,nosuid 1 2
UUID=94f05b71-5199-4b93-80cc-21fa1216f2eb /var/netwitness/logdecoder/logdecoder0/index0 xfs noatime,nosuid 1 2
UUID=8d40cf96-1a73-4fb7-8f59-5997e53415ec /var/netwitness/logdecoder/logdecoder0/sessiondb0 xfs noatime,nosuid 1 2
UUID=38661bd8-f690-4ed3-ba4c-0560161df106 /var/netwitness/logdecoder/logdecoder0/metadb0 xfs noatime,nosuid 1 2
UUID=b0dd95b5-ee01-49fa-aed8-baf838410750 /var/netwitness/logdecoder/logdecoder0/packetdb0 xfs noatime,nosuid 1 2
```

If looking for more information about the relationship between the UUID, PARTLABEL, mount points and the **fstab**, please see the [Relationship between UUID/PARTLABEL and Mount Points](#) section in the Appendix.

14. Reload the system daemon, confirm that the **fstab** is read successfully, and double check the partitions are visible. If any errors are produced, this means the changes done to the **fstab** file are not correct. Either a misspelling has occurred, or one or more device/directory locations are incorrect/missing. Use the backed up **fstab.backup** file if needed.

Do not reboot the device until any mounting issues are corrected, as this could cause the system to fail to reboot properly. If an issue occurs and you are unsure how to proceed, stop and contact NetWitness Support for assistance.

```
systemctl daemon-reload
mount -a
df -hP
```

/dev/sdc1	20G	303M	20G	2%	/var/netwitness/logdecoder
/dev/sdc2	30G	248M	30G	1%	/var/netwitness/logdecoder/index
/dev/sdc3	4.5T	32G	4.5T	1%	/var/netwitness/logdecoder/sessiondb
/dev/sdc4	13T	90G	13T	1%	/var/netwitness/logdecoder/metadb
/dev/sdc5	131T	931G	130T	1%	/var/netwitness/logdecoder/packetdb
/dev/sdd1	20G	303M	20G	2%	/var/netwitness/logdecoder/logdecoder0
/dev/sdd2	30G	248M	30G	1%	/var/netwitness/logdecoder/logdecoder0/index0
/dev/sdd3	4.5T	32G	4.5T	1%	/var/netwitness/logdecoder/logdecoder0/sessiondb0
/dev/sdd4	13T	90G	13T	1%	/var/netwitness/logdecoder/logdecoder0/metadb0
/dev/sdd5	131T	931G	130T	1%	/var/netwitness/logdecoder/logdecoder0/packetdb0

Updating the Log Decoder Database Configuration (First & Second)

This section will provide step by step instructions for how to adjust the Log Decoder's storage configuration to properly take advantage of the newly daisy-chained Ultrastar60 storage units. To see screenshots of the NetWitness UI, go to the Appendix under [Updating Log Decoder Database Configuration Screenshots](#).

1. Log into the NetWitness UI and go to the Explore View for the Log Decoder with the Ultrastar60 daisy-chained storage attached.
2. Once in Explore View, navigate to the database folder and select the config folder.

- Once in the config folder, look for the following nodes and replace their entries with the ones provided below. The semi-colon in the **Value** column is required, and each cell is a single continuous line.

Storage Unit	Node	Value
First & Second	packet.dir	/var/netwitness/logdecoder/packetdb=423 TB; /var/netwitness/logdecoder/logdecoder0/packetdb0=423 TB; /var/netwitness/logdecoder/logdecoder1/packetdb1=423 TB; /var/netwitness/logdecoder/logdecoder2/packetdb2=423 TB;
	meta.dir	/var/netwitness/logdecoder/metadb=16.5 TB; /var/netwitness/logdecoder/logdecoder0/metadb0=16.5 TB; /var/netwitness/logdecoder/logdecoder1/metadb1=16.5 TB; /var/netwitness/logdecoder/logdecoder2/metadb2=16.5 TB;
	session.dir	/var/netwitness/logdecoder/sessiondb=1 TB; /var/netwitness/logdecoder/logdecoder0/sessiondb0=1 TB; /var/netwitness/logdecoder/logdecoder1/sessiondb1=1 TB; /var/netwitness/logdecoder/logdecoder2/sessiondb2=1 TB;
	packet.free.space.min	1 TB
	meta.free.space.min	600 GB
	session.free.space.min	30 GB

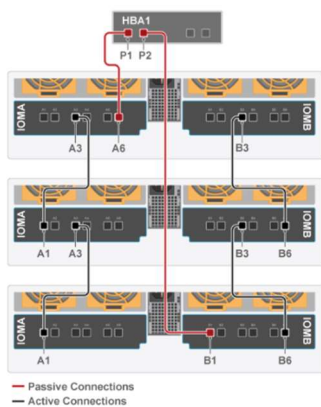
- Next go to the index folder and select its config folder. Replace the entry with the following from the **Value** column.

Storage Unit	Node	Value
First & Second	Index.dir	/var/netwitness/logdecoder/index=29 GB; /var/netwitness/logdecoder/logdecoder0/index0=29 GB; /var/netwitness/logdecoder/logdecoder1/index1=29 GB; /var/netwitness/logdecoder/logdecoder2/index2=29 GB;

- If no further Ultrastar60 storage units are to be configured, reboot the Log Decoder server for all changes to take full effect. All configurations are now complete for adding two Ultrastar60 daisy-chained storage units to the Log Decoder.
- If one or more additional Ultrastar60 units are to be added to the Log Decoder, please continue to the [Configuring Daisy-Chained Log Decoder Storage \(Third & Fourth\)](#) section.

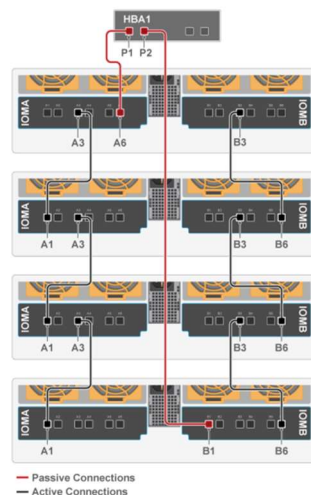
Configuring Daisy-Chained Log Decoder Storage (Third & Fourth)

Three Enclosures: One Host with a Single HBA



Source Device	Destination Device
Host1: P1	Enclosure1: IOMA, A6
Host1: P2	Enclosure3: IOMB, B1
Enclosure1: IOMA, A3	Enclosure2: IOMA, A1
Enclosure1: IOMB, B3	Enclosure2: IOMB, B6
Enclosure2: IOMA, A3	Enclosure3: IOMA, A1
Enclosure2: IOMB, B3	Enclosure3: IOMB, B6

Four Enclosures: One Host with a Single HBA



Source Device	Destination Device
Host1: P1	Enclosure1: IOMA, A6
Host1: P2	Enclosure4: IOMB, B1
Enclosure1: IOMA, A3	Enclosure2: IOMA, A1
Enclosure1: IOMB, B3	Enclosure2: IOMB, B6
Enclosure2: IOMA, A3	Enclosure3: IOMA, A1
Enclosure2: IOMB, B3	Enclosure3: IOMB, B6
Enclosure3: IOMA, A3	Enclosure4: IOMA, A1
Enclosure3: IOMB, B3	Enclosure4: IOMB, B6

This section provides the steps needed to attach a third and fourth Ultrastar60 storage unit to the Log Decoder. Please keep in mind that setting up the third and fourth storage units will be identical.

NetWitness currently supports two, three, and four storage units in a daisy-chain configuration. The instructions in this section will show how to configure the third and fourth at the same time. If there is only a third storage unit, do not run any of the fourth storage unit commands.

Please review the [How to Start Configuring Ultrastar60 Storage](#) section to get the output for the Enclosure, Connector, Physical ID, and LUN ID of the additional storage. The steps in this section are almost identical to setting up the first and second Ultrastar60 units on the Log Decoder. As such the steps will be more abbreviated but should align closely with the ones provided in the [Configuring Daisy-Chained Log Decoder Storage \(First & Second\)](#).

Important: When using the `arconf` utility to create block devices, the LUN IDs of the storage units must be entered in reverse order (0,59 0,58 0,57 ... 0,2 0,1 0,0), instead of normal order (0,0 0,1 0,2 ... 0,57 0,58 0,59). Pay particular attention when entering the commands of this section due to this reversal.

Creating Ultrastar60 Block Devices (Third & Fourth)

1. Perform a block device listing on the appliance. This command looks for any devices that are of size 442 TB. This will catch any configured Ultrastar60 storage. Only the first two Ultrastar60 storage units should appear as they are already configured.

```
lsblk | grep '442T'
```

2. Retrieve the Physical ID and LUN ID output for Connector 0 (CN0/First Storage Unit) and Connector 1 (CN1/Second Storage Unit), per the [How to Start Configuring Ultrastar60 Storage](#) section.
3. Break up the 120 LUNS into four groups of 30, remembering to use reverse order. If only configuring a third storage unit, there will be only 60 total LUNs.

Storage Unit	Groups	LUN IDs
Third	Group 1	0,149 0,148 0,147 ... 0,122 0,121 0,120
	Group 2	0,179 0,178 0,177 ... 0,152 0,151 0,150
Storage Unit	Groups	LUN IDs
Fourth	Group 3	0,209 0,208 0,207 ... 0,182 0,181 0,180
	Group 4	0,239 0,238 0,237 ... 0,212 0,211 0,210

4. Create the first RAID 50 Block Device 0 using the Group 1 LUNs from the **third** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,149 0,148 0,147 0,146 0,145 0,144 0,143 0,142 0,141 0,140 0,139 0,138
0,137 0,136 0,135 0,134 0,133 0,132 0,131 0,130 0,129 0,128 0,127 0,126
0,125 0,124 0,123 0,122 0,121 0,120 noprompt
```

5. Create the second RAID 50 Block Device 1 using Group 2 LUNs from the **third** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,179 0,178 0,177 0,176 0,175 0,174 0,173 0,172 0,171 0,170 0,169 0,168
0,167 0,166 0,165 0,164 0,163 0,162 0,161 0,160 0,159 0,158 0,157 0,156
0,155 0,154 0,153 0,152 0,151 0,150 noprompt
```

6. If there is a **fourth** storage unit, create the third RAID 50 Block Device 2 using Group 3 LUNs.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,209 0,208 0,207 0,206 0,205 0,204 0,203 0,202 0,201 0,200 0,199 0,198
0,197 0,196 0,195 0,194 0,193 0,192 0,191 0,190 0,189 0,188 0,187 0,186
0,185 0,184 0,183 0,182 0,181 0,180 noprompt
```

7. If there is a **fourth** storage unit, create the fourth RAID 50 Block Device 3 using Group 4 LUNs.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,239 0,238 0,237 0,236 0,235 0,234 0,233 0,232 0,231 0,230 0,229 0,228
0,227 0,226 0,225 0,224 0,223 0,222 0,221 0,220 0,219 0,218 0,217 0,216
0,215 0,214 0,213 0,212 0,211 0,210 noprompt
```

8. Run the **lsblk** command to find the newly created Ultrastar60 block devices. The command should return two or four new block devices of 442 TB in size, depending on the number of storage units being configured, similar to the output below.

```
lsblk | grep `442T`
```

	sdc	8:32	0	442T	0	disk
	sdd	8:32	0	442T	0	disk
	sde	8:32	0	442T	0	disk
	sdf	8:32	0	442T	0	disk
Third	sdg	8:32	0	442T	0	disk
	sdh	8:32	0	442T	0	disk
Fourth Storage	sdi	8:32	0	442T	0	disk
	sdj	8:32	0	442T	0	disk

Sometimes the block devices will not display immediately, this is normal. A rescan of the bus can be performed using the following command. Once it is complete, rerun the **lsblk** command above.

```
arcconf rescan 1
```

Create the Partitions and Mount Points for the New Block Devices (Third and Fourth)

This section will show how to create new partitions and set up mount points using the newly created block devices on the third and fourth Ultrastar60 storage units.

Please use the block devices that were discovered while running the **lsblk** command in the previous section. The discovered devices should take the place of `/dev/sdg` and `/dev/sdh` for storage unit three, and `/dev/sdi` and `/dev/sdj` for storage unit four in the following examples. The [LogDecoder-DB-Mount/Directory-Names Table](#) will be utilized for this section and should be referenced as needed.

If the appliance has been rebooted since the block device creation was performed, rerun the **lsblk** command again as the block device names may have changed. The operating system is responsible for device naming and can change the names between reboots.

The block device of `/dev/sdg`, `/dev/sdh`, `/dev/sdi`, and `/dev/sdj` will be used in examples throughout this section.* These are only place holders for the block devices within the customer's appliance.

If there are any questions, please stop and contact NetWitness Customer Support for assistance.

1. Set the disk label type to gpt for the new block devices (`/dev/sdg`, `/dev/sdh`, `/dev/sdi`, `/dev/sdj`)*.

Command:

```
parted -s <block device> mklabel gpt
```

Example:

Storage Unit	Command
Third	<code>parted -s /dev/sdg mklabel gpt</code>
	<code>parted -s /dev/sdh mklabel gpt</code>
Storage Unit	Command
Fourth	<code>parted -s /dev/sdi mklabel gpt</code>
	<code>parted -s /dev/sdj mklabel gpt</code>

2. Create the required Decoder partitions utilizing the **PART LABEL** column from the [LogDecoder-DB-Mount/Directory-Names Table](#). Run all the commands below for each block device (`/dev/sdc`, `/dev/sdd`, `/dev/sde`, `/dev/sdf`)* to create the required partitions.

Command:

```
parted --align minimal <block device> mkpart <part label> xfs start_size  
end_size
```

Example:

Third Storage Unit	parted --align minimal /dev/sdg mkpart logdecoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdg mkpart logdecondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdg mkpart logdecosessiondb xfs 50GiB 1.5TiB
	parted --align minimal /dev/sdg mkpart logdecometadb xfs 1.5TiB 50TiB
	parted --align minimal /dev/sdg mkpart logdecopacketdb xfs 50TiB 100%
	parted --align minimal /dev/sdh mkpart logdecoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdh mkpart logdecondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdh mkpart logdecosessiondb xfs 50GiB 1.5TiB
	parted --align minimal /dev/sdh mkpart logdecometadb xfs 1.5TiB 50TiB
	parted --align minimal /dev/sdh mkpart logdecopacketdb xfs 50TiB 100%
Fourth Storage Unit	parted --align minimal /dev/sdi mkpart logdecoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdi mkpart logdecondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdi mkpart logdecosessiondb xfs 50GiB 1.5TiB
	parted --align minimal /dev/sdi mkpart logdecometadb xfs 1.5TiB 50TiB
	parted --align minimal /dev/sdi mkpart logdecopacketdb xfs 50TiB 100%
	parted --align minimal /dev/sdj mkpart logdecoroot xfs 3MiB 20GiB
	parted --align minimal /dev/sdj mkpart logdecondex xfs 20GiB 50GiB
	parted --align minimal /dev/sdj mkpart logdecosessiondb xfs 50GiB 1.5TiB
	parted --align minimal /dev/sdj mkpart logdecometadb xfs 1.5TiB 50TiB
	parted --align minimal /dev/sdj mkpart logdecopacketdb xfs 50TiB 100%

- Confirm that all partitions have been created correctly for all block devices (/dev/sdc, /dev/sdd, /dev/sde, /dev/sdf)* by running the parted print command for all block devices. The output will be the same for all block devices.

Command:

```
parted <block device> print
```

Example:

Storage Unit	Command
Third	parted /dev/sdg print
	parted /dev/sdh print
Storage Unit	Command
Fourth	parted /dev/sdi print
	parted /dev/sdj print

All block device outputs should look identical to the output below.

