

This chapter describes how to program and configure Altera® CPLD, FPGA, and configuration devices with the Quartus® II Programmer.

The Quartus II software offers a complete software solution for system designers who design with Altera FPGA and CPLD devices. After you compile your design, you can use the Quartus II Programmer to program or configure your device, to test its functionality on a circuit board.

This chapter contains the following sections:

- “Programming Flow”
- “Quartus II Programmer GUI” on page 18–3
- “Programming and Configuration Modes” on page 18–5
- “Scripting Support” on page 18–10

- ❓ For more information about how to use the Quartus II Programmer GUI to program and configure your device, refer to *Programming Devices* in Quartus II Help.

## Programming Flow

The following steps describe the general overview of the programming flow:

1. Compile your design, such that the Quartus II Assembler generates the programming or configuration file.
2. Convert the programming or configuration file to target your configuration device and, optionally, create secondary programming files.
3. Program and configure the FPGA, CPLD, or configuration device using the programming or configuration file with the Quartus II Programmer.

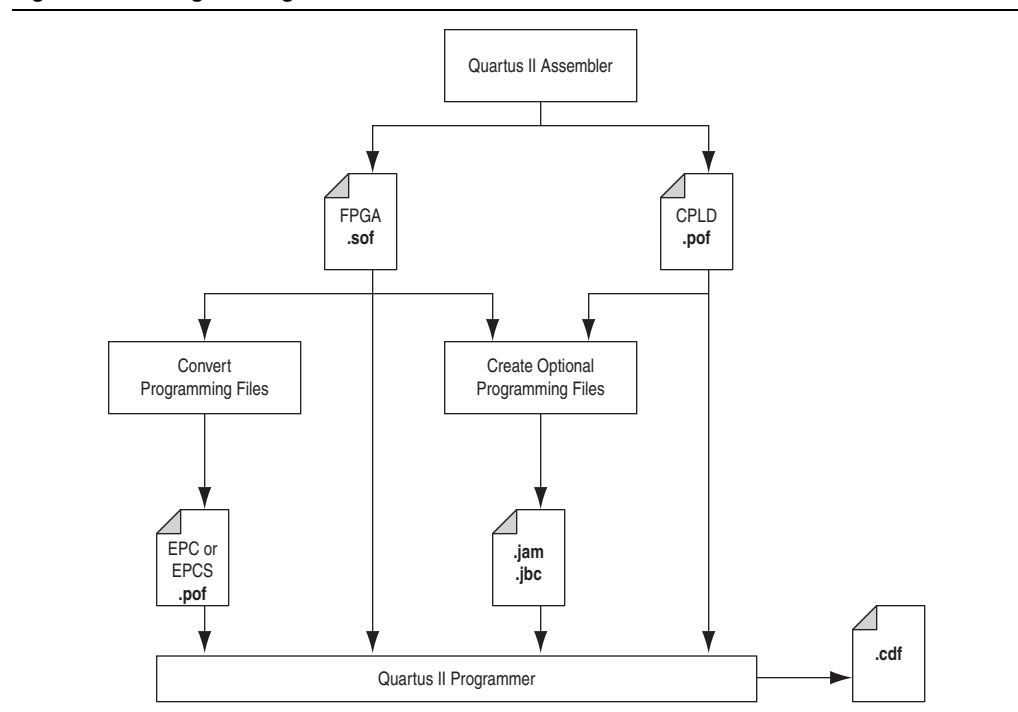
Table 18-1 lists the programming and configuration file formats supported by Altera FPGAs, CPLDs, and configuration devices.

**Table 18-1. Programming and Configuration File Format**

File Format	FPGA	CPLD	Configuration Device	Serial Configuration Device
SRAM Object File (.sof)	✓	—	—	—
Programmer Object File (.pof)	—	✓	✓	✓
JEDEC JESD71 STAPL Format File (.jam)	✓	✓	✓	—
Jam Byte Code File (.jbc)	✓	✓	✓	—

Figure 18-1 shows the programming file generation flow.

