

## Overview

## Overview

# Overview of the SLC-CLI

The Silicon Labs Configurator (SLC) is a metadata specification for the Simplicity SDK. It also describes methods of creating and configuring embedded software projects for Silicon Labs IoT devices using this metadata. Software is grouped into components (defined by .slcc files) that may provide features and/or require features provided by other components. Example projects (.slcp) describe a single software application (usually made up of multiple components plus application code) that can be used to generate an IDE project. See the [SLC Specification](#) for details about SLC.

The SLC Command Line Interface (SLC-CLI) tool, among other things, resolves project and component dependencies and generates a project for a specified embedded target and build system (for example, IAR Embedded Workbench or GNU tools via a Makefile), among other things. This section provides references to the most common operations done with SLC-CLI.

SLC-CLI is provided as a downloadable .zip file for three operating systems:

- Windows
- MacOS
- Linux

SLC-CLI may be used with the following SDKs and platforms:

- Gecko Bootloader
- Gecko Platform
- The Simplicity SDK

Example projects (defined in .slcp files) are installed with the SDK in a directory under the Simplicity SDK installed directory. The location varies depending on the SDK.

- Amazon AWS: <SiSDKpath>\app\amazon\example
- Bluetooth SDK: <SiSDKpath>\app\bluetooth\example
- Bluetooth Mesh SDK: <SiSDKpath>\app\btmesh\example
- OpenThread SDK: <SiSDKpath>\protocol\openthread\sample-apps
- 32-Bit MCU SDK: <SiSDKpath>\app\mcu\_example
- Proprietary (Flex) SDK: <SiSDKpath>\app\flex\example\<Connect or RAIL>
- Gecko Bootloader: <SiSDKpath>\platform\bootloader\sample-apps
- SDK Platform: <SiSDKpath>\app\common\example
- Wi-SUN SDK: <SiSDKpath>\app\wisun\example
- Z-Wave SDK: <SiSDKpath>\protocol\z-wave\apps
- Zigbee SDK: <SiSDKpath>\protocol\zigbee\app

SLC-compatible SDKs may also support extensions that may include example projects as well as components. By default, extensions are installed into the extension folder at the root of an SDK.

Extension: <SDKpath>\extension\<extension\_name>