LSDK 19.09 Documentation Web Page

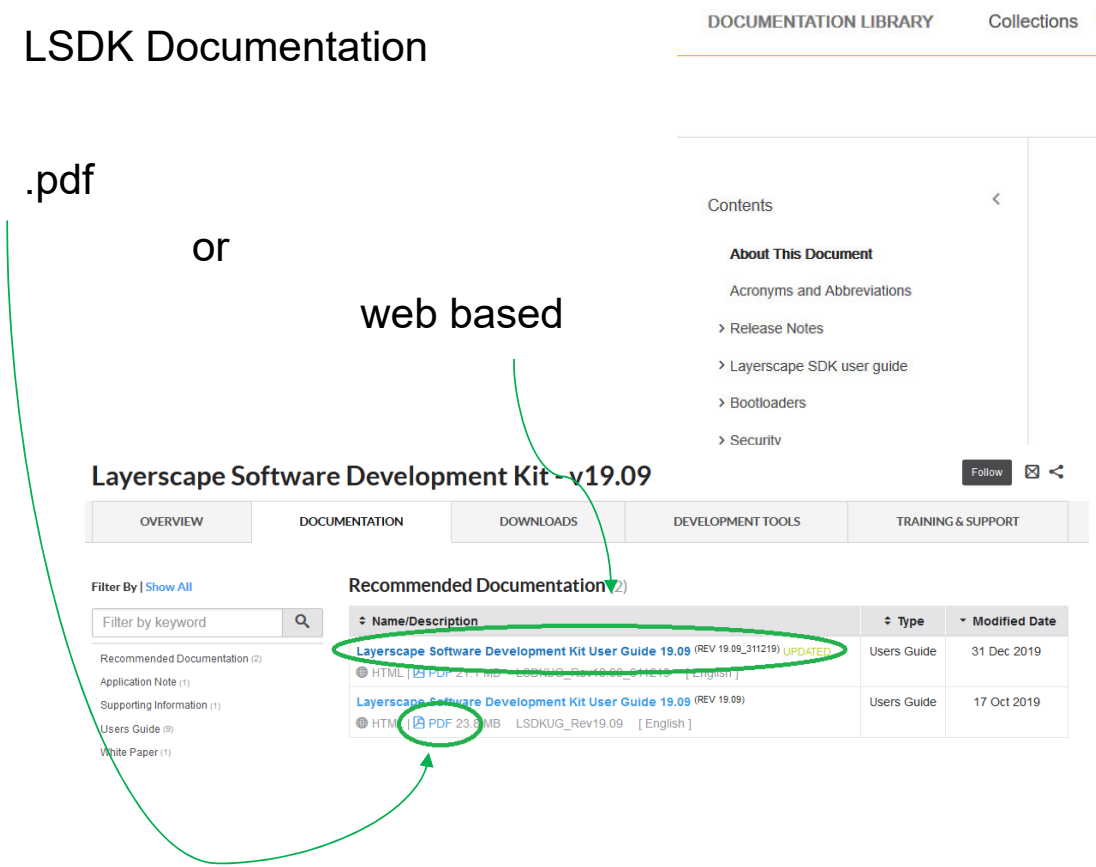
https://www.nxp.com/design/documentation:DOCUMENTATION#/

