

**LifeKeeper for Linux Security Enhancement  
DataKeeper for Linux  
with Encrypted Path and Disk**

---

**第 2 版**

## 目次

1.	目的 .....	3
2.	本ドキュメントのサポート範囲 .....	3
3.	本ドキュメントについて .....	4
4.	LifeKeeper の構築 .....	5
4.1.	LifeKeeper のインストールおよび設定 .....	5
4.2.	Firewall の確認 .....	5
4.3.	通信経路およびディスクの暗号化に必要なパッケージ .....	5
4.4.	本構成の留意事項 .....	5
5.	通信経路の暗号化 .....	6
5.1.	sysctl.conf ファイルの設定 .....	6
5.2.	ipsec.conf ファイルの編集 .....	6
5.3.	ipsec.secrets ファイルの編集 .....	7
5.4.	Openswan の起動 .....	7
5.5.	残りノードでの設定 .....	8
5.6.	tcpdump による動作確認 .....	8
5.6.1.	IPsec 設定前の tcpdump の出力例 .....	8
5.6.2.	IPsec 設定後の tcpdump の出力例 .....	9
6.	ディスクの暗号化 .....	10
6.1.	LUKS パーティションの作成 .....	10
6.2.	キーファイルによる認証の追加 .....	11
6.3.	LUKS パーティションの展開 .....	12
6.4.	/etc/crypttab ファイルの設定 .....	13
6.5.	残りノードでの設定 .....	13
7.	データレプリケーションリソースの作成 .....	14
8.	参考資料 .....	29
8.1.	ユーザーサイト .....	29
8.2.	ホワイトペーパー .....	29
9.	著者について .....	30
10.	免責事項 .....	31

## 改版履歴

2013 年 3 月 22 日 第 1 版

2013 年 6 月 20 日 第 2 版

### 1. 目的

---

本ドキュメントでは LifeKeeper for Linux およびその Recovery Kit の一つである DataKeeper for Linux を使用し、暗号化したローカルディスクに保存されたデータを暗号化した通信経路(IPsec)を経由してデータレプリケーションを行う方法についてまとめています。

ユーザやパートナーの皆様が製品のインストールを検討する際の判断材料として、本ドキュメントを参照いただくことを目的としております。

### 2. 本ドキュメントのサポート範囲

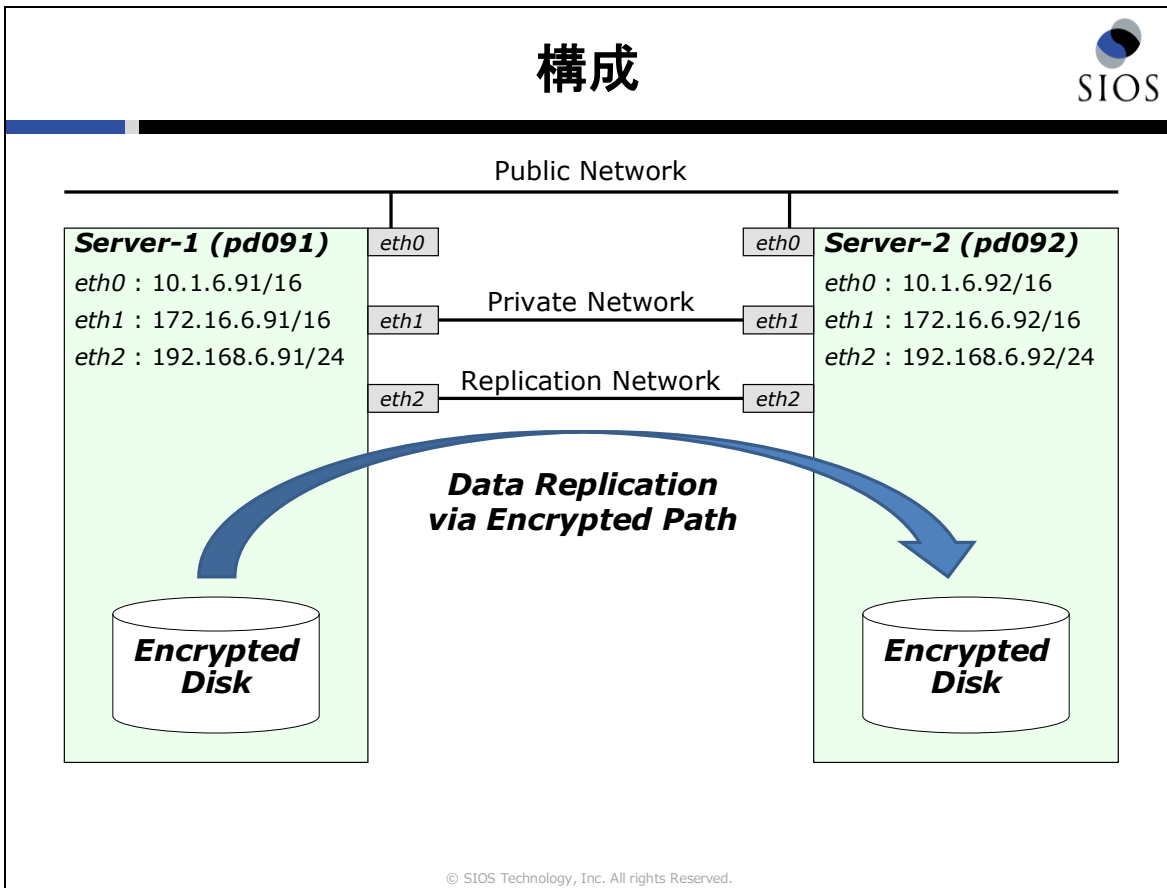
---

本ドキュメントは、国内向けに出荷された LifeKeeper for Linux に対してのみ有効です。製品に関する詳細は米国の SIOS Technology Corp.より提供されている各種技術ドキュメント (Release Notes, Technical Documentation)をご参照ください。

## 3. 本ドキュメントについて

本ドキュメントで想定している構成は以下の図の通りであり、使用する LifeKeeper および ディストリビューションは LifeKeeper for Linux v8.1.1 および 64 ビット版 Red Hat Enterprise Linux 6.3 です。

ローカルディスクの暗号化には LUKS (Linux Unified Key Setup) を使用し、通信経路の暗号化には Openswan を使用します。



# 4. LifeKeeper の構築

---

## 4.1. LifeKeeper のインストールおよび設定

---

LifeKeeper のインストールおよび設定の詳細につきましては、「LifeKeeper for Linux v8.1.1 スタートアップガイド」に記載されておりますので、そちらをご参照ください。

なお、本ドキュメントでは DataKeeper for Linux を使用します。スタートアップガイド「3.2.8. オプションの Recovery Kit パッケージのインストール」の手順にて「DataKeeper for Linux」を選択します。

## 4.2. Firewall の確認

---

本ドキュメントでは Firewall は無効にした状態で構築を行います。本番環境では適切に Firewall を適用して運用してください。LifeKeeper として使用するポートにつきましてはスタートアップガイド「2.3. Firewall の確認」に記載しております。

## 4.3. 通信経路およびディスクの暗号化に必要なパッケージ

---

本ドキュメントにおいて追加で必要となるパッケージは以下の通りです。tcpdump は IPsec の動作確認に使用します。

- openswan
- cryptsetup-luks
- tcpdump

## 4.4. 本構成の留意事項

---

本構成では、物理的なディスクに保存されるデータおよび物理的な通信経路を通るデータは暗号化されていますが、上位のアプリケーションや LifeKeeper 上で利用されるデータ自体は暗号化されていない平文のデータですので予めご了承ください。

## 5. 通信経路の暗号化

---

本項ではレプリケーションに使用するコミュニケーションパス(Replication Network)をトランスポートモードで事前共通鍵を使用して暗号化します。

### 5.1. sysctl.conf ファイルの設定

---

Openswan を使用し IPsec 通信を行うため、以下の様に sysctl.conf ファイルのパラメータを変更および追加し、システムに適用します。

- /etc/sysctl.conf ファイルの編集

```
net.ipv4.ip_forward = 1
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.eth0.send_redirects = 0
net.ipv4.conf.eth1.send_redirects = 0
net.ipv4.conf.eth2.send_redirects = 0
net.ipv4.conf.lo.send_redirects = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.eth0.accept_redirects = 0
net.ipv4.conf.eth1.accept_redirects = 0
net.ipv4.conf.eth2.accept_redirects = 0
net.ipv4.conf.lo.accept_redirects = 0
```
- システムに変更を適用する

```
# sysctl -p
```

### 5.2. ipsec.conf ファイルの編集

---

ノード間の IPsec に関する設定を/etc/ipsec.conf ファイルに記述します。

- /etc/ipsec.conf の書式

```
conn <接続名>
    left=<相手の IP>
    right=<自分の IP>
```

## DataKeeper for Linux with Encrypted Path and Disk

```
type=<接続モード>
authby=<認証の種類>
auto=<コネクション設定>
```

Server-1 における設定は以下の通りです。

- /etc/ipsec.conf の編集

```
conn replicationpath
left=192.168.6.92
right=192.168.6.91
type=transport
authby=secret
auto=start
```

### 5.3. ipsec.secrets ファイルの編集

---

ノード間で使用する事前共通鍵を/etc/ipsec.secrets ファイルに記述します。パスフレーズは各ノードで同一である必要があります。

- /etc/ipsec.secrets の書式

```
<相手の IP> <自分の IP> : PSK "<パスフレーズ>"
```

Server-1 における設定は以下の通りです。(任意のパスフレーズを入力してください)

- /etc/ipsec.secrets の編集

```
192.168.6.92 192.168.6.91 : PSK "_____"
```

### 5.4. Openswan の起動

---

以下のコマンドで Openswan を起動し IPsec を開始します。

```
# /etc/init.d/ipsec start
```

以下のコマンドで OS 起動時に Openswan が自動起動するように設定します

```
# chkconfig ipsec on
```

### 5.5. 残りノードでの設定

---

対向ノード(Server-2)でも同様の手順で IPsec の設定を行います。

### 5.6. tcpdump による動作確認

---

Server-1 において以下の tcpdump コマンドを実行し、Server-2 との間の Replication Network を通過するパケットをキャプチャします。

```
# tcpdump host 192.168.6.92 -tni eth2
```

