连接裸机设备

您可以将生成和运行工具包配置为使用安装在开发主机上的裸机工具链为裸机设备构建应用程序。您可以将设备连接到开发主机，以使用 GDB 或硬件调试器从 Qt Creator 运行和调试应用程序。这使您能够在通用远程 Linux 设备插件不支持的小型设备上进行调试。

> **注意：** 如果您使用qmake构建项目并且设备没有Qt库，则需要虚假的Qt安装。

构建应用程序支持以下工具链：

- GCC：微芯片技术（AVR、AVR32、PIC16、PIC32）、恩智浦半导体（冷火、M68K）、德州仪器仪表（MSP430）、美国国家半导体公司（CR16C）、瑞萨电子（M32R、M32C、RL78、RX、超级H、V850）、天基硅XTENSA（ESP8266、ESP32）、RISC-V、ARM
- IAR电子版：微芯片技术（AVR、AVR32）、恩智浦半导体（冷火、M68K）、德州仪器仪表（MSP430）、美国国家半导体（CR16C）、瑞萨电子（78K、M16/R8C、M32C、R32C、RH850、RL78、RX、超级H、V850）、意法半导体（STM8）、8051、RISC-V、臂
- 凯尔：手臂，C51（8051），C251（80251），C166（C16x，XC16x）
- SDCC：意法半导体（STM8），8051

