

# 生ディスクイメージを使用した、IBM Z DASDディスクへのSLE Microの展開

## 概要

SLE Microでは、デバイスストレージに直接展開できる生イメージ(「事前構築のイメージ」とも呼ばれる)が提供されています。

## 目的

この記事では、IBM Zマシン上にSLE Microを展開する手順について段階を追って説明します。

## 所要時間

この記事の理解には20分ほどを要します。

## 目標

SLE Microをシステムに正常に展開します。

## 要件

- Linuxが実行されているディスク。
- 生イメージを展開し、SLE Microが実行されるディスク。
- 設定メディアとして機能するディスク。

発行日: 11/12/2025

## 目次

- 1 事前構築のイメージについて 3
- 2 SLE Microの展開 4
- 3 マシンの準備 5
- 4 設定ディスクの準備 6
- 5 DASDディスク上への生ディスクイメージの展開 23
- 6 展開後の手順 24
- 7 法的事項 26
- A GNUフリー文書利用許諾契約書(GFDL) 26

# 1 事前構築のイメージについて

事前構築のイメージとは、実行中のオペレーティングシステムを、すぐに使用できる形で表したものです。これらのイメージは、インストーラを使用する従来の方法でインストールするのではなく、ターゲットホストのハードディスクにコピーします。このトピックでは、これらの事前構築のイメージに関する基本的な情報について説明します。

事前構築のイメージは、そのイメージで提供されているツールを使用して、最初のブート時に設定することを目的としています。ブートローダは、[1.2項「最初のブートの検出」](#)で説明されているように最初のブートを検出します。各イメージにはデフォルトでマウントされるサブボリュームが付属しており、最初のブート設定時に変更できます。サブボリュームに関する詳細については、[1.1項「デフォルトパーティショニング」](#)を参照してください。

## 1.1 デフォルトパーティショニング

事前構築のイメージは、デフォルトのパーティション方式で用意されています。このパーティションは、最初のブート時にIgnitionまたはCombustionを使用して変更できます。

### ！ 重要: Btrfsはルートファイルシステムに必須である

デフォルトのパーティション方式を変更する場合は、ルートファイルシステムをBtrfsにする必要があります。

イメージごとに次のサブボリュームがあります。

```
/home  
/root  
/opt  
/srv  
/usr/local  
/var
```

`/etc`ディレクトリはoverlayFSとしてマウントされ、そこでは上位ディレクトリが`/var/lib/overlay/1/etc/`にマウントされます。

デフォルトでマウントされているサブボリュームは、`/etc/fstab`の`x-initrd.mount`オプションで認識できます。その他のサブボリュームまたはパーティションは、IgnitionまたはCombustionのいずれかによって設定する必要があります。

## 1.2 最初のブートの検出

最初のブート時にのみ展開が設定されます。最初のブートとそれ以降のブートを区別するため、最初のブートの終了後にフラグファイル `/boot/writable/firstboot_happened` が作成されます。このファイルがファイルシステムに存在しないと、属性 `ignition.firstboot` がカーネルコマンドラインに渡され、それによって Ignition と Combustion の両方がトリガされて (initrd で) 実行されます。最初のブートの完了後、`/boot/writable/firstboot_happened` フラグファイルが作成されます。



### 注記: 必ず作成されるフラグファイル

設定ファイルが不適切であるか欠落しているために適切な設定にならない場合でも、`/boot/writable/firstboot_happened` フラグファイルが作成されます。

### 1.2.1 以降のブート時にシステムの再設定を強制する

最初のブートが発生した後で、システムを再設定する必要がある場合は、それ以降のブート時に再設定を強制することができます。ここでは2つのオプションがあります。

- `ignition.firstboot=1` 属性をカーネルコマンドラインに渡すことができます。
- フラグファイル `/boot/writable/firstboot_happened` を削除できます。

## 2 SLE Microの展開

SLE Microの展開手順は、3つのフェーズで構成されます。どのフェーズもスキップせず、次に示す順序に従ってください。

1. DASDディスクの有効化 - 展開を開始する前に、使用するすべてのディスクが利用可能でアクティブであることを確認する必要があります。詳細については、3項「マシンの準備」を参照してください。
2. 設定デバイスの準備 - ネットワークの設定、ユーザの追加、システムの登録を行うには、4項「設定ディスクの準備」に記載されている手順に従って設定を準備します。
3. SLE Microイメージのダウンロードとディスクへの展開。詳細については、5項「DASDディスク上への生ディスクイメージの展開」を参照してください。

## 3 マシンの準備

SLE Microの展開を開始する前に、必要なDASDディスク3つすべてがアクティブかどうかを確認する必要があります。ディスクの状態をチェックするには、次のコマンドを実行します。

```
# lsdasd
```

Bus-ID	Status	Name	Device	Type	BlkSz	Size	Blocks
0.0.0100	active	dasda	94:0	ECKD	4096	30720MB	7864380
0.0.0101	active	dasdc	94:8	ECKD	4096	20480MB	5243040
0.0.0102	active	dasdb	94:4	ECKD	4096	5120MB	1310760

5GBディスクは設定デバイスとして使用され、20GBディスクにはSLE Microが展開されます。コマンド出力にディスクが表示されない場合は、ディスクを有効にする必要があります。次の手順に従います。

1. **lszdev dasd**を実行してデバイスの全リストを表示します。

```
# lszdev dasd
```

TYPE	ID	ON	PERS	NAMES
dasd-eckd	0.0.0100	yes	no	dasda
dasd-eckd	0.0.0101	no	no	
dasd-eckd	0.0.0102	no	no	
dasd-eckd	0.0.0190	no	no	
dasd-eckd	0.0.019d	no	no	
dasd-eckd	0.0.019e	no	no	
dasd-eckd	0.0.0592	no	no	

2. 次のコマンドを使用してディスクを有効にします。

```
# chzdev -eDISK_ID
```

たとえば、ディスク0.0.0101を有効にするには、次のコマンドを実行します。

```
# chzdev -e 0.0.0101
```

3. **lsdasd**コマンドの出力でディスクがアクティブであることを確認します。

## 4 設定ディスクの準備



### ヒント: 仮想ディスクと物理ディスク

運用環境の展開には仮想ディスクが適しており、開発には物理ディスクが適していません。

インストールプロセス中に、ユーザやディレクトリを定義したり、SSHキーを提供したりするための複雑な設定を渡すことができます。そのためには、完全な設定を保存する設定デバイスを作成します。次の例で、このようなデバイスを作成する方法について説明します。

#### 1. 5GBディスクを特定します。

```
# lsdasd
```

Bus-ID	Status	Name	Device	Type	BlkSz	Size	Blocks
0.0.0100	active	dasda	94:0	ECKD	4096	30720MB	7864380
0.0.0101	active	dasdc	94:8	ECKD	4096	20480MB	5243040
0.0.0102	active	dasdb	94:4	ECKD	4096	5120MB	1310760