```
Model: Adaptec LOGICAL VOLUME (scsi)
Disk /dev/sdg: 486TB
Sector Size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File System  Name                  Flags
  1      3146kB  21.5GB  21.5GB                logdecoroot
  2      21.5GB  53.7GB  32.2GB                logdecondex
  3      53.7GB  4948GB  4894GB                logdecosessiondb
  4      4948GB  18.7TB  13.7TB                logdecometadb
  5      18.7TB  486TB   467TB                logdecopacketdb
```

4. Create the file systems on all the newly created Log Decoder partitions. The numbers under the **Number** column from the previous step's print outputs should be appended to the block device name as shown in the commands below.

Command:

```
mkfs.xfs -f <block device><partition Number>
```

Example:

Storage Unit	Block Device	Command
Third	/dev/sdg*	mkfs.xfs -f /dev/sdg1
		mkfs.xfs -f /dev/sdg2
		mkfs.xfs -f /dev/sdg3
		mkfs.xfs -f /dev/sdg4
		mkfs.xfs -f /dev/sdg5
	/dev/sdh*	mkfs.xfs -f /dev/sdh1
		mkfs.xfs -f /dev/sdh2
		mkfs.xfs -f /dev/sdh3
		mkfs.xfs -f /dev/sdh4
		mkfs.xfs -f /dev/sdh5

Storage Unit	Block Device	Command
Fourth	/dev/sdi*	mkfs.xfs -f /dev/sdi1
		mkfs.xfs -f /dev/sdi2
		mkfs.xfs -f /dev/sdi3
		mkfs.xfs -f /dev/sdi4
		mkfs.xfs -f /dev/sdi5
	/dev/sdj*	mkfs.xfs -f /dev/sdj1
		mkfs.xfs -f /dev/sdj2
		mkfs.xfs -f /dev/sdj3
		mkfs.xfs -f /dev/sdj4
		mkfs.xfs -f /dev/sdj5

5. Confirm that the first logdecoroot is mounted to /var/netwitness/logdecoder before continuing. This partition acts as the container for all other mounted storage partitions and must be available.

Command:

```
df -h
```

/dev/sdc1	20G	303M	20G	2%	/var/netwitness/logdecoder
/dev/sdc2	30G	248M	30G	1%	/var/netwitness/logdecoder/index
/dev/sdc3	4.5T	32G	4.5T	1%	/var/netwitness/logdecoder/sessiondb
/dev/sdc4	13T	90G	13T	1%	/var/netwitness/logdecoder/metadb
/dev/sdc5	131T	931G	130T	1%	/var/netwitness/logdecoder/packetdb
/dev/sdd1	20G	303M	20G	2%	/var/netwitness/logdecoder/logdecoder0
/dev/sdd2	30G	248M	30G	1%	/var/netwitness/logdecoder/logdecoder0/index0
/dev/sdd3	4.5T	32G	4.5T	1%	/var/netwitness/logdecoder/logdecoder0/sessiondb0
/dev/sdd4	13T	90G	13T	1%	/var/netwitness/logdecoder/logdecoder0/metadb0
/dev/sdd5	131T	931G	130T	1%	/var/netwitness/logdecoder/logdecoder0/packetdb0

6. Create the directories mount points for the new decoroot partitions.

Command:

```
mkdir /var/netwitness/logdecoder/logdecoder<#>
```

Example:

Storage Unit	Command
Third	mkdir /var/netwitness/logdecoder/logdecoder3
	mkdir /var/netwitness/logdecoder/logdecoder4
Storage Unit	Command
Fourth	mkdir /var/netwitness/logdecoder/logdecoder5
	mkdir /var/netwitness/logdecoder/logdecoder6

7. Mount the logdecoroot partitions to the appropriate folder locations (/dev/sdg1, /dev/sdh1, /dev/sdi1, /dev/sdj)*.

Command:

```
mount <partition> <mounting location>
```

Example:

Storage Unit	Command
Third	mount /dev/sdg1 /var/netwitness/logdecoder/logdecoder3
	mount /dev/sdh1 /var/netwitness/logdecoder/logdecoder4
Storage Unit	Command
Fourth	mount /dev/sdi1 /var/netwitness/logdecoder/logdecoder5
	mount /dev/sdj1 /var/netwitness/logdecoder/logdecoder6

8. With all the new logdecoroot partitions mounted, create all other mount point folders.

Command:

```
mkdir /var/netwitness/logdecoder/logdecoder<#>/<db>
```

Example:

Storage Unit	Command
Third	mkdir /var/netwitness/logdecoder/logdecoder3/index3
	mkdir /var/netwitness/logdecoder/logdecoder3/sessiondb3
	mkdir /var/netwitness/logdecoder/logdecoder3/metadb3
	mkdir /var/netwitness/logdecoder/logdecoder3/packetdb3
	mkdir /var/netwitness/logdecoder/logdecoder4/index4
	mkdir /var/netwitness/logdecoder/logdecoder4/sessiondb4
	mkdir /var/netwitness/logdecoder/logdecoder4/metadb4
	mkdir /var/netwitness/logdecoder/logdecoder4/packetdb4

Storage Unit	Command
Fourth	mkdir /var/netwitness/logdecoder/logdecoder5/index5
	mkdir /var/netwitness/logdecoder/logdecoder5/sessiondb5
	mkdir /var/netwitness/logdecoder/logdecoder5/metadb5
	mkdir /var/netwitness/logdecoder/logdecoder5/packetdb5
	mkdir /var/netwitness/logdecoder/logdecoder6/index6
	mkdir /var/netwitness/logdecoder/logdecoder6/sessiondb6
	mkdir /var/netwitness/logdecoder/logdecoder6/metadb6
	mkdir /var/netwitness/logdecoder/logdecoder6/packetdb6

9. Run the **blkid** command to list all the block device partitions on each of the block devices (/dev/sdg, /dev/sdh, /dev/sdi, /dev/sdj)*. Capture the output from all the commands as the UUIDs listed will be needed in later steps (see screenshot for an example of a single output). It is important to note which UUIDs are associated with which PARTLABELs from the output. These PARTLABELs directly correspond with the PART LABEL column in the [LogDecoder-DB-Mount/Directory-Names Table](#) section of this document.

Storage Unit	Command
Third	blkid egrep '/dev/sdg' awk '{print \$1, \$2, \$5}'
	blkid egrep '/dev/sdh' awk '{print \$1, \$2, \$5}'

Storage Unit	Command
Fourth	blkid egrep '/dev/sdi' awk '{print \$1, \$2, \$5}'
	blkid egrep '/dev/sdj' awk '{print \$1, \$2, \$5}'

```
/dev/sdg1: UUID="1af45678-12c4-f23a-12d4-cf234537a90e2" PARTLABEL="logdecoroot"
/dev/sdg2: UUID="2bc45786-1c42-a2af-4d32-23ff4537a912a" PARTLABEL="logdecondex"
/dev/sdg3: UUID="3de45878-2c41-113a-ff21-cf23ff37a11a2" PARTLABEL="logdecosessiondb"
/dev/sdg4: UUID="4fa46678-c4ef-3890-00cf-d123e537770e2" PARTLABEL="logdecometadb"
/dev/sdg5: UUID="5ee43678-3321-efc1-652c-00924537ab76d" PARTLABEL="logdecopacketdb"
```

10. Backup the operating system's *File System Table* file (*/etc/fstab*). Since this file will be edited shortly it is good practice to have a backup if anything goes wrong. This command will make a copy of the **fstab** and place it into the */root* directory, called **fstab.backup**.

```
cp /etc/fstab /root/fstab.backup
```

11. Edit the **fstab** (*/etc/fstab*) file and add the new UUIDs and mount locations for all block device partitions to the bottom of the file.

The example below shows how to use the UUIDs from the **blkid** outputs and combine them with the associate mount points. The customer should substitute their own UUIDs when adding the entries to the file. Each entry must be on its own line to function. Ten entries will be created for each storage unit attached to the appliance.

If unsure how to edit the **fstab** file, stop and request assistance from NetWitness Support.

If unsure of which mount points go to which block device partitions, review the **lsblk** outputs along with the [LogDecoder-DB-Mount/Directory-Names Table](#) for the appropriate storage.

Example of **fstab** Lines

```
UUID=bb10f793-f141-4e74-8021-cd953bdece83 /var/netwitness/logdecoder/logdecoder3 xfs noatime,nosuid 1 2
UUID=ef431ad6-0b5c-4ca0-9092-8a635d2b33ea /var/netwitness/logdecoder/logdecoder3/index3 xfs noatime,nosuid 1 2
UUID=f4ca8ff4-a87b-434b-add8-fe1a918aabf4 /var/netwitness/logdecoder/logdecoder3/sessiondb3 xfs noatime,nosuid 1 2
UUID=3116ab5c-e813-43e4-a503-200d49e529fc /var/netwitness/logdecoder/logdecoder3/metadb3 xfs noatime,nosuid 1 2
UUID=872ab261-c592-4416-9bdf-33727e4bb03d /var/netwitness/logdecoder/logdecoder3/packetdb3 xfs noatime,nosuid 1 2

UUID=8e95d2fc-2dbc-4961-88ed-2ce3c4eb07b4 /var/netwitness/logdecoder/logdecoder4 xfs noatime,nosuid 1 2
UUID=94f05b71-5199-4b93-80cc-21fa1216f2eb /var/netwitness/logdecoder/logdecoder4/index4 xfs noatime,nosuid 1 2
UUID=8d40cf96-1a73-4fb7-8f59-5997e53415ec /var/netwitness/logdecoder/logdecoder4/sessiondb4 xfs noatime,nosuid 1 2
UUID=38661bd8-f690-4ed3-ba4c-0560161df106 /var/netwitness/logdecoder/logdecoder4/metadb4 xfs noatime,nosuid 1 2
UUID=b0dd95b5-ee01-49fa-aed8-baf838410750 /var/netwitness/logdecoder/logdecoder4/packetdb4 xfs noatime,nosuid 1 2
```

If looking for more information about the relationship between the UUID, PARTLABEL, mount points and the **fstab**, please see the [Relationship between UUID/PARTLABEL and Mount Points](#) section in the Appendix.

12. Reload the system daemon, confirm that the **fstab** is read successfully, and double check the partitions are visible. If any errors are produced, this means the changes done to the **fstab** file are not correct. Either a misspelling has occurred, or one or more device/directory locations are incorrect/missing. Use the backed up **fstab.backup** file if needed.

Do not reboot the device until any mounting issues are corrected as this could cause the system to fail to reboot properly. If an issue occurs and you are unsure how to proceed, stop and contact NetWitness Support for assistance.

```
systemctl daemon-reload
mount -a
df -hP
```

Third Storage Unit

/dev/sdg1	20G	303M	20G	2%	/var/netwitness/logdecoder/logdecoder3
/dev/sdg2	30G	248M	30G	1%	/var/netwitness/logdecoder/logdecoder3/index3
/dev/sdg3	4.5T	32G	4.5T	1%	/var/netwitness/logdecoder/logdecoder3/sessiondb3
/dev/sdg4	13T	90G	13T	1%	/var/netwitness/logdecoder/logdecoder3/metadb3
/dev/sdg5	131T	931G	130T	1%	/var/netwitness/logdecoder/logdecoder3/packetdb3
/dev/sdh1	20G	303M	20G	2%	/var/netwitness/logdecoder/logdecoder4
/dev/sdh2	30G	248M	30G	1%	/var/netwitness/logdecoder/logdecoder4/index4
/dev/sdh3	4.5T	32G	4.5T	1%	/var/netwitness/logdecoder/logdecoder4/sessiondb4
/dev/sdh4	13T	90G	13T	1%	/var/netwitness/logdecoder/logdecoder4/metadb4
/dev/sdh5	131T	931G	130T	1%	/var/netwitness/logdecoder/logdecoder4/packetdb4

Fourth Storage Unit

/dev/sdi1	20G	303M	20G	2%	/var/netwitness/logdecoder/logdecoder5
/dev/sdi2	30G	248M	30G	1%	/var/netwitness/logdecoder/logdecoder5/index5
/dev/sdi3	4.5T	32G	4.5T	1%	/var/netwitness/logdecoder/logdecoder5/sessiondb5
/dev/sdi4	13T	90G	13T	1%	/var/netwitness/logdecoder/logdecoder5/metadb5
/dev/sdi5	131T	931G	130T	1%	/var/netwitness/logdecoder/logdecoder5/packetdb5
/dev/sdj1	20G	303M	20G	2%	/var/netwitness/logdecoder/logdecoder6
/dev/sdj2	30G	248M	30G	1%	/var/netwitness/logdecoder/logdecoder6/index6
/dev/sdj3	4.5T	32G	4.5T	1%	/var/netwitness/logdecoder/logdecoder6/sessiondb6
/dev/sdj4	13T	90G	13T	1%	/var/netwitness/logdecoder/logdecoder6/metadb6
/dev/sdj5	131T	931G	130T	1%	/var/netwitness/logdecoder/logdecoder6/packetdb6

Updating the Decoder Database Configuration (Third & Fourth)

This section will provide step by step instructions for how to adjust the Log Decoder's storage configuration to properly take advantage of the newly daisy-chained Ultrastar60 storage units. To see screenshots of the NetWitness UI, go to the Appendix under [Updating Log Decoder Database Configuration Screenshots](#).

1. Log into the NetWitness UI and go to the Explore View for the Decoder with the Ultrastar60 daisy-chained storage attached.
2. Once in Explore View, navigate to the database folder and select the config folder.
3. Once in the config folder, look for the following nodes. The first and second Ultrastar60 partition entries should already be in these nodes. Add the entries in the **Value** column to the end of the existing entry. If only the Third storage is present, use only those entries. If the Third and Fourth storage are present, combine the **Value** column entries in the table and then add them to the node. Make sure that none of the semi-colons are missing as they are required.

Storage Unit	Node	Value
Third & Fourth	packet.free.space.min	1 TB
	meta.free.space.min	600 GB
	session.free.space.min	30 GB

Storage Unit	Node	Value
Third	packet.dir	/var/netwitness/logdecoder/logdecoder3/packetdb3=390 TB; /var/netwitness/logdecoder/logdecoder4/packetdb4=390 TB;
	meta.dir	/var/netwitness/logdecoder/logdecoder3/metadb3=48.5 TB; /var/netwitness/logdecoder/logdecoder4/metadb4=48.5 TB;
	session.dir	/var/netwitness/logdecoder/logdecoder3/sessiondb3=1.5 TB; /var/netwitness/logdecoder/logdecoder4/sessiondb4=11.5 TB;

Storage Unit	Node	Value
Fourth	packet.dir	/var/netwitness/logdecoder/logdecoder5/packetdb5=390 TB; /var/netwitness/logdecoder/logdecoder6/packetdb6=390 TB;
	meta.dir	/var/netwitness/logdecoder/logdecoder5/metadb5=48.5 TB; /var/netwitness/logdecoder/logdecoder6/metadb6=48.5 TB;
	session.dir	/var/netwitness/logdecoder/logdecoder5/sessiondb5=1.5 TB; /var/netwitness/logdecoder/logdecoder6/sessiondb6=1.5 TB;

4. Next go to the index folder and select its config folder. The first and second Ultrastar60 partition entries should already be in this node. Add the entries in the **Value** column to the end of the existing entry. If only the Third storage is present, use only that entry. If the Third and Fourth storage are present, combine the **Value** column entries in the table and then add them to the node. Make sure that none of the semi-colons are missing as they are required.

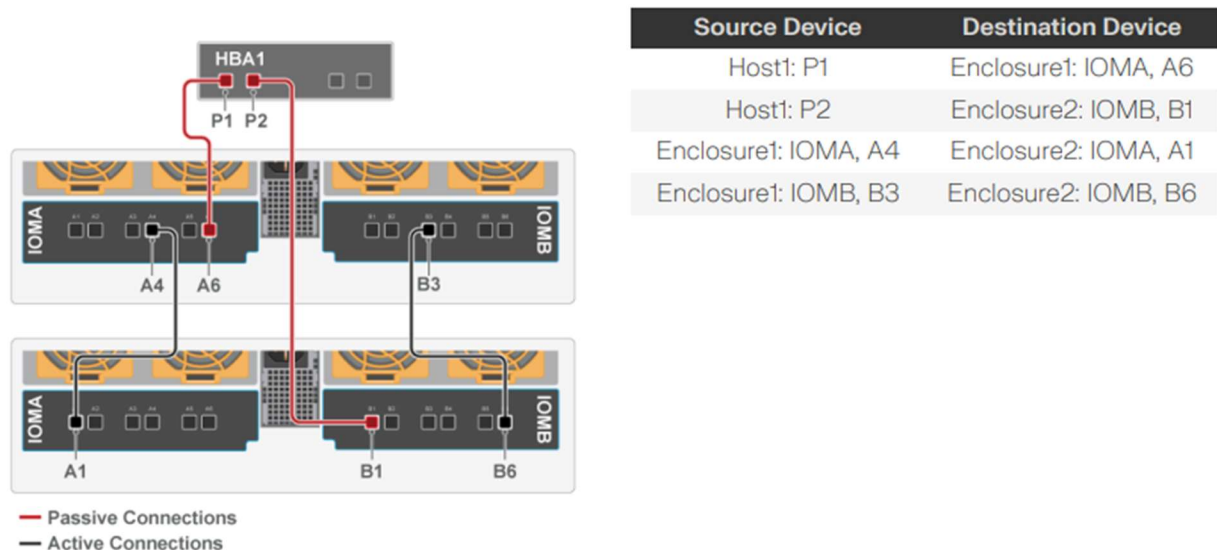
Storage Unit	Node	Value
Third	index.dir	/var/netwitness/logdecoder/logdecoder3/index3=29 GB; /var/netwitness/logdecoder/logdecoder4/index4=29 GB;

Storage Unit	Node	Value
Fourth	index.dir	/var/netwitness/logdecoder/logdecoder5/index5=29 GB; /var/netwitness/logdecoder/logdecoder6/index6=29 GB;

5. Once all the configurations have been completed, reboot the Log Decoder server for all changes to take full effect.
6. All steps have now been completed for daisy-chaining storage to the Log Decoder.

Configuring Daisy-Chained Network Hybrid Storage (First & Second)

Two Enclosures: One Host with a Single HBA



This section will go through the steps needed to set up and connect two Ultrastar60 storage units to a Network Hybrid. Please review the [How to Start Configuring Ultrastar60 Storage](#) section before proceeding as the output for Enclosure, Connector, Physical ID, and LUN ID will be required. A maximum of two Ultrastar60 storage units can be connected to a Hybrid appliance.

Important: The `arcconf` utility requires that the LUN IDs captured earlier must be entered in reverse order (0,59 0,58 0,57 ... 0,2 0,1 0,0), instead of normal order (0,0 0,1 0,2 ... 0,57 0,58 0,59). Pay particular attention when entering the commands of this section due to this reversal.

Creating Ultrastar60 Block Devices (First & Second)

1. Perform a block device listing on the appliance. This command looks for any devices that are of size 442 TB. This will catch any configured Ultrastar60 storage. Since all connected units are unconfigured, no block devices should be returned.