#### 5.6.1. IPsec 設定前の tcpdump の出力例

---

IPsec 設定が適用される以前は以下の様に、コミュニケーションパスの通信が行われている事がわかります。

```
IP 192.168.6.91.33680 > 192.168.6.92.lcm-server: Flags [P.], seq 240:244, ack 3, win 115, options [nop,nop,TS val 607787980 ecr 652514744], length 4
IP 192.168.6.92.lcm-server > 192.168.6.91.33680: Flags [.], ack 244, win 114, options [nop,nop,TS val 652519713 ecr 607787980], length 0
IP 192.168.6.91.33680 > 192.168.6.92.lcm-server: Flags [P.], seq 244:360, ack 3, win 115, options [nop,nop,TS val 607788020 ecr 652519713], length 116
IP 192.168.6.92.lcm-server > 192.168.6.91.33680: Flags [.], ack 360, win 114, options [nop,nop,TS val 652519713 ecr 607788020], length 0
IP 192.168.6.92.lcm-server > 192.168.6.91.33680: Flags [P.], seq 3:4, ack 360, win 114, options [nop,nop,TS val 652519744 ecr 607788020], length 1
IP 192.168.6.91.33680 > 192.168.6.92.lcm-server: Flags [.], ack 4, win 115, options [nop,nop,TS val 607788052 ecr 652519744], length 0
IP 192.168.6.92.34017 > 192.168.6.91.lcm-server: Flags [P.], seq 240:244, ack 3, win 115, options [nop,nop,TS val 652521488 ecr 607784860], length 4
IP 192.168.6.91.lcm-server > 192.168.6.92.34017: Flags [.], ack 244, win 114, options [nop,nop,TS val 607789836 ecr 652521488], length 0
IP 192.168.6.92.34017 > 192.168.6.91.lcm-server: Flags [P.], seq 244:360, ack 3, win 115, options [nop,nop,TS val 652521528 ecr 607789836], length 116
IP 192.168.6.91.lcm-server > 192.168.6.92.34017: Flags [.], ack 360, win 114, options [nop,nop,TS val 607789836 ecr 652521528], length 0
IP 192.168.6.91.lcm-server > 192.168.6.92.34017: Flags [P.], seq 3:4, ack 360, win 114, options [nop,nop,TS val 607789860 ecr 652521528], length 1
IP 192.168.6.92.34017 > 192.168.6.91.lcm-server: Flags [.], ack 4, win 115, options [nop,nop,TS val 652521553 ecr 607789860], length 0
```



### 5.6.2. IPsec 設定後の tcpdump の出力例

---

IPsec 設定が適用された以降は以下の様に全ての通信が ESP で暗号化され、通信の内容を知る事ができなくなります。

```
IP 192.168.6.91 > 192.168.6.92: ESP(spi=0xf1a56fa6, seq=0x91), length 84
IP 192.168.6.92 > 192.168.6.91: ESP(spi=0xa37fa419, seq=0x90), length 84
IP 192.168.6.91 > 192.168.6.92: ESP(spi=0xf1a56fa6, seq=0x92), length 196
IP 192.168.6.92 > 192.168.6.91: ESP(spi=0xa37fa419, seq=0x91), length 84
IP 192.168.6.92 > 192.168.6.91: ESP(spi=0xa37fa419, seq=0x92), length 84
IP 192.168.6.91 > 192.168.6.92: ESP(spi=0xf1a56fa6, seq=0x93), length 84
IP 192.168.6.92 > 192.168.6.91: ESP(spi=0xa37fa419, seq=0x93), length 84
IP 192.168.6.91 > 192.168.6.92: ESP(spi=0xf1a56fa6, seq=0x94), length 84
IP 192.168.6.92 > 192.168.6.91: ESP(spi=0xa37fa419, seq=0x94), length 196
IP 192.168.6.91 > 192.168.6.92: ESP(spi=0xf1a56fa6, seq=0x95), length 84
IP 192.168.6.91 > 192.168.6.92: ESP(spi=0xf1a56fa6, seq=0x96), length 84
IP 192.168.6.92 > 192.168.6.91: ESP(spi=0xa37fa419, seq=0x95), length 84
```

# 6. ディスクの暗号化

---

本項ではレプリケーションに使用するローカルディスクを暗号化します。以降の手順では、ローカルディスクを暗号化するにあたり、ローカルディスク内の既存のデータは全て破棄し、新たにファイルシステムを作成しますので、データが保存されている場合はバックアップを取得してください。

## 6.1. LUKS パーティションの作成

---

以下のコマンドでは aes-xts-plain 方式、256bit の鍵長で LUKS パーティションを作成します。「YES」は**大文字**で入力し、パスワードは **2 回**入力します。

```
# cryptsetup --cipher aes-xts-plain --key-size 256 luksFormat /dev/sdb1
```

```
WARNING!
=====
This will overwrite data on /dev/sdb1 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase:
Verify passphrase:
```

各ノードで異なるパスワードである事を推奨します。(任意のパスワードを入力してください)

- Server-1 : \_\_\_\_\_
- Server-2 : \_\_\_\_\_

LUKS にはキースロットが複数あるため、複数のパスワードやキーファイルを登録できます。デバイスの展開に使用可能なキースロットは、以下のコマンドで確認できます。最初指定したパスワードによる認証は Key Slot 0 が使用されます。

- スロット状態の確認  
# cryptsetup luksDump /dev/sdb1

## DataKeeper for Linux with Encrypted Path and Disk

```
LUKS header information for /dev/sdb1

Version:          1
Cipher name:      aes
Cipher mode:      xts-plain
Hash spec:        sha1
Payload offset:   4096
MK bits:          256
MK digest:        XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
MK salt:          XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
                  XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
MK iterations:    42125
UUID:             XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

Key Slot 0: ENABLED
  Iterations:      168521
  Salt:           XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
                  XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
  Key material offset: 8
  AF stripes:     4000
Key Slot 1: DISABLED
Key Slot 2: DISABLED
Key Slot 3: DISABLED
Key Slot 4: DISABLED
Key Slot 5: DISABLED
Key Slot 6: DISABLED
Key Slot 7: DISABLED
```

### 6.2. キーファイルによる認証の追加

キーサイズが 256bit (32byte) のキーファイルを各ノードで作成し、LUKS パーティションに追加します。キーファイルの追加の際は前項で入力したパスフレーズを入力します。

- キーファイルの作成

```
# mkdir /etc/luks
# dd if=/dev/urandom of=/etc/luks/encrypted_sdb1.key bs=1 count=32
```
- キーファイルの追加

```
# cryptsetup luksAddKey /dev/sdb1 /etc/luks/encrypted_sdb1.key
```

キーファイルによる認証を追加する事により、新たに Key Slot 1 が ENABLED となり、2 種類のうちいずれかの方式での認証が可能となります。

## DataKeeper for Linux with Encrypted Path and Disk

- スロット状態の確認

```
# cryptsetup luksDump /dev/sdb1
```

```
LUKS header information for /dev/sdb1

Version:          1
Cipher name:      aes
Cipher mode:      xts-plain
Hash spec:        sha1
Payload offset:   4096
MK bits:          256
MK digest:        XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
MK salt:          XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
                  XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
MK iterations:    42125
UUID:             XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

Key Slot 0: ENABLED
  Iterations:      168521
  Salt:            XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
                  XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
  Key material offset: 8
  AF stripes:      4000
Key Slot 1: ENABLED
  Iterations:      164813
  Salt:            XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
                  XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
  Key material offset: 264
  AF stripes:      4000
Key Slot 2: DISABLED
Key Slot 3: DISABLED
Key Slot 4: DISABLED
Key Slot 5: DISABLED
Key Slot 6: DISABLED
Key Slot 7: DISABLED
```

### 6.3. LUKS パーティションの展開

---

本項で可能な LUKS パーティションの展開方法は以下の 2 つがあります。

- パスフレーズによる展開

```
# cryptsetup luksOpen /dev/sdb1 encrypted_sdb1
```

- キーファイルによる展開

```
# cryptsetup luksOpen /dev/sdb1 encrypted_sdb1 ¥
```

## DataKeeper for Linux with Encrypted Path and Disk

```
> --key-file /etc/luks/encrypted_sdb1.key
```

上記何れ方法で展開した場合でも以下のマッパーデバイスへのリンクファイルが作成され、当該ファイルをブロックデバイスとして使用することで、暗号化したローカルディスク内のデータを複合化した状態で読み書きできます。データレプリケーションではこのリンクファイルが指すマッパーデバイスを使用します。

```
/dev/mapper/encrypted_sdb1
```

### 6.4. /etc/crypttab ファイルの設定

---

/etc/cyprttab ファイルを作成する事で、OS 起動時に LUKS パーティションの展開を行います。

- /etc/crypttab ファイルの作成  
encrypted\_sdb1 /dev/sdb1 /etc/luks/encrypted\_sdb1.key luks

### 6.5. 残りノードでの設定

---

対向ノード(Server-2)でも同様の手順で LUKS の設定を行います。

## 7. データレプリケーションリソースの作成

データレプリケーションリソース作成対照のファイルシステムを任意のディレクトリにマウントします。本ドキュメントでは、`/dev/mapper/encrypted_sdb1` を `/mnt/dr` にマウントしています。

#	df	Filesystem	1K-ブロック	使用	使用可	使用%	マウント位置
		/dev/mapper/vg_pd091-lv_root	11941808	5625872	5709320	50%	/
		tmpfs	961308	264	961044	1%	/dev/shm
		/dev/sda1	495844	37562	432682	8%	/boot
		/dev/sr0	3592530	3592530	0	100%	/media/RHEL_6.3_x86_64_D
		isc 1					
		/dev/mapper/encrypted_sdb1	16506524	176108	15491928	2%	/mnt/dr

データレプリケーションリソースの作成は下記の順序で行います。

表 1 ファイルシステムリソースの設定値

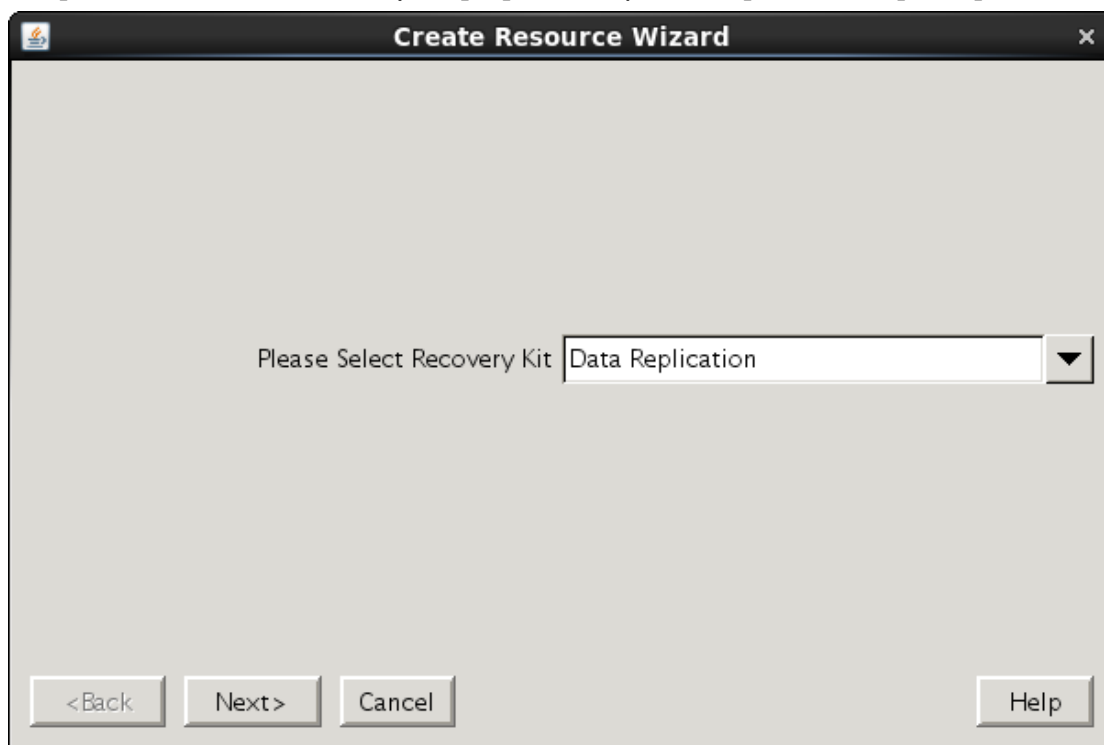
番号	項目	入力もしくは選択する値
1	Please Select Recovey Kit	Data Replication を選択
2	Switchback Type (ソースサーバ)	intelligent を選択
3	Server	ソースサーバ名を選択
4	Hierarchy Type	Replicate Existing Filesystem を選択
5	Existing Mount Point	レプリケーション元となるマウントポイントを選択
6	Data Repolication Resource Tag (ソースサーバ)	タグ名を選択もしくは入力
7	File System Resource Tag (ソースサーバ)	タグ名を選択もしくは入力
8	Bitmap File (ソースサーバ)	ビットマップファイルのパスの選択もしくは入力
9	Enable Asynchronous Replication ?	yes もしくは no を選択
10	Target Server	ターゲットサーバ名を選択
11	Switchback Type (ターゲットサーバ)	intelligent を選択
12	Template Priority (ソースサーバ)	デフォルト値を選択
13	Target Priority (ターゲットサーバ)	デフォルト値を選択