上記の出力では、設定デバイスとして使用するディスクのBus-IDの値は**0.0.0102**です。デバイス名(/dev/dasdX)は再起動のたびに変わる可能性があるため、ディスクの特定にはBus-IDの値またはディスクサイズを使用します。

#### 2. ディスクをフォーマットします。フォーマットすると、そのディスク上のデータはすべて削除されることに注意してください。

```
# dasdfmt -b 4096 -y -p /dev/dasdb
```

#### 3. パーティションを作成します。

```
# parted /dev/dasdb mkpart ext2 0% 100%
```

#### 4. パーティションをフォーマットします。

```
# mkfs.ext4 /dev/dasdb1
```

#### 5. パーティションにignitionというラベルを付けます。

```
# e2label /dev/dasdb1 ignition
```

#### 6. ファイルシステムを確認します。

```
# blkid
```

- パーティションをマウントします。

```
# mount /dev/dasdbl /mnt
```

- IgnitionまたはCombustion、あるいはその両方のディレクトリ構造を準備します。Ignitionの場合、次のコマンドを実行します。

```
# mkdir -p /mnt/ignition
```

Combustionの場合、次のコマンドを実行します。

```
# mkdir -p /mnt/combustion
```

- SSH公開鍵を `/mnt/combustion/ssh_key.pub` としてコピーします。
- 説明に従ってさらに設定を準備します。

## 4.1 Combustionを使用したSLE Micro展開の設定

Combustionは、最初のブート時にシステムを設定できるdracutモジュールです。Combustionを使用すると、デフォルトパーティションの変更、ユーザのパスワードの設定、ファイルの作成、パッケージのインストールなどができます。

### 4.1.1 Combustionの動作

`ignition.firstboot`引数をカーネルのコマンドラインに渡した後でCombustionが呼び出されます。Combustionは、`script`という名前の付属のファイルを読み込み、そこに含まれるコマンドを実行して、ファイルシステムに変更を加えます。`script`にネットワークフラグを記述していると、Combustionによってネットワークの設定が試みられます。`/sysroot`がマウントされた後、Combustionは`/etc/fstab`にあるマウントポイントをすべて有効にしようと試み、その後`transactional-update`を呼び出して、`root`パスワードの設定やパッケージのインストールなどの他の変更を適用します。

設定ファイル`script`は、`combustion`をラベルとする設定メディアの`combustion`サブディレクトリに置く必要があります。ディレクトリ構造は次のようになる必要があります。

```
<root directory>
└─ combustion
```

```
└─ script
└─ other files
```



## ヒント: IgnitionとCombustionの併用

Combustionは、Ignitionとともに使用することができます。その場合は、設定メディアに`ignition`というラベルを付け、`ignition`ディレクトリと`config.ign`を次に示すようなディレクトリ構造に含めます。

```
<root directory>
└─ combustion
  └─ script
  └─ other files
└─ ignition
  └─ config.ign
```

このシナリオでは、IgnitionはCombustionの前に実行されます。

## 4.1.2 Combustionの設定例

### 4.1.2.1 script設定ファイル

`script`設定ファイルは、Combustionによって`transactional-update`シェルで解析および実行される一連のコマンドです。この記事では、Combustionで実行される設定タスクの例を示します。



### 重要: インタープリタ宣言の指定

`script`ファイルはシェルで解釈されることから、その先頭行には必ずインタープリタ宣言を記述します。たとえば、Bashの場合は次のように記述します。

```
#!/bin/bash
```

実際のシステムにログインするには、少なくとも`root`パスワードを指定します。ただし、SSHキーを使用した認証を確立することをお勧めします。`root`パスワードを使用する必要がある場合は、セキュアなパスワードを設定してください。ランダムに生成されたパスワードを使用する場合は、少なくとも10文字を使用してください。パスワードを手動で作成する場合は、11文字以上を使用し、大文字、小文字、および数字を組み合わせてください。

#### 4.1.2.1.1 ネットワーク設定

最初のブート時にネットワーク接続を設定および使用するには、次のステートメントを `script` に追加します。

```
# combustion: network
```

このステートメントを使用すると、`rd.neednet=1` 引数が `dracut` に渡されます。ネットワーク設定はデフォルトで DHCP を使用します。別のネットワーク設定が必要な場合は、[4.1.2.1.2項「initramfsでの変更の実行」](#)の説明に従って進めます。

このステートメントを使用しない場合、システムはネットワーク接続がない設定のままになります。

#### 4.1.2.1.2 initramfsでの変更の実行

`initramfs` 環境の変更が必要になる場合があります。たとえば、NetworkManager用のカスタムネットワーク設定を `/etc/NetworkManager/system-connections/` に書き込む場合などです。そのためには、`prepare` ステートメントを使用します。

たとえば、静的IPアドレスで接続を作成し、DNSを設定する場合は、次のようにします。

```
#!/bin/bash
# combustion: network prepare
set -euxo pipefail

nm_config() {
  umask 077 # Required for NM config
  mkdir -p /etc/NetworkManager/system-connections/
  cat >/etc/NetworkManager/system-connections/static.nmconnection <<-EOF
  [connection]
  id=static
  type=ethernet
  autoconnect=true

  [ipv4]
  method=manual
  dns=192.168.100.1
  address1=192.168.100.42/24,192.168.100.1
EOF
}

if [ "${1-}" = "--prepare" ]; then
  nm_config # Configure NM in the initrd
  exit 0
fi
```

```
# Redirect output to the console
exec > >(exec tee -a /dev/tty0) 2>&1

nm_config # Configure NM in the system
curl example.com
# Leave a marker
echo "Configured with combustion" > /etc/issue.d/combustion
```

#### 4.1.2.1.3 パーティショニング

SLE Micro生イメージは、1.1項「[デフォルトパーティショニング](#)」で説明されるように、デフォルトのパーティショニングスキームで提供されます。異なるパーティショニングを使用することもできます。次の一連のスニペット例では、[/home](#)を別のパーティションに移動します。



### 注記: スナップショットに含まれるディレクトリの外部で変更を実行する

次のスクリプトは、スナップショットに含まれていない変更を実行します。このスクリプトが失敗してスナップショットが破棄された場合、[/dev/vdb](#)デバイスへの変更など、特定の変更は表示されたままになり、元に戻すことはできません。

次のスニペットは、[/dev/vdb](#)デバイス上にパーティションが1つのみのGPTパーティショニングスキームを作成します。

```
sfdisk /dev/vdb <<EOF
label: gpt
type=linux
EOF

partition=/dev/vdb1
```

パーティションはBtrfsにフォーマットされます。

```
wipefs --all ${partition}
mkfs.btrfs ${partition}
```

次のスニペットによって、[/home](#)の移動可能なコンテンツが新しい[/home](#)フォルダの場所に移動されます。

```
mount /home
mount ${partition} /mnt
rsync -aAXP /home/ /mnt/
umount /home /mnt
```