```
lsblk | grep '442T'
```

2. Retrieve the Physical ID and LUN ID output for Connector 0 (CN0/First Storage Unit) and Connector 1 (CN1/Second Storage Unit), per the [How to Start Configuring Ultrastar60 Storage](#) section.

3. Break up the 120 LUNS into four groups of 30, remembering to use reverse order.

Storage Unit	Groups	LUN IDs
First	Group 1	0,29 0,28 0,27 ... 0,2 0,1 0,0
	Group 2	0,59 0,58 0,57 ... 0,32 0,31 0,30
Storage Unit	Groups	LUN IDs
Second	Group 3	0,89 0,88 0,87 ... 0,62 0,61 0,60
	Group 4	0,119 0,118 0,117 ... 0,92 0,91 0,90

4. Create the first RAID 50 Block Device 0 using the Group 1 LUNs from the **first** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,29 0,28 0,27 0,26 0,25 0,24 0,23 0,22 0,21 0,20 0,19 0,18 0,17 0,16 0,15
0,14 0,13 0,12 0,11 0,10 0,9 0,8 0,7 0,6 0,5 0,4 0,3 0,2 0,1 0,0 noprompt
```

5. Create the second RAID 50 Block Device 1 using Group 2 LUNs from the **first** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,59 0,58 0,57 0,56 0,55 0,54 0,53 0,52 0,51 0,50 0,49 0,48 0,47 0,46 0,45
0,44 0,43 0,42 0,41 0,40 0,39 0,38 0,37 0,36 0,35 0,34 0,33 0,32 0,31 0,30
noprompt
```

6. Create the third RAID 50 Block Device 2 using Group 3 LUNs from the **second** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,89 0,88 0,87 0,86 0,85 0,84 0,83 0,82 0,81 0,80 0,79 0,78 0,77 0,76 0,75
0,74 0,73 0,72 0,71 0,70 0,69 0,68 0,67 0,66 0,65 0,64 0,63 0,62 0,61 0,60
noprompt
```

7. Create the fourth RAID 50 Block Device 3 using Group 4 LUNs from the **second** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,119 0,118 0,117 0,116 0,115 0,114 0,113 0,112 0,111 0,110 0,109 0,108
0,107 0,106 0,105 0,104 0,103 0,102 0,101 0,100 0,99 0,98 0,97 0,96 0,95
0,94 0,93 0,92 0,91 0,90 noprompt
```

8. Run the **lsblk** command to find the newly created Ultrastar60 block devices. The command should return four devices of 442 TB in size, similar to the output below.

```
lsblk | grep '442T'
```

First Storage	sda	8:32	0	442T	0	disk
	sdb	8:32	0	442T	0	disk
Second Storage	sdc	8:32	0	442T	0	disk
	sdd	8:32	0	442T	0	disk

Sometimes the block devices will not display immediately. This is normal. A rescan of the bus can be performed using the below command. Once it is complete, rerun the **lsblk** command above.

```
arcconf rescan 1
```

Create the Partitions and Mount Points for the New Block Devices (First and Second)

This section will show how to create new partitions and set up mount points using the newly created block devices on the first and second Ultrastar60 storage units.

Please use the block devices that were discovered while running the **lsblk** command in the previous section. The discovered devices should take the place of `/dev/sda` and `/dev/sdb`, for the first storage unit, and `/dev/sdc` and `/dev/sdd`, for the second storage unit in the following examples. The [NetworkHybrid-DB-Mount/Directory-Names Table](#) will be utilized for this section and should be referenced as needed.

If the appliance has been rebooted since the block device creation was performed, rerun the **lsblk** command again as the block device names may have changed. The operating system is responsible for device naming and can change the names between reboots.

The block device of `/dev/sda`, `/dev/sdb`, `/dev/sdc`, and `/dev/sdd` will be used in examples throughout this section.* These are only place holders for the block devices within the customer's appliance.

If there are any questions, please stop and contact NetWitness Customer Support for assistance.

1. Set the disk label type to gpt for the new block devices (`/dev/sda`, `/dev/sdb`, `/dev/sdc`, `/dev/sdd`)*.

Command:

```
parted -s <block device> mklabel gpt
```

Example:

Storage Unit	Command
First	<code>parted -s /dev/sda mklabel gpt</code>
	<code>parted -s /dev/sdb mklabel gpt</code>
Storage Unit	Command
Second	<code>parted -s /dev/sdc mklabel gpt</code>
	<code>parted -s /dev/sdd mklabel gpt</code>

2. Create the required Network Hybrid partitions utilizing the **PART LABEL** column from the [NetworkHybrid-DB-Mount/Directory-Names Table](#). Run all the commands below for each block device (`/dev/sda`, `/dev/sdb`, `/dev/sdc`, `/dev/sdd`)* to create the required partitions.

Command:

```
parted --align minimal <block device> mkpart <part label> xfs start_size  
end_size
```

Example:

First Storage Unit	parted --align minimal /dev/sda mkpart concsessiondb xfs 3MiB 5TiB
	parted --align minimal /dev/sda mkpart concmetadb xfs 5TiB 80TiB
	parted --align minimal /dev/sda mkpart decopacketdb xfs 80TiB 100%
	parted --align minimal /dev/sdb mkpart concsessiondb xfs 3MiB 5TiB
	parted --align minimal /dev/sdb mkpart concmetadb xfs 5TiB 80TiB
	parted --align minimal /dev/sdb mkpart decopacketdb xfs 80TiB 100%
Second Storage Unit	parted --align minimal /dev/sdc mkpart concsessiondb xfs 3MiB 5TiB
	parted --align minimal /dev/sdc mkpart concmetadb xfs 5TiB 80TiB
	parted --align minimal /dev/sdc mkpart decopacketdb xfs 80TiB 100%
	parted --align minimal /dev/sdd mkpart concsessiondb xfs 3MiB 5TiB
	parted --align minimal /dev/sdd mkpart concmetadb xfs 5TiB 80TiB
	parted --align minimal /dev/sdd mkpart decopacketdb xfs 80TiB 100%

- Confirm that all partitions have been created correctly for all block devices (/dev/sda, /dev/sdb, /dev/sdc, /dev/sdd)* by running the parted print command for all block devices. The output will be the same for all block devices.

Command:

parted <block device> print

Example:

Storage Unit	Command
First	parted /dev/sda print
	parted /dev/sdb print
Storage Unit	Command
Second	parted /dev/sdc print
	parted /dev/sdd print

All block device outputs should look identical to the output below.

```
Model: Adaptec LOGICAL VOLUME (scsi)
Disk /dev/sda: 486TB
Sector Size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File System	Name	Flags
1	3146kB	5498GB	5498GB		concsessiondb	
2	5498GB	88.0TB	82.5TB		concmetadb	
3	88.0TB	486TB	398TB		decopacketdb	

4. Create the file systems on all the newly created Network Hybrid partitions. The numbers under the **Number** column from the previous step's print outputs should be appended to the block device name as shown in the commands below.

Command:

```
mkfs.xfs -f <block device><partition number>
```

Example:

Storage Unit	Block Device	Command
First	/dev/sda*	mkfs.xfs -f /dev/sda1
		mkfs.xfs -f /dev/sda2
		mkfs.xfs -f /dev/sda3
	/dev/sdb*	mkfs.xfs -f /dev/sdb1
		mkfs.xfs -f /dev/sdb2
		mkfs.xfs -f /dev/sdb3

Storage Unit	Block Device	Command
Second	/dev/sdc*	mkfs.xfs -f /dev/sdc1
		mkfs.xfs -f /dev/sdc2
		mkfs.xfs -f /dev/sdc3
	/dev/sdd*	mkfs.xfs -f /dev/sdd1
		mkfs.xfs -f /dev/sdd2
		mkfs.xfs -f /dev/sdd3

5. Create the directories where the new partitions will be mounted.

Command:

```
mkdir <folder location>
```

Example:

Storage Unit	Command
First	mkdir /var/netwitness/decoder/packetdb0
	mkdir /var/netwitness/concentrator/sessiondb0
	mkdir /var/netwitness/concentrator/metadb0
	mkdir /var/netwitness/decoder/packetdb1
	mkdir /var/netwitness/concentrator/sessiondb1
	mkdir /var/netwitness/concentrator/metadb1

Storage Unit	Command
Second	mkdir /var/netwitness/decoder/packetdb2
	mkdir /var/netwitness/concentrator/sessiondb2
	mkdir /var/netwitness/concentrator/metadb2
	mkdir /var/netwitness/decoder/packetdb3
	mkdir /var/netwitness/concentrator/sessiondb3
	mkdir /var/netwitness/concentrator/metadb3

6. Run the **blkid** command to list all the block device partitions on the configured block devices (*/dev/sda, /dev/sdb, /dev/sdc, /dev/sdd*)*. Capture the output from the commands as the UUIDs listed will be needed in later steps (see screenshot for example output). It is important to note which UUIDs are associated with which PARTLABELs from the output. These PARTLABELs directly correspond with the PART LABEL column from the [NetworkHybrid-DB-Mount/Directory-Names Table](#) section of this document.

Storage Unit	Command
First	<code>blkid egrep '/dev/sda' awk '{print \$1, \$2, \$5}'</code>
	<code>blkid egrep '/dev/sdb' awk '{print \$1, \$2, \$5}'</code>

Storage Unit	Command
Second	<code>blkid egrep '/dev/sdc' awk '{print \$1, \$2, \$5}'</code>
	<code>blkid egrep '/dev/sdd' awk '{print \$1, \$2, \$5}'</code>

```
/dev/sda1: UUID="1af45678-12c4-f23a-12d4-cf234537a90e2" PARTLABEL="concsessiondb"
/dev/sda2: UUID="2bc45786-1c42-a2af-4d32-23ff4389a912a" PARTLABEL="concmetadb"
/dev/sda3: UUID="5ee43678-f23a-113a-ff21-00924537ab76d2" PARTLABEL="decopacketdb"
```

7. Backup the operating system's *File System Table* file (*/etc/fstab*). Since this file will be edited shortly it is good practice to have a backup if anything goes wrong. This command will make a copy of the **fstab** and place it into the */root* directory, called **fstab.backup**.

```
cp /etc/fstab /root/fstab.backup
```

8. Edit the **fstab** (/etc/fstab) file and add the new UUIDs and mount locations for all block device partitions to the bottom of the file.

The example below shows how to use the UUIDs from the **blkid** outputs and combine them with the associate mount points. The customer should substitute their own UUIDs when adding the entries to the file. Each entry must be on its own line to function. Three entries will be created for each storage unit attached to the appliance.

If unsure how to edit the **fstab** file, stop and request assistance from NetWitness Support.

If unsure of which mount points go to which block device partitions, review the **lsblk** outputs along with the [NetworkHybrid-DB-Mount/Directory-Names Table](#) for the appropriate storage unit.

Example of **fstab** Lines

```
UUID= bb10f793-f141-4e74-8021-cd953bdece83 /var/netwitness/decoder/packetdb0 xfs noatime,nosuid 1
UUID=f4ca8ff4-a87b-434b-add8-fe1a918aabf4 /var/netwitness/concentrator/sessiondb0 xfs noatime,nosuid 1 2
UUID=3116ab5c-e813-43e4-a503-200d49e529fc /var/netwitness/concentrator/metadb0 xfs noatime,nosuid 1 2

UUID= 8e95d2fc-2dbc-4961-88ed-2ce3c4eb07b4 /var/netwitness/decoder/packetdb1 xfs noatime,nosuid 1 2
UUID=8d40cf96-1a73-4fb7-8f59-5997e53415ec /var/netwitness/concentrator/sessiondb1 xfs noatime,nosuid 1 2
UUID=38661bd8-f690-4ed3-ba4c-0560161df106 /var/netwitness/concentrator/metadb1 xfs noatime,nosuid 1 2
```

If looking for more information about the relationship between the UUID, PARTLABEL, mount points and the **fstab**, please see the [Relationship between UUID/PARTLABEL and Mount Points](#) section in the Appendix.

9. Reload the system daemon, confirm that the **fstab** is read successfully, and double check the partitions are visible. If any errors are produced, this means the changes done to the **fstab** file are not correct. Either a misspelling has occurred, or one or more device/directory locations are incorrect/missing. Use the backed up **fstab.backup** file if needed.

Do not reboot the device until any mounting issues are corrected as this could cause the system to fail to reboot properly. If an issue occurs and you are unsure how to proceed, stop and contact NetWitness Support for assistance.

```
systemctl daemon-reload
mount -a
df -hP
```

/dev/sda1	5.0T	36G	5.0T	1%	/var/netwitness/concentrator/sessiondb0
/dev/sda2	75T	538G	75T	1%	/var/netwitness/concentrator/metadb0
/dev/sda3	363T	2.6T	360T	1%	/var/netwitness/decoder/packetdb0

Updating the Network Hybrid Database Configuration (First & Second)

This section will provide step by step instructions for how to adjust the Network Hybrid's storage configuration to properly take advantage of the newly daisy-chained Ultrastar60 storage units. To see screenshots of the NetWitness UI, go to the Appendix under [Updating Network Hybrid Database Configuration Screenshots](#).

1. Log into the NetWitness UI and go to the Explore View for the Decoder service on the Network Hybrid with the Ultrastar60 daisy-chained storage attached.
2. Once in Explore View, navigate to the database folder and select the config folder.
3. Once in the config folder, look for the following node. There will already be an entry in the node. Combine the **Value** column entries in the table together and then add them to the end of the node entry. Making sure that none of the semi-colons are missing as they are required.

Storage Unit	Node	Value
First	packet.dir	/var/netwitness/decoder/packetdb0=355 TB; /var/netwitness/decoder/packetdb1=355 TB;

Storage Unit	Node	Value
Second	packet.dir	/var/netwitness/decoder/packetdb2=355 TB; /var/netwitness/decoder/packetdb3=355 TB;

4. Restart the Decoder service via the system console to accept the configuration changes.

```
systemctl restart nwdecoder
```

5. Go back to the NetWitness UI, select the Concentrator service of the Network Hybrid. Then go to the Explore View for the service.
6. Once in the Explore View, navigate to the database folder and select the config folder.

7. Once in the config folder, look for the following nodes. There will already be an entry in the nodes. Combine the **Value** column entries for the same node together and then add them to the end of the node entry. Make sure that none of the semi-colons are missing as they are required.