## DataKeeper for Linux with Encrypted Path and Disk

14	Target Disk	レプリケーション先となるデバイスを選択
15	Data Replication Resource Tag (ターゲットサーバ)	デフォルト値を選択
16	Bitmap File (ターゲットサーバ)	ビットマップファイルのパスの選択もしくは入力
17	Replication Path	同期経路を選択
18	Replication Type	同期モードを選択
19	Mount Pint (ターゲットサーバ)	デフォルト値を選択
20	Root Tag (ターゲットサーバ)	デフォルト値を選択

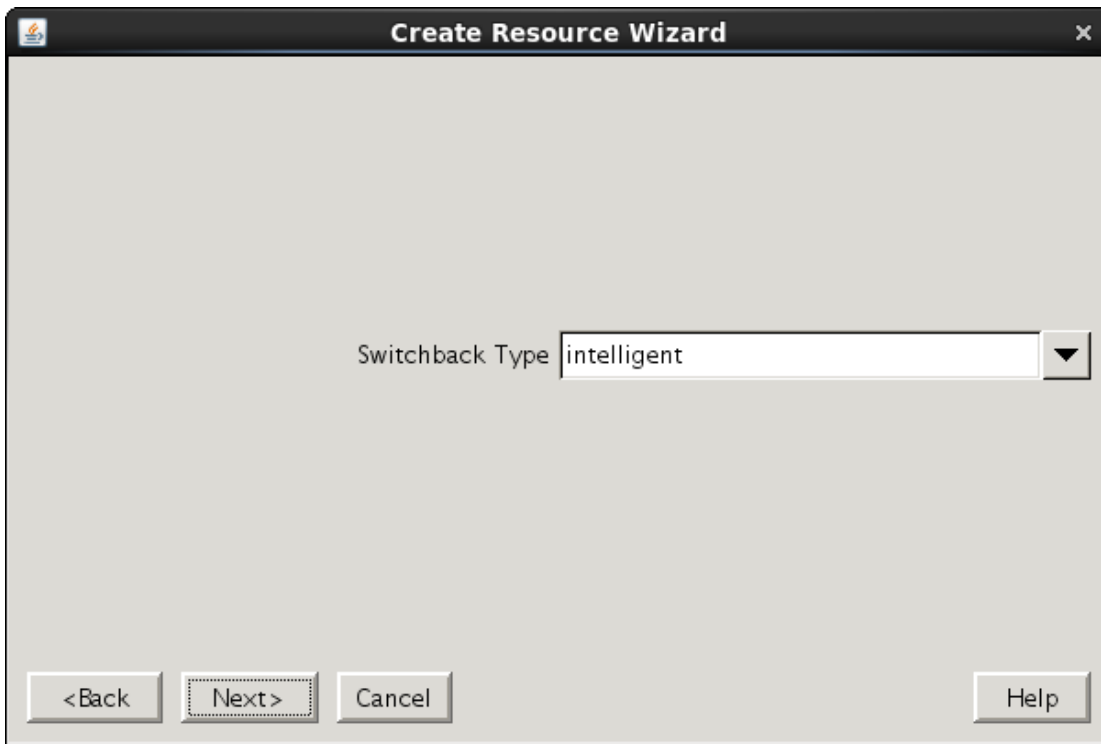
リソースを作成するためのウィザードを起動した後は、以下の順序で進めます。

1. [Please Select Recovery Kit]で[Data Replication]を選択し、[Next]をクリック

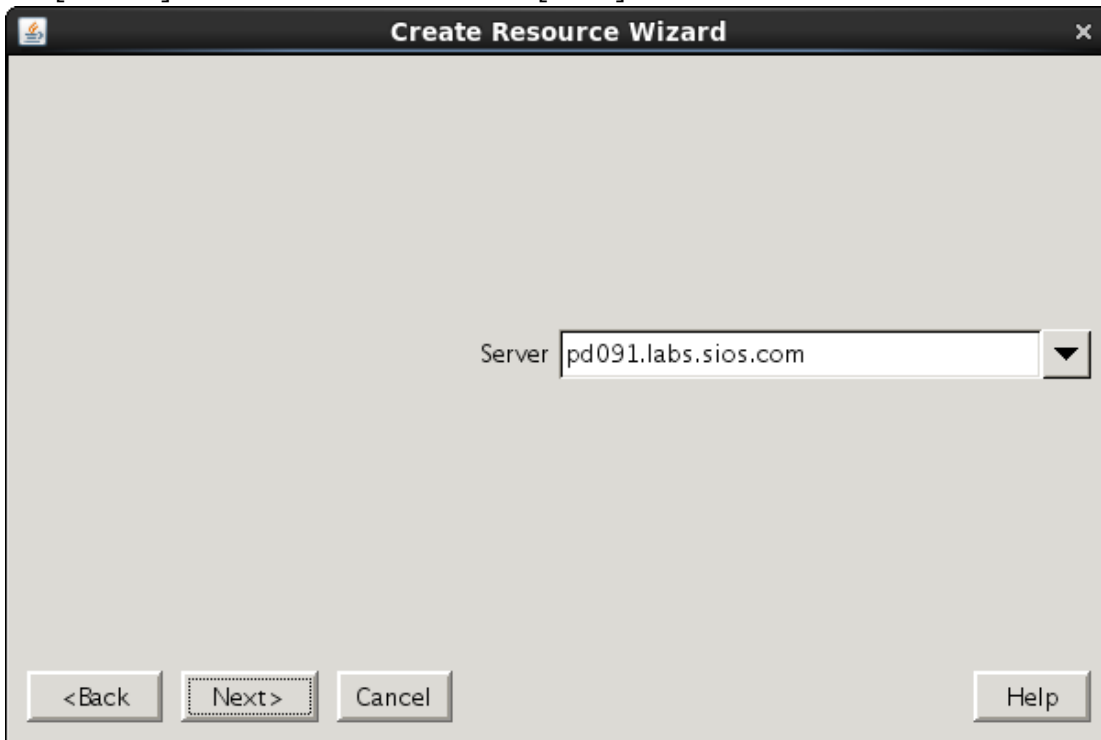


## DataKeeper for Linux with Encrypted Path and Disk

2. [Switchback Type]で[intelligent]を選択し、[Next]をクリック



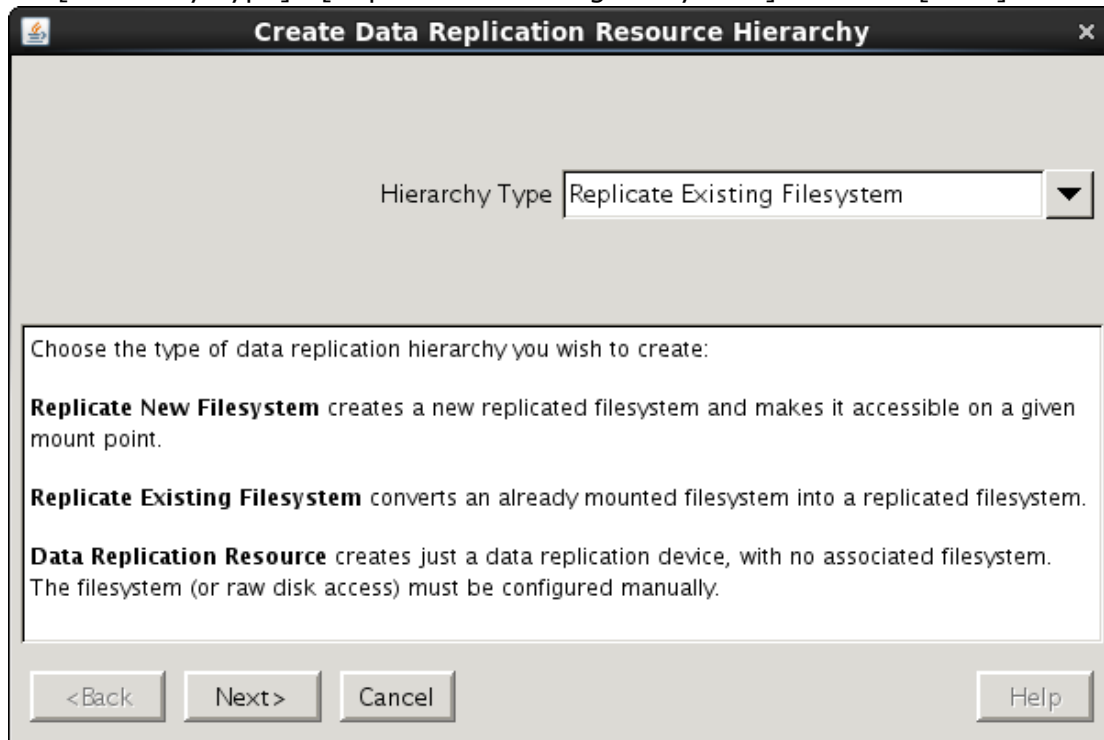
3. [Server]でソースサーバ名を選択し、[Next]をクリック



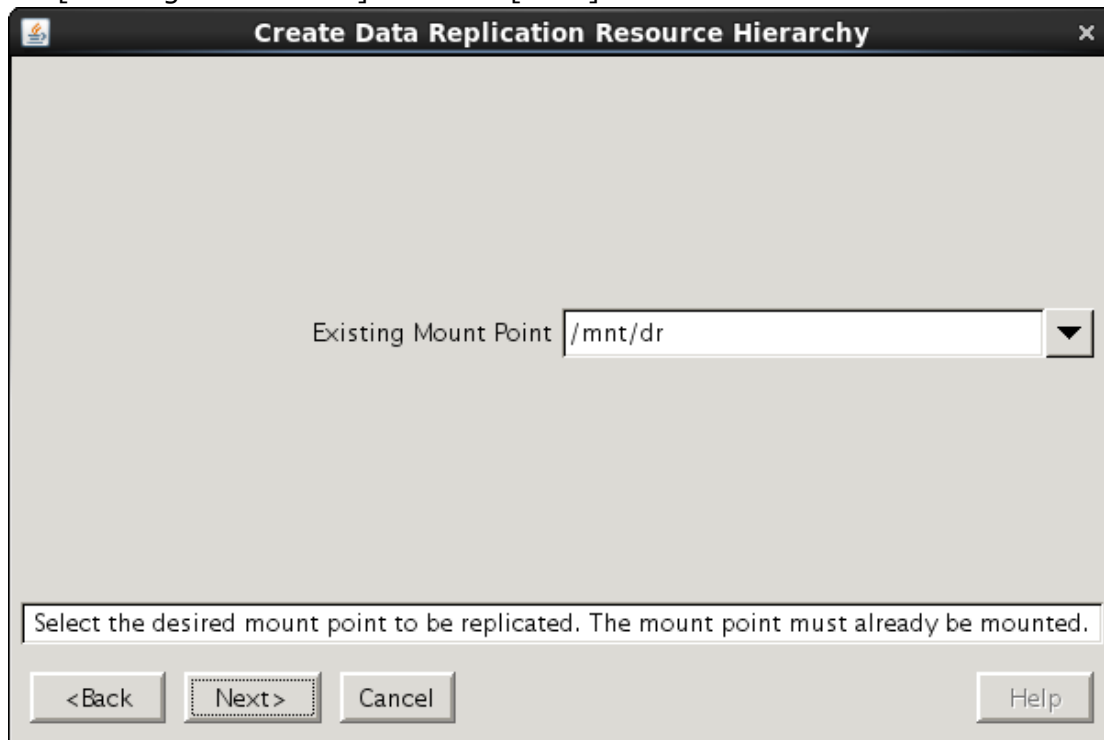


## DataKeeper for Linux with Encrypted Path and Disk

4. [Hierarchy Type]で[Replicate Existing Filesystem]を選択し、[Next]をクリック

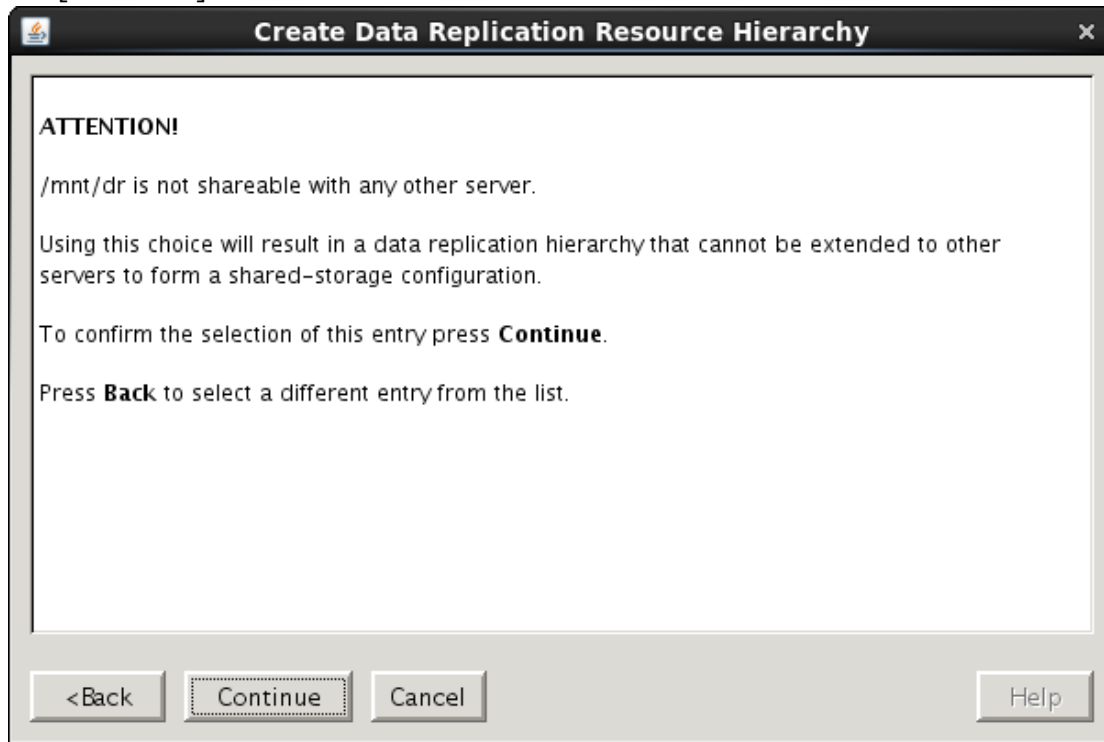


5. [Existing Mount Point]を選択し、[Next]をクリック

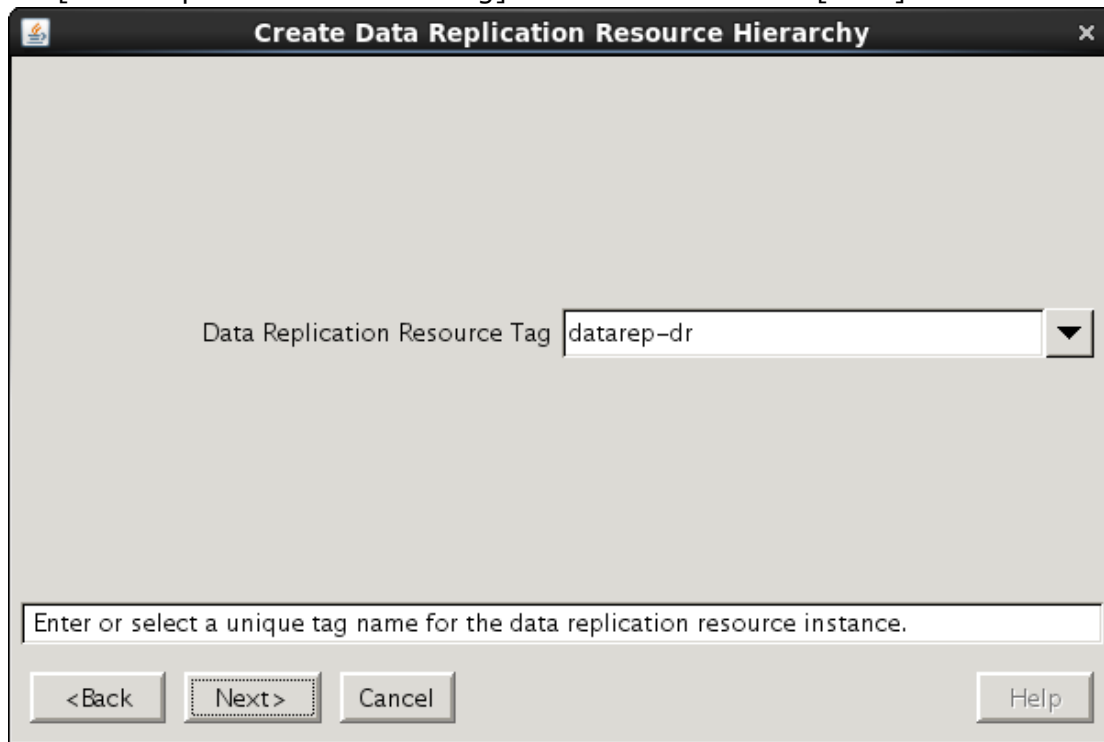


## DataKeeper for Linux with Encrypted Path and Disk

[Continue]をクリック



6. [Data Replication Resource Tag]を選択もしくは入力し、[Next]をクリック



## DataKeeper for Linux with Encrypted Path and Disk

7. [File System Resource Tag]を選択もしくはは入力し、[Next]をクリック

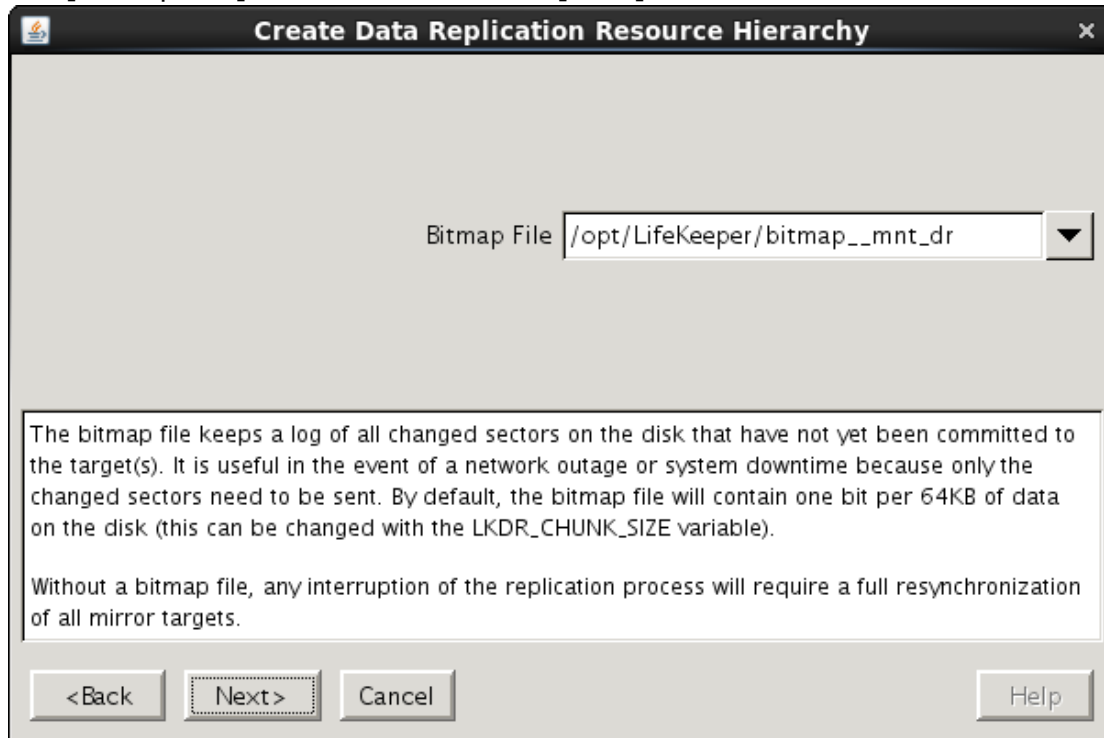


File System Resource Tag

Enter or select a unique tag name for the filesystem resource instance.