次のスニペットは、`/etc/fstab`の古いエントリを削除して新しいエントリを作成します。

```
awk -i inplace '$2 != "/home"' /etc/fstab
echo "$(blkid -o export ${partition} | grep ^UUID=) /home btrfs defaults 0 0" >>/etc/fstab
```

#### 4.1.2.1.4 新しいユーザの作成

Cockpitなどの一部のサービスでは、`root`以外のユーザを使用してログインする必要があるため、ここで非特権ユーザを少なくとも1人定義します。または、6.2項「[ユーザの追加](#)」の説明に従って、実行中のシステムからこのようなユーザを作成できます。

新しいユーザアカウントを追加するには、まず、ユーザのパスワードを表すハッシュ文字列を作成します。`openssl passwd -6`コマンドを使用します。

パスワードハッシュを取得したら、次の行を`script`に追加します。

```
mount /home
useradd -m EXAMPLE_USER
echo 'EXAMPLE_USER:PASSWORD_HASH' | chpasswd -e
```

#### 4.1.2.1.5 `root`のパスワードの設定

`root`パスワードを設定する前に、`openssl passwd -6`などを使用して、そのパスワードのハッシュを生成します。パスワードを設定するには、`script`に次の行を追加します。

```
echo 'root:PASSWORD_HASH' | chpasswd -e
```

#### 4.1.2.1.6 SSHキーの追加

次のスニペットは、`root`のSSHキーを保存するディレクトリを作成してから、設定デバイス上にあるSSH公開鍵を`authorized_keys`ファイルにコピーします。

```
mkdir -pm700 /root/.ssh/
cat id_rsa_new.pub >> /root/.ssh/authorized_keys
```



### 注記

SSHを介したりモートログインを使用する必要がある場合は、SSHサービスを有効にする必要があります。詳細については、4.1.2.1.7項「[サービスの有効化](#)」を参照してください。

#### 4.1.2.1.7 サービスの有効化

SSHサービスなどのシステムサービスを有効にするには、`script`に次の行を追加します。

```
systemctl enable sshd.service
```

#### 4.1.2.1.8 パッケージのインストール

### ！ 重要: ネットワーク接続とシステムの登録が必要な場合がある

特定のパッケージでは追加のサブスクリプションが必要になる場合があるので、システムの事前登録が必要なことがあります。また、追加のパッケージをインストールするには、利用可能なネットワーク接続が必要になる場合もあります。

最初のブート設定時に、システムに追加パッケージをインストールできます。たとえば、以下を追加して、`vim`エディタをインストールできます。

```
zypper --non-interactive install vim-small
```

### 📝 注記

設定が完了し、設定したシステムでブートすると、`zypper`を使用できなくなることに注意してください。後で変更を実行するには、`transactional-update`コマンドを使用して、変更されたスナップショットを作成する必要があります。

#### 4.1.2.2 `script`ファイルの完全な例

次の`script`は、Combustionの独自の設定を記述する方法に関するガイドとして役立つ完全な設定を提供します。この例では、Ignitionの設定はこれ以上必要ありません。

```
#!/bin/bash
# combustion: network prepare

set -euxo pipefail

## The OSA subchannels to enable
ZNET_SUBCHANNELS=0.0.1000,0.0.1001,0.0.1002

## Network information to configure
```

```

IPADDRESS="10.144.64.155/24" ## Formet is ipaddress/cidr
GATEWAY="10.144.64.254"
NAMESERVERS="10.144.53.53;10.144.53.54" ## A semicolon-separated list of name servers

## Hostname information
NODE_HOSTNAME="micro6"

## Add password for root user
## Use either 'openssl passwd -6' or 'mkpasswd --method=sha-512' to encrypt the
password.
ROOT_USER_PASSWORD='PASSWORD_HASH'
SSH_ROOT_PUBLIC_KEY=ssh_key.pub

## Add a regular user, because root login may be disallowed in some services.
CREATE_NORMAL_USER=user ## Replace the "user" with a desired username here.
NORMAL_USER_PASSWORD='PASSWORD_HASH'
SSH_USER_PUBLIC_KEY=ssh_key.pub

## Register to SUSE Customer Center and install additional packages
REG_EMAIL='tux@suse.com' ## Email address for product registration
SLMICRO_REGCODE='REGISTRATIONCODE' ## A registration code required to install
additional packages
ADDITIONAL_PACKAGES='' ## A space separated list of additional packages to install

nm_config() {
    umask 077 # Required for Network Manager configuration
    mkdir -p /etc/NetworkManager/system-connections/
    cat >/etc/NetworkManager/system-connections/Wired connection 1.nmconnection' <<EOF
    [connection]
    id=static
    type=ethernet
    autoconnect=true

    [ipv4]
    method=manual
    address1=$IPADDRESS
    gateway=$GATEWAY
    dns=$NAMESERVERS
EOF
}

if [ "${1-}" = "--prepare" ]; then
    # Configure Network Manager in the initrd
    nm_config
    # Enable OSA network devices
    chzdev qeth $ZNET_SUBCHANNELS -ep
    chzdev qeth $ZNET_SUBCHANNELS -e
    exit 0
fi

```

```

## Post output on stdout
exec > >(exec tee -a /dev/ttyS0) 2>&1

## Set hostname
echo $NODE_HOSTNAME > /etc/hostname

## Set root password
echo root:$ROOT_USER_PASSWORD | chpasswd -e
## Add ssh public key as authorized key for the root user
mkdir -pm700 /root/.ssh/
cat $SSH_ROOT_PUBLIC_KEY >> /root/.ssh/authorized_keys

## Mount /var and /home so user can be created smoothly
if [ "$CREATE_NORMAL_USER" ]
then
    mount /var && mount /home
fi
## User creation
if [ "$CREATE_NORMAL_USER" ]
then
    echo "User creation is requested, creating user."
    useradd -m $CREATE_NORMAL_USER -s /bin/bash -g users
    echo $CREATE_NORMAL_USER:$NORMAL_USER_PASSWORD | chpasswd -e
    echo $CREATE_NORMAL_USER "ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers.d/adminusers
    mkdir -pm700 /home/$CREATE_NORMAL_USER/.ssh/
    chown -R $CREATE_NORMAL_USER:users /home/$CREATE_NORMAL_USER/.ssh/
    cat $SSH_USER_PUBLIC_KEY >> /home/$CREATE_NORMAL_USER/.ssh/authorized_keys
    echo "Requested user has been created, requested password has been set."
else
    echo "No user will be created"
fi

# Configure NM in the system
nm_config
# Enable OSA network device
chzdev qeth $ZNET_SUBCHANNELS -ep
chzdev qeth $ZNET_SUBCHANNELS -e

## Enable services
echo "Enabling services."
systemctl enable cockpit.socket
systemctl enable sshd

## Unmount var and home
if [ "$CREATE_NORMAL_USER" ]
then
    umount /var && umount /home
fi

```

```
echo "Configured with Combustion at $(date)" > /etc/issue.d/combustion
```