Storage Unit	Node	Value
First	meta.dir	/var/netwitness/concentrator/metadb0=74 TB; /var/netwitness/concentrator/metadb1=74 TB;
	session.dir	/var/netwitness/concentrator/sessiondb0=4.95 TB; /var/netwitness/concentrator/sessiondb1=4.95 TB;

Storage Unit	Node	Value
Second	meta.dir	/var/netwitness/concentrator/metadb2=74 TB; /var/netwitness/concentrator/metadb3=74 TB;
	session.dir	/var/netwitness/concentrator/sessiondb2=4.95 TB; /var/netwitness/concentrator/sessiondb3=4.95 TB;

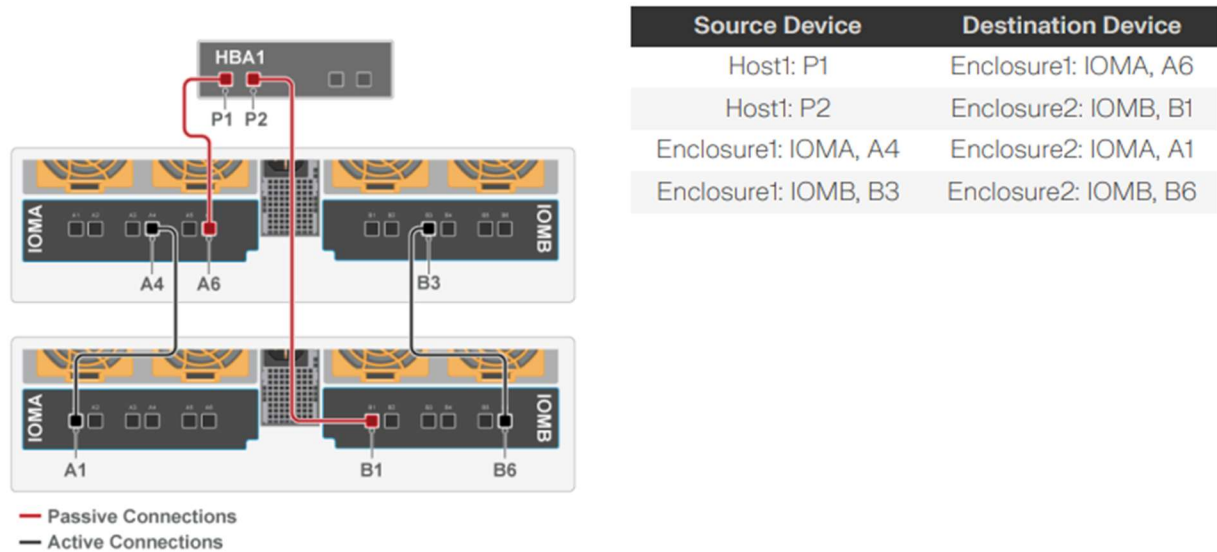
8. Restart the Concentrator service via the system console to accept the configuration changes.

```
systemctl restart nwconcentrator
```

9. All steps have now been completed for the configuration of the Network Hybrid.

Configuring Daisy-Chained Log Hybrid Storage (First & Second)

Two Enclosures: One Host with a Single HBA



This section will go through the steps needed to set up and connect two Ultrastar60 storage units to a Log Hybrid. Please review the [How to Start Configuring Ultrastar60 Storage](#) section before proceeding as the output for Enclosure, Connector, Physical ID, and LUN ID will be required. A maximum of two Ultrastar60 storage units can be connected to a Hybrid appliance.

Important: The `arcconf` utility requires that the LUN IDs captured earlier must be entered in reverse order (0,59 0,58 0,57 ... 0,2 0,1 0,0), instead of normal order (0,0 0,1 0,2 ... 0,57 0,58 0,59). Pay particular attention when entering the commands of this section due to this reversal.

Creating Ultrastar60 Block Devices (First & Second)

1. Perform a block device listing on the appliance. This command looks for any devices that are of size 442 TB. This will catch any configured Ultrastar60 storage. Since all connected units are unconfigured, no block devices should be returned.

```
lsblk | grep '442T'
```

2. Retrieve the Physical ID and LUN ID output for Connector 0 (CN0/First Storage Unit) and Connector 1 (CN1/Second Storage Unit), per the [How to Start Configuring Ultrastar60 Storage](#) section.

3. Break up the 120 LUNS into four groups of 30, remembering to use reverse order.

Storage Unit	Groups	LUN IDs
First	Group 1	0,29 0,28 0,27 ... 0,2 0,1 0,0
	Group 2	0,59 0,58 0,57 ... 0,32 0,31 0,30
Storage Unit	Groups	LUN IDs
Second	Group 3	0,89 0,88 0,87 ... 0,62 0,61 0,60
	Group 4	0,119 0,118 0,117 ... 0,92 0,91 0,90

4. Create the first RAID 50 Block Device 0 using the Group 1 LUNs from the **first** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,29 0,28 0,27 0,26 0,25 0,24 0,23 0,22 0,21 0,20 0,19 0,18 0,17 0,16 0,15
0,14 0,13 0,12 0,11 0,10 0,9 0,8 0,7 0,6 0,5 0,4 0,3 0,2 0,1 0,0 noprompt
```

5. Create the second RAID 50 Block Device 1 using Group 2 LUNs from the **first** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,59 0,58 0,57 0,56 0,55 0,54 0,53 0,52 0,51 0,50 0,49 0,48 0,47 0,46 0,45
0,44 0,43 0,42 0,41 0,40 0,39 0,38 0,37 0,36 0,35 0,34 0,33 0,32 0,31 0,30
noprompt
```

6. Create the third RAID 50 Block Device 2 using Group 3 LUNs from the **second** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,89 0,88 0,87 0,86 0,85 0,84 0,83 0,82 0,81 0,80 0,79 0,78 0,77 0,76 0,75
0,74 0,73 0,72 0,71 0,70 0,69 0,68 0,67 0,66 0,65 0,64 0,63 0,62 0,61 0,60
noprompt
```

7. Create the fourth RAID 50 Block Device 3 using Group 4 LUNs from the **second** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,119 0,118 0,117 0,116 0,115 0,114 0,113 0,112 0,111 0,110 0,109 0,108
0,107 0,106 0,105 0,104 0,103 0,102 0,101 0,100 0,99 0,98 0,97 0,96 0,95
0,94 0,93 0,92 0,91 0,90 noprompt
```

8. Run the **lsblk** command to find the newly created Ultrastar60 block devices. The command should return four devices of 442 TB in size, similar to the output below.

```
lsblk | grep '442T'
```

First Storage	sda	8:32	0	442T	0	disk
	sdb	8:32	0	442T	0	disk
Second Storage	sdc	8:32	0	442T	0	disk
	sdd	8:32	0	442T	0	disk

Sometimes the block devices will not display immediately. This is normal. A rescan of the bus can be performed using the following command. Once it is complete, rerun the **lsblk** command above.

```
arcconf rescan 1
```

Create the Partitions and Mount Points for the New Block Devices (First & Second)

This section will show how to create new partitions and set up mount points using the newly created block devices on the first and second Ultrastar60 storage units.

Please use the block devices that were discovered while running the **lsblk** command in the previous section. The discovered devices should take the place of `/dev/sda` and `/dev/sdb`, for the first storage unit , and `/dev/sdc` and `/dev/sdd`, for the second storage unit in the following examples. The [LogHybrid-DB-Mount/Directory-Names Table](#) will be utilized for this section and should be referenced as needed.

If the appliance has been rebooted since the block device creation was performed, rerun the **lsblk** command again as the block device names may have changed. The operating system is responsible for device naming and can change the names between reboots.

The block device of `/dev/sda`, `/dev/sdb`, `/dev/sdc`, and `/dev/sdd` will be used in examples throughout this section.* These are only place holders for the block devices within the customer's appliance.

If there are any questions, please stop and contact NetWitness Customer Support for assistance.

1. Set the disk label type to gpt for the new block devices (`/dev/sda`, `/dev/sdb`, `/dev/sdc`, `/dev/sdd`)*.

Command:

```
parted -s <block device> mklabel gpt
```

Example:

Storage Unit	Command
First	<code>parted -s /dev/sda mklabel gpt</code>
	<code>parted -s /dev/sdb mklabel gpt</code>
Storage Unit	Command
Second	<code>parted -s /dev/sdc mklabel gpt</code>
	<code>parted -s /dev/sdd mklabel gpt</code>

2. Create the required Log Hybrid partitions utilizing the **PART LABEL** column from the [LogHybrid-DB-Mount/Directory-Names Table](#). Run all the commands below for each block device (`/dev/sda`, `/dev/sdb`, `/dev/sdc`, `/dev/sdd`)* to create the required partitions.

Command:

```
parted --align minimal <block device> mkpart <part label> xfs start_size  
end_size
```

Example:

First Storage Unit	parted --align minimal /dev/sda mkpart concsessionondb xfs 3MiB 13TiB
	parted --align minimal /dev/sda mkpart concmetagdb xfs 13TiB 218TiB
	parted --align minimal /dev/sda mkpart logdecopacketdb xfs 218TiB 100%
	parted --align minimal /dev/sdb mkpart concsessionondb xfs 3MiB 13TiB
	parted --align minimal /dev/sdb mkpart concmetagdb xfs 13TiB 218TiB
	parted --align minimal /dev/sdb mkpart logdecopacketdb xfs 218TiB 100%
Second Storage Unit	parted --align minimal /dev/sdc mkpart concsessionondb xfs 3MiB 13TiB
	parted --align minimal /dev/sdc mkpart concmetagdb xfs 13TiB 218TiB
	parted --align minimal /dev/sdc mkpart logdecopacketdb xfs 218TiB 100%
	parted --align minimal /dev/sdd mkpart concsessionondb xfs 3MiB 13TiB
	parted --align minimal /dev/sdd mkpart concmetagdb xfs 13TiB 218TiB
	parted --align minimal /dev/sdd mkpart logdecopacketdb xfs 218TiB 100%

- Confirm that all partitions have been created correctly for all block devices (/dev/sda, /dev/sdb, /dev/sdc, /dev/sdd)* by running the parted print command for all block devices. The output will be the same for all block devices.

Command:

parted <block device> print

Example:

Storage Unit	Command
First	parted /dev/sda print
	parted /dev/sdb print
Storage Unit	Command
Second	parted /dev/sdc print
	parted /dev/sdd print

All block device outputs should look identical to the output below.

```

Model: Adaptec LOGICAL VOLUME (scsi)
Disk /dev/sda: 486TB
Sector Size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File System  Name                Flags
  1      3146kB  5498GB  5498GB                concsessionondb
  2      5498GB  88.0TB  82.5TB                concmetagdb
  3      88.0TB  486TB   398TB                logdecopacketdb

```

4. Create the file systems on all the newly created LogHybrid partitions. The numbers under the **Number** column from the previous step's print outputs should be appended to the block device name as shown in the commands below.

Command:

```
mkfs.xfs -f <block device><partition number>
```

Example:

Storage Unit	Block Device	Command
First	/dev/sda*	mkfs.xfs -f /dev/sda1
		mkfs.xfs -f /dev/sda2
		mkfs.xfs -f /dev/sda3
	/dev/sdb*	mkfs.xfs -f /dev/sdb1
		mkfs.xfs -f /dev/sdb2
		mkfs.xfs -f /dev/sdb3
Storage Unit	Block Device	Command
Second	/dev/sdc*	mkfs.xfs -f /dev/sdc1
		mkfs.xfs -f /dev/sdc2
		mkfs.xfs -f /dev/sdc3
	/dev/sdd*	mkfs.xfs -f /dev/sdd1
		mkfs.xfs -f /dev/sdd2
		mkfs.xfs -f /dev/sdd3

5. Create the directories where the new partitions will be mounted.

Command:

```
mkdir <folder location>
```

Example:

Storage Unit	Command
First	mkdir /var/netwitness/logdecoder/packetdb0
	mkdir /var/netwitness/concentrator/sessiondb0
	mkdir /var/netwitness/concentrator/metadb0
	mkdir /var/netwitness/logdecoder/packetdb1
	mkdir /var/netwitness/concentrator/sessiondb1
	mkdir /var/netwitness/concentrator/metadb1
Storage Unit	Command
Second	mkdir /var/netwitness/logdecoder/packetdb2
	mkdir /var/netwitness/concentrator/sessiondb2
	mkdir /var/netwitness/concentrator/metadb2
	mkdir /var/netwitness/logdecoder/packetdb3
	mkdir /var/netwitness/concentrator/sessiondb3
	mkdir /var/netwitness/concentrator/metadb3

- Run the **blkid** command to list all the block device partitions on the configured block devices (*/dev/sda, /dev/sdb, /dev/sdc, /dev/sdd*)*. Capture the output from the commands as the UUIDs listed will be needed in later steps (see screenshot for example output). It is important to note which UUIDs are associated with which PARTLABELs from the output. These PARTLABELs directly correspond with the PART LABEL column from the [LogHybrid-DB-Mount/Directory-Names Table](#) section of this document.

Storage Unit	Command
First	blkid egrep '/dev/sda' awk '{print \$1, \$2, \$5}' blkid egrep '/dev/sdb' awk '{print \$1, \$2, \$5}'

Storage Unit	Command
Second	blkid egrep '/dev/sdc' awk '{print \$1, \$2, \$5}' blkid egrep '/dev/sdd' awk '{print \$1, \$2, \$5}'

```
/dev/sda1: UUID="1af45678-12c4-f23a-12d4-cf234537a90e2" PARTLABEL="concsessiondb"
/dev/sda2: UUID="2bc45786-1c42-a2af-4d32-23ff4389a912a" PARTLABEL="concmatab"
/dev/sda3: UUID="5ee43678-f23a-113a-ff21-00924537ab76d2" PARTLABEL="logdecopacketdb"
```

- Backup the operating system's *File System Table* file (*/etc/fstab*). Since this file will be edited shortly it is good practice to have a backup if anything goes wrong. This command will make a copy of the **fstab** and place it into the */root* directory, called **fstab.backup**.

```
cp /etc/fstab /root/fstab.backup
```

8. Edit the **fstab** (/etc/fstab) file and add the new UUIDs and mount locations for all block device partitions to the bottom of the file.

The example below shows how to use the UUIDs from the **blkid** outputs and combine them with the associate mount points. The customer should substitute their own UUIDs when adding the entries to the file. Each entry must be on its own line to function. Three entries will be created for each storage unit attached to the appliance.

If unsure how to edit the **fstab** file, stop and request assistance from NetWitness Support.

If unsure of which mount points go to which block device partitions, review the **lsblk** outputs along with the [LogHybrid-DB-Mount/Directory-Names Table](#) for the appropriate storage unit.

Example of **fstab** Lines

```
UUID= bb10f793-f141-4e74-8021-cd953bdece83 /var/netwitness/logdecoder/packetdb0 xfs noatime,nosuid 1
UUID=f4ca8ff4-a87b-434b-add8-fe1a918aabf4 /var/netwitness/concentrator/sessiondb0 xfs noatime,nosuid 1 2
UUID=3116ab5c-e813-43e4-a503-200d49e529fc /var/netwitness/concentrator/metadb0 xfs noatime,nosuid 1 2

UUID= 8e95d2fc-2dbc-4961-88ed-2ce3c4eb07b4 /var/netwitness/logdecoder/packetdb1 xfs noatime,nosuid 1 2
UUID=8d40cf96-1a73-4fb7-8f59-5997e53415ec /var/netwitness/concentrator/sessiondb1 xfs noatime,nosuid 1 2
UUID=38661bd8-f690-4ed3-ba4c-0560161df106 /var/netwitness/concentrator/metadb1 xfs noatime,nosuid 1 2
```

If looking for more information about the relationship between the UUID, PARTLABEL, mount points and the **fstab**, please see the [Relationship between UUID/PARTLABEL and Mount Points](#) section in the Appendix.

9. Reload the system daemon, confirm that the **fstab** is read successfully, and double check the partitions are visible. If any errors are produced, this means the changes done to the **fstab** file are not correct. Either a misspelling has occurred, or one or more device/directory locations are incorrect/missing. Use the backed up **fstab.backup** file if needed.

Do not reboot the device until any mounting issues are corrected as this could cause the system to fail to reboot properly. If an issue occurs and you are unsure how to proceed, stop and contact NetWitness Support for assistance.

```
systemctl daemon-reload
mount -a
df -hP
```

First Storage Unit					
/dev/sda1	13T	93G	13T	1%	/var/netwitness/concentrator/sessiondb0
/dev/sda2	205T	1.5G	204T	1%	/var/netwitness/concentrator/metadb0
/dev/sda3	225T	1.6T	223T	1%	/var/netwitness/logdecoder/packetdb0
/dev/sdb1	13T	93G	13T	1%	/var/netwitness/concentrator/sessiondb1
/dev/sdb2	205T	1.5G	204T	1%	/var/netwitness/concentrator/metadb1
/dev/sdb3	225T	1.6T	223T	1%	/var/netwitness/logdecoder/packetdb1

Second Storage Unit					
/dev/sdc1	13T	93G	13T	1%	/var/netwitness/concentrator/sessiondb2
/dev/sdc2	205T	1.5G	204T	1%	/var/netwitness/concentrator/metadb2
/dev/sdc3	225T	1.6T	223T	1%	/var/netwitness/logdecoder/packetdb2
/dev/sdd1	13T	93G	13T	1%	/var/netwitness/concentrator/sessiondb3
/dev/sdd2	205T	1.5G	204T	1%	/var/netwitness/concentrator/metadb3
/dev/sdd3	225T	1.6T	223T	1%	/var/netwitness/logdecoder/packetdb3

Updating the Log Hybrid Database Configuration (First & Second)

This section will provide step by step instructions for how to adjust the Network Hybrid's storage configuration to properly take advantage of the newly daisy-chained Ultrastar60 storage units. To see screenshots of the NetWitness UI, go to the Appendix under [Updating Log Hybrid Database Configuration Screenshots](#).

1. Log into the NetWitness UI and go to the Explore View for the Log Decoder service on the Log Hybrid with the Ultrastar60 daisy-chained storage attached.
2. Once in Explore View, navigate to the database folder and select the config folder.

- Once in the config folder, look for the following node. There will already be an entry in the node. Combine the **Value** column entries in the table together and then add them to the end of the node entry. Make sure that none of the semi-colons are missing as they are required.