<Back Next > Cancel Help

8. [Bitmap File]を選択もしくはは入力し、[Next]をクリック



Bitmap File

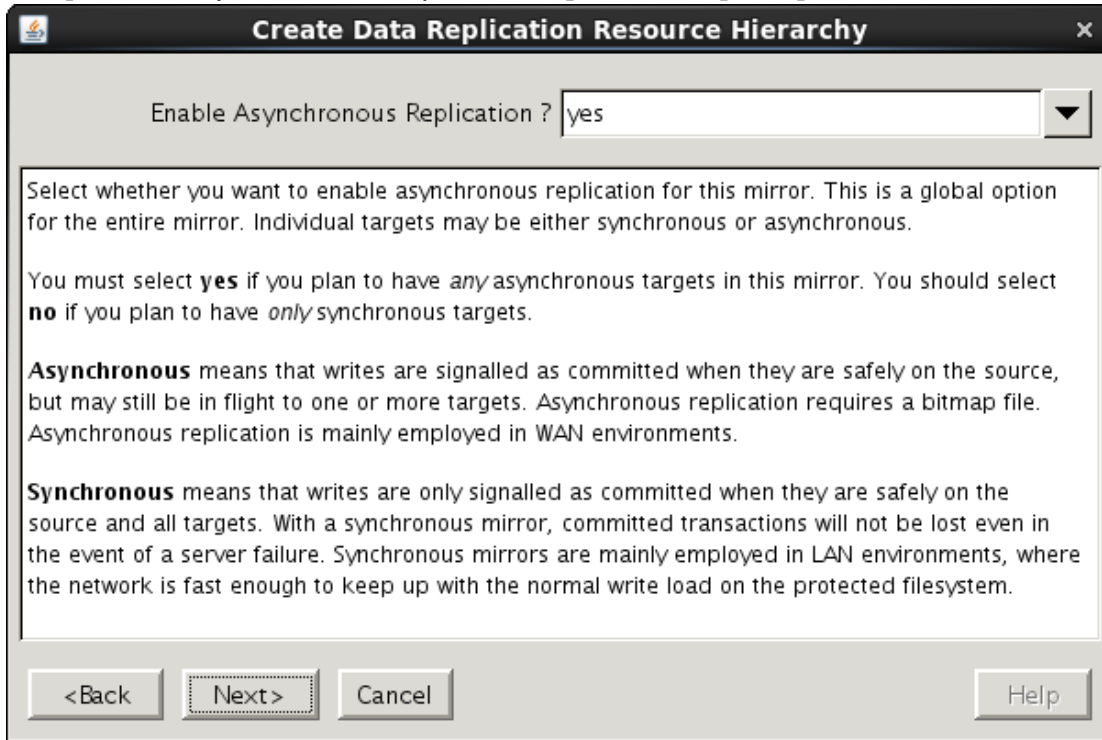
The bitmap file keeps a log of all changed sectors on the disk that have not yet been committed to the target(s). It is useful in the event of a network outage or system downtime because only the changed sectors need to be sent. By default, the bitmap file will contain one bit per 64KB of data on the disk (this can be changed with the LKDR\_CHUNK\_SIZE variable).

Without a bitmap file, any interruption of the replication process will require a full resynchronization of all mirror targets.