## 4.2 Ignitionを使用したSLE Micro展開の設定

Ignition (<https://coreos.github.io/ignition/>) は、最初のブート時に仕様に従ってシステムを設定できるプロビジョニングツールです。

### 4.2.1 Ignitionの動作

システムを初めてブートすると、Ignitionが`initramfs`の一部としてロードされ、設定ファイルを特定のディレクトリ(USBフラッシュディスク上、またはURLを指定可能)内で検索します。カーネルによってファイルシステムが一時ファイルシステムから実際のルートファイルシステムに切り替わる前(`switch_root`コマンドが発行される前)に、すべての変更処理が実行されます。

Ignitionでは、JSON形式の設定ファイル`config.ign`が使用されます。その設定は手動で記述できるほか、<https://ignite.opensuse.org>にあるFuel Ignition Webアプリケーションを使用して生成することもできます。

#### ! 重要

Fuel IgnitionはIgnitionの語彙をまだ完全には網羅していません。そのため、結果として生成されるJSONファイルをさらに手動で微調整しなければならない場合があります。

#### 4.2.1.1 config.ign


設定メディアの`ignition`サブディレクトリに設定ファイル`config.ign`を置く必要があります。このような設定メディアとして、たとえば`ignition`をラベルとするUSBメモリが考えられます。ディレクトリ構造は次のようになります。

```
<root directory>
├── ignition
│   └── config.ign
```



## ヒント

Ignition設定でディスクイメージを作成するために、<https://ignite.opensuse.org>にあるFuel Ignition Webアプリケーションを使用できます。

`config.ign`には、複数のデータタイプ(オブジェクト、文字列、整数、ブール、オブジェクトのリスト)が含まれています。仕様の全詳細については、[Ignition specification v3.3.0 \(https://coreos.github.io/ignition/configuration-v3\\_3/\)](https://coreos.github.io/ignition/configuration-v3_3/) を参照してください。

`version`属性は必須であり、SLE Microの場合、その値は3.3.0またはそれ以下のバージョンに設定する必要があります。そのように設定しないとIgnitionの実行に失敗します。

`root`としてシステムにログインするには、少なくとも`root`のパスワードを記述する必要があります。ただし、SSHキーを使用してアクセスを確立することをお勧めします。パスワードの設定では、必ずセキュアなパスワードを使用します。ランダムに生成されたパスワードを使用する場合は、少なくとも10文字を使用してください。パスワードを手動で作成する場合は、11文字以上を使用し、大文字、小文字、および数字を組み合わせてください。

## 4.2.2 Ignitionの設定例

### 4.2.2.1 設定例

このセクションでは、組み込みのJSON形式でのIgnition設定の例をいくつか示します。



### 重要

1.1項「[デフォルトパーティショニング](#)」では、事前構築されたイメージを実行するときに、デフォルトでマウントされるサブボリュームが列挙されます。デフォルトではマウントされないサブボリューム上で新しいユーザを追加する場合やファイルを変更する場合は、まずそのサブボリュームを宣言して、それもマウントされるようにする必要があります。ファイルシステムのマウントに関する詳細については[4.2.2.1.1.3項「`filesystems`属性](#)」を参照してください。



### 注記: `version`属性は必須

各`config.fcc`はバージョン1.4.0以下を含む必要があります。バージョンは対応するIgnition仕様に変換されます。

#### 4.2.2.1.1 ストレージの設定

`storage`属性は、パーティションの設定、RAID、ファイルシステムの定義、ファイルの作成などに使用されます。パーティションを定義するには、`disks`属性を使用します。`filesystems`属性は、パーティションをフォーマットし、特定のパーティションのマウントポイントを定義するために使用されます。`files`属性は、ファイルシステムのファイルの作成に使用できます。先に述べた属性のそれぞれについて、次のセクションで説明します。

##### 4.2.2.1.1.1 `disks`属性

`disks`属性は、これらのデバイス上でパーティションを定義できるデバイスのリストです。`disks`属性には、少なくとも1つの`device`属性が含まれている必要があり、その他の属性はオプションです。次の例では、単一の仮想デバイスを使用し、ディスクを4つのパーティションに分割します。

```
{
  "ignition": {
    "version": "3.0.0"
  },
  "storage": {
    "disks": [
      {
        "device": "/dev/vda",
        "partitions": [
          {
            "label": "root",
            "number": 1,
            "typeGuid": "4F68BCE3-E8CD-4DB1-96E7-FBCAF984B709"
          },
          {
            "label": "boot",
            "number": 2,
            "typeGuid": "BC13C2FF-59E6-4262-A352-B275FD6F7172"
          },
          {
            "label": "swap",
            "number": 3,
            "typeGuid": "0657FD6D-A4AB-43C4-84E5-0933C84B4F4F"
          },
          {
            "label": "home",
            "number": 4,
            "typeGuid": "933AC7E1-2EB4-4F13-B844-0E14E2AEF915"
          }
        ]
      },
    ],
    "wipeTable": true
  }
}
```

```
}
]
}
}
```

#### 4.2.2.1.1.2 **raid属性**

raidは、RAIDアレイのリストです。raidの次の属性は必須です:

##### **level**

特定のRAIDアレイのレベル(linear、raid0、raid1、raid2、raid3、raid4、raid5、raid6)

##### **devices**

絶対パスで参照されるアレイ内のデバイスのリスト

##### **name**

mdデバイスに使用される名前

次に例を示します。

```
{
  "ignition": {
    "version": "3.0.0"
  },
  "storage": {
    "raid": [
      {
        "devices": [
          "/dev/sda",
          "/dev/sdb"
        ],
        "level": "raid1",
        "name": "system"
      }
    ]
  }
}
```

#### 4.2.2.1.1.3 **filesystems属性**

filesystemsには次の属性が含まれている必要があります:

##### **device**

デバイスへの絶対パス。通常、物理ディスクの場合は/dev/sda

## format

ファイルシステム形式(Btrfs、Ext4、xfs、vfat、またはswap)



### 注記

SLE Microの場合、rootファイルシステムはBtrfsにフォーマットする必要があります。

次の例は、filesystems属性の使用方法を示しています。/optディレクトリは、/dev/sda1パーティションにマウントされ、Btrfsにフォーマットされます。デバイスは消去されません。

次に例を示します。

```
{
  "ignition": {
    "version": "3.0.0"
  },
  "storage": {
    "filesystems": [
      {
        "device": "/dev/sda1",
        "format": "btrfs",
        "path": "/opt",
        "wipeFilesystem": false
      }
    ]
  }
}
```

普通、通常のユーザのホームディレクトリは/home/USER\_NAMEディレクトリにあります。デフォルトでは/homeがinitrdにマウントされないため、ユーザを正しく作成するには、このマウントを明示的に定義する必要があります。

```
{
  "ignition": {
    "version": "3.1.0"
  },
  "passwd": {
    "users": [
      {
        "name": "root",
        "passwordHash": "PASSWORD_HASH",
        "sshAuthorizedKeys": [
          "ssh-rsa SSH_KEY_HASH"
        ]
      }
    ]
  }
}
```

```

    }
  ]
},
"storage": {
  "filesystems": [
    {
      "device": "/dev/sda3",
      "format": "btrfs",
      "mountOptions": [
        "subvol=@/home"
      ],
      "path": "/home",
      "wipeFilesystem": false
    }
  ]
}
}
}