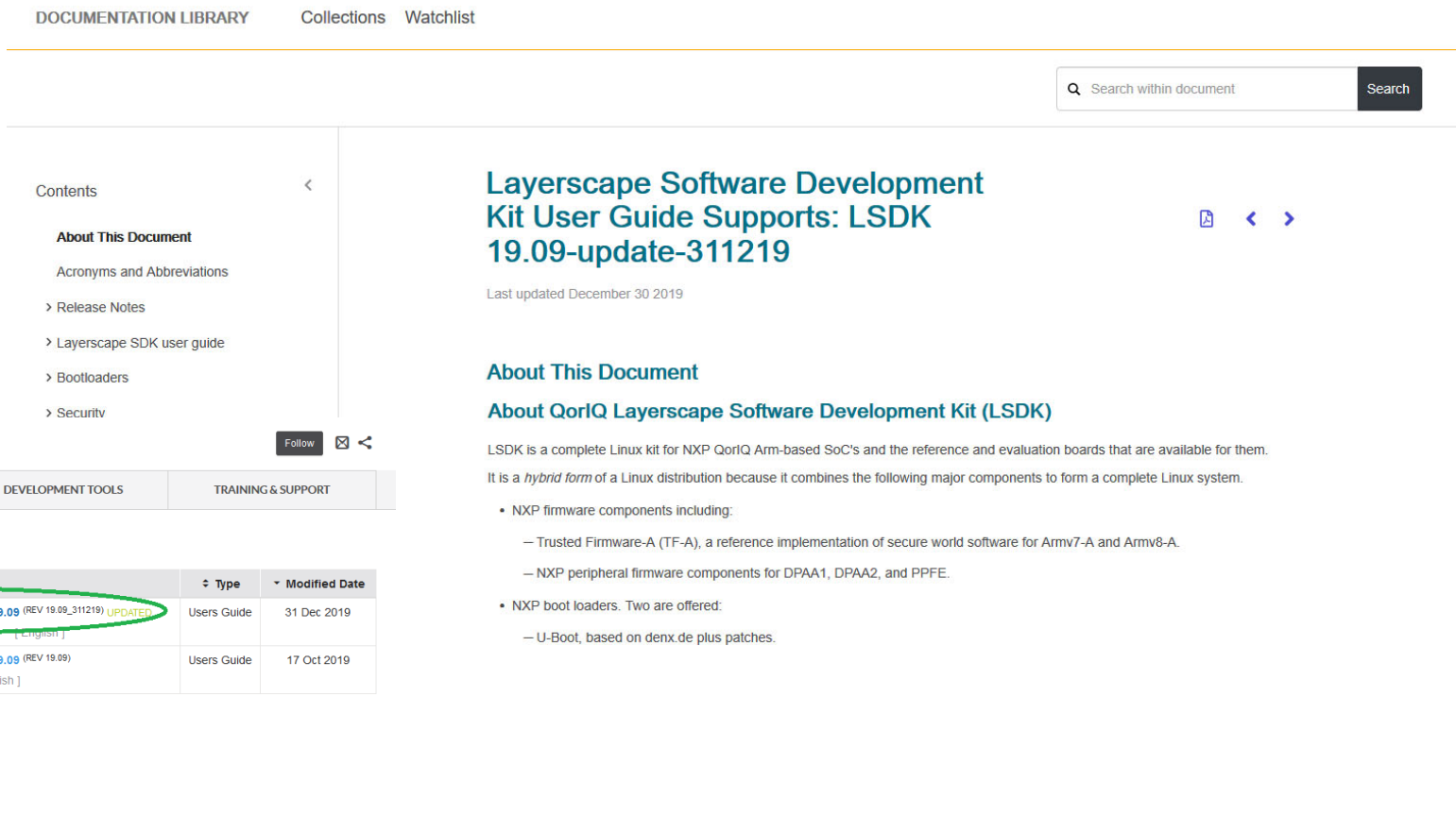
LSDK Documentation

.pdf

or

web based





LSDK 19.09 Generation

https://www.nxp.com/design/software/embedded-software/linux-software-and-development-tools/layerscape-software-development-kit-v19.09:LAYERSCAPE-SDK

```
$ tar xvzf flexbuild_lsdk1909_update_221019.tgz
$ cd flexbuild_lsdk1909_update_221019
$ source setup.env
$ flex-builder -h
```

Two main commands

flex-builder

commands to make and build the images to be downloaded on LS evaluation board.

flex-installer

to install generated or downloaded Linux on the SD or on the board itself.

flex-builder

```
Usage: flex-builder [-c <component>]   [-m <machine>] [-a <arch>] [-b <boottype>]
   or: flex-builder [-i <instruction>] [-m <machine>] [-a <arch>] [-b <boottype>]

Most used example with autobuild:
 flex-builder -m ls1046ardb -a arm64     # auto build all firmware, linux, apps components and LSDK userland for ls1046ardb
 flex-builder -i auto -a arm64           # auto build all firmware, linux, apps components and LSDK userland for all arm64 machines

Most used example with separate command:
 flex-builder -i mkrfs                # generate Ubuntu main arm64 userland by default
 flex-builder -i mkrfs -r ubuntu:mate       # generate Ubuntu-Mate arm64 GUI desktop userland
 flex-builder -i mkrfs -r yocto:tiny        # generate Yocto-base arm64 tiny userland
 flex-builder -i mkrfs -r centos            # generat CentOS arm64 userland
 flex-builder -i mklinux -a arm64           # generate lsdk_linux_arm64_LS_tiny.itb including rootfs_yocto_arm64_tiny.cpio.gz
 flex-builder -c linux -a arm64             # build linux component with default linux repo and default branch/tag for all arm64 machines
 flex-builder -c atf -m ls1046ardb -b sd    # build ATF images for SD boot on LS1046ardb
 flex-builder -i mkfw -m ls1046ardb -b sd   # generate composite firmware for SD boot on ls1046ardb
 flex-builder -i mkbootpartition -a arm64   # generate boot partition tarball appliable for a variety of userland used on arm64 platforms
 flex-builder -c apps -a arm64              # buld all apps components (dpdk, fmc, restool, optee_os, secure_obj, edgescale, etc) for arm64
 flex-builder -i merge-component -a arm64   # merge all components packages and kernel modules into target userland
 flex-builder -i packrfs -a arm64    # pack and compress target userland as .tgz
 flex-builder -i download -m ls1043ardb     # download prebuilt userland
 flex-builder -i list    # show the list of enabled config, machines and components
 flex-builder -i repo-fetch    # fetch all git repositories of components from remote repos if not exist locally
 flex-builder -i repo-update    # update all components to the latest TOP commmits of current branches
 flex-builder docker    # create or attach to Ubuntu docker container to run Flexbuild in docker
 flex-builder clean    # clean all previously generated images except distro rootfs, optionally

Most used options:
 -m, --machine       target machine, supports ls1012afrwy,ls1021atwr,ls1028ardb,ls1043ardb,ls1046ardb,ls1088ardb_pb,ls2088ardb,lx2160ardb, etc
 -a, --arch       target arch of processor, valid argument: arm64, arm64:be, arm32, arm32:be, ppc64, ppc32, arm64 by default
 -b, --boottype       type of boot media, valid argument: nor, sd, emmc, qspi, xspi, nand, default all types if unspecified
 -c, --component       component to be built, valid argument: firmware, apps, linux, uboot, atf, rcw, mc_utils, restool,
     edgescale, fmc, openssl, vpp, dpdk, ovs_dpdk, pktgen_dpdk, openstack_nova, optee_os, libpkcs11, secure_obj, etc
 -r, --rootfs          specify flavor of target rootfs, valid argument: ubuntu|yocto|centos:main|devel|lite|tiny|edgescale|cloud
 -i, --instruction     instruction to do for dedicated operation
 -s, --secure       enable security feature in case of secure boot

See docs/flexbuild_usage.txt and docs/lsdk_build_install.txt for more information about the available commands.
```

