AN1499: Configuring OpenThread Applications for Thread 1.4

Thread 1.4 builds on Thread's robust foundation. Thread 1.4 border routers seamlessly allow onboarding devices by making use of a standardized way to share credentials and join Thread networks. Device manufacturers can add compelling new features and offer dynamic support and cloud-powered features. Thread 1.4 also allows enhanced mesh diagnostics by providing more thorough visibility into a Thread mesh topology. Additionally, for professional installations and commercial building scenarios, Thread 1.4 defines secure commissioning at scale by using authenticated transport layers over standards such as Bluetooth.

Note: Silicon Labs includes the OpenThread stack with the current default protocol version 1.4 (=5). All mandatory features in 1.3, 1.2, and 1.1 are automatically enabled, and the stack is backwards compatible.

Silicon Labs provides components and configuration options that enable you to configure Thread 1.4 features with sample applications. These features are compatible with EFR32MG1x and EFR32MG2x SoCs, RCPs, and modules. This application note assumes you have a basic understanding of how Thread is implemented on EFR32 devices. For more information, see *UG103.11: Thread Fundamentals*.


KEY POINTS

- Including Thread 1.4 features in SoC Applications
- Including Thread 1.4 features in an OpenThread Border Router

---

1 Including Thread 1.4 features in SoC Applications

Silicon Labs provides a number of sample SoC OpenThread applications. You can modify these to include Thread 1.4 features (most of which are enabled by default). This chapter assumes you are familiar with creating and modifying OpenThread projects in Simplicity Studio 5. If you need more information, see the Simplicity Studio 5 User's Guide and *QSG170: Silicon Labs OpenThread Quick Start Guide*.

As an example, the following procedure shows how to configure 1.4 features:

1. Create a project based on the example: **OpenThread – SoC CLI (FTD)**.

2. On the **SOFTWARE COMPONENTS** tab, search for and select the **Stack (FTD)** entry. Depending on your application, you may have to do this on a **Stack (MTD)** or **Stack (RCP)** component (this example is for an FTD application).
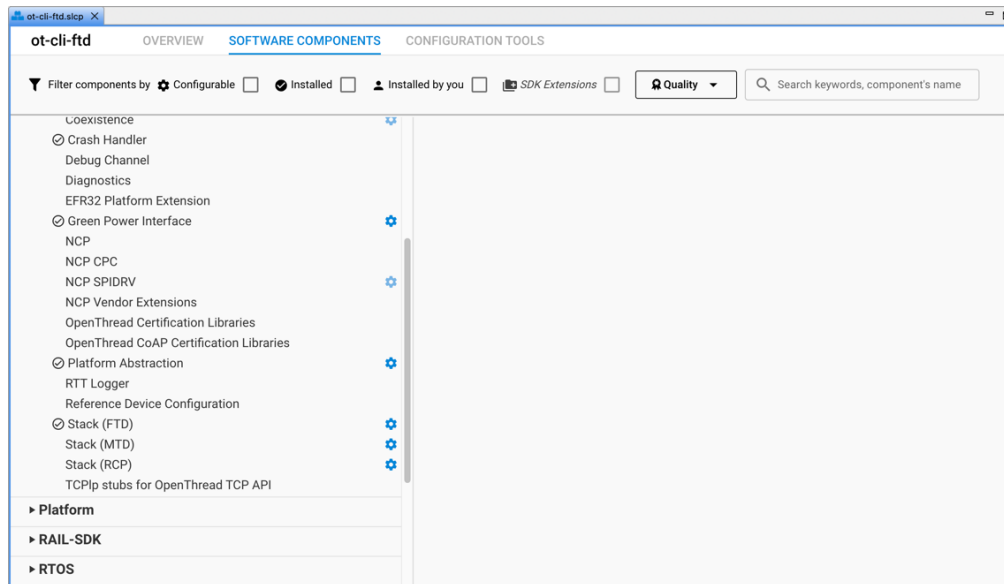
3. In the stack component, make sure to configure the **Thread Stack Protocol Version** to 1.4.
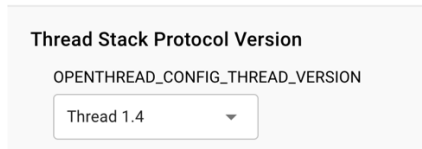
The following are border router features defined in the 1.4 specification of the Thread standard. The OpenThread stack allows configuration of these features for any Thread stack protocol version and on any router. However, a Thread 1.4 compliant border router device must enable these features.