```

#### 4.2.2.1.1.4 files属性

`files`属性を使用して、マシンに任意のファイルを作成できます。デフォルトのパーティション方式の外部でファイルを作成するには、`filesystems`属性を使用してディレクトリを定義する必要がありますことに注意してください。

次の例では、ホスト名が`files`属性を使用して作成されます。ファイル`/etc/hostname`は`sl-micro1`のホスト名で作成されます。



### 重要

JSONでは、ファイルモードを`420`などの10進数で指定できることに注意してください。

JSON:

```

{
  "ignition": {
    "version": "3.0.0"
  },
  "storage": {
    "files": [
      {
        "overwrite": true,
        "path": "/etc/hostname",
        "contents": {
          "source": "data:,sl-micro1"
        }
      },
    ],
  },
}

```

```
    "mode": 420
  }
]
}
}
```

#### 4.2.2.1.1.5 directories属性

`directories`属性は、ファイルシステムに作成されるディレクトリのリストです。`directories`属性には、少なくとも1つの`path`属性が含まれている必要があります。次に例を示します。

```
{
  "ignition": {
    "version": "3.0.0"
  },
  "storage": {
    "directories": [
      {
        "path": "/home/tux",
        "user": {
          "name": "tux"
        }
      }
    ]
  }
}
```

#### 4.2.2.1.2 ユーザ管理

`passwd`属性は、ユーザを追加するために使用されます。Cockpitなどの一部のサービスでは、`root`以外のユーザを使用してログインする必要があるため、ここで非特権ユーザを少なくとも1人定義します。または、6.2項「[ユーザの追加](#)」の説明に従って、実行中のシステムからこのようなユーザを作成できます。

システムにログインするには、`root`と通常のユーザを作成し、それぞれのパスワードを設定します。たとえば、`openssl`コマンドを使用して、パスワードをハッシュする必要があります。

```
openssl passwd -6
```

このコマンドは、選択したパスワードのハッシュを作成します。このハッシュを`password_hash`属性の値として使用します。

次に例を示します。

```
{
  "ignition": {
    "version": "3.0.0"
  },
  "passwd": {
    "users": [
      {
        "name": "root",
        "passwordHash": "PASSWORD_HASH",
        "sshAuthorizedKeys": [
          "ssh-rsa SSH_KEY_HASH USER@HOST"
        ]
      }
    ]
  }
}
```

`users`属性には、少なくとも1つの`name`属性が含まれている必要があります。`ssh_authorized_keys`は、ユーザのsshキーのリストです。

#### 4.2.2.1.3 `systemd`サービスの有効化

`systemd`属性で指定して、`systemd`サービスを有効にできます。

次に例を示します。

```
{
  "ignition": {
    "version": "3.0.0"
  },
  "systemd": {
    "units": [
      {
        "enabled": true,
        "name": "sshd.service"
      }
    ]
  }
}
```

#### 4.2.2.2 YAML形式ファイルからJSON形式ファイルへの変換

JSONは、構造化データを保存するための汎用的なファイル形式です。Ignitionなどのアプリケーションは、設定の保存と取得にJSONを使用します。JSONの構文は複雑で人間には読みにくいため、YAMLという、よりわかりやすい形式で設定を記述し、それをJSONに変換することができます。

##### 4.2.2.2.1 YAMLファイルからJSON形式への変換

YAMLファイルに記述したIgnition固有の文法をJSON形式に変換するにはbutaneツールを使用します。このツールではYAMLファイルの構文も検証されるので、構造に潜むエラーを把握できます。butaneの最新バージョンを使用する場合は、次のリポジトリを追加します。

```
> sudo zypper ar -f \
  https://download.opensuse.org/repositories/devel:/kubic:/ignition/openSUSE_Tumbleweed/ \
  devel_kubic_ignition
```

openSUSE\_Tumbleweedを、使用しているディストリビューションに応じて次のいずれかに置き換えます。

- 'openSUSE\_Leap\_\${releasever}'
- 15.5

これで、butaneツールをインストールできるようになりました。

```
> sudo zypper ref && zypper in butane
```

そのインストールが完了すると、次のコマンドを実行してbutaneを呼び出すことができます。

```
> butane -p -o config.ign config.fcc
```

- config.fccは、YAML設定ファイルへのパスです。
- config.ignは、出力するJSON設定ファイルへのパスです。
- -pコマンドオプションを使用すると、出力ファイルに改行が追加されるため、読みやすくなります。

## 5 DASDディスク上への生ディスクイメージの展開

SLE Microを20GB DASDディスクに展開するには、次の手順に従います。

1. **wget**または**curl**を使用して生ディスクイメージをダウンロードします。次に例を示します。

```
> curl -L0k0
```

2. イメージを抽出します。

```
> unpack xz -dBUILD_IDENTIFICATION.raw.xz
```

3. 20GBのデバイス名を検索して保存する変数をエクスポートします。

```
# export SMDASD=$(lsdasd -s | grep 20480MB | tr -s [:blank:] | cut -d' ' -f3)
```

4. ディスクをフォーマットします。

- ディスクを初めてフォーマットする場合、次のコマンドを実行します。

```
# dasdfmt -b 4096 -y -p /dev/$SMDASD
```

- ディスクがフォーマット済みである場合、次のコマンドを実行します。

```
# dasdfmt -b 4096 -M quick -y -p /dev/$SMDASD
```

5. 生ディスクイメージをディスクにコピーします(この場合、ディスク名は/dev/dasdcです)。

```
dd if=IMAGE_NAME.raw status=progress of=/dev/dasdc bs=4k
```

6. 実行中のLinuxをシャットダウンします。

```
# init 0
```

7. 2番目のDASDミニディスクをx3270端末で起動して、SLE Microを起動します。

```
# ipl 101
```

## 6 展開後の手順

### 6.1 CLIからのSLE Microの登録

展開プロセス中にCombustion scriptを使用してシステムを登録しなかった場合は、実行中のシステムから登録できます。

SUSE Customer CenterでSLE Microを登録するには、次の手順に従います。

1. **transactional-update register**を次のように実行します。

```
# transactional-update register -rREGISTRATION_CODE -e EMAIL_ADDRESS
```

ローカル登録サーバで登録するには、さらに次のようにサーバへのURLも入力します。

```
# transactional-update register -rREGISTRATION_CODE -e EMAIL_ADDRESS \  
--url "https://suse_register.example.com/"
```

`REGISTRATION_CODE`は、SLE Microのコピーで受け取った登録コードで置き換えます。`EMAIL_ADDRESS`は、各自または各自の組織が登録の管理に使用しているSUSEアカウントに関連付けられたEメールアドレスで置き換えます。

2. システムを再起動して、最新のスナップショットに切り替えます。
3. これで、SLE Microが登録されました。



### 注記: その他の登録オプション

このセクションの説明対象以外の情報については、**SUSEConnect --help**でのインラインドキュメントを参照してください。

## 6.2 ユーザの追加

SLE Microでは、SSH経由でのログインやCockpitへのアクセスに非特権ユーザが必要であるため、該当するアカウントを作成する必要があります。

非特権ユーザをCombustionで定義している場合、この手順はオプションです。

1. 次のように、**useradd**コマンドを実行します。

```
# useradd -m USER_NAME
```

2. そのアカウントのパスワードを設定します。

```
# passwd USER_NAME
```

3. 必要に応じて、ユーザをwheelグループに追加します。

```
# usermod -aG wheel USER_NAME
```

## 7 法的事項

Copyright © 2006–2025 SUSE LLC and contributors. All rights reserved.