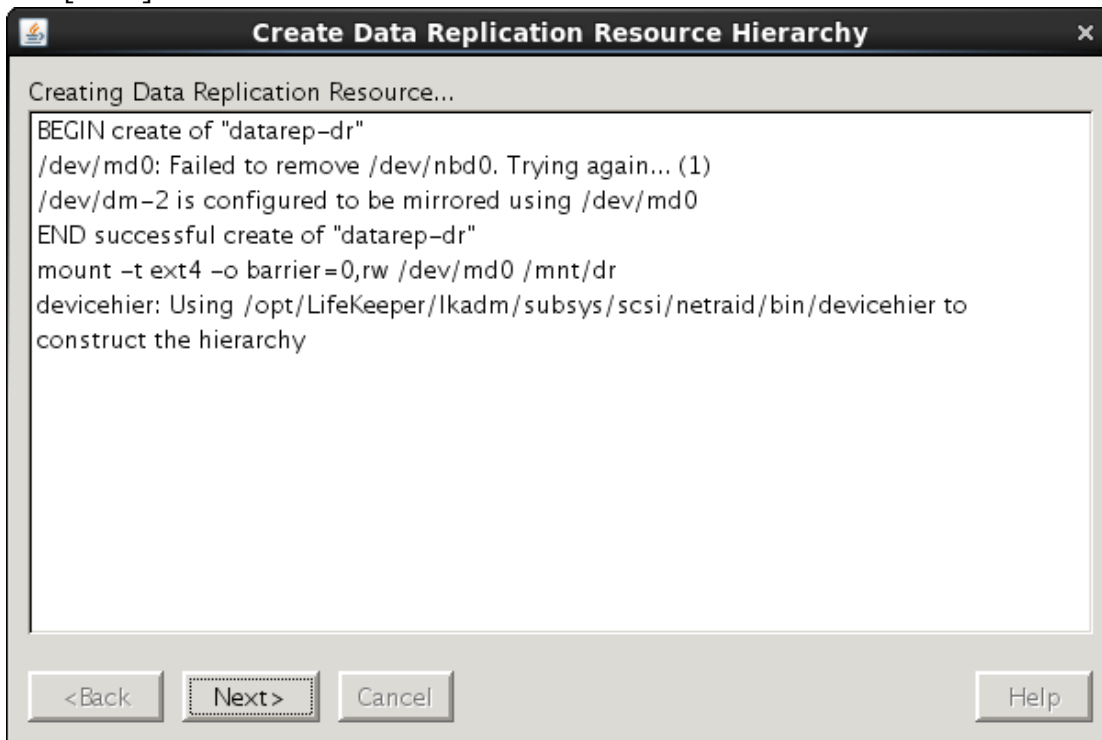
<Back Next > Cancel Help

## DataKeeper for Linux with Encrypted Path and Disk

9. [Enable Asynchronous Replication ?]を選択し、[Next]をクリック

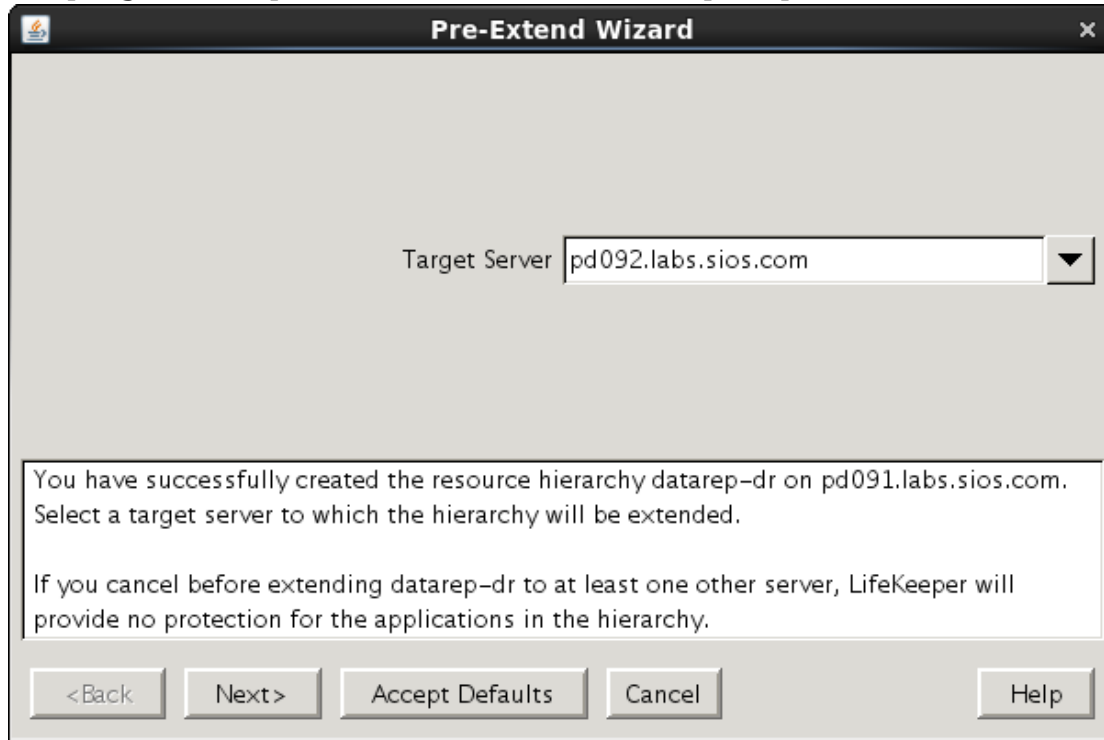


[Next]をクリック

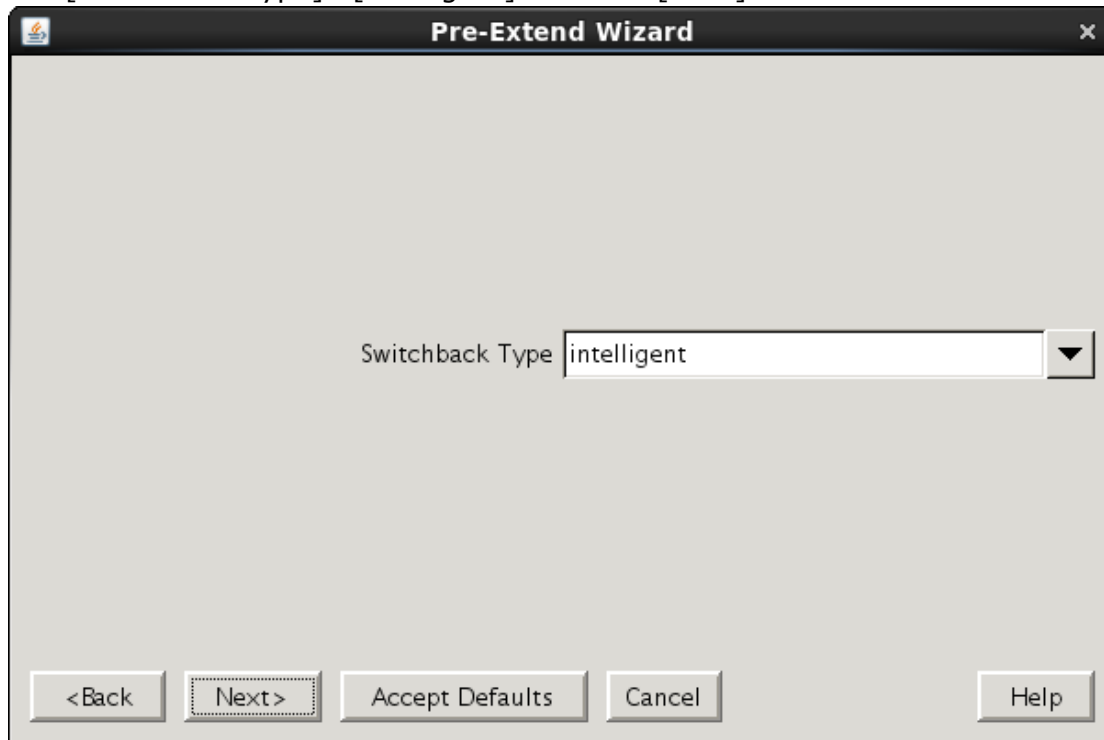


## DataKeeper for Linux with Encrypted Path and Disk

10. [Target Server]でターゲットサーバ名を選択し、[Next]をクリック

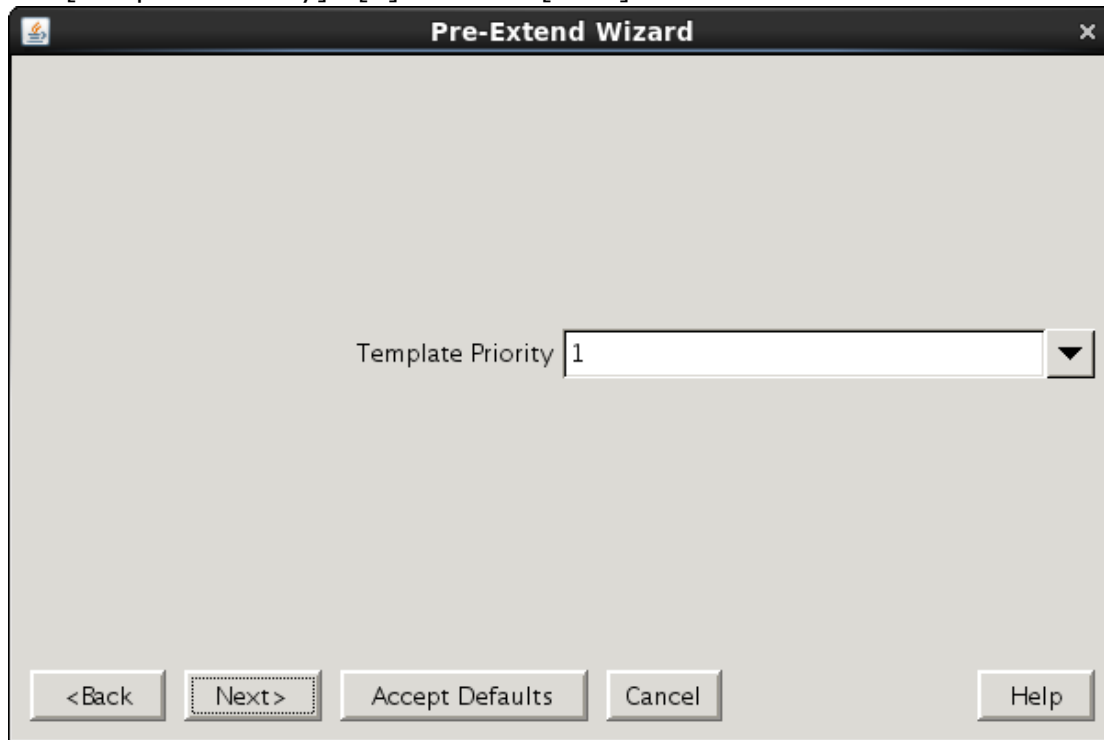


11. [Switchback Type]で[intelligent]を選択し、[Next]をクリック

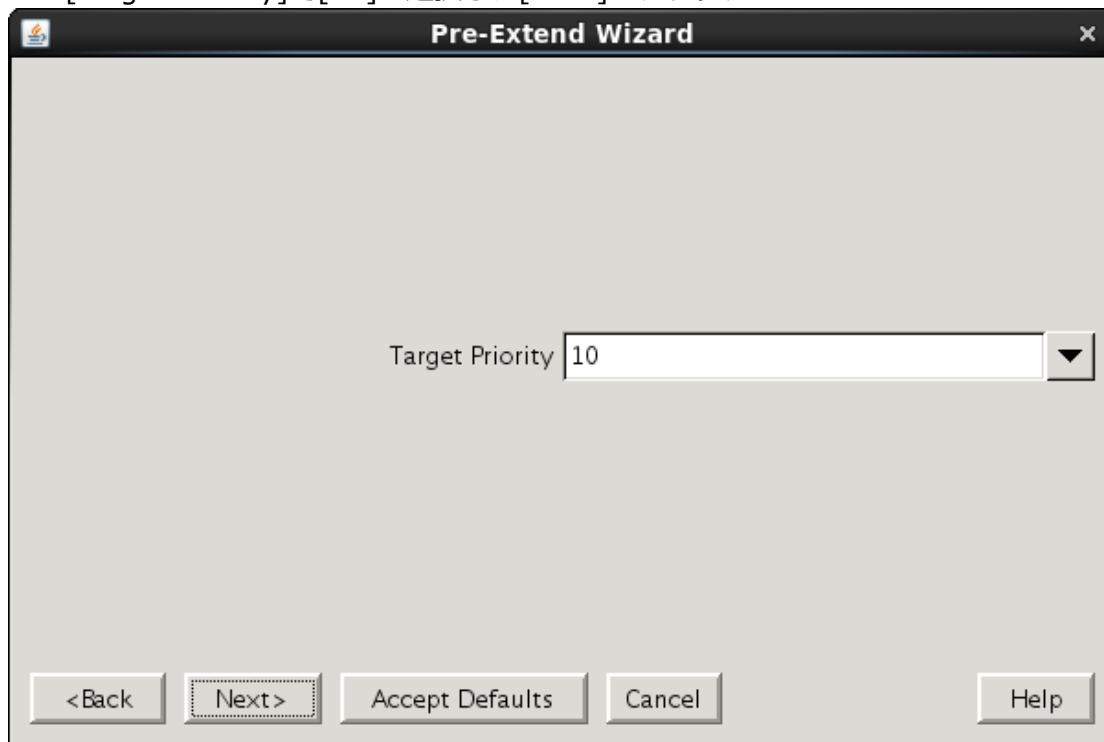


## DataKeeper for Linux with Encrypted Path and Disk

12. [Template Priority]で[1]を選択し、[Next]をクリック

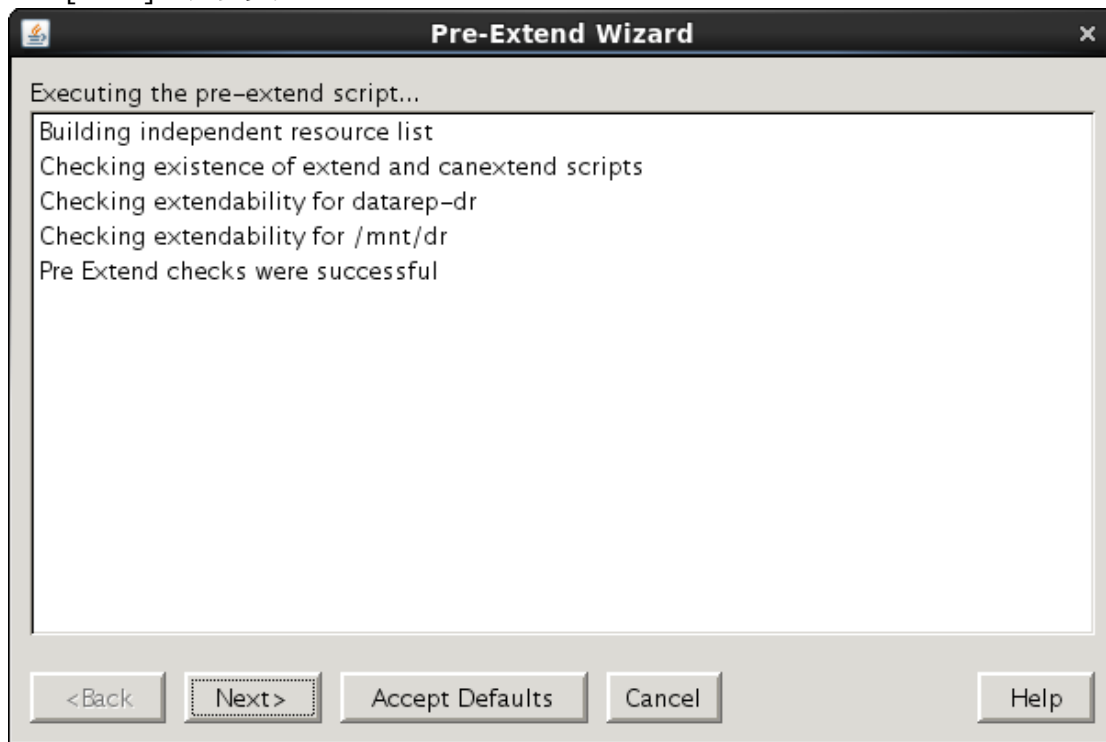


13. [Target Priority]で[10]を選択し、[Next]をクリック



## DataKeeper for Linux with Encrypted Path and Disk

[Next]をクリック



14. [Target Disk]を選択し、[Next]をクリック



## DataKeeper for Linux with Encrypted Path and Disk

15. [Data Replication Resource Tag]を入力し、[Next]をクリック

Extend Data Replication Resource

Template Server: pd091.labs.sios.com  
Tag to Extend: datarep-dr  