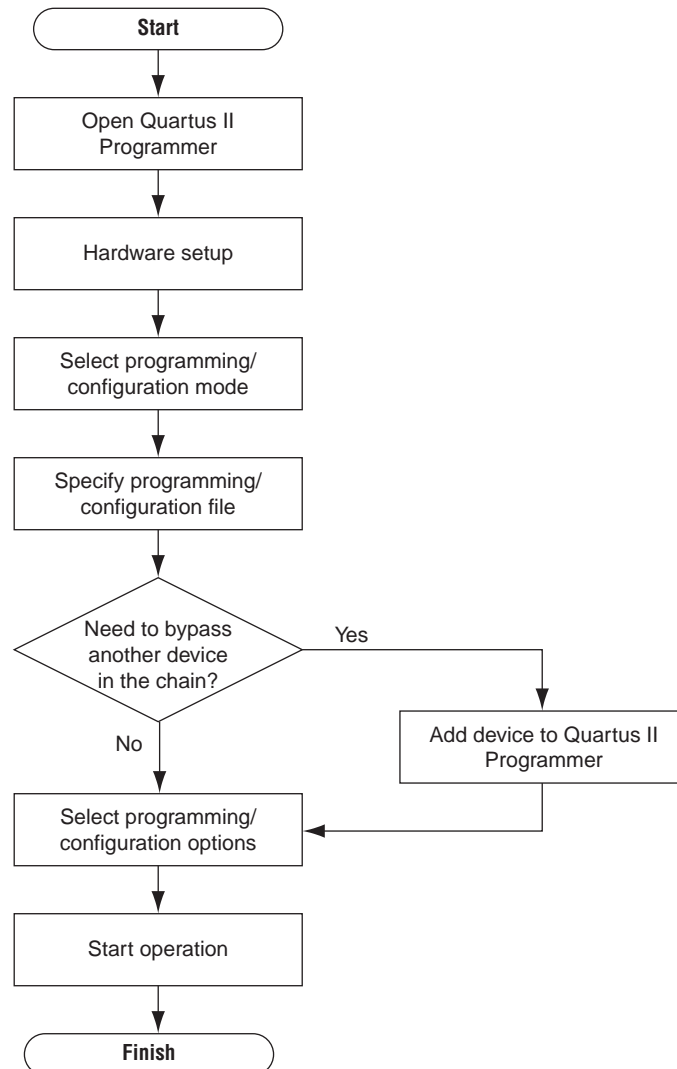
**Figure 18-1. Programming File Generation Flow**



- ❓ For more information about Chain Description Files (.cdf), refer to *About Programming* in Quartus II Help.

Figure 18-2 shows the Quartus II Programmer programming flow.

**Figure 18-2. Programming Flow**



## Quartus II Programmer GUI

The Quartus II Programmer GUI is a window in which you can add your programming and configuration files, specify programming options and hardware, and then proceed with the programming or configuration of the device.


To open the Programmer window, on the Tools menu, click **Programmer**. As you proceed through the programming flow, the Quartus II Message window reports the status of each operation.

If the Quartus II Programmer automatically detects devices with shared JTAG IDs, the Programmer prompts you to specify the correct device in the JTAG chain.

- ② For a description of the Programmer window, refer to *Programmer Window* in Quartus II Help. For a description of options in the Tools menu, refer to *Programmer Page (Options Dialog Box)* in Quartus II Help.

## Hardware Setup

The Quartus II Programmer provides the flexibility to choose a download cable or programming hardware. Before you can program or configure your device, you must have the correct hardware setup.

- ② For hardware settings, refer to *Setting Up Programming Hardware* in Quartus II Help.
-  For more information about programming hardware driver installation, refer to the *Setting up Programming Hardware in Quartus II Software* page on the Altera website.

## JTAG Settings

The JTAG server allows the Quartus II Programmer to access the JTAG hardware. You can also access the JTAG download cable or programming hardware connected to a remote computer through the JTAG server of that computer. With the JTAG server, you can control the programming or configuration of devices from a single computer through other computers at remote locations. The JTAG server uses the TCP/IP communications protocol.

- ② For more information about JTAG settings, refer to *Using the JTAG Server* in Quartus II Help.