この文書は、GNUフリー文書ライセンスのバージョン1.2または(オプションとして)バージョン1.3の条項に従って、複製、頒布、および/または変更が許可されています。ただし、この著作権表示およびライセンスは変更せずに記載すること。ライセンスバージョン1.2のコピーは、「GNUフリー文書ライセンス」セクションに含まれています。

SUSEの商標については、<https://www.suse.com/company/legal/> を参照してください。その他の第三者のすべての商標は、各社の所有に帰属します。商標記号(®、™など)は、SUSEおよび関連会社の商標を示します。アスタリスク(\*)は、第三者の商標を示します。

本書のすべての情報は、細心の注意を払って編集されています。しかし、このことは正確性を完全に保証するものではありません。SUSE LLC、その関係者、著者、翻訳者のいずれも誤りまたはその結果に対して一切責任を負いかねます。

## A GNUフリー文書利用許諾契約書(GFDL)

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. この使用許諾書を一字一句そのままの複製および頒布することは許可されますが、変更は許可されません。

### 0. 序文

この利用許諾契約書の目的は、マニュアル、テキストブック、またはその他の機能的で有用な文書を、自由という意味で「フリー」にすることです。つまり、そのような文書を、変更の有無や商用非商用に関わらず、コピーまたは再配布する実効的な自由をすべての人々に保証することです。第二に、本利用許諾契約書は、作者または発行者が他者によって行われた変更について責任を負わないとともに、その著作物の功績が確保されるように意図されています。

本利用許諾契約書は、「コピーレフト」(著作物を自由に複製および改変できるようにすること)の一種であり、文書の派生著作物は、それ自体が同じ意味においてフリーでなければなりません。フリーソフトウェア向けに考慮されたコピーレフト利用許諾であるGNU一般公衆利用許諾契約書(GPL)を補足するものです。

弊社は、この利用許諾契約書をフリーソフトウェアのマニュアルに使用するために設計しました。それは、フリーソフトウェアにはフリーマニュアルが必要であるためです。つまり、フリープログラムには、そのソフトウェアと同じ自由を提供するマニュアルが付属しなければなりません。ただし、本利用許諾契約書は、ソフトウェアマニュアルに制限されるものでは

ありません。主題であるか否か、または印刷された本として発行されるか否かに関わらず、任意のテキスト著作物に使用することができます。本利用許諾契約書は、その目的が指示または参照に置かれている著作物に主に使用することを推奨します。

## 1. 適用範囲と定義

本利用許諾契約書は、この利用許諾の条項に従って頒布できることを定めた著作権者の通告が記載されている任意のメディアにおけるマニュアルまたは他の著作物に適用されます。そのような通告は、その著作物をここに記載されている条件に従って使用するための世界的な無償の利用許諾を無期限で付与します。次に示す「文書」は、そのような任意のマニュアルまたは著作物を指します。その公衆ユーザはいずれも被許諾者であり、「利用者」と呼ばれます。利用者は、著作権法に従った許可が必要になるような方法で著作物を複製、変更または頒布する場合に、利用許諾を受け入れます。

文書の「変更された版」とは、そのまま複製されるか、変更または別の言語に翻訳された(またはその両方)文書あるいはその一部を含んだ著作物のことです。

「二次セクション」は、文書の発行者または作者と文書の全体的な主題(または関連事項)との関係のみを示す文書の名前付き付録または前付け部分です。総体的な主題に直接関わる内容は含まれていません。(したがって、文書が部分的に数学のテキストブックになっている場合、二次セクションでは数学について説明されない場合があります)。関係には、主題または関連事項との歴史的なつながり、あるいはそれらに関する法的、商的、哲学的、倫理的、政治的位置付けが含まれる場合があります。

「不変セクション」は、文書が本利用許諾契約書の条件の下でリリースされる旨を述べている通告において、そのタイトルが不変セクションのものとして指定されている、ある特定の二次セクションです。セクションが、すでに説明した二次セクションの定義に一致しない場合は、不変として指定することはできません。文書には、不変セクションが含まれない場合があります。文書で不変セクションを特定しない場合、不変セクションは含まれません。

「カバーテキスト」とは、文書が本利用許諾契約書の条件の下でリリースされる旨を述べている通告において、表カバーテキストまたは裏カバーテキストとして列挙されている、ある一定の短い文章のことです。表カバーテキストは、最大で5語、裏カバーテキストは、最大で25語によって構成できます。

文書の「透過的な複製」とは、その仕様が一般の利用者にとって入手可能で、一般的なテキストエディタまたは一般的な描画プログラム(画素で構成される画像用)、あるいは広く使用されている図面エディタ(図面用)で文書を直接改訂するのに適した形式で表される機械可読の複製のことです。テキストフォーマッタへの入力またはテキストフォーマッタへの入力に適したさまざまな形式への変換に適していることも前提になります。読者による以後の変更を阻止または妨げるようにマークアップまたはマークアップのない状態が調整されている、他の点では

透過的なファイル形式で行われた複製は、透過的な複製ではありません。イメージ形式は、相当量のテキストに使用されている場合、透過的ではありません。「透過的」ではない複製は、「不透明」と呼ばれます。

透過的な複製に適した形式として、マークアップのないプレーンなASCII、Texinfo入力形式、LaTeX入力形式、一般に取得可能なDTDを使用するSGMLまたはXML、標準に準拠したHTML、人為的変更用のPostScriptまたはPDFがあります。透過的なイメージ形式には、PNG、XCF、JPGが含まれます。不透明な形式には、独自のワードプロセッサのみで読み取りおよび編集を行える独自の形式、DTDまたは処理(またはその両方)ツールを一般に取得できないSGMLまたはXML、機械生成HTML、出力のみを目的として一部のワードプロセッサによって作成されるPostScriptまたはPDFが含まれます。

「タイトルページ」とは、印刷された本の場合、タイトルページ自体、および本利用許諾契約書でタイトルページに表示することが要求されるマテリアルを読みやすいように保持するために必要な以降のページのことを指します。そのようなタイトルページがない形式の著作物の場合、「タイトルページ」は、本文の開始部分に先行する、著作物のタイトルを最も顕著に表している部分の近くにあるテキストのことを指します。