- Ephemeral Key support for Credential Sharing
  - Not needed on non-border routers
- Mesh Diagnostics: Mandatory and automatically enabled for border routers
  - Can be enabled on any router capable device
- Support for DHCPv6 Prefix Delegation: Mandatory and automatically enabled for border routers
  - Not needed on non-border routers

There is one optional exception that is omitted from configuration:
- Support for Thread Commissioning over Bluetooth (TCAT over Bluetooth): Commercial feature

Additional information about these features is included in the following table.

Table 1-1. Thread 1.4 Configuration Flags

Flag	Note
Ephemeral Key for Credential Sharing (OPENTHREAD_CONFIG_BORDER_AGENT_EPHEMERAL_KEY_ENABLE)	Enables setting an ephemeral PSKc to allow a commissioner (from a different network or ecosystem for example) to establish a secure connection with a border router. Once a connection is established with an ePSKc, the commissioner can set or program a different network active dataset, thus allowing multiple Thread networks to form a single, larger ranging mesh. This feature is only meant for devices with a border agent, such as a Thread border router. For more information, see: https://github.com/openthread/openthread/blob/main/src/cli/README.md#ba-ephemeralkey
Mesh Diagnostics (OPENTHREAD_CONFIG_MESH_DIAG_ENABLE)	Allow sending diagnostic requests and queries to other nodes and process the responses. Includes support for enhanced network diagnostics in Thread 1.4 to enable better insight into network topology and neighbor links, helping one visualize network state. For more information, see commands starting at: https://github.com/openthread/openthread/blob/main/src/cli/README.md#meshdiag-topology-ip6-addrs-children
DHCP6 PD feature (OPENTHREAD_CONFIG_BORDER_ROUTING_DHCP6_PD_ENABLE)	Supports handling platform generated ND messages. The prefix can be allocated by other software on the interface, which will advertise the assigned prefix to the thread interface via router advertisement messages.
Thread Commissioning over Authenticated TLS (implemented over Bluetooth) (OPENTHREAD_CONFIG_BLE_TCAT_ENABLE)	This is an optional feature meant only for commercial networks. To read more about Thread Commissioning at Scale, see example client at: https://github.com/openthread/openthread/tree/main/tools/tcat_ble_client

2 Including Thread 1.4 features in an OpenThread Border Router

Silicon Labs provides several sample OpenThread RCP applications. By default, the RCP applications on supported Silicon Labs hardware automatically support Thread 1.4 features if they are present on the host (the Border Router, which is Silicon Labs' supported RCP model). None of the Thread 1.4 features are RCP-specific, so turning them on or off for the RCP sample application has no effect.

Refer to *AN1256: Using the Silicon Labs RCP with the OpenThread Border Router* for detailed instructions on how to build an OpenThread Border Router for Raspberry Pi 3B+ or above. You must use a Thread protocol version 1.4 RCP with a Border Router that is also running a stack at protocol version 1.4.

Most Thread 1.4 features can be enabled on the border router simply by setting the protocol version to Thread 1.4, as they will be enabled by default. However, you can review the following CMake flags for more information. (See Table 1-1. Thread 1.4 Configuration Flags for more information on their purpose.)

Table 2-1. Border Router CMake Mappings

CMake Flag	Thread 1.4 Configuration Flag
OT_BORDER_AGENT_EPSKC	OPENTHREAD_CONFIG_BORDER_AGENT_EPHEMERAL_KEY_ENABLE
OT_MESH_DIAG	OPENTHREAD_CONFIG_MESH_DIAG_ENABLE
OT_BORDER_ROUTING_DHCP6_PD	OPENTHREAD_CONFIG_BORDER_ROUTING_DHCP6_PD_ENABLE
OT_BLE_TCAT	OPENTHREAD_CONFIG_BLE_TCAT_ENABLE

You can install a pre-built Docker container with OpenThread Border Router:

https://hub.docker.com/r/siliconlabsinc/openthread-border-router/tags

Or you can manually install an OpenThread Border Router by following the steps in *AN1256: Using the Silicon Labs RCP with the OpenThread Border Router* or https://openthread.io/guides/border-router/build.

Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

SILICON LABS

www.silabs.com