Available from shell linux command prompt once **source setup.env**

flex-installer

```
Usage:  flex-installer [ -i <instruction> ] [ -b <bootpartition> ] [ -r <rootfs> ] [ -f <firmware> ] [ -d <device> ] [ -
m <machine> ]
OPTION:
 -i, --instruction  Instruction to execute, valid argument as below:
     'auto'    Automatically partition and format the target storage drive, then download and install distro
images
     'pf'          Only partition and format the target storage drive without installing distro images
     'download'     Only download distro images without installation
     'install'      Only install the specified image, can be omitted by default
     'list'         Show the list of supported machines and installation invironment
     'mksdcard'     Create sdcard.img including composite firmware and distro images
 -b, --bootpart    Boot partition image to be programmed into SD/USB/SATA storage drive
 -r, --rootfs    The first distro rootfs image by default to be programmed into target storage drive
 -R, --rootfs2    The second distro rootfs image for dual distros installation
 -d, --device    Device name of the target SD/USB/SATA storage drive in Linux
 -p, --partition    Specify configurable partitions of target disk, default as "-p 4P=100M:1G:6G:-1" if not
specified
 -f, --firmware    Composite firmware to be programmed into SD card
 -F, --force        Force partition and format target disk regardless of the data in disk
 -e, --efi    Used for the case of UEFI as bootloader instead of U-BOot, valid argument: dtb or acpi
 -m, --machine    Target machine name to specify the name of composite firmware for automatical
deployment
 -u, --url    Specify URL of distro webserver to override the default one for automatically downloading distro
 -v, --version    Print version info
 -h, --help    Print help info
```

Available from shell linux command prompt once **source setup.env**

flex-installer

Examples:
- Automatically download aauto -m ls1046ardb -d /dev/mmcblk0 (default latest LSDK distro)
  $ flex-installer -i auto:lsdk1906 -m lx2160ardb -d /dev/sdx   (specify LSDK version)
  You can specify one or several of '-b <bootpartition> -r <rootfs> -R <second-rootfs> -f <firmware> -u <url>' options to override the default settings

- Install local distro images with single distro:
  $ flex-installer -b bootpartition_arm64_lts_4.19.tgz -r rootfs_lsdk1906_LS_arm64_main.tgz -f firmware_ls1046ardb_uboot_sdboot.img -d /dev/sdx

- Install local distro images with dual distros:
  $ flex-installer -b bootpartition_arm64_lts_4.19.tgz -r rootfs_lsdk1906_LS_arm64_main.tgz -R rootfs_buildroot_LS_arm64_devel.tgz -f <firmware> -d /dev/sdx
  (run 'setenv devpart_root 3;boot' in U-Boot to boot the second distro from partition 3)

- On ARM board running TinyDistro, first partition target disk, then download local distro images onto board and install as below:
  $ flex-installer -i pf -d /dev/mmcblk0 (or /dev/sdx)
  $ cd /run/media/mmcblk0p3 nd install LSDK distro images to target storage drive on host machine or ARM board:
  $ flex-installer -i (or sdx3) and download distro images to this partition via wget or scp
  $ flex-installer -b bootpartition_arm64_lts_xx.tgz -r rootfs_lsdk1906_LS_arm64_main.tgz -d /dev/mmcblk0 (or /dev/sdx)

- Only download disto images:
  $ flex-installer -i download -m ls1046ardb

- only install composite firmware:
  $ flex-installer -f firmware_lx2160ardb_uboot_sdboot.img -d /dev/mmcblk0 (or /dev/sdx)

- Install distro into sdcard.img with loop device
  $ flex-installer -i mksdcard -m ls1046ardb

Note: '-e dtb' or '-e acpi' option is needed if UEFI is used as bootloader, no need in case of U-Boot.

Available from shell linux command prompt once **source setup.env**

References

- Basic flexbuild usage instructions in the sources: flexbuild/docs
  - **Layerscape Software Developmnt Kit User Guide 19.09** (REV 19.09_311219) updated
- Plenty of git resources online
  - Basic one: http://rogerdudler.github.io/git-guide/