Storage Unit	Node	Value
First	packet.dir	/var/netwitness/logdecoder/packetdb0=218 TB; /var/netwitness/logdecoder/packetdb1=218 TB;
Second	packet.dir	/var/netwitness/logdecoder/packetdb2=218 TB; /var/netwitness/logdecoder/packetdb3=218 TB;

- Restart the Log Decoder service via the system console to accept the configuration changes.

```
systemctl restart nwlogdecoder
```

- Go back to the NetWitness UI, select the Concentrator service of the Log Hybrid. Then go to the Explore View for the service.
- Once in the Explore View, navigate to the database folder and select the config folder.
- Once in the config folder, look for the following nodes. There will already be an entry in the nodes. Combine the **Value** column entries for the same node together and then add them to the end of the node entry. Making sure that none of the semi-colons are missing as they are required.

Storage Unit	Node	Value
First	meta.dir	/var/netwitness/concentrator/metadb0=204.9 TB; /var/netwitness/concentrator/metadb1=204.9 TB;
	session.dir	/var/netwitness/concentrator/sessiondb0=12.9 TB; /var/netwitness/concentrator/sessiondb1=12.9 TB;
Second	meta.dir	/var/netwitness/concentrator/metadb2=204.9TB; /var/netwitness/concentrator/metadb3=204.9 TB;
	session.dir	/var/netwitness/concentrator/sessiondb2=12.9 TB; /var/netwitness/concentrator/sessiondb3=12.9 TB;

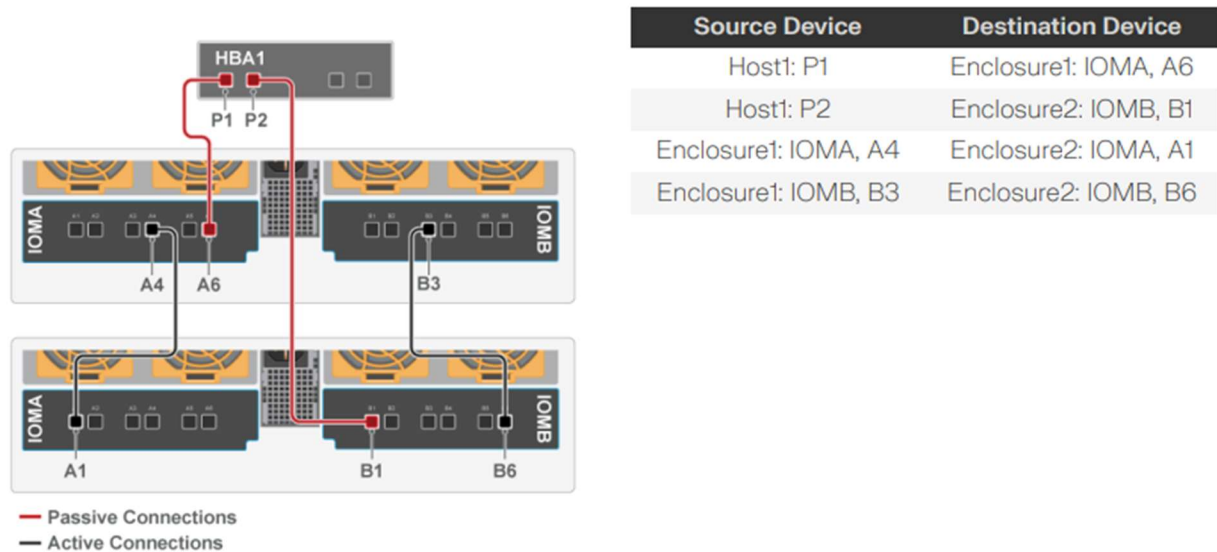
- Restart the Concentrator service via the system console to accept the configuration changes.

```
systemctl restart nwconcentrator
```

- All steps have now been completed for the configuration of the Log Hybrid.

Configure Daisy-Chained Archiver Storage (First & Second)

Two Enclosures: One Host with a Single HBA



This section will go through the steps needed to set up and connect two Ultrastar60 storage units to an Archiver. Please review the [How to Start Configuring Ultrastar60 Storage](#) section before proceeding as the output for Enclosure, Connector, Physical ID, and LUN ID will be required. A maximum of four Ultrastar60 storage units can be connected to an Archiver appliance.

Important: The `arcconf` utility requires that the LUN IDs captured earlier must be entered in reverse order (0,59 0,58 0,57 ... 0,2 0,1 0,0), instead of normal order (0,0 0,1 0,2 ... 0,57 0,58 0,59). Pay particular attention when entering the commands of this section due to this reversal.

Creating Ultrastar60 Block Devices (First & Second)

1. Perform a block device listing on the appliance. This command looks for any devices that are of size 442 TB. This will catch any configured Ultrastar60 storage. Since all connected units are unconfigured, no block devices should be returned.

```
lsblk | grep '442T'
```

2. Retrieve the Physical ID and LUN ID output for Connector 0 (CN0/First Storage Unit) and Connector 1 (CN1/Second Storage Unit), per the [How to Start Configuring Ultrastar60 Storage](#) section.

3. Break up the 120 LUNS into four groups of 30, remembering to use reverse order.

Storage Unit	Groups	LUN IDs
First	Group 1	0,29 0,28 0,27 ... 0,2 0,1 0,0
	Group 2	0,59 0,58 0,57 ... 0,32 0,31 0,30
Storage Unit	Groups	LUN IDs
Second	Group 3	0,89 0,88 0,87 ... 0,62 0,61 0,60
	Group 4	0,90 0,91 0,92 ... 117 0,118 0,119

4. Create the first RAID 50 Block Device 0 using the Group 1 LUNs from the **first** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,29 0,28 0,27 0,26 0,25 0,24 0,23 0,22 0,21 0,20 0,19 0,18 0,17 0,16 0,15
0,14 0,13 0,12 0,11 0,10 0,9 0,8 0,7 0,6 0,5 0,4 0,3 0,2 0,1 0,0 noprompt
```

5. Create the second RAID 50 Block Device 1 using Group 2 LUNs from the **first** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,59 0,58 0,57 0,56 0,55 0,54 0,53 0,52 0,51 0,50 0,49 0,48 0,47 0,46 0,45
0,44 0,43 0,42 0,41 0,40 0,39 0,38 0,37 0,36 0,35 0,34 0,33 0,32 0,31 0,30
noprompt
```

6. Create the third RAID 50 Block Device 2 using Group 3 LUNs from the **second** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,89 0,88 0,87 0,86 0,85 0,84 0,83 0,82 0,81 0,80 0,79 0,78 0,77 0,76 0,75
0,74 0,73 0,72 0,71 0,70 0,69 0,68 0,67 0,66 0,65 0,64 0,63 0,62 0,61 0,60
noprompt
```

7. Create the fourth RAID 50 Block Device 3 using Group 4 LUNs from the **second** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,119 0,118 0,117 0,116 0,115 0,114 0,113 0,112 0,111 0,110 0,109 0,108
0,107 0,106 0,105 0,104 0,103 0,102 0,101 0,100 0,99 0,98 0,97 0,96 0,95
0,94 0,93 0,92 0,91 0,90 noprompt
```

8. Run the **lsblk** command to find the newly created Ultrastar60 block devices. The command should return four devices of 442 TB in size, similar to the output below.

```
lsblk | grep '442T'
```

First Storage	sdc	8:32	0	442T	0	disk
	sdd	8:32	0	442T	0	disk
Second Storage	sde	8:32	0	442T	0	disk
	sdf	8:32	0	442T	0	disk

Sometimes the block devices will not display immediately. This is normal. A rescan of the bus can be performed using the following command. Once it is complete, rerun the **lsblk** command above.

```
arcconf rescan 1
```

Create the Partitions and Mount Points for the New Block Devices (First & Second)

This section will show how to create new partitions and set up mount points using the newly created block devices on the first and second Ultrastar60 storage units.

Please use the block devices that were discovered while running the **lsblk** command in the previous section. The devices discovered should take the place of `/dev/sdc` and `/dev/sdd`, for the first storage unit. The devices `/dev/sde` and `/dev/sdf` should also be replaced for the second storage unit, in the following examples. The [Archiver-DB-Mount/Directory-Names Table](#) will be utilized for this section and should be referenced as needed.

If the appliance has been rebooted since the block device creation was performed, rerun the **lsblk** command again as the block device names may have changed. The operating system is responsible for device naming and can change the names between reboots.

The block device of `/dev/sdc`, `/dev/sdd`, `/dev/sde`, and `/dev/sdf` will be used in examples throughout this section.* These are only place holders for the block devices within the customer's appliance.

If there are any questions, please stop and contact NetWitness Customer Support for assistance.

1. Set the disk label type to gpt for the new block devices (`/dev/sdc`, `/dev/sdd`, `/dev/sde`, `/dev/sdf`)*.

Command:

```
parted -s <block device> mklabel gpt
```

Example:

Storage Unit	Command
First	<code>parted -s /dev/sdc mklabel gpt</code>
	<code>parted -s /dev/sdd mklabel gpt</code>
Storage Unit	Command
Second	<code>parted -s /dev/sde mklabel gpt</code>
	<code>parted -s /dev/sdf mklabel gpt</code>

2. Create the required Decoder partitions utilizing the **PART LABEL** column from the [Archiver-DB-Mount/Directory-Names Table](#). Run all the commands below for each block device (`/dev/sdc`, `/dev/sdd`, `/dev/sde`, `/dev/sdf`)* to create the required partitions.

Command:

```
parted --align minimal <block device> mkpart <part label> xfs start_size end_size
```

Example:

First Storage Unit	<code>parted --align minimal /dev/sdc mkpart database xfs 3MiB 100%</code>
	<code>parted --align minimal /dev/sdd mkpart database xfs 3MiB 100%</code>
Second Storage Unit	<code>parted --align minimal /dev/sde mkpart database xfs 3MiB 100%</code>
	<code>parted --align minimal /dev/sdf mkpart database xfs 3MiB 100%</code>

3. Confirm that all partitions have been created correctly for all block devices (`/dev/sdc`, `/dev/sdd`, `/dev/sde`, `/dev/sdf`)* by running the parted print command for all block devices. The output will be the same for all block devices.

Command:

`parted <block device> print`

Example:

Storage Unit	Command
First	<code>parted /dev/sdc print</code>
	<code>parted /dev/sdd print</code>
Storage Unit	Command
Second	<code>parted /dev/sde print</code>
	<code>parted /dev/sdf print</code>

All block device outputs should look identical to the output below.

```
Model: Adaptec LOGICAL VOLUME (scsi)
Disk /dev/sdc: 486TB
Sector Size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File System  Name      Flags
  1      3146kB  486TB   486TB                   database
```

4. Create the file systems on all the newly created Archiver partitions. The numbers under the Number column from the previous step's print outputs should be appended to the block device name as shown in the commands below.

Command:

```
mkfs.xfs -f <block device><partition number>
```

Example:

Storage Unit	Block Device	Command
First	/dev/sdc*	mkfs.xfs -f /dev/sdc1
	/dev/sdd*	mkfs.xfs -f /dev/sdd1

Storage Unit	Block Device	Command
Second	/dev/sde*	mkfs.xfs -f /dev/sde1
	/dev/sdf*	mkfs.xfs -f /dev/sdf1

5. Create the directories where the database partitions will be mounted.

Command:

```
mkdir /var/netwitness/archiver/database<#>
```

Example:

Storage Unit	Command
First	mkdir /var/netwitness/archiver/database0
	mkdir /var/netwitness/archiver/database1

Storage Unit	Command
Second	mkdir /var/netwitness/archiver/database2
	mkdir /var/netwitness/archiver/database3

- Run the **blkid** command to list all the block device partitions on the configured block devices (*/dev/sdc, /dev/sdd, /dev/sde, /dev/sdf*)*. Capture the output from the commands as the UUIDs listed will be needed in later steps (see screenshot for example output). It is important to note which UUIDs are associated with which PARTLABELs from the output. These PARTLABELs directly correspond with the PART LABEL column from the [Archiver-DB-Mount/Directory-Names Table](#) section of this document.

Storage Unit	Command
First	<code>blkid egrep '/dev/sdc' awk '{print \$1, \$2, \$5}'</code>
	<code>blkid egrep '/dev/sdd' awk '{print \$1, \$2, \$5}'</code>
Storage Unit	Command
Second	<code>blkid egrep '/dev/sde' awk '{print \$1, \$2, \$5}'</code>
	<code>blkid egrep '/dev/sdf' awk '{print \$1, \$2, \$5}'</code>

```
/dev/sdc1:  UUID="1a770e78-3890-f23a-12d4-cf234537a90e2"  PARTLABEL="database"
```

- Backup the operating system's *File System Table* file (*/etc/fstab*). Since this file will be edited shortly it is good practice to have a backup if anything goes wrong. This command will make a copy of the **fstab** and place it into the */root* directory, called **fstab.backup**.

```
cp /etc/fstab /root/fstab.backup
```

8. Edit the **fstab** (/etc/fstab) file and add the new UUIDs and mount locations for all block device partitions to the bottom of the file.

The example below shows how to use the UUIDs from the **blkid** outputs and combine them with the associated mount points. The customer should substitute their own UUIDs when adding the entries to the file. Each entry must be on its own line to function. Two entries will be created for each storage unit attached to the appliance.

If unsure how to edit the fstab file, stop and request assistance from NetWitness Support.

If unsure of which mount points go to which block device partitions, review the outputs from the previous three steps along with the [Archiver-DB-Mount/Directory-Names Table](#).

Example fstab Lines:

```
UUID=1a770e78-3890-f23a-12d4-cf234537a90e2 /var/netwitness/archiver/database0 xfs noatime,nosuid 1 2
UUID= 2bc45786-1c42-a2af-4d32-009245f23a76d2 /var/netwitness/archiver/database1 xfs noatime,nosuid 1 2
```

If looking for more information about the relationship between the UUID, PARTLABEL, mount points and the **fstab**, please see the [Relationship between UUID/PARTLABEL and Mount Points](#) section in the Appendix.

9. Reload the system daemon, confirm that the **fstab** is read successfully, and double check the partitions are visible. If any errors are produced, this means the changes done to the **fstab** file are not correct. Either a misspelling has occurred, or one or more device/directory locations are incorrect/missing. Use the backed up **fstab.backup** file if needed.

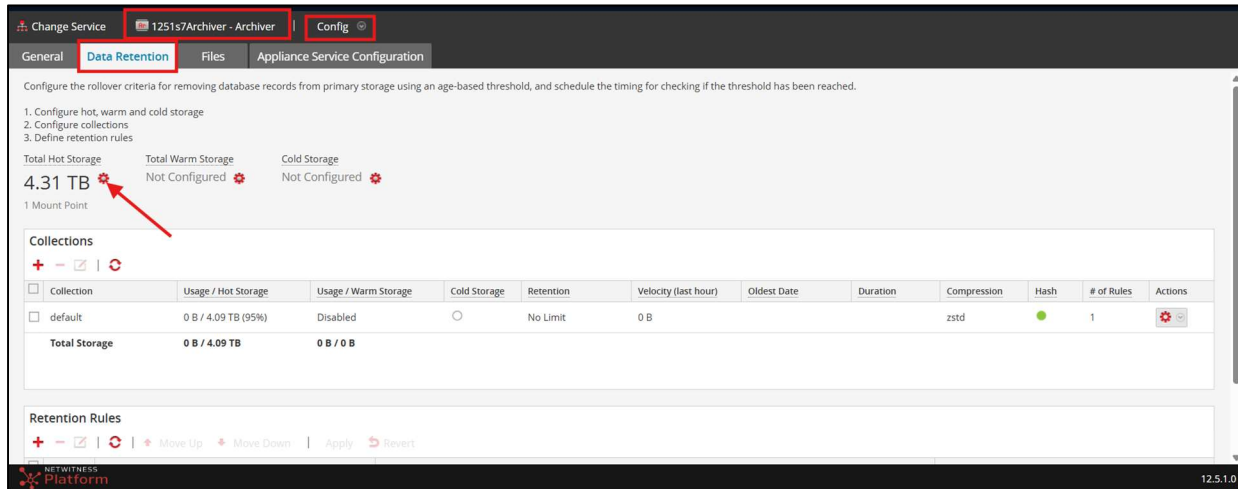
```
systemctl daemon-reload
mount -a
df -hP
```

/dev/sda2	1014M	183M	832M	19%	/boot
/dev/sda1	205T	1.5G	204T	1%	/boot/efi
/dev/mapper/netwitness_vg00-varlog	225T	1.6T	223T	1%	/var/log
/dev/mapper/netwitness_vg00-usrhome	10G	175M	9.9G	1%	/home
/dev/mapper/netwitness_vg00-nwhome	4.4T	469G	3.9T	11%	/var/netwitness
/dev/sdcl	443T	3.1T	439T	1%	/var/netwitness/archiver/database0

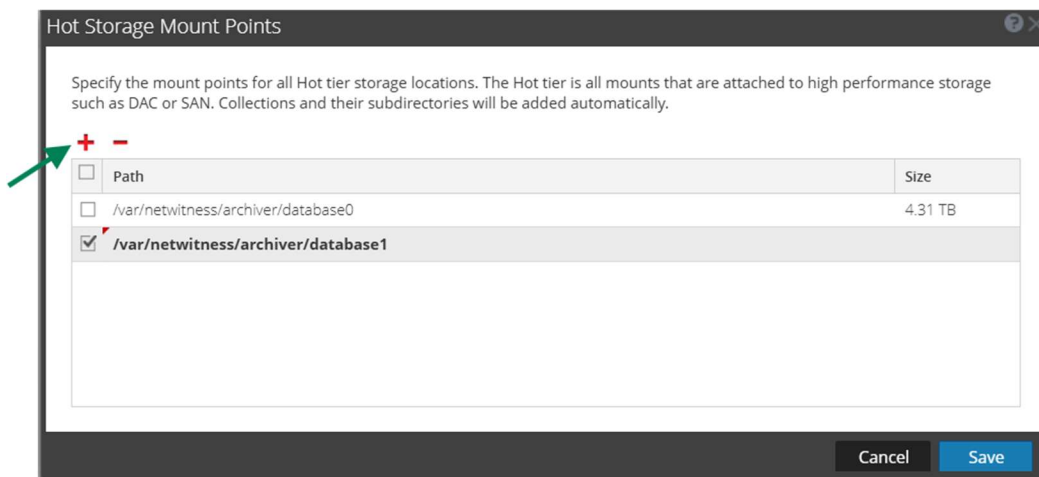
Update the Archiver Database/Collection Configuration (First & Second)

This section will provide step by step instructions for how to adjust the Archiver's storage configuration to properly take advantage of the newly daisy-chained Ultrastar60 storage units.