Target Server: pd092.labs.sios.com

Data Replication Resource Tag

Enter or select a unique tag name for the data replication resource instance.

<Back Next > Accept Defaults Cancel Help

16. [Bitmap File]を選択もしくは入力し、[Next]をクリック

Extend Data Replication Resource

Template Server: pd091.labs.sios.com  
Tag to Extend: datarep-dr  
Target Server: pd092.labs.sios.com

Bitmap File

The bitmap file keeps a log of all changed sectors on the disk that have not yet been committed to the target(s). It is useful in the event of a network outage or system downtime because only the changed sectors need to be sent. By default, the bitmap file will contain one bit per 64KB of data on the disk (this can be changed with the LKDR\_CHUNK\_SIZE variable).

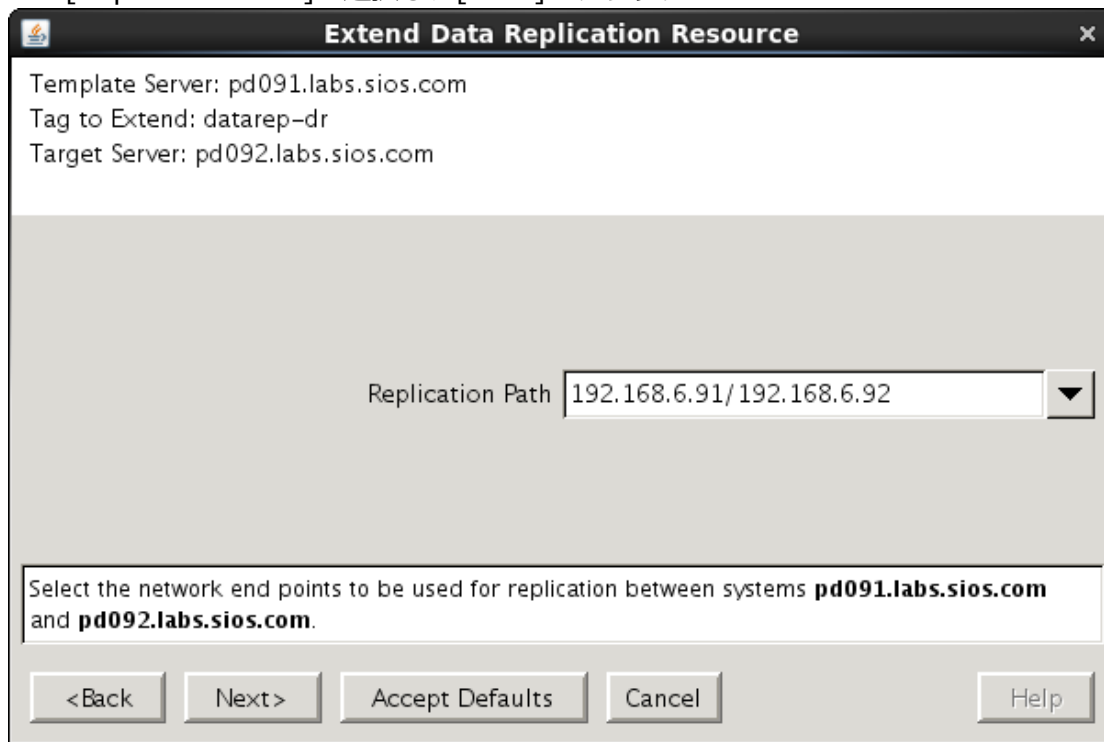
On the target, the bitmap file is necessary when replication changes roles (i.e., on switchover or failover). Without a bitmap file, it is impossible to switch back again to the source without transmitting all the data from the target.

