

SDT-MEDIA(1) SUSE Driver Tools

Scott Bahling

12-08-2020

NAME

sdt media - tool for creating kISO and driver kit images

SYNOPSIS

sdt media kiso -updatedir=DIR -base=DIR_OR_ISO [options]

sdt media driverkit -updatedir=DIR [options]

sdt media fixiso ISOFILE

DESCRIPTION

kISO

The kISO is an ISO image containing a copy of a portion of the /boot section of the SUSE Linux Enterprise installation media with updates to the kernel, kernel modules, or other files based on the updates provided by the user.

Driver kits

A driver kit is an add-on product that delivers kernel drivers for a specific target distribution and is the recommended way to deliver drivers to end users. Being a standard SUSE add-on product, the driver kit can be delivered on CD/DVD or as a package repository via network or local file system.

COMMON OPTIONS

-b *BASE_MEDIA*, -base *BASE_MEDIA* Points to the contents of the initial installation media of the SUSE Linux Enterprise Product that the driver kit is designed to work with. Option can point to either a directory

containing the contents of the installation media, or the media ISO image itself.

This option will cause the driver kit to be built as a installation kit.

-u *UPDATEDIR*, --updatedir=*UPDATEDIR* Directory to be scanned for update files to be added to driver kit. Directory will be scanned for kernel packages, KMP packages, YaST2 modules (.ybc), and tar files (.tgz). (Default: `./updates`)

-w *WORKDIR*, --workdir *WORKDIR* Work directory. Created if non-existent and remains after completion. If this option is not specified, temporary directory is created and deleted on exit.

-m *MODSPEC*, --modspec *MODSPEC* Additional module to add to module.config such that it's added to initrd and linuxrc considers it while probing hardware. MODSPEC is of the following format:

`SECTION:MODULE,DESCR,PARAM,PRE_INST,POST_INST,INITRD,AUTO`

If the kernel doesn't contain any new driver which should be loaded for installation compared to the kernel contained in the base ISO, this option isn't necessary.

-M, --merge Merge driver kit with base SUSE Linux Enterprise installation media. A new ISO image is created containing the standard SUSE Linux Enterprise installation repository in addition to the driver kit repository along with the boot loader and installer. The installation process will automatically add the driver kit as an add-on product to be installed in tandem with the base SUSE Linux Enterprise product.

-o *OUTPUTPATH*, --output *OUTPUTPATH* Specifies the path to place the resulting media image. If OUTPUTPATH is a directory, an auto-generated filename will be used. If OUTPUTPATH is not an existing directory, but resides in an existing directory, the path will be interpreted as a full path plus filename. If this option is not provided, the image will be created in the current working directory using an auto-generated filename.

-N, --name Name of driver kit or kISO. Ignored with --addon option.

-V *VOLUME*, --volume *VOLUME* ISO9660 Volume Identification for the ISO image.

-p *PREPARER*, --preparer *PREPARER* ISO9660 Preparer Identification for the ISO image.

-P *PUBLISHER*, --publisher *PUBLISHER* ISO9660 Publisher Identification for ISO image.

-A *APPLICATION*, --application *APPLICATION* ISO9660 Application spec for the ISO image.

--threads *MAX_THREADS* Max number of threads to use for certain processes. Currently passed to the xz command when compressing the initrd. [default: 1]

DRIVERKIT OPTIONS

The following options are only valid for the *driverkit* command

- a *ADD-ON*, --addon *ADD-ON*** Path to add-on product to be used as basis of a driverkit. The path can either point to a directory containing the contents of the add-on product, or an ISO image of the add-on product. The add-on will be scanned same as update directory for kernel or other package updates. When used together with the **-b|--base** option, the add-on product will be copied without modification together with the boot structure from the base media thereby generating a installation kit.
- e, --embed-dk** Embed driver kit repository into initrd. This can make the initial ram disk quite large, and cause long ram disk load times.
- D, --no-dk** Exclude driver kit repository on base of media. Driver Kit will be embedded in initrd.

KISO & INSTALLATION DRIVER KIT OPTIONS

The following options only apply when using the *kiso* command or creating an installation kit using the *-base* option.

- boot-param *BOOTPARAMS*** Specify a parameter to add to the isolinux.cfg/elilo.conf file. Format is "label:param" where label is the label of the section in the config file to add the param to.

Example: Add "install=" param to the section labeled "linux":

```
--boot-param="linux:install=http://installserver/repo"
```

Hidden parameters

Using the format "label::param" (with double colon) will cause the parameter to be hidden from the boot command line.

