

Contents

1	NAME	2
2	SYNOPSIS	2
3	DESCRIPTION	2
4	OPTIONS	3
5	EXAMPLES	3
6	SEE ALSO	4
7	GENERAL OPENSIX OPTIONS	4

1 NAME

slxos-export - OpenSLX-script to generate an export from a vendor-OS.

2 SYNOPSIS

slxos-export [options] <action>

Options

<code>--help</code>	brief help message
<code>--log-level=<int></code>	level of logging verbosity (0-3)
<code>--man</code>	show full documentation
<code>--version</code>	show version

Actions

export <vendor-OS-name> <export-type>

exports the vendor-OS with the given name using the given export type and adds it to the config-DB, too. The export will be named as the vendor-OS, but with an additional '-<X>' appended to it (where <X> will be replaced by the chosen export-type).

list-exported

list all exported vendor-OSes

list-installed

list all installed vendor-OSes

list-types

list all supported export types

remove <export-name>

removes the export with the given name from disk and config-DB

3 DESCRIPTION

slxos-export converts an installed vendor-OS into a form that can be accessed via network by booting clients.

The resulting form of such a conversion is called an *export* and those come in different flavors:

Export Type 'nfs'

NFS (network file system) is a well established networking file system, which is supported by LINUX since long.

Export Type 'sqfs-nbd'

Squash-FS is a rather modern filesystem providing very good compression, resulting in considerably reduced network traffic during boot (and execution). However, in order to mount a squash-fs that resides on the server, the client has to get access to it first. This can be established via a network block device, which basically "transports" a block device over the network (from server to client), making it possible to use more or less any file system over the network. So, this example translates to 'use a squashfs on a network block device'.

When invoking `slxos-export`, you have to pass it a vendor-OS name and the export type you want to use and it will do the conversion (which can take a while, so please be patient).

The resulting export will be stored under `/srv/openslx/export`.

4 OPTIONS

`-help`

Prints a brief help message and exits.

`-man`

Prints the manual page and exits.

`-version`

Prints the version and exits.

5 EXAMPLES

Exporting a Vendor-OS via NFS

```
slxos-export export suse-10.2 nfs
```

Exports the installed vendor-OS `suse-10.2` via `nfs`, the resulting NFS-export will live in `/srv/openslx/export/nfs/suse-10.2`.

Exporting a Vendor-OS via NBD

```
slxos-export export ubuntu-6.10 sqfs-nbd
```

Exports the installed vendor-OS `ubuntu-6.10` via `nbd`, the resulting Squash-FS will live in `/srv/openslx/export/nbd/ubuntu-6.10`.

Removing an Export

slxos-export remove ubuntu-6.10 nbd

Wipes the squash-FS of the export named 'ubuntu-6.10' from disk (i.e. the file /srv/openslx/export/nbd/ubuntu-6.10 will be deleted) and removes that export from the config-DB, too.

6 SEE ALSO

slxsettings, slxos-setup, slxconfig, slxconfig-demuxer

7 GENERAL OPENSXLX OPTIONS

Being a part of OpenSLX, this script supports several other options which can be used to overrule the OpenSLX settings:

<code>--db-name=<string></code>	name of database
<code>--db-spec=<string></code>	full DBI-specification of database
<code>--db-type=<string></code>	type of database to connect to
<code>--locale=<string></code>	locale to use for translations
<code>--log-level=<int></code>	level of logging verbosity (0-3)
<code>--logfile=<string></code>	file to write logging output to
<code>--private-path=<string></code>	path to private data
<code>--public-path=<string></code>	path to public (client-accessible) data
<code>--temp-path=<string></code>	path to temporary data

Please refer to the `slxsettings`-manpage for a more detailed description of these options.