1. Log into the NetWitness UI and go to the **Config** area of the Archiver with the Ultrastar60 daisy-chained storage attached, and select the **Data Retention** tab.
2. Click the **red gear** next to the **Total Hot Storage**, as shown in the screenshot.



3. This will open the **Hot Storage Mount Points** requestor. Click the **plus (+)** icon and add the `/var/netwitness/archiver/database1` storage to the **Path** field.



Note: The default path of `/var/netwitness/archiver/database0` is setup during the orchestration of the Archiver. The default size of 4.31 TB is already set but will update to 442.02 TB after the **Save** button has been clicked.

4. Continue to click the plus (+) icon until all database partitions are added (database2 and database3). Once done, click the **Save** button.

Hot Storage Mount Points

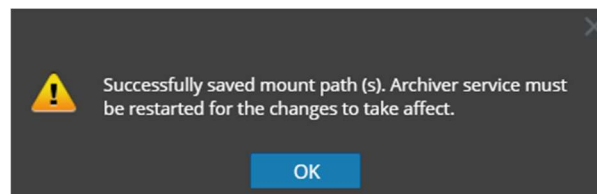
Specify the mount points for all Hot tier storage locations. The Hot tier is all mounts that are attached to high performance storage such as DAC or SAN. Collections and their subdirectories will be added automatically.

+ -

☐	Path	Size
☐	/var/netwitness/archiver/database0	4.31 TB
☐	/var/netwitness/archiver/database1	442.02 TB
☐	/var/netwitness/archiver/database2	442.02 TB
☐	/var/netwitness/archiver/database3	442.02 TB

Cancel Save

5. A confirmation box will appear confirming that the paths have been saved to the Archiver. Click **OK** to continue. It will also say that the Archiver service will need restarted before the storage can be used.



- When returned to the Data Retention tab within the Archiver, the Total Hot Storage display should have increased by the amount of storage added from all the database paths. This can be used to confirm that all storage paths have been correctly recognized by the Archiver service.

Change Service
1251s/Archiver - Archiver
Config

General
Data Retention
Files
Appliance Service Configuration

Configure the rollover criteria for removing database records from primary storage using an age-based threshold, and schedule the timing for checking if the threshold has been reached.

1. Configure hot, warm and cold storage
2. Configure collections
3. Define retention rules

Total Hot Storage
1643.76 TB

Total Warm Storage
Not Configured

Cold Storage
Not Configured

5 Mount Points

Collections

Collection	Usage / Hot Storage	Usage / Warm Storage	Cold Storage	Retention	Velocity (last hour)	Oldest Date	Duration	Compression	Hash	# of Rules	Actions
default	328.76 GB / 1683.76 TB (...)	Disabled	○	No Limit	6.18 GB	2025-04-21 20:32	7 days 18 hour...	zstd	●	1	⚙️
Total Storage	328.76 GB / 1683.76 TB	0 B / 0 B									

Retention Rules

+ - ↺
Move Up Move Down Apply Revert

Note: The Total Hot Storage value (1643.76 TB) is the sum of all 4 mount points which includes the updated default path of /var/netwitness/archiver/database0.

- Restart the Archiver service by going to the Archiver's system console and restarting the service.

```
systemctl restart nwarchiver
```

If the Archiver does not reboot as expected, or the storage is not appearing correctly, stop and contact NetWitness Support for assistance.

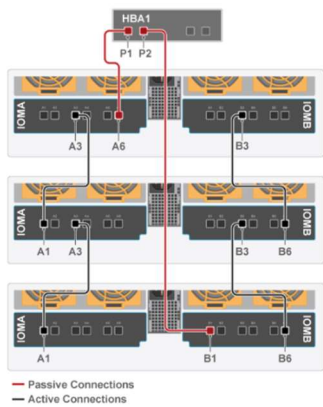
- The storage configuration can also be checked by going to the Archiver's Explore View, under the archiver -> collections -> default -> database -> config folder. The sizes after the path entries should all equal 0 B. This is normal.

Parameter	Value
Meta Compression Level (meta.compression.level)	0
Meta Database Directory (meta.dir)	/var/netwitness/archiver/database0/default/metadb=0 B;/var/netwitness/archiver/database1/default/metadb=0 B;/var/netwitness/archiver/database2/default/metadb=0 B;/var/netwitness/archiver/database3/default/metadb=0 B;/var/netwitness/archiver/database4/default/metadb=0 B
Cold Meta Database Directory (meta.dir.cold)	
Warm Meta Database Directory (meta.dir.warm)	
Meta File Size (meta.file.size)	auto
Meta Open Files (meta.files)	auto
Meta Minimum Free Space (meta.free.space.min)	1 GB
Meta Index Fidelity (meta.index.fidelity)	25
Meta Integrity Flush (meta.integrity.flush)	sync
Meta Write Block Size (meta.write.block.size)	8 MB
Packet Compression (packet.compression)	zstd
Packet Compression Level (packet.compression.level)	0
Packet/Log Database Directory (packet.dir)	/var/netwitness/archiver/database0/default/packetdb=0 B;/var/netwitness/archiver/database1/default/packetdb=0 B;/var/netwitness/archiver/database2/default/packetdb=0 B;/var/netwitness/archiver/database3/default/packetdb=0 B;/var/netwitness/archiver/database4/default/packetdb=0 B
Cold Packet/Log Database Directory (packet.dir.cold)	
Warm Packet/Log Database Directory (packet.dir.warm)	
Packet File Size (packet.file.size)	auto

Parameter	Value
Packet/Log Database Directory (packet.dir)	/var/netwitness/archiver/database0/default/packetdb=0 B;/var/netwitness/archiver/database1/default/packetdb=0 B;/var/netwitness/archiver/database2/default/packetdb=0 B;/var/netwitness/archiver/database3/default/packetdb=0 B;/var/netwitness/archiver/database4/default/packetdb=0 B
Cold Packet/Log Database Directory (packet.dir.cold)	
Warm Packet/Log Database Directory (packet.dir.warm)	
Packet File Size (packet.file.size)	auto
Packet Open Files (packet.files)	auto
Packet Minimum Free Space (packet.free.space.min)	3 GB
Packet Index Fidelity (packet.index.fidelity)	15
Packet Integrity Flush (packet.integrity.flush)	sync
Packet Write Block Size (packet.write.block.size)	8 MB
Session Database Directory (session.dir)	/var/netwitness/archiver/database0/default/sessiondb=0 B;/var/netwitness/archiver/database1/default/sessiondb=0 B;/var/netwitness/archiver/database2/default/sessiondb=0 B;/var/netwitness/archiver/database3/default/sessiondb=0 B;/var/netwitness/archiver/database4/default/sessiondb=0 B
Cold Session Database Directory (session.dir.cold)	
Warm Session Database Directory (session.dir.warm)	
Session File Size (session.file.size)	auto
Session Open Files (session.files)	auto
Session Minimum Free Space (session.free.space.min)	266 MB
Session Integrity Flush (session.integrity.flush)	sync
Session Write Block Size (session.write.block.size)	32 KB

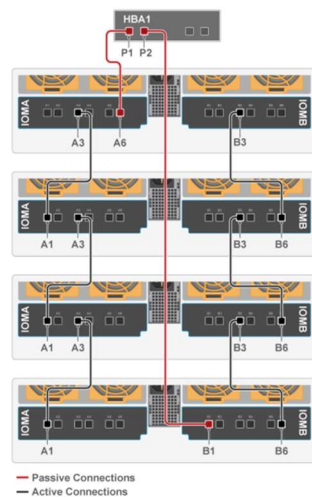
Configure Daisy-Chained Archiver Storage (Third & Fourth)

Three Enclosures: One Host with a Single HBA



Source Device	Destination Device
Host1: P1	Enclosure1: IOMA, A6
Host1: P2	Enclosure3: IOMB, B1
Enclosure1: IOMA, A3	Enclosure2: IOMA, A1
Enclosure1: IOMB, B3	Enclosure2: IOMB, B6
Enclosure2: IOMA, A3	Enclosure3: IOMA, A1
Enclosure2: IOMB, B3	Enclosure3: IOMB, B6

Four Enclosures: One Host with a Single HBA



Source Device	Destination Device
Host1: P1	Enclosure1: IOMA, A6
Host1: P2	Enclosure4: IOMB, B1
Enclosure1: IOMA, A3	Enclosure2: IOMA, A1
Enclosure1: IOMB, B3	Enclosure2: IOMB, B6
Enclosure2: IOMA, A3	Enclosure3: IOMA, A1
Enclosure2: IOMB, B3	Enclosure3: IOMB, B6
Enclosure3: IOMA, A3	Enclosure4: IOMA, A1
Enclosure3: IOMB, B3	Enclosure4: IOMB, B6

This section provides the steps needed to attach a third and fourth Ultrastar60 storage unit to the Archiver. Please keep in mind that setting up the third and fourth storage units will be identical. NetWitness currently supports two, three, and four storage units in a daisy-chain configuration. The instructions in this section will show how to configure the third and fourth at the same time. If there is only a third storage unit, do not run any of the fourth storage unit commands.

Please review the [How to Start Configuring Ultrastar60 Storage](#) section to get the output for the Enclosure, Connector, Physical ID, and LUN ID of the additional storage. The steps in this section are almost identical to setting up the first and second Ultrastar60 units on the Archiver. As such the steps will be more abbreviated but should align closely with the ones provided in the [Configuring Daisy-Chained Archiver Storage \(First & Second\)](#).

Creating Ultrastar60 Block Devices

1. Perform a block device listing on the appliance. This command looks for any devices that are of size 442 TB. This will catch any configured Ultrastar60 storage. Since all connected units are unconfigured, no block devices should be returned.

```
lsblk | grep '442T'
```

- Retrieve the Physical ID and LUN ID output for Connector 0 (CN0/First Storage Unit) and Connector 1 (CN1/Second Storage Unit), per the [How to Start Configuring Ultrastar60 Storage](#) section.

Important: The `arcconf` utility requires that the LUN IDs captured earlier must be entered in reverse order (0,59 0,58 0,57 ... 0,2 0,1 0,0), instead of normal order (0,0 0,1 0,2 ... 0,57 0,58 0,59). Pay particular attention when entering the commands of this section due to this reversal.

- Break up the 120 LUNS into four groups of 30, remembering to use reverse order. If only a third storage unit is added, the total LUNs will be 60.

Storage Unit	Groups	LUN IDs
Third	Group 1	0,149 0,148 0,147 ... 0,122 0,121 0,120
	Group 2	0,179 0,178 0,177 ... 0,152 0,151 0,150
Storage Unit	Groups	LUN IDs
Fourth	Group 3	0,209 0,208 0,207 ... 0,182 0,181 0,180
	Group 4	0,239 0,238 0,237 ... 0,212 0,211 0,210

- Create the first RAID 50 Block Device 0 using the Group 1 LUNs from the **third** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,149 0,148 0,147 0,146 0,145 0,144 0,143 0,142 0,141 0,140 0,139 0,138
0,137 0,136 0,135 0,134 0,133 0,132 0,131 0,130 0,129 0,128 0,127 0,126
0,125 0,124 0,123 0,122 0,121 0,120 noprompt
```

- Create the second RAID 50 Block Device 1 using Group 2 LUNs from the **third** storage unit.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,179 0,178 0,177 0,176 0,175 0,174 0,173 0,172 0,171 0,170 0,169 0,168
0,167 0,166 0,165 0,164 0,163 0,162 0,161 0,160 0,159 0,158 0,157 0,156
0,155 0,154 0,153 0,152 0,151 0,150 noprompt
```

- If a **fourth** storage unit is attached, create the third RAID 50 Block Device 2 using Group 3 LUNs.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,209 0,208 0,207 0,206 0,205 0,204 0,203 0,202 0,201 0,200 0,199 0,198
0,197 0,196 0,195 0,194 0,193 0,192 0,191 0,190 0,189 0,188 0,187 0,186
0,185 0,184 0,183 0,182 0,181 0,180 noprompt
```

7. If a **fourth** storage unit is attached, create the fourth RAID 50 Block Device 3 using Group 4 LUNs.

```
arcconf create 1 logicaldrive stripsize 1024 legs 3 method default max 50
0,239 0,238 0,237 0,236 0,235 0,234 0,233 0,232 0,231 0,230 0,229 0,228
0,227 0,226 0,225 0,224 0,223 0,222 0,221 0,220 0,219 0,218 0,217 0,216
0,215 0,214 0,213 0,212 0,211 0,210 noprompt
```

8. Run the **lsblk** command to find the newly created Ultrastar60 block devices. The command should return four devices of 442 TB in size, similar to the output below.

```
lsblk | grep `442T`
```

	sdc	8:32	0	442T	0	disk
	sdd	8:32	0	442T	0	disk
	sde	8:32	0	442T	0	disk
	sdf	8:32	0	442T	0	disk
Third Storage	sdg	8:32	0	442T	0	disk
	sdh	8:32	0	442T	0	disk
Fourth Storage	sdi	8:32	0	442T	0	disk
	sdj	8:32	0	442T	0	disk

Sometimes the block devices will not display immediately. This is normal. A rescan of the bus can be performed using the following command. Once it is complete, rerun the **lsblk** command above.

```
arcconf rescan 1
```

Create the Partitions and Mount Points for the New Block Devices (Third & Fourth)

This section will show how to create new partitions and set up mount points using the newly created block devices on the third and fourth Ultrastar60 storage units. If there is only a third unit, ignore any commands associated with a fourth storage unit.

Please use the block devices that were discovered while running the **lsblk** command in the previous section. The devices discovered should take the place of `/dev/sdg` and `/dev/sdh`, for the third storage unit, and `/dev/sdi` and `/dev/sdj`, for the fourth storage unit in the following examples. The [Archiver-DB-Mount/Directory-Names Table](#) will be utilized for this section and should be referenced as needed.

If the appliance has been rebooted since the block device creation was performed, rerun the **lsblk** command again as the block device names may have changed. The operating system is responsible for device naming and can change the names between reboots.

The block device of `/dev/sdg`, `/dev/sdh`, `/dev/sdi`, and `/dev/sdj` will be used in examples throughout this section.* These are only place holders for the block devices within the customer's appliance.

If there are any questions, please stop and contact NetWitness Customer Support for assistance.

1. Set the disk label type to gpt for the new block devices (`/dev/sdg`, `/dev/sdh`, `/dev/sdi`, `/dev/sdj`)*.

Command:

```
parted -s <block device> mklabel gpt
```

Example:

Storage Unit	Command
First	<code>parted -s /dev/sdg mklabel gpt</code>
	<code>parted -s /dev/sdh mklabel gpt</code>
Storage Unit	Command
Second	<code>parted -s /dev/sdi mklabel gpt</code>
	<code>parted -s /dev/sdj mklabel gpt</code>

2. Create the required Decoder partitions utilizing the **PART LABEL** column from the [Archiver-DB-Mount/Directory-Names Table](#). Run all the commands below for each block device (`/dev/sdg`, `/dev/sdh`, `/dev/sdi`, `/dev/sdj`)* to create the required partitions.

Command:

```
parted --align minimal <block device> mkpart <part label> xfs start_size  
end_size
```

Example:

Third Storage Unit	parted --align minimal /dev/sdg mkpart database xfs 3MiB 100%
	parted --align minimal /dev/sdh mkpart database xfs 3MiB 100%
Fourth Storage Unit	parted --align minimal /dev/sdi mkpart database xfs 3MiB 100%
	parted --align minimal /dev/sdj mkpart database xfs 3MiB 100%

3. Confirm that all partitions have been created correctly for all block devices (/dev/sdg, /dev/sdh, /dev/sdi, /dev/sdj)* by running the parted print command for all block devices. The output will be the same for all block devices.

Command:

parted <block device> print

Example:

Storage Unit	Command
Third	parted /dev/sdg print
	parted /dev/sdh print
Storage Unit	Command
Fourth	parted /dev/sdi print
	parted /dev/sdj print

All block device outputs should look identical to the output below.