裸机设备类型接受您在设备首选项中指定的自定义 GDB 命令。您可以指定在使用特定调试服务器提供程序进行连接时要执行的命令。

使用 GDB 时，支持以下调试服务器提供程序：

- 断续器
- 捷联
- 开放文档开发
- 科通

ST-Link 和 J-Link 调试服务器提供程序可以与 uVision IDE 一起使用。

启用裸机设备插件

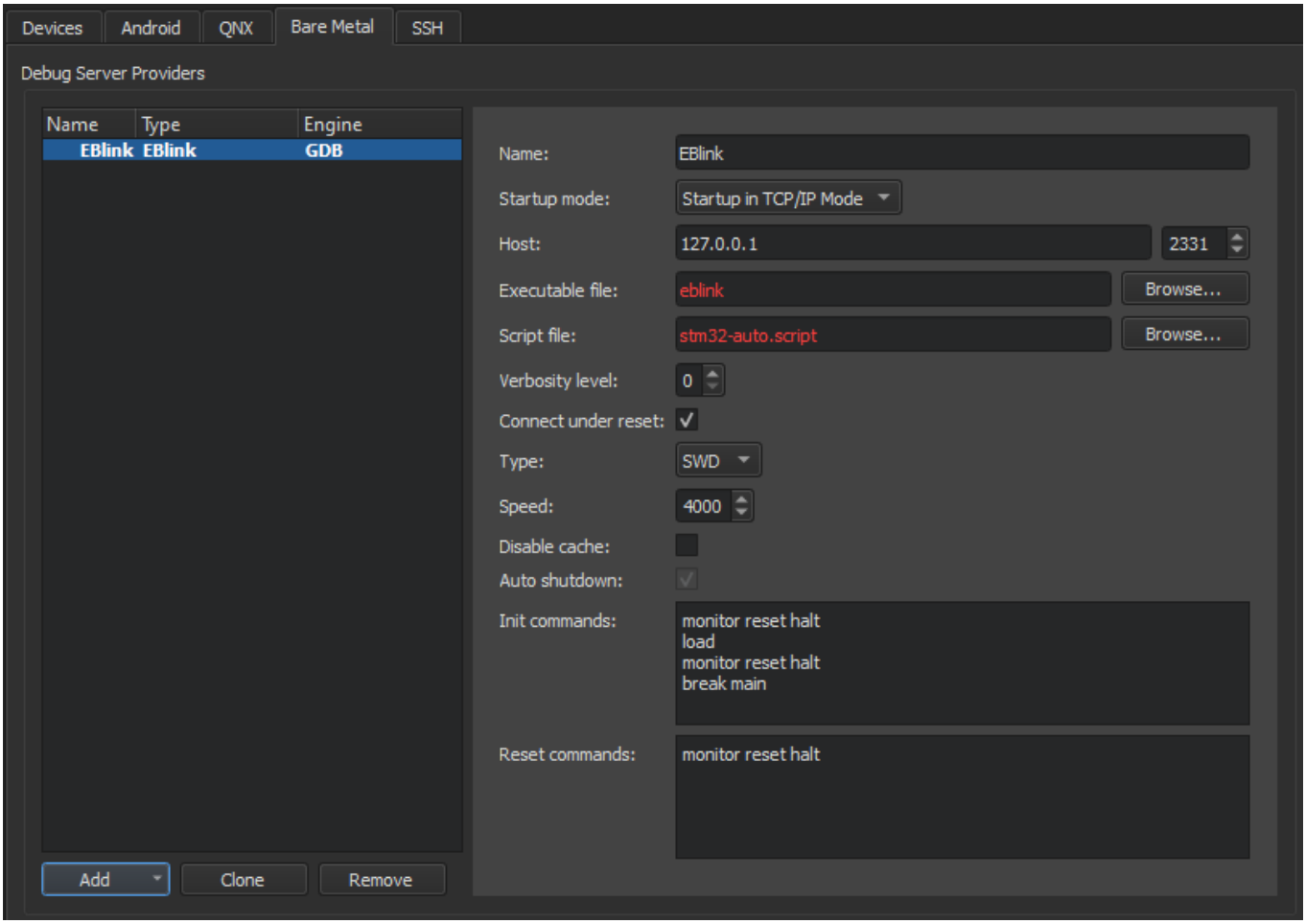
要启用裸机设备插件：

1. 选择"**帮助**">"**关于插件**>"设备**支持**>**裸机**"。
2. 选择**立即重新启动**以重新启动Qt创建器并加载插件。

若要使用调试服务器提供程序创建与裸机设备的连接，请选择“**编辑**>**首选项**”>**设备**>“**裸机**>**添加**>**默认值**”。可用设置取决于调试服务器提供程序。