「XYZという表題の付いた」セクションとは、そのタイトルが正確にXYZになっているか、またはXYZを別の言語に翻訳しているテキストに続いてカッコ付きのXYZが含まれている文書の名前付きサブユニットのことです。(ここで、XYZは、次に示すように、「謝辞」、「献辞」、「推薦」、「履歴」などの特定のセクション名を表します)。文書を変更するときに、そのようなセクションの「タイトルを保存する」とは、この定義に従って「XYZという表題の付いた」セクションが残されることを表します。

文書では、本利用許諾契約書が文書に適用される旨を述べている通告の付近に保証の放棄を含めることができます。保証の放棄条項は、本利用許諾契約書内の参照によって、保証の放棄に関してのみ組み込まれると見なされます。つまり、これらの保証の放棄条項がもつ可能性のある他のいかなる含意も無効であり、本利用許諾契約書の意味にまったく影響を与えません。

## 2. そのままの複製

利用者は、商用か否かを問わず、任意のメディアにおいて文書を複製または頒布することができます。その際に、本利用許諾契約書、著作権表示、および本利用許諾契約書が文書に適用される旨を述べる利用許諾通告をすべての複製で再生し、本利用許諾契約書の条件に他のいかなる条件も追加しないことが前提条件になります。利用者は、技術的手段によって、作成または頒布する複製の読み込みまたはさらなる複製を妨げたり、制御したりすることはできません。ただし、複製と引き換えに対価を受け取ることができます。十分に大量の複製を頒布する場合は、セクション3の条件に従う必要もあります。

すでに述べた同じ条件に従って複製を貸与したり、複製を公開したりすることもできます。

### 3. 大量の複製

発行する文書の印刷した複製(または、通常、印刷したカバーをもつメディアに含まれた複製)が100部を超え、文書の利用許諾通告でカバーテキストを必要とする場合は、すべてのカバーテキスト(表カバーの表カバーテキスト、裏カバーの裏カバーテキスト)を明瞭かつ読みやすく記載したカバーに文書の複製を同封する必要があります。また、両方のカバーでは、これらの複製の発行者として、利用者を読みやすい状態で明確に識別しなければなりません。表カバーには、フルタイトルを記述し、タイトルのすべての語が同等に目立つようにする必要があります。カバーには他のマテリアルを追加することもできます。カバーに限って変更を行った場合の複製は、文書のタイトルが保持されていて、これらの条件を満たしている限り、他の点に関してそのままの複製と見なすことができます。

いずれかのカバーで、必要なテキストが多すぎて、読みやすい状態に収まらない場合は、列挙されている最初の部分(問題なく収まる分)を実際のカバーに記載し、残りの部分を隣接ページに入れます。

文書の不透明な複製を100部以上公開または頒布する場合は、それぞれの不透明な複製とともに機械可読の透過的な複製を含めるか、それぞれの不透明な複製内あるいはその複製とともに、ネットワークの一般利用者が標準的な一般ネットワークプロトコルを使用して、追加マテリアルのない文書の完全な透過的な複製をダウンロードするときにアクセスできるコンピュータネットワークの場所を明記する必要があります。後者のオプションを使用する場合は、不透明な複製の大量頒布を開始するときに十分慎重な手順を取り、この透過的な複製が、その版の不透明な複製を最後に一般頒布した後(直接またはエージェントや小売業者を通じて)少なくとも1年間、指定した場所で継続的にアクセス可能となるように配慮する必要があります。

大量の複製を再頒布する時点よりもかなり前に、文書の作者に連絡して、文書の更新版を提供する機会を与えることが要求されますが、必須ではありません。

### 4. 変更

文書の変更された版を、すでに述べた第2項および第3項の条件に従って複製および頒布することができます。その際は、本利用許諾契約書に確実に従って、変更された版をリリースし、変更された版が文書の役割を担うようにして、その複製を所要する任意の利用者に変更された版の頒布および変更の利用許諾を与えることが前提になります。また、変更された版で次のことを行う必要があります。

- A. タイトルページ(カバーがある場合はカバー上も含める)で、文書、および以前の版の文書(以前の版がある場合は、その旨、文書の履歴セクションに列挙する)と識別されるタイトルを使用します。前の版と同じタイトルは、その版の元の発行者が許可を与えた場合に、使用することができます。
- B. タイトルページ上に、この要件から解放されない限り、変更された版において変更の著者としての責任を担う1人以上の人またはエンティティとともに、文書の筆頭著者を少なくとも5人、作者として列挙します(5人に満たない場合は、その筆頭著者のすべて)。
- C. タイトルページ上に、変更された版の発行者の名前を、発行者として記載します。
- D. 文書のすべての著作権表示を保持します。
- E. 変更に関する適切な著作権表示を、他の著作権表示の隣に追加します。
- F. 著作権表示の直後に、本利用許諾契約書の条項に従って変更された版を利用するための許可を一般利用者に与える利用許諾通告を、次の補遺に示す形式で含めます。
- G. その利用許諾通告に、不変セクションの全リスト、および文書の利用許諾通告で指定されている必須カバーテキストを保持します。
- H. 本利用許諾契約書の変更されていない複製を含めます。
  - I. 「履歴」という表題のセクションを保持して、そのタイトルを保持し、タイトルページに記載されているとおりに、変更された版のタイトル、年度、新しい作者、発行者を少なくとも示す項目を追加します。文書に履歴というセクションがない場合は、そのタイトルページに記載されているとおりに文書のタイトル、年度、作者、発行者を示すセクションを作成し、前の文章に記載されているとおりに変更された版を示す項目を追加します。
  - J. 文書の透過的な複製に一般利用者がアクセスできるように文書で指定されている場合は、そのネットワークの場所、およびその文書の基盤となった前の版に対応して文書で指定されているネットワークの場所を保持します。これらは、「履歴」セクションに配置することができます。文書自体よりも4年以上前に発行された著作物の場合、または参照されているその版の元の発行者が許可を与えている場合は、そのネットワークの場所を省略することができます。
  - K. 「謝辞」または「献辞」という表題のセクションの場合は、そのセクションのタイトルを保持し、セクション内に、それぞれの貢献者謝辞またはその中の献辞(またはその両方)のすべての内容と意味合いを保持します。
  - L. 文書のすべての不変セクションを保持し、そのテキストおよびタイトルを未変更のままにします。セクション番号またはそれと同等の要素は、セクションタイトルの一部と見なされません。

- M. 「推薦」という表題の任意のセクションを削除します。そのようなセクションは、変更された版に含めることはできません。
- N. 既存のセクションのタイトルを変更して、「推薦」という表題にしたり、タイトルが不変セクションと矛盾したりしないようにします。
- O. 保証の放棄を保持します。