```

Model: Adaptec LOGICAL VOLUME (scsi)
Disk /dev/sdg: 486TB
Sector Size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File System  Name      Flags
  1      3146kB  486TB   486TB   xfs          database

```

4. Create the file systems on all the newly created Archiver partitions. The numbers under the **Number** column from the previous step's print outputs should be appended to the block device name as shown in the commands below.

Command:

```
mkfs.xfs -f <block device><partition number>
```

Example:

Storage Unit	Block Device	Command
Third	/dev/sdg*	mkfs.xfs -f /dev/sdg1
	/dev/sdh*	mkfs.xfs -f /dev/sdh1

Storage Unit	Block Device	Command
Fourth	/dev/sdi*	mkfs.xfs -f /dev/sdi1
	/dev/sdj*	mkfs.xfs -f /dev/sdj1

5. Create the directories where the database partitions will be mounted.

Command:

```
mkdir /var/netwitness/archiver/database<#>
```

Example:

Storage Unit	Command
Third	mkdir /var/netwitness/archiver/database4
	mkdir /var/netwitness/archiver/database5

Storage Unit	Command
Fourth	mkdir /var/netwitness/archiver/database6
	mkdir /var/netwitness/archiver/database7

- Run the **blkid** command to list all the block device partitions on the configured block devices (*/dev/sdg, /dev/sdh, /dev/sdi, /dev/sdj*)*. Capture the output from the commands as the UUIDs listed will be needed in later steps (see screenshot for example output). It is important to note which UUIDs are associated with which PARTLABELs from the output. These PARTLABELs directly correspond with the PART LABEL column from the [Archiver-DB-Mount/Directory-Names Table](#) section of this document.

Storage Unit	Command
Third	<code>blkid egrep '/dev/sdg' awk '{print \$1, \$2, \$5}'</code>
	<code>blkid egrep '/dev/sdh' awk '{print \$1, \$2, \$5}'</code>
Storage Unit	Command
Fourth	<code>blkid egrep '/dev/sdi' awk '{print \$1, \$2, \$5}'</code>
	<code>blkid egrep '/dev/sdj' awk '{print \$1, \$2, \$5}'</code>

```
/dev/sdg1:  UUID="5ee4ff28-f23a-113a-0924-009245456b76d2"  PARTLABEL="database"
```

- Backup the operating system's *File System Table* file (*/etc/fstab*). Since this file will be edited shortly it is good practice to have a backup if anything goes wrong. This command will make a copy of the **fstab** and place it into the */root* directory, called **fstab.backup**.

```
cp /etc/fstab /root/fstab.backup
```

8. Edit the **fstab** (/etc/fstab) file and add the new UUIDs and mount locations for all block device partitions to the bottom of the file.

The example below shows how to use the UUIDs from the **blkid** outputs and combine them with the associated mount points. The customer should substitute their own UUIDs when adding the entries to the file. Each entry must be on its own line to function. Two entries will be created for each storage unit attached to the appliance.

If unsure how to edit the fstab file, stop and request assistance from NetWitness Support.

If unsure of which mount points go to which block device partitions, review the outputs from the previous three steps along with the [Archiver-DB-Mount/Directory-Names Table](#).

Example **fstab** Lines:

```
UUID=5ee4ff28-f23a-113a-0924-009245456b76d2 /var/netwitness/archiver/database4 xfs noatime,nosuid 1 2
UUID=4fa46678-c4ef-3890-00cf-d1e43677770e21 /var/netwitness/archiver/database5 xfs noatime,nosuid 1 2
```

If looking for more information about the relationship between the UUID, PARTLABEL, mount points and the **fstab**, please see the [Relationship between UUID/PARTLABEL and Mount Points](#) section in the Appendix.

9. Reload the system daemon, confirm that the **fstab** is read successfully, and double check the partitions are visible. If any errors are produced, this means the changes done to the **fstab** file are not correct. Either a misspelling has occurred, or one or more device/directory locations are incorrect/missing. Use the backed up **fstab.backup** file if needed.

```
systemctl daemon-reload
mount -a
df -hP
```

/dev/sda2	1014M	183M	832M	19%	/boot
/dev/sda1	205T	1.5G	204T	1%	/boot/efi
/dev/mapper/netwitness_vg00-varlog	225T	1.6T	223T	1%	/var/log
/dev/mapper/netwitness_vg00-usrhome	10G	175M	9.9G	1%	/home
/dev/mapper/netwitness_vg00-nwhome	4.4T	469G	3.9T	11%	/var/netwitness
/dev/sdc1	443T	3.1T	439T	1%	/var/netwitness/archiver/database0
/dev/sdd1	443T	3.1T	439T	1%	/var/netwitness/archiver/database1
/dev/sde1	443T	3.1T	439T	1%	/var/netwitness/archiver/database2
/dev/sdf1	443T	3.1T	439T	1%	/var/netwitness/archiver/database3
/dev/sdg1	443T	3.1T	439T	1%	/var/netwitness/archiver/database4
/dev/sdh1	443T	3.1T	439T	1%	/var/netwitness/archiver/database5
/dev/sdi1	443T	3.1T	439T	1%	/var/netwitness/archiver/database6
/dev/sdj1	443T	3.1T	439T	1%	/var/netwitness/archiver/database7

Update the Archiver Database/Collection Configuration (Third & Fourth)

This section will provide step by step instructions for how to adjust the Archiver's storage configuration to properly take advantage of the newly daisy-chained Ultrastar60 storage units.

Important: There is a limit of 255 characters for the Hot Tier Storage Path value by default. When adding multiple mount points this character limit can be reached. Daisy-chaining more than two Ultrastar60 storage units can reach this character limit, as such, the default limit must be increased.

If unfamiliar with using text editing tools, or unsure about adjusting configuration files, stop and contact NetWitness Support for assistance.

1. To adjust the maximum length of the Hot Tier Storage path, access the Archiver system console via SSH.
2. Make sure that the Archiver service is stopped before continuing, otherwise the changes may not be saved.

```
systemctl stop nwarchiver
```

3. Once the service is stopped, go to the **/var/netwitness/ng** directory.

```
cd /var/netwitness/ng
```

4. Make a backup copy of the **NwArchiver.cfg** file as it will need to be edited. This command will back up the file to /root. It is important to always have a backup of every critical system file before making any changes.

```
cp NwArchiver.cfg /root/NwArchiver.cfg.bak
```

5. Edit the file using a text editor such as VI and search for **prettyName="Mount Directories (Hot Tier)"** and/or **name="mounts.dir"**. Both lead to the same configuration line.
6. Once the entry is found, change the attribute of **maxLength="255"** to **maxLength="1024"**. This will not affect performance in any way but will allow for more mount paths to be added.

```
<config name="mounts.dir" prettyName="Mount Directories (Hot Tier)" value="/var/netwitness/archiver/  
database0;/var/netwitness/archiver/database1;/var/netwitness/archiver/database2;/var/netwitness/archiver/da  
tabase3" instance="config" hash="12291705766769975358" getRoles="everyone" setRoles="archiver.manage"  
maxLength="255" />
```

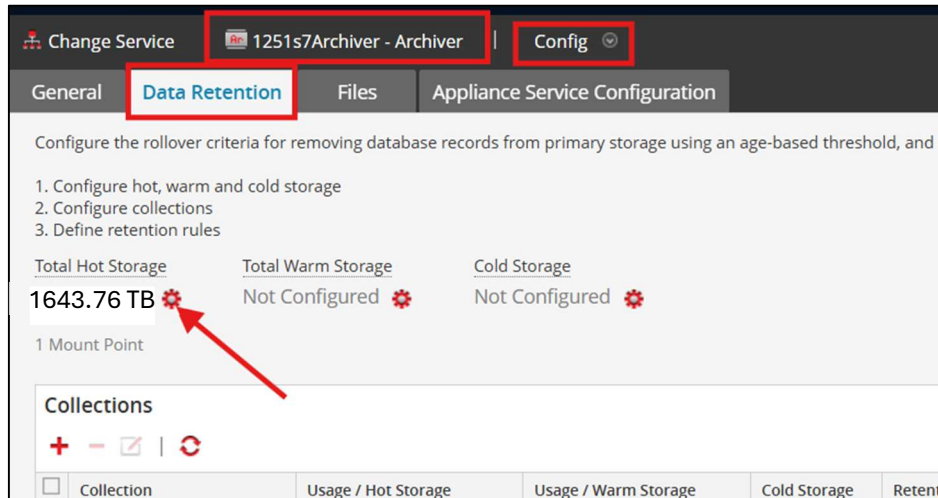
7. Once the **maxLength** has been adjusted, save the **NwArchiver.cfg** file.

- Restart the Archiver service.

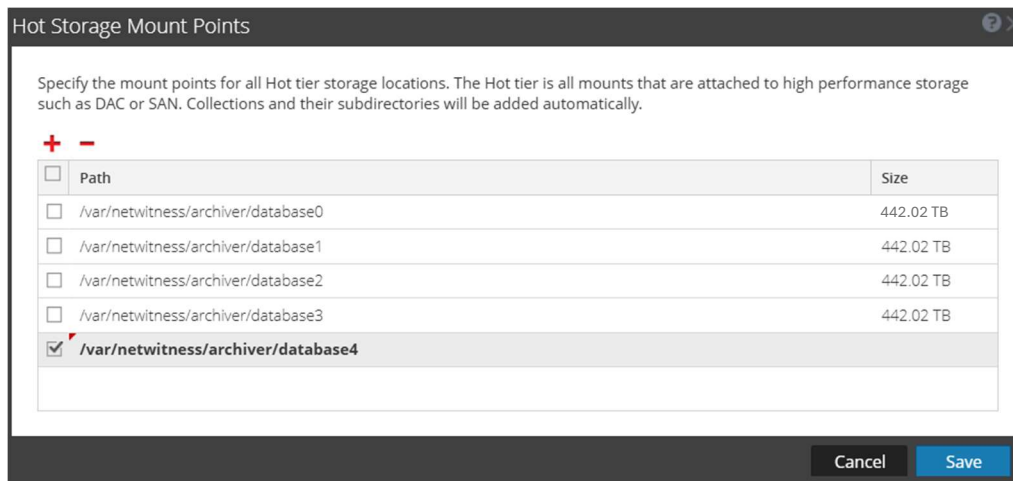
```
systemctl start nwarchiver
```

- Once the service is up and responding, log into the NetWitness UI and go to the **Config** area of the Archiver with the Ultrastar60 daisy-chained storage attached, and select the **Data Retention** tab.

- Click the **red gear** next to the **Total Hot Storage**, as shown in the screenshot.

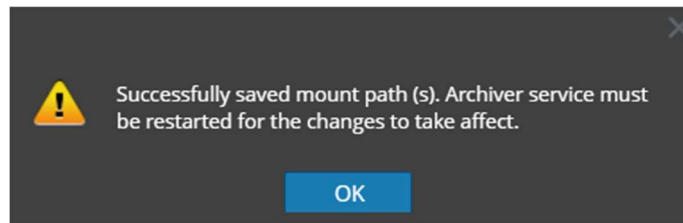


- This will open the **Hot Storage Mount Points** requestor. Click the **plus (+)** icon and add the `/var/netwitness/archiver/database4` storage to the **Path** field, exactly like adding the first two Ultrastar60 storage units mount points.



- Continue to click the plus (+) icon until all database partitions are added. If only adding a third storage unit, add `/var/netwitness/archiver/database5`. If a fourth storage unit is attached, also add `/var/netwitness/archiver/database6` and `/var/netwitness/archiver/database7`. Once done, click the **Save** button.

13. A confirmation box will appear confirming that the paths have been saved to the Archiver. Click **OK** to continue. It will also say that the Archiver service will need restarted before the storage can be used.



14. When returned to the Data Retention tab within the Archiver, the **Total Hot Storage** display should have increased by the amount of storage added from all the database paths. This can be used to confirm that all storage paths have been correctly recognized by the Archiver service.
15. Restart the Archiver service by going to the Archiver's system console and restarting the service.