断续器

易闪是一款 ARM Cortex-M 调试工具，支持松鼠脚本、实时变量和热插拔。

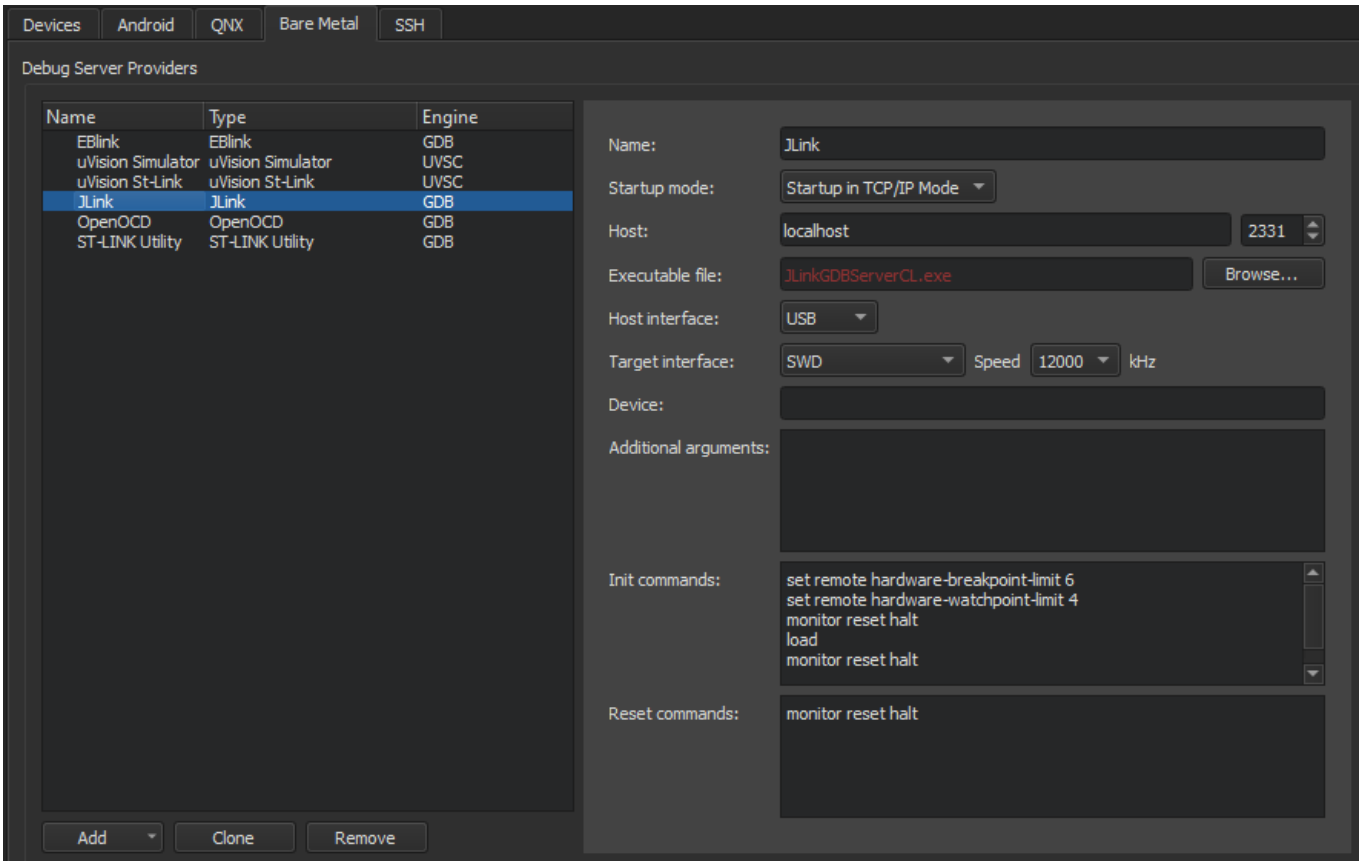


To specify settings for EBlink:

1. In the **Name** field, enter a name for the connection.
2. In the **Startup mode** field, select the mode to start the debug server provider in.
3. In the **Host** field, select the host name and port number to connect to the debug server provider.
4. In the **Executable file** field, enter the path to the debug server provider executable.
5. In the **Script file** field, enter the path to a device script file.
6. In the **Verbosity level** field, enter the level of verbose logging.
7. Select the **Connect under reset** check box to use the ST-Link interface. Deselect the check box for hot-plugging.
8. In the **Type** field, select the interface type.
9. In the **Speed** field, enter the interface speed between 120 and 8000 kiloherz (kHz).
10. Select the **Disable cache** check box to disable the EBlink flash cache.
11. Select the **Auto shutdown** check box to automatically shut down the EBlink server after disconnecting.
12. In the **Init commands** field, enter the commands to execute when initializing the connection.
13. In the **Reset commands** field, enter the commands to execute when resetting the connection.

J-Link

J-Link is a line of debug probes by Segger.

