SDK2

Aug 15, 2023

Contents

CHAPTER 1

Overview

The SDK^2 (*SDK-squared*, or *SDK-SDK*) is a software development kit (SDK) for building software development kits. Amongst other things, it is used to build the MESA SDK and MadSDK. It was developed by the Massive Stars Group at the University of Wisconsin-Madison.

CHAPTER 2

Basic Usage

To use the SDK2, first clone the repository from GitHub:

```
$ git clone https://github.com/rhdtownsend/sdk2.git
```

Set the `SDK2_ROOT` environment variable to point to the root directory of the repository:

```
$ export SDK2_ROOT=~/sdk2
```

Also, set the `SDK2_TMP` environment variable to point to a temporary directory that will be used to build SDKs (if possible, place this directory on a fast storage device, such as an SSD):

```
$ export SDK2_TMP=~/sdk2-tmp
```

Next, choose an appropriate *profile*. The profile determines which SDK will be built, and for what platform. Current choices include:

`mesasdk-x86_64-macos` MESA SDK running on Intel 64-bit OSX (10.10 onward)

`mesasdk-x86_64-linux` MESA SDK running on Intel 64-bit Linux

`mesasdk-x86_64-linux` MESA SDK running on ARM 64-bit Linux

`madsdk-x86_64-macos` Mad SDK running on Intel 64-bit OSX (10.10 onward)

`madsdk-x86_64-linux` Mad SDK running on Intel 64-bit Linux

See the profile directory for the complete set of profiles (each profile is stored in its own subdirectory; `common` is a special directory used to store info common to all profiles).

Once you've chosen a profile, set the `SDK2_PROFILE` environment variable accordingly, e.g.

```
$ export SDK2_PROFILE=mesasdk-x86_64-linux
```

Finally, set the `SDK2_RELEASE` environment variable to the relase number of the SDK. Convention is to use `Y.M.N`, where `Y` is the two-digit year, `M` is the month number, and `N` is an index counting upward from `1` for each release made in that month. So, for the third release in November 2020, you would use

```
$ export SDK2_RELEASE=20.11.3
```

With these three environment variables set, you can now build the SDK via

```
$ $SDK2_ROOT/exec/sdk2 all
```

After some time (typically, an hour or two, depending on the speed of your system), the SDK2 will complete the build process, and you'll have a fresh SDK sitting in the directory *$SDK2_TMP/$PROFILE_NAME*. Here, PROFILE_NAME is the name of the profile — mesasdk for the MESA SDK, and madsdk for the Mad SDK.

CHAPTER 3

Customization

3.1 Changing Packages

To change which packages are built for a given profile, edit the list of packages given in the file *$SDK2_ROOT*/profile/<profile_name>/packages. Lines can be commented out using the # symbol.

3.2 Adding Profiles

To add a profile, simply copy one of the existing profiles in *$SDK2_ROOT*/profile and edit the files *$SDK2_ROOT*/profile/<profile_name>/settings and *$SDK2_ROOT*/profile/<profile_name>/packages as necessary.

3.3 Adding Packages

To add a package (for subsequent listing in *$SDK2_ROOT*/profile/<profile_name>/packages), you need to create a package description (PD) file in the directory *$SDK2_ROOT*/package (or a subdirectory thereof). The PD file is written in the Bash scripting language. Perhaps the best way to learn about PD files is to look at the existing files within the subdirectories of $SDK2_ROOT/package. Briefly, at minimum a PD file must define the following variables:

SRC_FILE Name of file containing source code for package (must exist in *$SDK2_TMP*/src directory, or on the server with the URL prefix given by the PROFILE_URL profile variable)

SRC_DIR Name of directory to unpack source code into (typically, the same as the package name)

Additionally, the PD file must define a Bash function build() which contains the commands necessary to build the package. It may also define functions unpack() and install(), which handle unpacking the source code and installing the package; if these are not defined, then defaults are used (see the script *$SDK2_ROOT*/exec/sdk2 for details).

CHAPTER 4

Portability

If you're creating an SDK for use on multiple Linux systems, you should build the SDK on the oldest system you plan on using. This is to ensure that the SDK doesn't end up using functionality in newer releases of glibc (the GNU shared C library) that is absent on older systems. For the record, the MESA and Mad SDKs are built in a Docker container based on CentOS 5.11. To build the image for this container, use the *$SDK2_ROOT*/docker/create_docker_image script; and to run the container, use the *$SDK2_ROOT*/docker/run_docker_container script.

CHAPTER 5

Support

The SDK2 project comes without any guarantee of support whatsoever. However, we would like to hear about any problems you encounter when using it.

Index