```
systemctl restart nwarchiver
```

If the Archiver does not reboot as expected, or the storage is not appearing correctly, stop and contact NetWitness Support for assistance.

16. The storage configuration can also be checked by going to the Archiver's Explore View, under the archiver -> collections -> default -> database -> config folder. This is the same process used when confirming the database paths for the first and second storage units. The size after each path entry should equal 0 B. This is a normal configuration.

Appendix

Identifying the Physical IDs, LUN IDs, and Connectors

This screenshot is an example output from the `arcconf list` command. Here you can see the Physical IDs, LUN IDs, and Connector IDs.

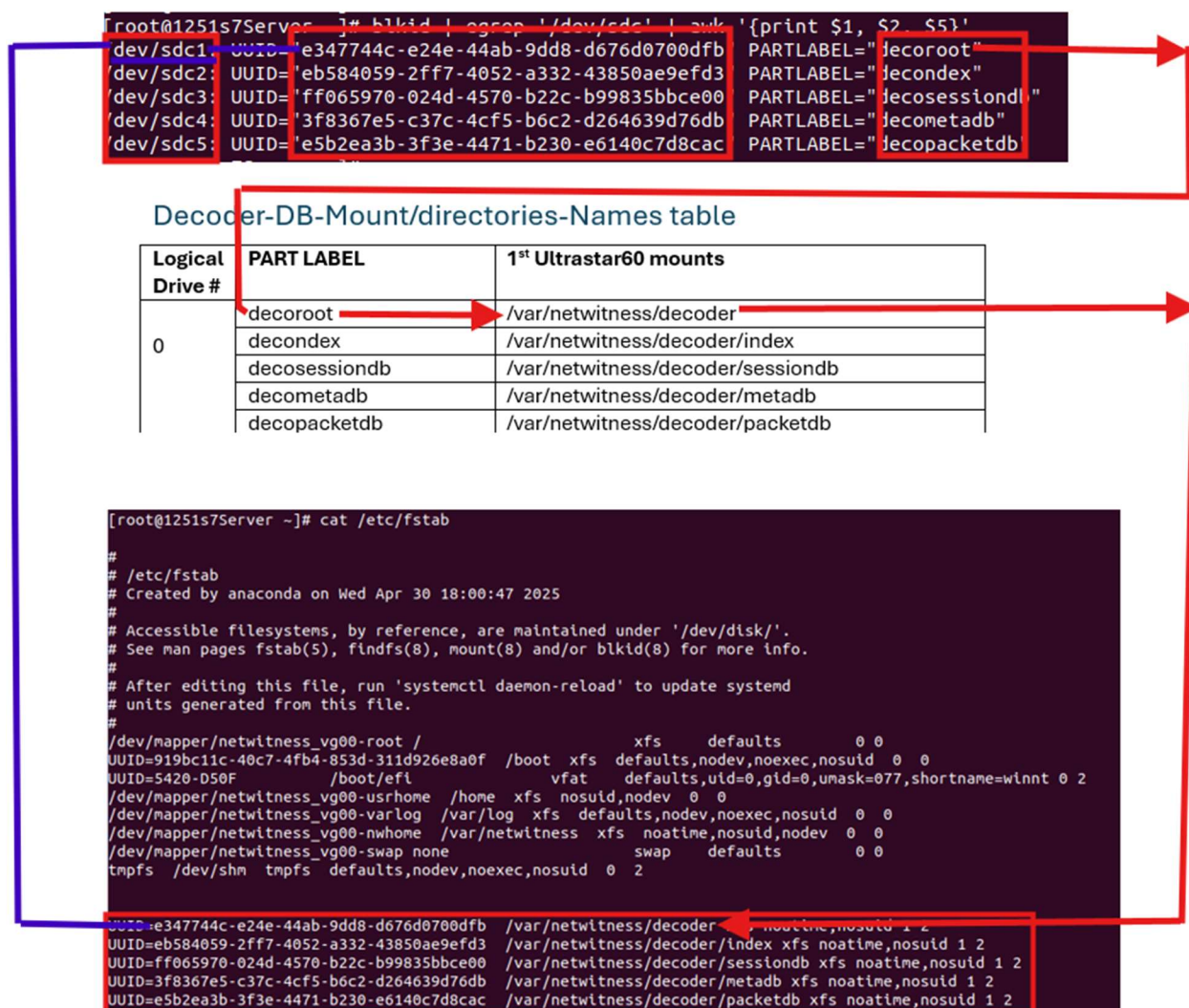
First Storage Unit			
Physical	0,0	[Enclosure 1, Slot 1	(Connector 0:CN0)]
Physical	0,1	[Enclosure 1, Slot 1	(Connector 0:CN0)]
Physical	0,2	[Enclosure 1, Slot 1	(Connector 0:CN0)]
Physical	0,3	[Enclosure 1, Slot 1	(Connector 0:CN0)]
Physical	0,4	[Enclosure 1, Slot 1	(Connector 0:CN0)]
Physical	0,5	[Enclosure 1, Slot 1	(Connector 0:CN0)]
Physical	0,6	[Enclosure 1, Slot 1	(Connector 0:CN0)]
Physical			
Physical			
Physical			
Physical	0,54	[Enclosure 1, Slot 1	(Connector 0:CN0)]
Physical	0,55	[Enclosure 1, Slot 1	(Connector 0:CN0)]
Physical	0,56	[Enclosure 1, Slot 1	(Connector 0:CN0)]
Physical	0,57	[Enclosure 1, Slot 1	(Connector 0:CN0)]
Physical	0,58	[Enclosure 1, Slot 1	(Connector 0:CN0)]
Physical	0,59	[Enclosure 1, Slot 1	(Connector 0:CN0)]

Second Storage Unit			
Physical	0,60	[Enclosure 1, Slot 1	(Connector 1:CN1)]
Physical	0,61	[Enclosure 1, Slot 1	(Connector 1:CN1)]
Physical	0,62	[Enclosure 1, Slot 1	(Connector 1:CN1)]
Physical	0,63	[Enclosure 1, Slot 1	(Connector 1:CN1)]
Physical	0,64	[Enclosure 1, Slot 1	(Connector 1:CN1)]
Physical	0,65	[Enclosure 1, Slot 1	(Connector 1:CN1)]
Physical	0,66	[Enclosure 1, Slot 1	(Connector 1:CN1)]
Physical			
Physical			
Physical			
Physical	0,114	[Enclosure 1, Slot 1	(Connector 1:CN1)]
Physical	0,115	[Enclosure 1, Slot 1	(Connector 1:CN1)]
Physical	0,116	[Enclosure 1, Slot 1	(Connector 1:CN1)]
Physical	0,117	[Enclosure 1, Slot 1	(Connector 1:CN1)]
Physical	0,118	[Enclosure 1, Slot 1	(Connector 1:CN1)]
Physical	0,119	[Enclosure 1, Slot 1	(Connector 1:CN1)]

Relationship between UUID/PARTLABEL and Mount Points

The following shows the relationship between block device partitions (ex. `/dev/sdc1`), the associated UUID (ex. `e347744c-e24e-44ab-9dd8-d676d0700dfb`), PARTLABEL (ex. `decoroot`), mount point (ex. `/var/netwitness/decoder`), and how it is represented in the `/etc/fstab` file system table.

This diagram is to provide a visual representation of how all these pieces interact with each other when setting up the Ultrastar60 storage units. The **red** line shows the mapping of the PARTLABEL for the block device partition to the mount directory. While the **blue** line shows the mapping of the UUID of the partition and the mount directory in the fstab for the specified partition.



Updating Decoder Database Configuration Screenshots

These screenshots show the Explore View for the Decoder when adjusting the database size configurations.

Decoder Meta and Packet Configuration Explore View

Change Service	1251s7LogDecoder - Decoder	Explore
1251s7LogDecoder - ...	/database/config	1251s7LogDecoder - Decoder
1251s7LogDecoder - Decoder (DECODER)	Meta Compression (meta.compression)	none
connections	Meta Compression Level (meta.compression.level)	0
database	Meta Database Directory (meta.dir)	/var/netwitness/decoder/metadb=16.5 TB;/var/netwitness/decoder/decoder0/metadb0=16.5 TB;
config	Cold Meta Database Directory (meta.dir.cold)	
stats	Warm Meta Database Directory (meta.dir.warm)	
decoder	Meta File Size (meta.file.size)	auto
deviceappliance	Meta Open Files (meta.files)	auto
index	Meta Minimum Free Space (meta.free.space.min)	150 GB
logs	Meta Index Fidelity (meta.index.fidelity)	4
rest	Meta Integrity Flush (meta.integrity.flush)	normal
sdk	Meta Write Block Size (meta.write.block.size)	64 KB
services	Packet Compression (packet.compression)	none
storedproc	Packet Compression Level (packet.compression.level)	0
sys	Packet/Log Database Directory (packet.dir)	/var/netwitness/decoder/packetdb=423 TB;/var/netwitness/decoder/decoder0/packetdb0=423 TB;
users	Cold Packet/Log Database Directory (packet.dir.cold)	
	Warm Packet/Log Database Directory (packet.dir.warm)	
	Packet File Size (packet.file.size)	auto

Decoder Session Configuration Explore View

Change Service	1251s7LogDecoder - Decoder	Explore
1251s7LogDecoder - ...	/database/config	1251s7LogDecoder - Decoder
1251s7LogDecoder - Decoder (DECODER)	Packet/Log Database Directory (packet.dir)	/var/netwitness/decoder/packetdb=423 TB;/var/netwitness/decoder/decoder0/packetdb0=423 TB;
connections	Cold Packet/Log Database Directory (packet.dir.cold)	
database	Warm Packet/Log Database Directory (packet.dir.warm)	
config	Packet File Size (packet.file.size)	auto
stats	Packet File Type (packet.file.type)	netwitness
decoder	Packet Open Files (packet.files)	auto
deviceappliance	Packet Minimum Free Space (packet.free.space.min)	1 TB
index	Packet Index Fidelity (packet.index.fidelity)	1
logs	Packet Integrity Flush (packet.integrity.flush)	normal
rest	Packet Write Block Size (packet.write.block.size)	filesize
sdk	Session Database Directory (session.dir)	/var/netwitness/decoder/sessiondb=1 TB;/var/netwitness/decoder/decoder0/sessiondb0=1 TB
services	Cold Session Database Directory (session.dir.cold)	
storedproc	Warm Session Database Directory (session.dir.warm)	
sys	Session File Size (session.file.size)	auto

Decoder Index Configuration Explore View

Change Service	1251s7Server - Decoder	Explore
1251s7Server - Deco...	/index/config	1251s7Server - Decoder
1251s7Server - Decoder (DECODER)	Index Directory (index.dir)	/var/netwitness/decoder/index=29 GB;/var/netwitness/decoder/decoder0/index0=29 GB
connections	Index Cold Storage Directory (index.dir.cold)	
database	Index Warm Storage Directory (index.dir.warm)	
config	Index Open Slice Count (index.slices.open)	42
stats	Page Compression (page.compression)	huffhybrid
decoder	Enable Background Reindexing (reindex.enable)	true
deviceappliance	Reindex Meta Readers Count (reindex.meta.readers.count)	2
index	Save Session Count (save.session.count)	auto
config	Slice Memory Max (slice.memory.max)	4.1 GB
stats		
logs		

Updating Log Decoder Database Configuration Screenshots

These screenshots show the Explore View for the Decoder when adjusting the database size configurations.

Decoder Meta and Packet Configuration Explore View

Parameter	Value
Meta Compression Level (meta.compression.level)	0
Meta Database Directory (meta.dir)	/var/netwitness/logdecoder/metadb=12.4 TB;/var/netwitness/logdecoder/logdecoder0/metadb0=12.4 TB
Cold Meta Database Directory (meta.dir.cold)	
Warm Meta Database Directory (meta.dir.warm)	
Meta File Size (meta.file.size)	auto
Meta Open Files (meta.files)	auto
Meta Minimum Free Space (meta.free.space.min)	38 GB
Meta Index Fidelity (meta.index.fidelity)	4
Meta Integrity Flush (meta.integrity.flush)	sync
Meta Write Block Size (meta.write.block.size)	64 KB
Packet Compression (packet.compression)	none
Packet Compression Level (packet.compression.level)	0
Packet/Log Database Directory (packet.dir)	/var/netwitness/logdecoder/packetdb=420 TB;/var/netwitness/logdecoder/logdecoder0/packetdb0=420 TB
Cold Packet/Log Database Directory (packet.dir.cold)	
Warm Packet/Log Database Directory (packet.dir.warm)	
Packet File Size (packet.file.size)	auto

Decoder Session Configuration Explore View

Parameter	Value
Meta Write Block Size (meta.write.block.size)	64 KB
Packet Compression (packet.compression)	none
Packet Compression Level (packet.compression.level)	0
Packet/Log Database Directory (packet.dir)	/var/netwitness/logdecoder/packetdb=420 TB;/var/netwitness/logdecoder/logdecoder0/packetdb0=420 TB
Cold Packet/Log Database Directory (packet.dir.cold)	
Warm Packet/Log Database Directory (packet.dir.warm)	
Packet File Size (packet.file.size)	auto
Packet Open Files (packet.files)	auto
Packet Minimum Free Space (packet.free.space.min)	38 GB
Packet Index Fidelity (packet.index.fidelity)	1
Packet Integrity Flush (packet.integrity.flush)	sync
Packet Write Block Size (packet.write.block.size)	64 KB
Session Database Directory (session.dir)	/var/netwitness/logdecoder/sessiondb=4.4 TB;/var/netwitness/logdecoder/logdecoder0/sessiondb0=4.4 TB
Cold Session Database Directory (session.dir.cold)	
Warm Session Database Directory (session.dir.warm)	
Session File Size (session.file.size)	

The directory where session files will be moved for cold storage (e.g. staged directory for backup). The following format specifiers are allowed: %y = year, %m = month, %d = day, %H = hour, %M = minute, %S = second, %f = nanosecond.

Decoder Index Configuration Explore View

Parameter	Value
Index Directory (index.dir)	/var/netwitness/logdecoder/index=29 GB;/var/netwitness/logdecoder/logdecoder0/index0=29 GB
Index Cold Storage Directory (index.dir.cold)	
Index Warm Storage Directory (index.dir.warm)	
Index Open Slice Count (index.slices.open)	42
Page Compression (page.compression)	huffybrid
Enable Background Reindexing (reindex.enable)	true
Reindex Meta Readers Count (reindex.meta.readers.count)	6
Save Session Count (save.session.count)	auto
Slice Memory Max (slice.memory.max)	4.1 GB

Updating Network Hybrid Database Configuration Screenshots

These screenshots show the Explore View for the Network Hybrid when adjusting the database size configurations.

Decoder Configuration Explore View

Change Service	1251s7hybrid - Decoder	Explore
1251s7hybrid - Deco...	/database/config	1251s7hybrid - Decoder
1251s7hybrid - Decoder (DECODER)	meta.write.block.size (meta.write.block.size)	04 KB
connections	Packet Compression (packet.compression)	none
database	Packet Compression Level (packet.compression.level)	0
config	Packet/Log Database Directory (packet.dir)	/var/netwitness/decoder/packetdb/packetdb==55.3 TB;/var/netwitness/decoder/packetdb0==355 TB;/var/netwitness/decoder/packetdb1==355 TB;/var/netwitness/decoder/packetdb2==355 TB
stats	Cold Packet/Log Database Directory (packet.dir.cold)	
decoder	Warm Packet/Log Database Directory (packet.dir.warm)	
deviceappliance	Packet File Size (packet.file.size)	auto
index	Packet File Type (packet.file.type)	pcapng
logs	Packet Open Files (packet.files)	auto
rest	Packet Minimum Free Space (packet.free.space.min)	518 GB
sdk	Packet Index Fidelity (packet.index.fidelity)	1
services	Packet Integrity Flush (packet.integrity.flush)	sync
storedproc	Packet Write Block Size (packet.write.block.size)	64 KB
sys	Session Database Directory (session.dir)	/var/netwitness/decoder/sessiondb==1.24 TB
	Cold Session Database Directory (session.dir.cold)	

Concentrator Configuration Explore View

Change Service	1251s7hybrid - Concentrator	Explore
1251s7hybrid - Conc...	/database/config	1251s7hybrid - Concentrator
1251s7hybrid - Concentrator (CONCENTR...)	Hash Algorithm (hash.algorithm)	none
concentrator	Hash Databases (hash.databases)	session,meta
connections	Hash Directory (hash.dir)	
database	Manifest Directory (manifest.dir)	
config	Meta Compression (meta.compression)	none
stats	Meta Compression Level (meta.compression.level)	0
deviceappliance	Meta Database Directory (meta.dir)	/var/netwitness/concentrator/metadb==25.13 TB;/var/netwitness/concentrator/metadb0=74 TB;/var/netwitness/concentrator/metadb1=74 TB;/var/netwitness/concentrator/metadb2=74 TB
index	Cold Meta Database Directory (meta.dir.cold)	
logs	Warm Meta Database Directory (meta.dir.warm)	
rest	Meta File Size (meta.file.size)	auto
sdk	Meta Open Files (meta.files)	auto
services	Meta Minimum Free Space (meta.free.space.min)	259 GB
storedproc	Meta Index Fidelity (meta.index.fidelity)	4
sys	Meta Integrity Flush (meta.integrity.flush)	sync
users	Meta Write Block Size (meta.write.block.size)	64 KB
	Session Database Directory (session.dir)	/var/netwitness/concentrator/sessiondb==2.49 TB;/var/netwitness/concentrator/sessiondb0=4.95 TB;/var/netwitness/concentrator/sessiondb1=4.95 TB;/var/netwitness/concentrator/sessiondb2=4.95 TB

Updating Log Hybrid Database Configuration Screenshots

These screenshots show the Explore View for the Log Hybrid when adjusting the database size configurations.

Log Decoder Configuration Explore View

Change Service	1251LogHybrid - Log Decoder	Explore
1251LogHybrid - Log ...	/database/config	1251LogHybrid - Log Decoder
1251LogHybrid - Log Decoder (LOG_DECO...	Warm Meta Database Directory (meta.dir.warm)	
collections	Meta File Size (meta.file.size)	auto
connections	Meta Open Files (meta.files)	auto
database	Meta Minimum Free Space (meta.free.space.min)	129 GB
config	Meta Index Fidelity (meta.index.fidelity)	4
stats	Meta Integrity Flush (meta.integrity.flush)	sync
decoder	Meta Write Block Size (meta.write.block.size)	64 KB
deviceappliance	Packet Compression (packet.compression)	none
index	Packet Compression Level (packet.compression.level)	0
logs	Packet/Log Database Directory (packet.dir)	/var/netwitness/logdecoder/packetdb/packetdb==41.47 TB;/var/netwitness/logdecoder/packetdb0=218 TB;/var/netwitness/logdecoder/packetdb1=218 TB
rest	Cold Packet/Log Database Directory (packet.dir.cold)	
sdk	Warm Packet/Log Database Directory (packet.dir.warm)	
services	Packet File Size (packet.file.size)	auto
storedproc	Packet Open Files (packet.files)	auto
sys	Packet Minimum Free Space (packet.free.space.min)	388 GB
users	Packet Index Fidelity (packet.index.fidelity)	1
	Packet Integrity Flush (packet.integrity.flush)	sync

Concentrator Configuration Explore View

Change Service	1251LogHybrid - Concentrator	Explore
1251LogHybrid - Con...	/database/config	1251LogHybrid - Concentrator
1251LogHybrid - Concentrator (CONCENT...	Manifest Directory (manifest.dir)	
concentrator	Meta Compression (meta.compression)	none
connections	Meta Compression Level (meta.compression.level)	0
database	Meta Database Directory (meta.dir)	/var/netwitness/concentrator/metadb==37.71 TB;/var/netwitness/concentrator/metadb0=204.9 TB;/var/netwitness/concentrator/metadb1=204.9 TB
config	Cold Meta Database Directory (meta.dir.cold)	
stats	Warm Meta Database Directory (meta.dir.warm)	
deviceappliance	Meta File Size (meta.file.size)	auto
index	Meta Open Files (meta.files)	auto
logs	Meta Minimum Free Space (meta.free.space.min)	388 GB
rest	Meta Index Fidelity (meta.index.fidelity)	4
sdk	Meta Integrity Flush (meta.integrity.flush)	sync
services	Meta Write Block Size (meta.write.block.size)	64 KB
storedproc	Session Database Directory (session.dir)	/var/netwitness/concentrator/sessiondb==3.73 TB;/var/netwitness/concentrator/sessiondb0=12.9 TB;/var/netwitness/concentrator/sessiondb1=12.9 TB
sys	Cold Session Database Directory (session.dir.cold)	
users	Warm Session Database Directory (session.dir.warm)	
	Session File Size (session.file.size)	auto
	Session Open Files (session.files)	auto