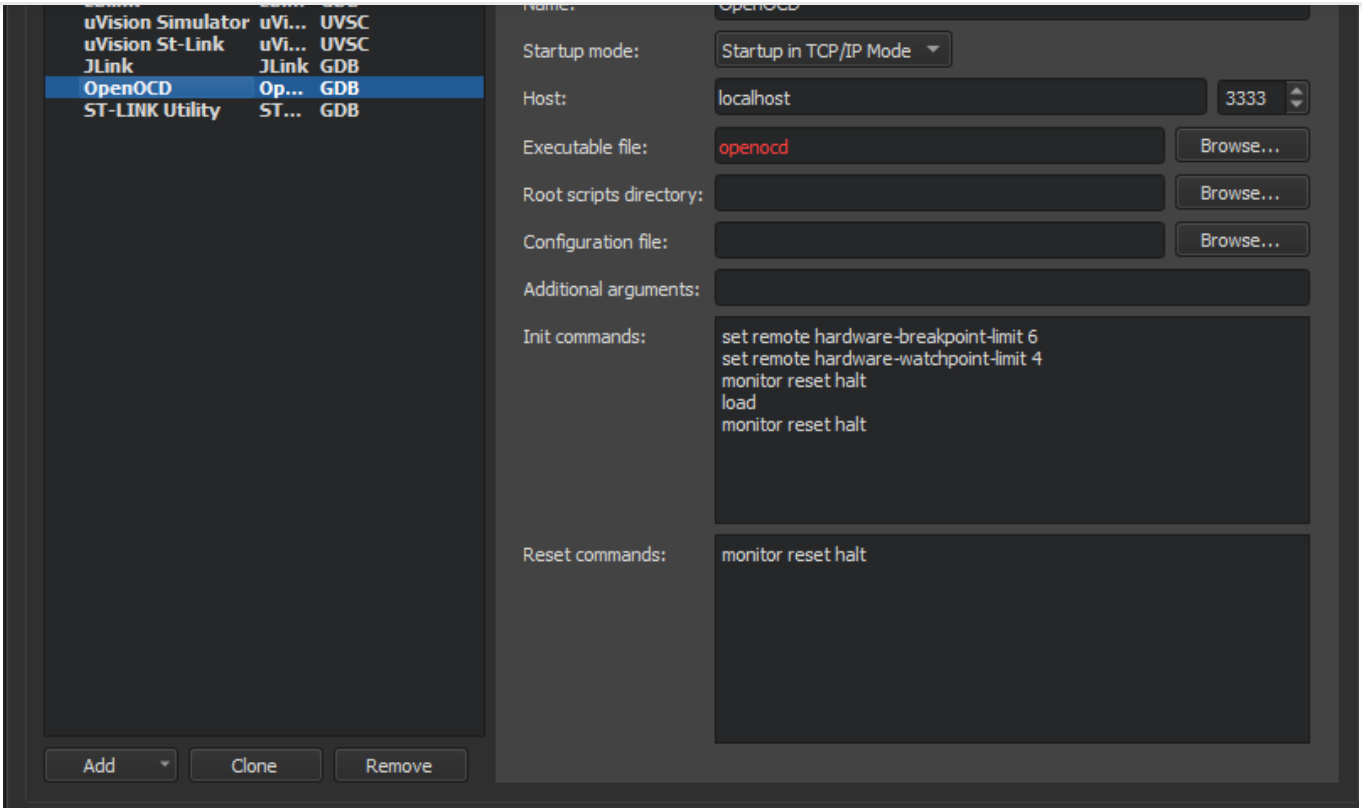


To specify settings for J-Link debug probes:

1. In the **Name** field, enter a name for the connection.
2. In the **Startup mode** field, select the mode to start the debug server provider in.
3. In the **Host** field, select the host name and port number to connect to the debug server provider.
4. In the **Executable file** field, enter the path to the debug server provider executable.
5. In the **Host interface** field, select the connection type, IP or USB, or use the default connection.
6. In the **Target interface** field, select the target interface type.
7. In the **Speed** field, enter the interface speed in kHz.
8. In the **Device** field, select the device to connect to.
9. In the **Additional arguments** field, enter arguments for the commands.
10. In the **Init commands** field, enter the commands to execute when initializing the connection.
11. In the **Reset commands** field, enter the commands to execute when resetting the connection.
12. Select **Apply** to add the debug server provider.

OpenOCD

OpenOCD (Open On-Chip Debugger) is an on-chip debug solution for targets based on the ARM7 and ARM9 family with Embedded-ICE (JTAG) facility. It enables source level debugging with the GDB compiled for the ARM architecture.