<Back Next > Accept Defaults Cancel Help

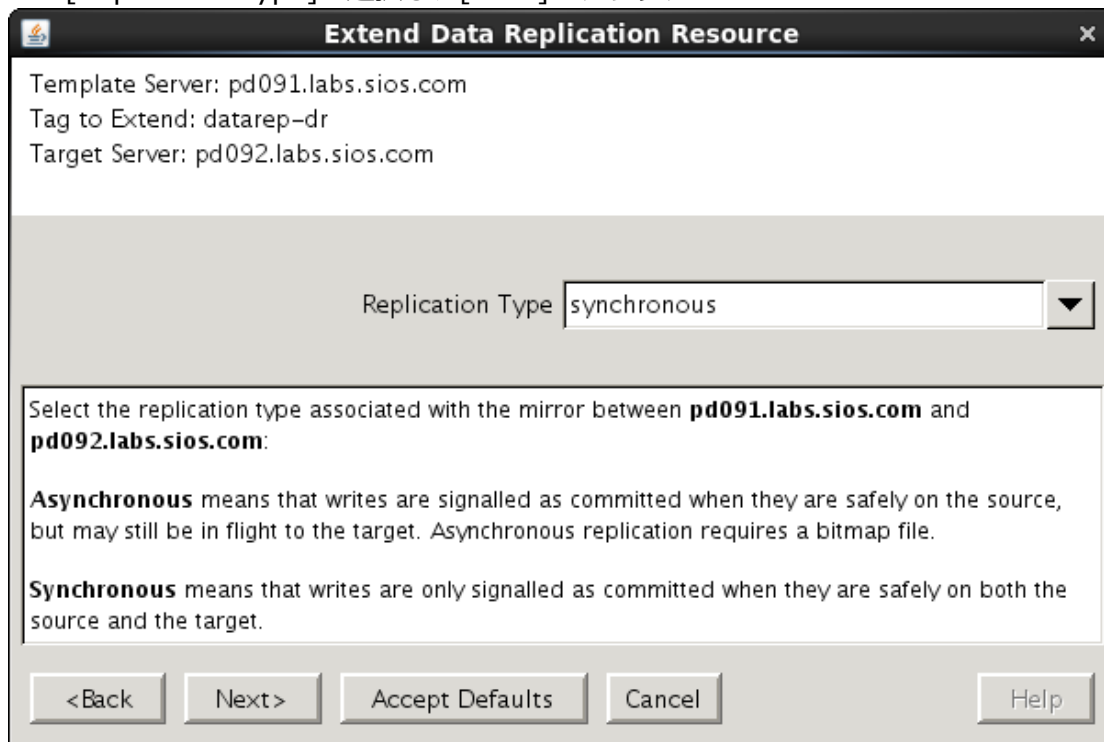


## DataKeeper for Linux with Encrypted Path and Disk

17. [Replication Path]を選択し、[Next]をクリック

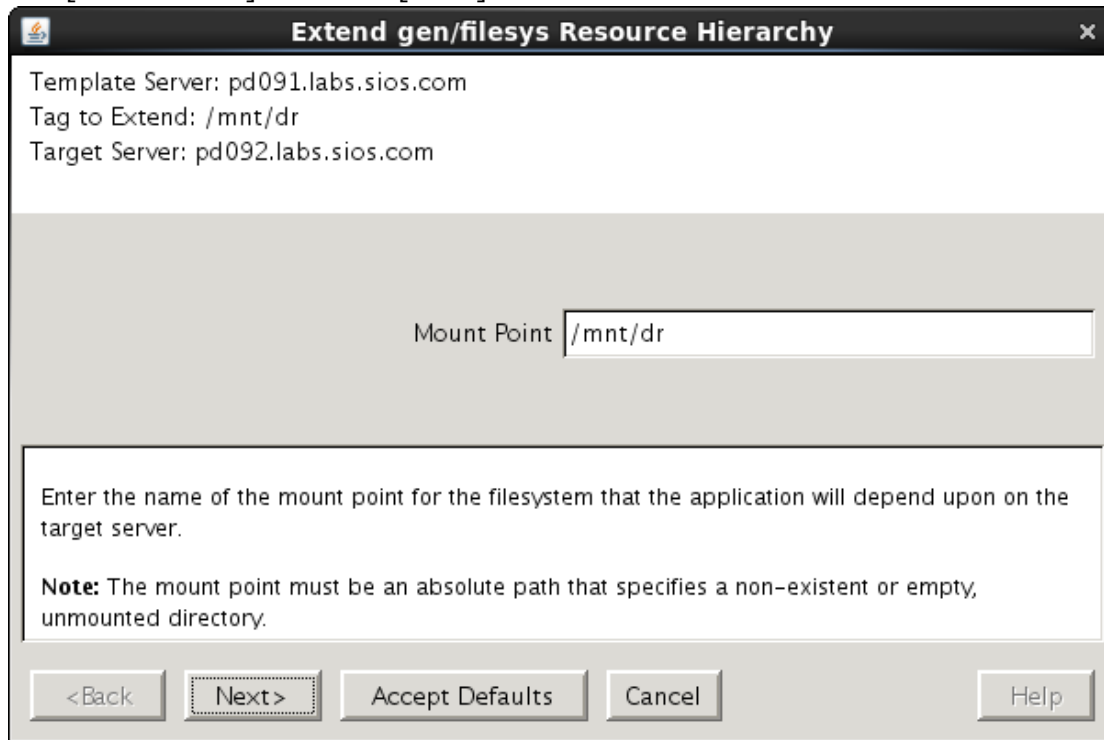


18. [Replication Type]を選択し、[Next]をクリック



## DataKeeper for Linux with Encrypted Path and Disk

19. [Mount Point]を入力し、[Next]をクリック



Extend gen/filesys Resource Hierarchy

Template Server: pd091.labs.sios.com  
Tag to Extend: /mnt/dr  
Target Server: pd092.labs.sios.com

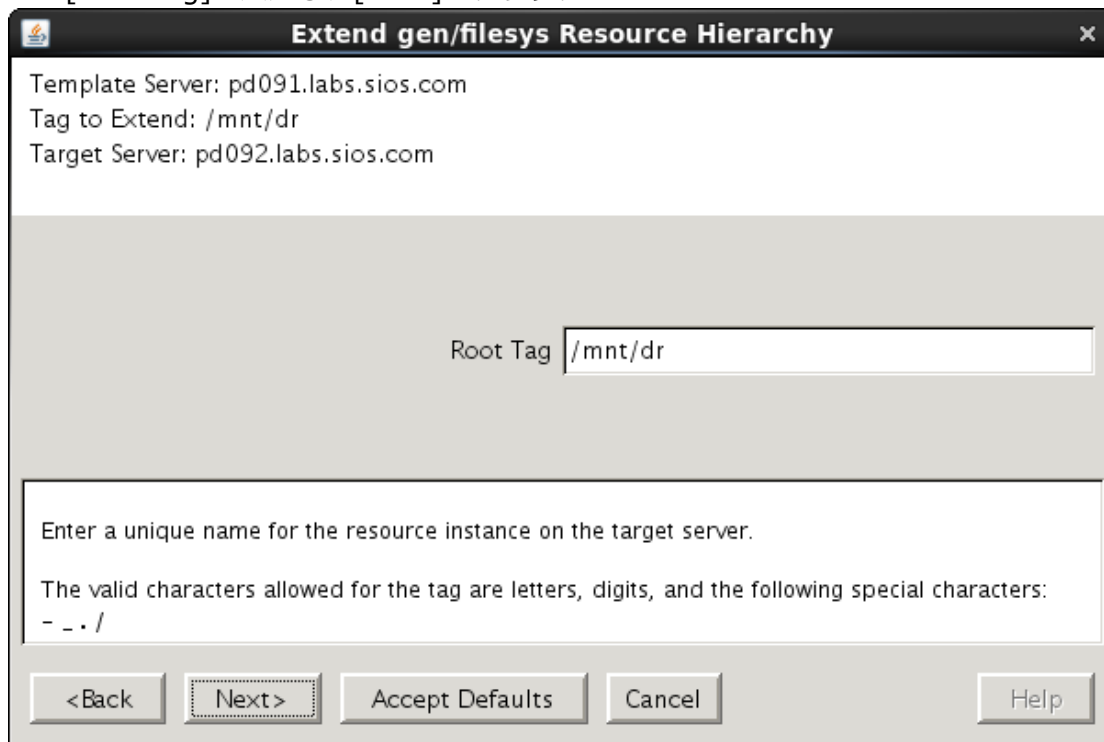
Mount Point

Enter the name of the mount point for the filesystem that the application will depend upon on the target server.

**Note:** The mount point must be an absolute path that specifies a non-existent or empty, unmounted directory.

<Back Next> Accept Defaults Cancel Help

20. [Root Tag]を入力し、[Next]をクリック



Extend gen/filesys Resource Hierarchy

Template Server: pd091.labs.sios.com  
Tag to Extend: /mnt/dr  
Target Server: pd092.labs.sios.com

Root Tag

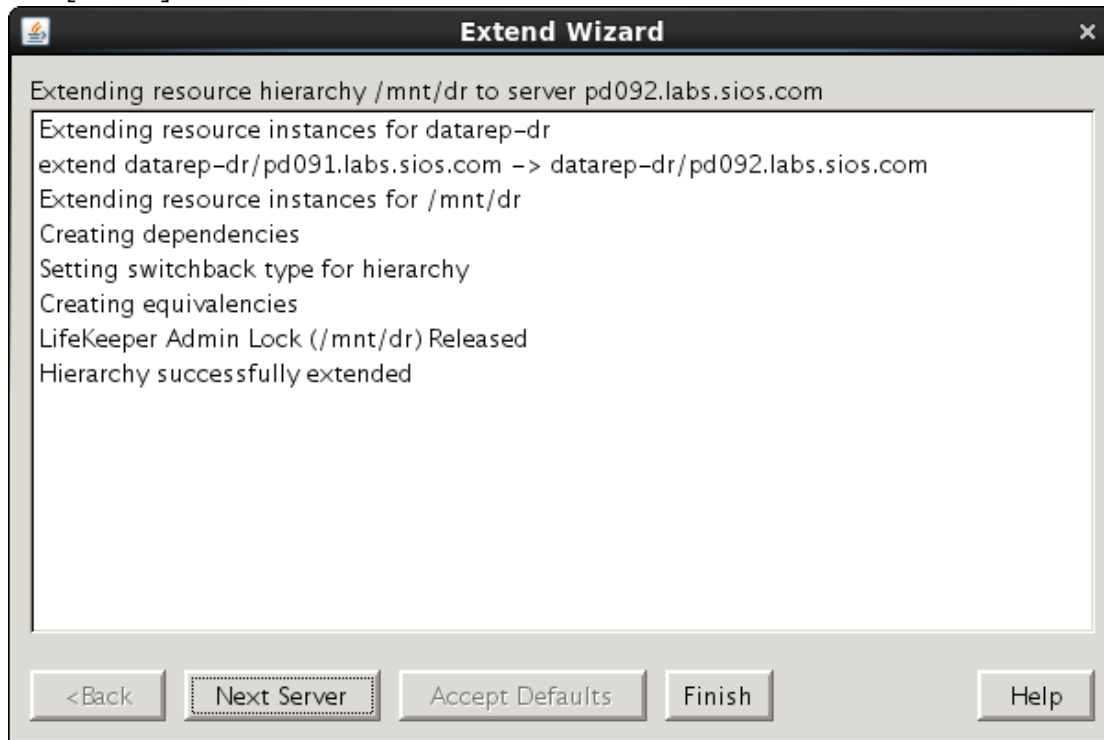
Enter a unique name for the resource instance on the target server.