変更された版に、二次セクションと見なされ、文書から複製されたマテリアルを含まない新しい前付けセクションまたは付録が含まれる場合は、これらの一部またはすべてを任意に「不変」として指定することができます。これを行うには、変更された版の利用許諾表示内で列挙されている不変セクションにそのタイトルを追加します。これらのタイトルは、他のすべてのセクションタイトルと異なっている必要があります。

「推薦」という表題のセクションを追加することができますが、その際は、変更された版のさまざまな当事者による推薦以外の要素が含まれていないことが前提になります。たとえば、校正者によるコメント、または文が標準的な信頼できる定義として組織によって承認されていることを示すという宣言文などが相当します。

表カバーテキストとしての最大5語の短い文、および裏カバーテキストとしての最大25語の短い文を、変更された版のカバーテキストのリストの終わりに追加できます。表カバーテキストの短い1文および裏カバーテキストの短い1文のみを、1つのエンティティが追加できます(またはエンティティによって行われた調整を通じて)。文書に、利用者または利用者が関わる同じエンティティによって行われた調整を通じて前に追加された同じカバーのカバーテキストがすでに含まれている場合は、別のものを追加することはできませんが、古いものを置き換えることができます。ただし、その古いものを追加した前の発行者から明示的な許可を得る必要があります。

文書の作者および発行者は、本利用許諾契約書により、その名前を得るために使用したり、変更された版の推薦を主張または暗示したりする許可を与えるものではありません。

## 5. 文書の結合

文書は、すでに述べた変更された版に関するセクション4の条件に従って、本利用許諾契約書の下でリリースされた他の文書と結合することができます。その際は、その組み合わせの中に、元の全文書のすべての不変セクションを未変更のまま含めて、そのすべてを結合された著作物の不変セクションとしてその利用許諾表示に列挙し、そのすべての保証の放棄を保持することが前提となります。

結合された著作物には、本利用許諾契約書の複製を1つのみ含める必要があります。複数の同一の不変セクションは、単一の複製で置き換えることができます。同じ名前だが内容の異なる複数の不変セクションがある場合は、そのような各セクションのタイトルを固有なものにし

ます。その際は、その終わりに、カッコ付きで、そのセクションの元の作者または発行者の名前(既知の場合)、あるいは固有の番号を追加します。不変セクションのリスト内のセクションタイトルには、結合された著作物の利用許諾表示の場合と同じ調整を加えます。

組み合わせでは、さまざまな元の文書の「履歴」という表題のセクションを結合して、1つの「履歴」というセクションを構築する必要があります。同じように、「謝辞」という表題のセクション、および「献辞」という表題のセクションも結合します。「推薦」という表題のすべてのセクションを削除する必要があります。

## 6. 文書のコレクション

文書および本利用許諾契約書の下でリリースされた他の文書から成るコレクションを作成して、さまざまな文書に含まれる本利用許諾契約書の個々の複製を、コレクションに含まれる単一の複製で置き換えることができますが、他のすべての点での各文書のそのままの複製に関する本利用許諾契約書の規則に従うことが前提になります。

そのようなコレクションから単一の文書を抽出して、その文書を本利用許諾契約書に従って個々に頒布することができますが、その際は、本利用許諾契約書の複製を抽出した文書に挿入して、その文書のそのままの複製に関するその他のすべての点で、本利用許諾契約書に従う必要があります。

## 7. 独立した著作物の集積

文書またはその派生物を他の個別および独立した文書または著作物とともに、ストレージまたは頒布メディア内またはそのボリューム上に蓄積することを「集積」と呼びます。その場合は、個々の著作物の許可を超えてその蓄積の利用者の法的権限を制限することに、蓄積による著作権を使用しないことが前提になります。文書が集積に含まれる場合、本利用許諾契約書は、それ自体が文書の派生著作物ではない集積内の他の著作物に適用されません。

セクション3のカバーテキスト要件が文書のこれらの複製に適用可能であり、文書が集積全体の半分に満たない場合は、文書のカバーテキストを、集積内の文書のカバー、または文書が電子形式の場合は、電子的な同等のカバーに配置することができます。それ以外の場合は、集積全体の印刷されたカバー上に表示する必要があります。

## 8. 翻訳

翻訳は一種の変更と見なされるため、セクション4の条件に従って文書の翻訳を配布することができます。不変セクションを翻訳で置き換えるには、著作権者からの特別な許可が必要ですが、これらの不変セクションの元のバージョンのほかに、一部またはすべての不変セクションの翻訳を含めることができます。本利用許諾契約書、文書内のすべての利用許諾表示、お

よび保証の放棄の翻訳を含めることができますが、その際は、本利用許諾契約書の元の英語版、およびそれらの利用許諾表示と保証の放棄の元の版も含めることが前提になります。本利用許諾契約書、利用許諾表示、または保証の放棄の翻訳と元の版との間に不一致がある場合は、元の版が優先されます。

文書内のセクションに、「謝辞」、「献辞」、または「履歴」という表題が付いている場合は、そのタイトルを保持する(セクション1)ための要件(セクション4)により、通常は実際のタイトルを変更する必要があります。

## 9. 終了

利用者は、本利用許諾契約書に明示的に記載されている形態を除き、文書を複製、改変、二次利用許諾、および頒布してはなりません。別の方法で文書を複製、改変、二次利用許諾、または頒布しようとするのは無効であり、本利用許諾契約書の下で利用者の権利は自動的に消滅します。ただし、本使用許諾の下で利用者から複製物または権利を受領した関係者は、条項を遵守している限り、権利が消滅することはありません。

## 10. 本利用許諾契約書の将来の改訂

フリーソフトウェア財団は、GNUフリー文書利用許諾契約書(GFDL)の新版または改訂版を随時公表することがあります。そのような新版は、性格的には現行版と似たものになりますが、新たな問題や懸案事項に対応するために細部が異なる可能性があります。<https://www.gnu.org/copyleft/>を参照してください。

本利用許諾契約書の各版には、区別するための版番号が設定されます。文書に、それに適用される本利用許諾契約書の特定の版番号と「後継版」が指定されている場合、利用者は、選択によって、その指定された版の条項またはフリーソフトウェア財団から公開される後継版の条項(ドラフトではない)に従うことになります。文書に、本利用許諾契約書の版番号が指定されていない場合、利用者は、フリーソフトウェア財団からこれまでに公開された任意の版(ドラフトではない)を選択することができます。

## 補遺: 本利用許諾書をご使用の文書に使用する方法

```
Copyright (c) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

不変セクション、表カバーテキスト、および裏カバーテキストがある場合は、「with...Texts」の行を次のように置き換えます。

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

カバーテキストのない不変セクションが含まれている場合、またはこの3つの他の組み合わせの場合は、その2つの代替要素をマージして状況に合わせます。

文書にプログラムコードの重要な例が含まれている場合は、GNU一般公衆利用許諾契約書 (GPL) などの選択したフリーソフトウェアの利用許諾に従って、これらの例を平行してリリースし、フリーソフトウェアでのその利用を許可することを推奨します。