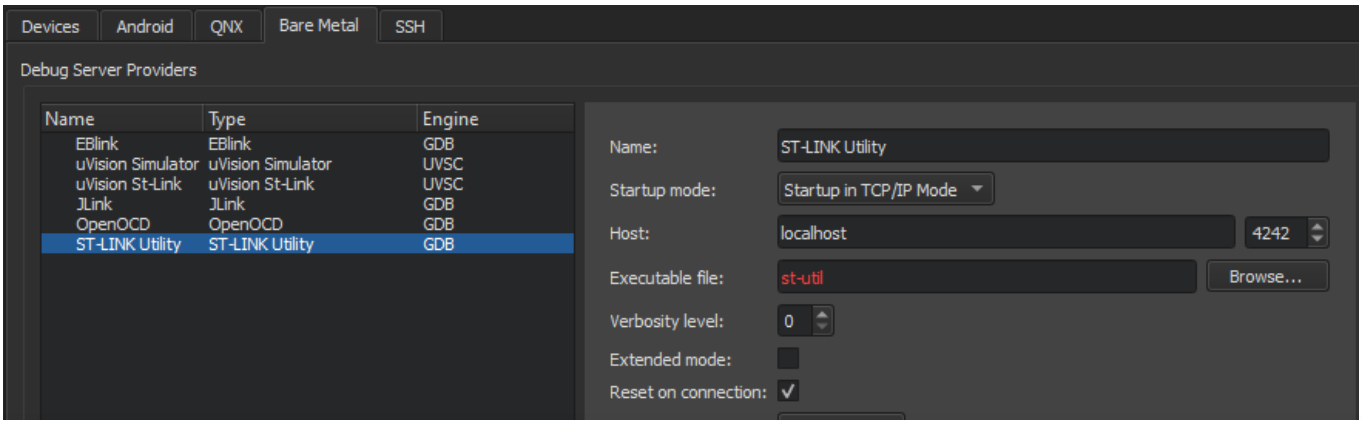


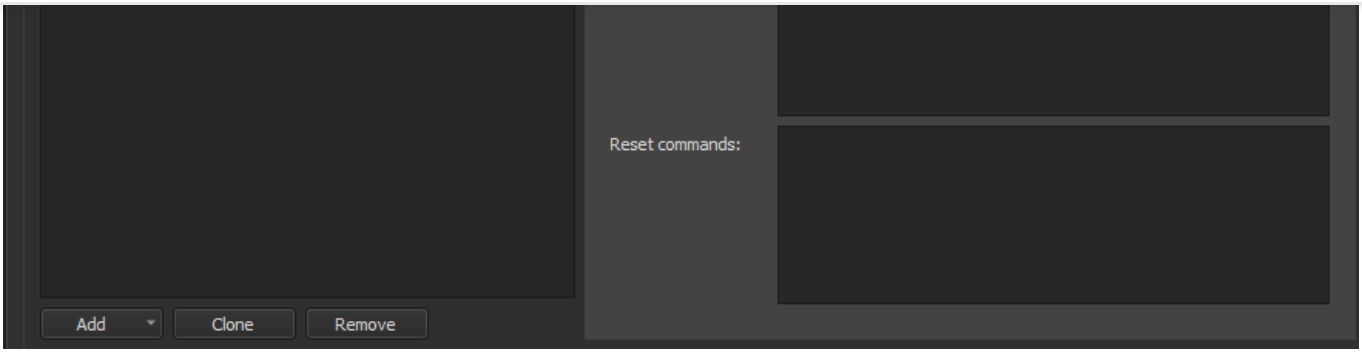
To specify settings for OpenOCD:

1. In the **Name** field, enter a name for the connection.
2. In the **Startup mode** field, select the mode to start the debug server provider in.
3. In the **Host** field, select the host name and port number to connect to the debug server provider.
4. In the **Executable file** field, enter the path to the debug server provider executable.
5. In the **Root scripts directory** field, enter the path to the directory that contains configuration scripts.
6. In the **Configuration file** field, enter the path to the device configuration file.
7. In the **Additional arguments** field, enter arguments for the commands.
8. In the **Init commands** field, enter the commands to execute when initializing the connection.
9. In the **Reset commands** field, enter the commands to execute when resetting the connection.
10. Select **Apply** to add the debug server provider.

St-Link

ST-LINK Utility is used for programming STM32 microcontrollers.





To specify settings for St-Link:

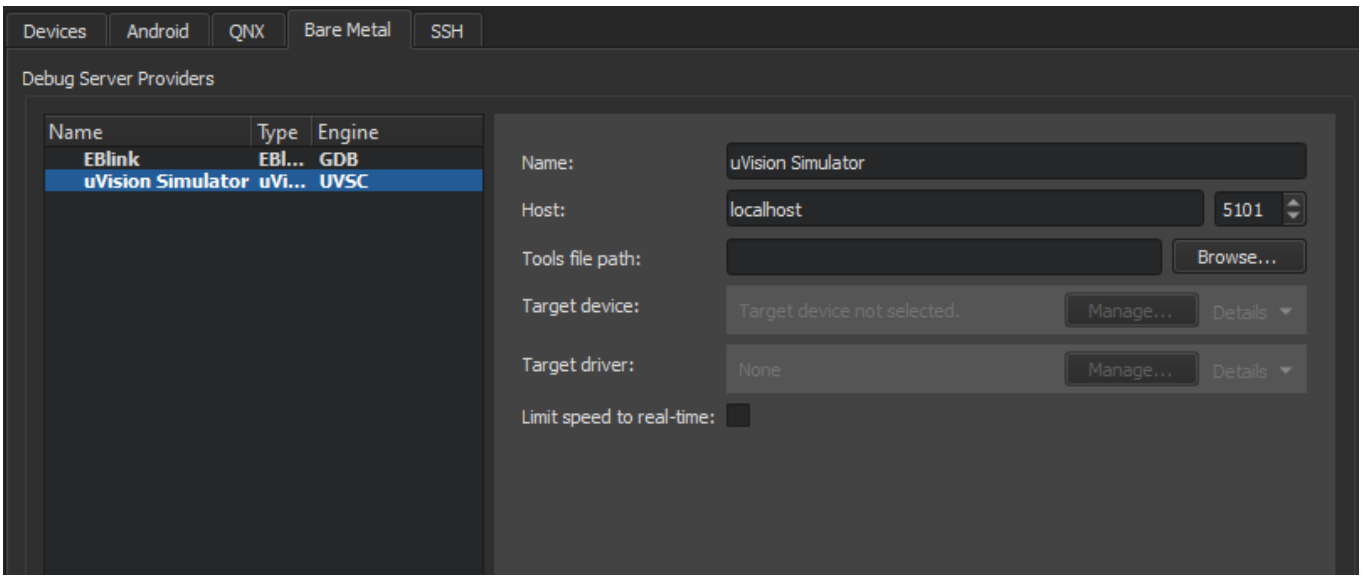
1. In the **Name** field, enter a name for the connection.
2. In the **Startup mode** field, select the mode to start the debug server provider in.
3. In the **Host** field, select the host name and port number to connect to the debug server provider.
4. In the **Executable file** field, enter the path to the debug server provider executable.
5. In the **Verbosity level** field, enter the level of verbose logging.
6. Select the **Extended mode** check box to continue listening for connection requests after after the connection is closed.
7. Select the **Reset on connection** check box to reset the board when the connection is created.
8. In the **Version** field, select the transport layer type supported by the device.
9. In the **Init commands** field, enter the commands to execute when initializing the connection.
10. In the **Reset commands** field, enter the commands to execute when resetting the connection.
11. Select **Apply** to add the debug server provider.