The valid characters allowed for the tag are letters, digits, and the following special characters:  
- \_ . /

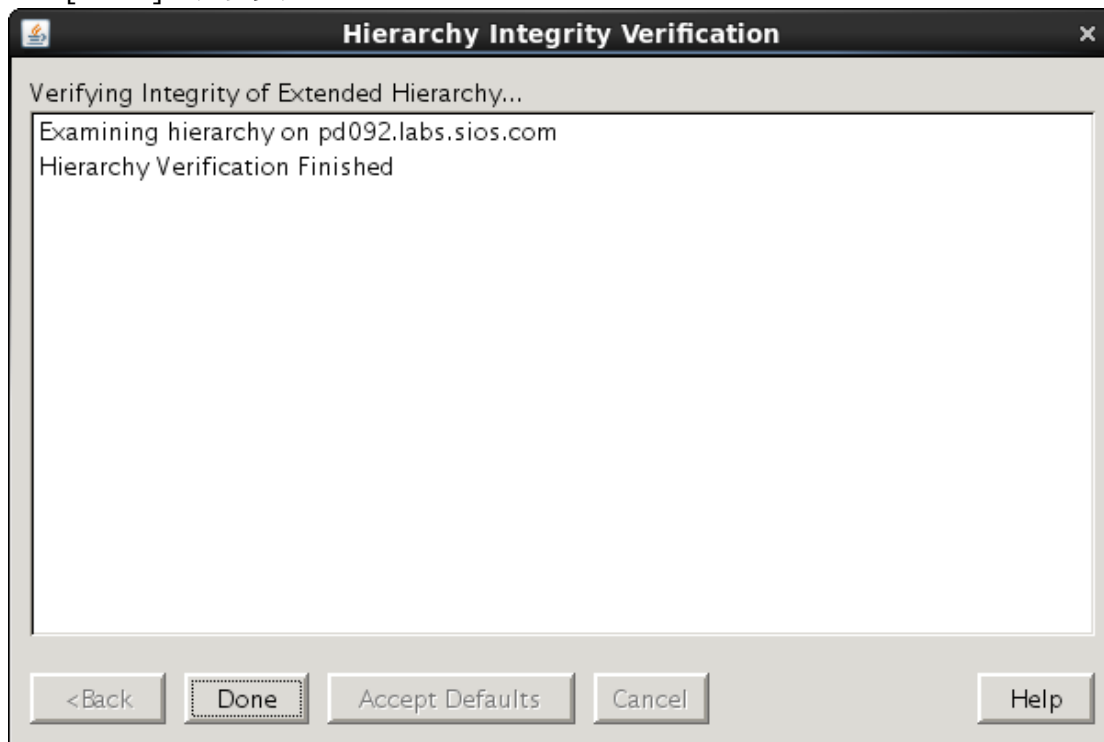
<Back Next> Accept Defaults Cancel Help

## DataKeeper for Linux with Encrypted Path and Disk

[Finish]をクリック

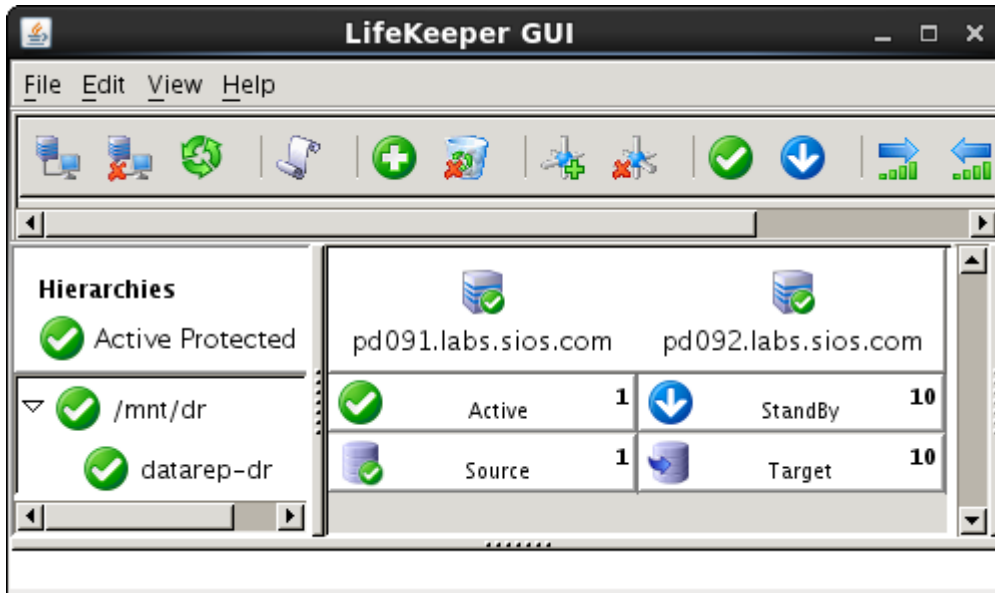


[Done]をクリック

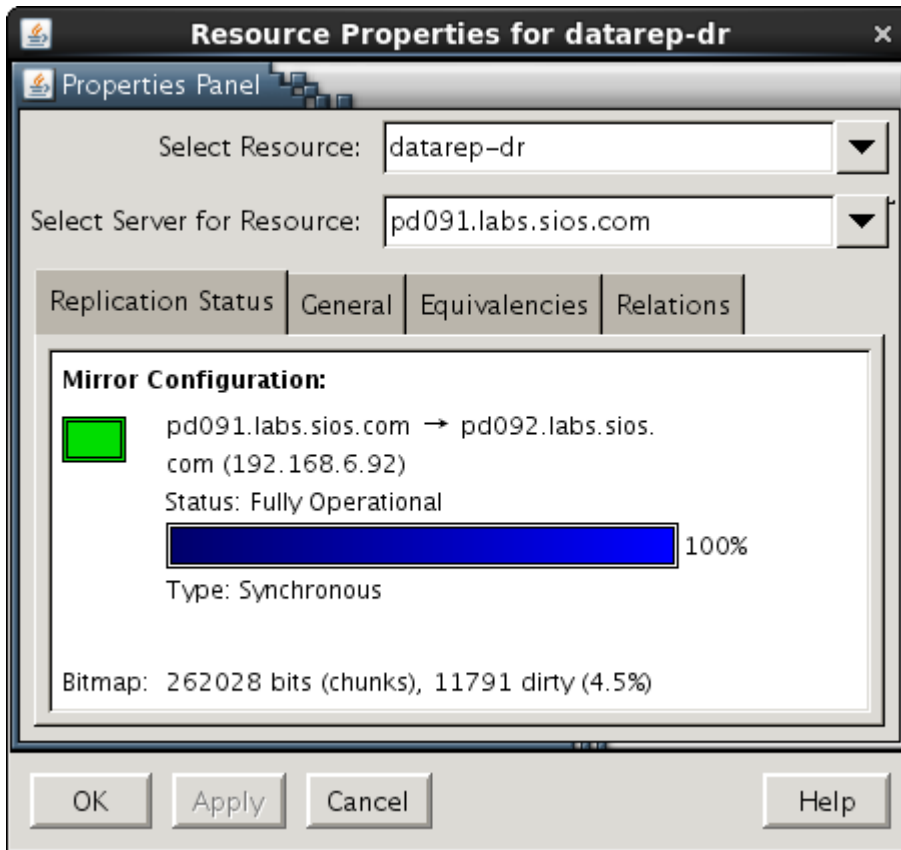


## DataKeeper for Linux with Encrypted Path and Disk

データレプリケーションリソースを作成し初期同期が完了すると、以下の様にリソースが表示されます。



データレプリケーションリソースのプロパティにおける同期完了状態は以下の通りです。



## 8. 参考資料

---

### 8.1. ユーザーサイト

---

/etc/default/LifeKeeper 設定ファイルの和訳

<http://lk.sios.com/?p=1291>

[Linux]ERROR 104052: Cannot get the hardware ID of device "デバイス名"について

<http://lk.sios.com/?p=866>

[Linux]Cannot allocate memory"エラーにより DataKeeper(SDR)リソースの作成に失敗する

<http://lk.sios.com/?p=862>

[Linux]ミラーボリュームで作成できる最大の容量は？

<http://lk.sios.com/?p=1491>

[Linux]SteelEye DataReplication(SDR)リソースの起動、停止の順序を教えてください。

<http://lk.sios.com/?p=872>

[Linux]データレプリケーション構成で膨大なトラフィックが発生する事があるのですが？

<http://lk.sios.com/?p=860>

[Linux]データレプリケーション構成においてスタンバイ側のファイルシステム(データ)に不整合が発生することがある

<http://lk.sios.com/?p=1027>

### 8.2. ホワイトペーパー

---

LifeKeeper と Fusion-io 社 ioDrive を活用した高速データベース HA ソリューション

[http://www.sios.com/products/bcp/lkdk/product/pdf/LK-FIO\\_whitepaper.pdf](http://www.sios.com/products/bcp/lkdk/product/pdf/LK-FIO_whitepaper.pdf)

LifeKeeper + ioDrive2 性能測定結果資料

<http://www.sios.com/products/bcp/lkdk/product/pdf/LK-FIO-Performce.pdf>

## **9. 著者について**

---

花島 直裕は 2009 年にソフトウェア・エンジニアとしてサイオステクノロジーに入社。  
LifeKeeper 製品の QA 業務に従事した後、2011 年より同製品のサポート業務に従事。

## 10. 免責事項

---

- 本書に記載された情報は予告なしに変更、削除される場合があります。最新のものをご確認ください。
- 本書に記載された情報は、全て慎重に作成され、記載されていますが、本書をもって、その妥当性や正確性についていかなる種類の保証もするものではありません。
- 本書に含まれた誤りに起因して、本書の利用者に生じた損害については、サイオステクノロジー株式会社は一切の責任を負うものではありません。
- 第三者による本書の記載事項の変更、削除、ホームページ及び本書等に対する不正なアクセス、その他第三者の行為により本書の利用者に生じた一切の損害について、サイオステクノロジー株式会社は一切の責任を負うものではありません。
- システム障害などの原因によりメールフォームからのお問い合わせが届かず、または延着する場合がありますので、あらかじめご了承ください。お問い合わせの不着及び延着に関し、サイオステクノロジー株式会社は一切の責任を負うものではありません。

### 【著作権】

本書に記載されているコンテンツ（情報・資料・画像等種類を問わず）に関する知的財産権は、サイオステクノロジー株式会社に帰属します。その全部、一部を問わず、サイオステクノロジー株式会社の許可なく本書を複製、転用、転載、引用、公衆への送信、販売、翻案その他の二次利用をすることはいずれも禁止されます。またコンテンツの改変、削除についても一切認められません。

本書では、製品名、ロゴなど、他社が保有する商標もしくは登録商標を使用しています。

---

サイオステクノロジー株式会社

住所：〒106-0047

東京都港区南麻布 2 丁目 12-3 サイオスビル

電話：03-6401-5161

FAX：03-6401-5162

URL：<http://www.sios.com>