- no-initrd-mod-updates** Do not update the kernel modules in the initrd. When updating the initrd, the kernel modules and related files will be copied as-is from the base media to the installation initrd.
- uni-media** Generate media to work with both SUSE Linux Enterprise Server and Desktop products.
- E, --no-efi** Do **not** generate efi boot loader. Only applies to x86_64 arch.

kISO ONLY OPTIONS

The following options only apply with the *kiso* command.

- classic** Build DUD in “Classic Mode”. By default the DUD will create register the directory containing packages as a add-on product repository during installation. Using the –classic option will cause the DUD to use the old style bash script for installing the packages.
- freshen-only** With classic DUD type, packages are installed/updated unconditionally (rpm -U). With –freshen-only the packages will be updated only if previously installed (rpm -F). Only useful together with –classic option.

See the sdt-dud.1 for more information on the classic mode and freshen only options.

USAGE

Creating kISO images

To create a kISO kit use the **kiso** sub-command of the sdt tool.

```
sdt media kiso -u updates -b ./distro/sles11sp1-x86_64 -o mykiso.iso
```

The **-b/--base** option points to the distro installation media used as a basis of the installation kit.

The **-u/--updatedir** option specifies a directory where the sdt tool will find the files to be added to the Update Medium. Any KMPs, kernel packages, or other files found in the directory will be added to the kISO boot structure appropriately. If the **--updatedir** option is not specified, the **updates** subdirectory in the current working directory will be used by default.

Creating driver kits

To create a driver kit use the **driverkit** sub-command of the sdt tool.

```
sdt media driverkit -u updates -o mydriverkit.iso
```

The **-u/--updatedir** option specifies a directory where the sdt tool will find the files to be added to the Update Medium. As with the kISO command, the **updates** directory in the current working directory will be used by default if not otherwise specified.

Creating installation kits

To create an installation kit use the **-b/--base** option to point to the distro installation media used as a basis of the installation kit.

```
sdt media driverkit -u updates -b ./distro/sles11sp1-x86_64 -o mydriverkit.iso
```

This will copy the boot structure from the distro installation media to the newly created driver kit. In addition, any KMPs, kernel packages, or other files found in the directory specified by `-u/--updatedir` option will be added to the boot structure appropriately.

Creating a *merged* media

To create an ISO image that *merges* the kISO or driver kit contents with the base SUSE product onto a single media, use the `-M/--merge` option:

The following will generate the kISO as usual and copy its content along with the base SUSE Product repository to the new ISO image. The ISO boot structure will be updated to include any updated kernel, kernel modules or other updates just as with the installation kit.

```
sdt media kiso -M -u updates -b ./distro/sles11sp1-x86_64 -o mykiso.iso
```

The following will generate the driver-kit as usual and copy it along with the base SUSE Product repository to the new ISO image. The ISO boot structure will be updated to include any updated kernel, kernel modules or other updates just as with the installation kit.

```
sdt media driverkit -M -u updates -b ./distro/sles11sp1-x86_64 -o mydriverkit.iso
```

Using existing add-on products

Existing add-on product media can be used when creating an installation kit or merged media. There are two ways to achieve that. The first method uses the `-a/--addon PATH_TO_ADDON` option to point to an existing add-on product ISO image or repository directory structure.

```
sdt media driverkit -a isos/my-addon-1.0.iso -u updates -b ./distro/sles11sp1-x86_64 -o myir
```

The second method is to place a copy of the add-on product ISO image in the updates directory with the filename `addon.iso`. With this method the `-a/--addon` option is not required. The `sdt media` tool will automatically detect files named `addon.iso` and use them to create the resulting media.

Note: when using an existing add-on product, any rpm packages in the `--updatedir` location will be ignored.

FILES DETECTED IN UPDATEDIR

When scanning the `UPDATEDIR` for files to be added to the driver kit, certain files will be automatically detected and specific actions taken. See `sdt-file-detection` (7) for more information on how files are detected.

FIXISO COMMAND

Most versions of mkisofs or genisoimage do not generate proper El Torito boot catalogs resulting in iso images that do not boot alternate boot loaders (i.e. uEFI) properly. The `sdt media fixiso` command will attempt to fix the boot catalog on the ISO image.

Usage: `sdt media fixiso PATH_TO_ISO_IMAGE`

The image will be fixed **in place** overwriting the bits in the ISO image file.

LIST COMMAND

The `sdt media list` command will list the contents of the ISO media. Currently it only outputs the list of rpm packages and their versions. It is useful for quickly validating that a driverkit image contains the proper packages.

SEE ALSO

`sdt(1)`, `sdt-file-detection(7)`