uVision IDE

uVision is an IDE for developing applications for embedded devices. Applications can be debugged by using uVision Simulator or directly on hardware by using St-Link and J-Link.

You can view the current state of peripheral registers in the **Peripheral Registers** view in Debug mode. The view is hidden by default.

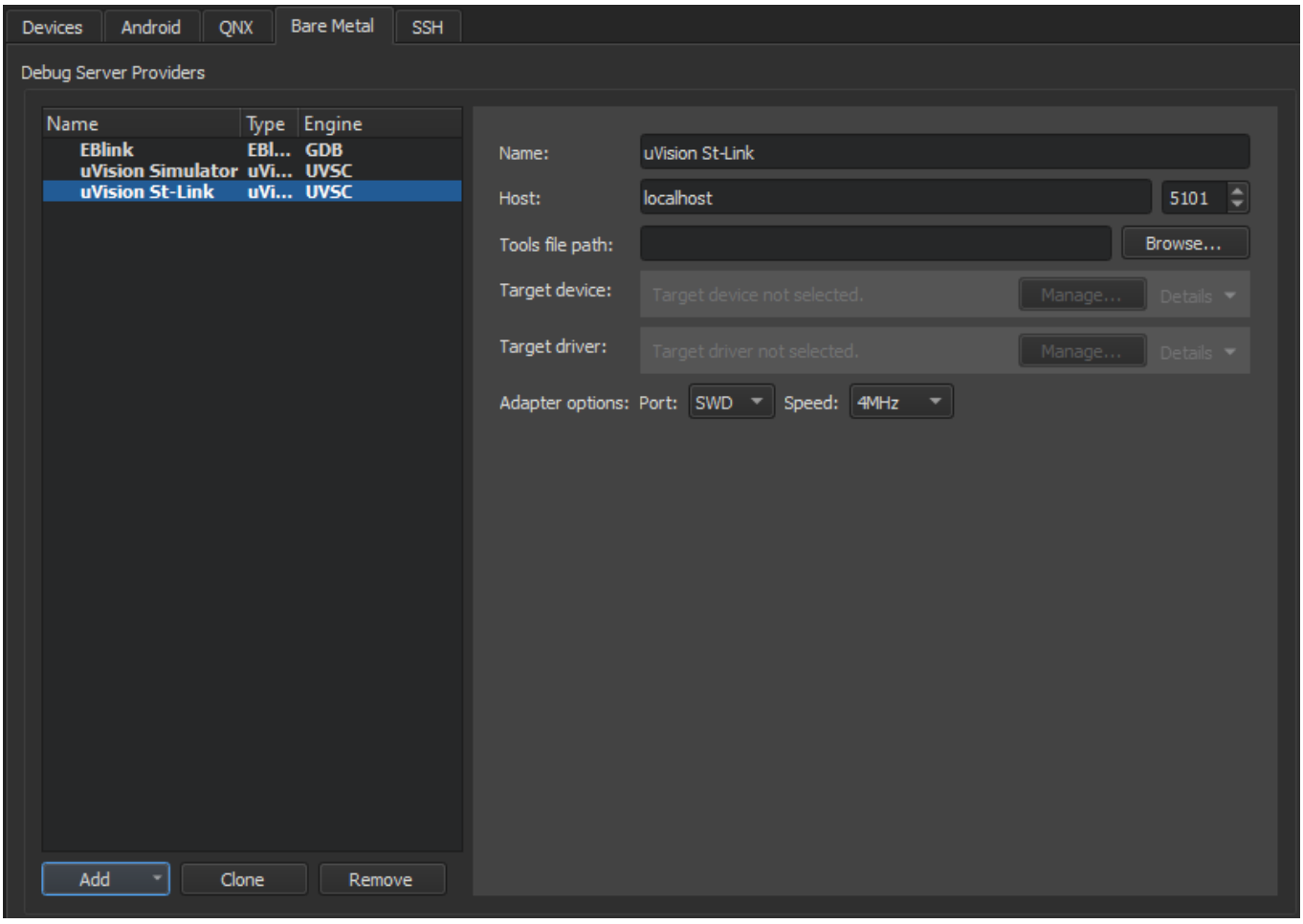
uVision Simulator



To specify settings for uVision Simulator or uVision St-Link Debugger:

1. In the **Name** field, enter a name for the connection.
2. In the **Host** field, select the host name and port number to connect to the debug server provider.
3. In the **Tools file path** field, enter the path to the Keil toolset configuration file.
4. In the **Target device** field, select the device to debug.
5. In the **Target driver** field, select the driver for connecting to the target device.
6. Select the **Limit speed to real-time** check box to limit the connection speed.
7. Select **Apply** to add the debug server provider.

uVision St-Link Debugger

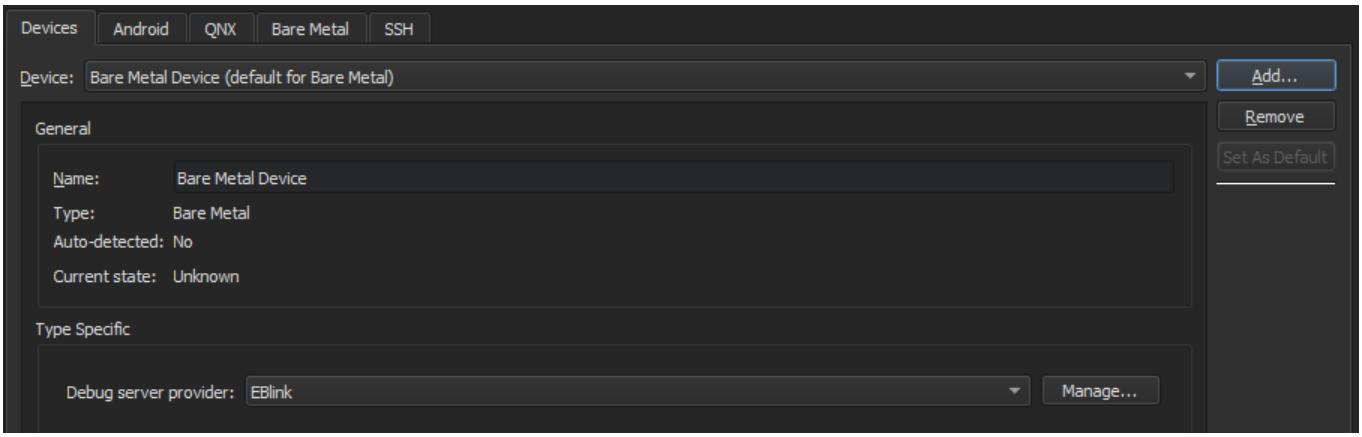


To specify settings for uVision St-Link Debugger:

1. In the **Name** field, enter a name for the connection.
2. In the **Host** field, select the host name and port number to connect to the debug server provider.

5. In the **Target driver** field, select the driver for connecting to the target device.
6. In the **Adapter options** field specify the adapter interface type and speed in MHz.
7. Select **Apply** to add the debug server provider.

Adding Bare Metal Devices

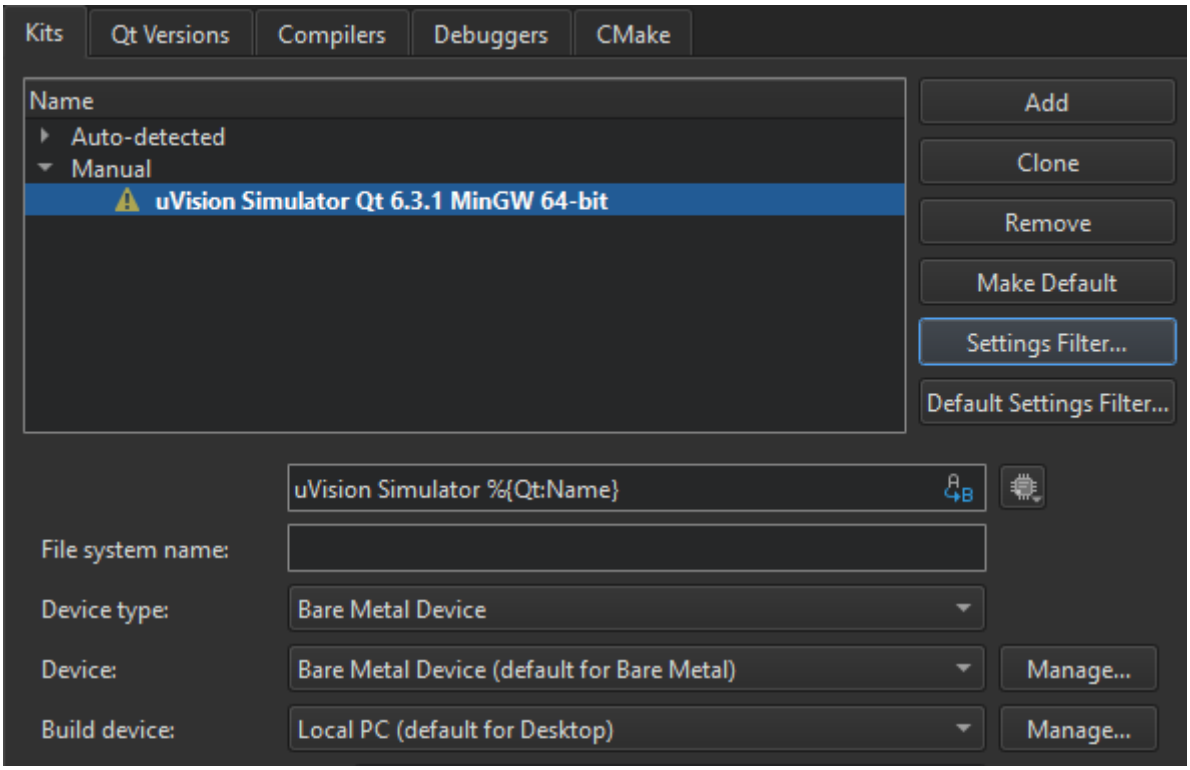


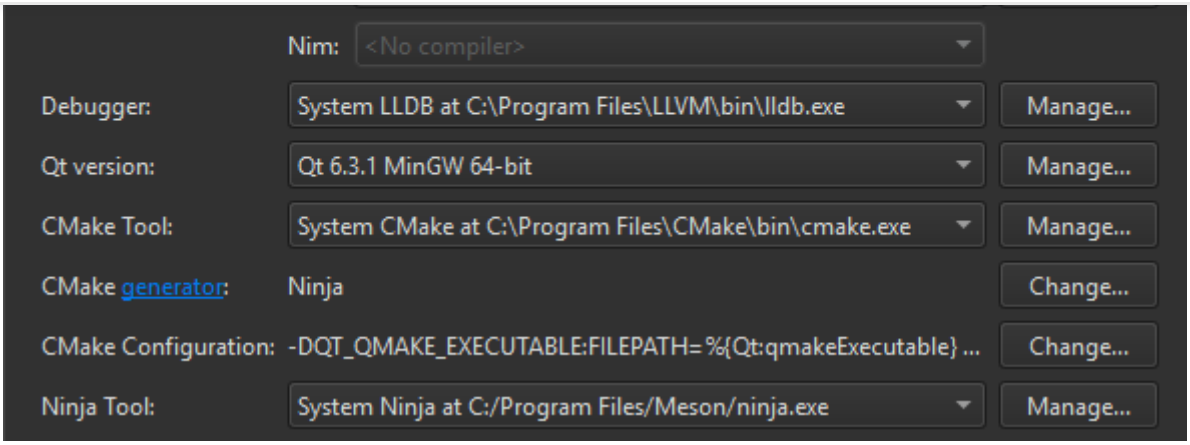
To add a bare metal device:

1. Select **Edit** > **Preferences** > **Devices** > **Add** > **Bare Metal Device** > **Start Wizard**.
2. In the **Debug server provider** field, select a debug server provider.
3. Select **Apply** to add the device.

Building for and Running on Bare Metal Devices

To add a kit for building applications and running them on bare metal devices, select **Edit** > **Preferences** > **Kits** > **Add**. For more information, see Adding Kits.





You can build applications for and run them on bare metal devices in the same way as for and on the desktop. For more information, see Building for Multiple Platforms and Running on Multiple Platforms.

‹ Connecting Android Devices

Connecting Boot2Qt Devices ›

Qt DOCUMENTATION

Community

Contribute to Qt
Forum
Wiki
Downloads
Marketplace

Feedback Sign In