## JTAG Chain Debugger Tool

The JTAG Chain Debugger tool allows you to test the JTAG chain integrity and detect intermittent failures of the JTAG chain. In addition, the tool allows you to shift in JTAG instructions and data through the JTAG interface and step through the test access port (TAP) controller state machine for debugging purposes. You access the tool from the Tools menu on the main menu of the Quartus II software.

- ② For more information, refer to *Using the JTAG Chain Debugger* in Quartus II Help.

## Other Programming Tools

The following section describes other programming tools in more detail.

### Stand-Alone Quartus II Programmer

Altera offers the free stand-alone Quartus II Programmer, which has the same full functionality as the Quartus II Programmer in the Quartus II software. The stand-alone Quartus II Programmer is useful when programming your devices with another workstation, so you do not need two full licenses. You can download the stand-alone Quartus II Programmer from the [Download Center](#) on the Altera website.

## Programming and Configuration Modes

The following section describes the Quartus II Programmer and the Programmer configuration modes.

### Configuration Modes

The Quartus II Programmer supports five configuration modes, including JTAG, passive serial (PS), active serial (AS), Configuration via Protocol (CvP), and in-socket modes (ISM).





Table 18–2 lists the programming and configuration modes supported by Altera devices.

**Table 18–2. Programming and Configuration Modes**

Mode	FPGA	CPLD	Configuration Device	Serial Configuration Device
JTAG	✓	✓	✓	—
PS	✓	—	—	—
AS	—	—	—	✓
CvP	✓	—	—	—
In-Socket Programming	—	✓ <sup>(1)</sup>	✓	✓

**Note to Table 18–2:**

(1) MAX II CPLDs do not support in-socket programming mode.

-  For more information about programming and configuration modes, refer to *About Programming* in Quartus II Help.
-  For more information about CvP configuration mode, refer to the *Configuration via Protocol (CvP) Implementation in Altera FPGAs User Guide*.
-  For more information about JTAG, PS, and AS configuration modes and in-socket programming mode, refer to the *Configuration Handbook*, or the device handbook or data sheet for the respective FPGA, CPLD, or configuration device.
-  For a list of programming adapters available for Altera devices, refer to [www.altera.com](http://www.altera.com).



## Design Security Keys

The Quartus II Programmer supports the generation of encryption key programming files and encrypted configuration files for Altera FPGAs that support the design security feature. You can also use the Quartus II Programmer to program the encryption key into the FPGA.

-  For more information about using the design security feature with the Quartus II software, refer to *AN 341: Using the Design Security Feature in Stratix II and Stratix II GX Devices* and *AN 512: Using the Design Security Feature in Stratix III Devices*.

## Optional Programming or Configuration Files

The Quartus II software can generate optional programming or configuration files in various formats that you can use with programming tools other than the Quartus II Programmer. When you compile a design in the Quartus II software, the Assembler automatically generates either a **.sof** or **.pof**. The Assembler also allows you to convert FPGA configuration files to programming files for configuration devices.

-  For more information, refer to *About Optional Programming Files* in Quartus II Help.
-  For more information about the programming and configuration file formats, refer to file format topics in the Quartus II Help or the *Configuration File Formats* chapter of the *Configuration Handbook*. For more information about using the **.jam** and **.jbc** programming files with the Jam STAPL Player, Jam STAPL Byte-Code Player, and the `quartus_jli` command-line executable, refer to *AN 425: Using Command-Line Jam STAPL Solution for Device Programming*.

## Secondary Programming Files

The Quartus II software generates programming files in various formats for use with different programming tools.

Table 18-3 lists the file types generated by the Quartus II software and supported by the Quartus II Programmer.

**Table 18-3. File Types Generated by the Quartus II Software and Supported by the Quartus II Programmer (Part 1 of 2)**

File Type	Generated by the Quartus II Software	Supported by the Quartus II Programmer
<b>.sof</b>	✓	✓
<b>.pof</b>	✓	✓
<b>.jam</b>	✓	✓
<b>.jbc</b>	✓	✓
JTAG Indirect Configuration File ( <b>.jic</b> )	✓	✓
Serial Vector Format File ( <b>.svf</b> )	✓	—
In System Configuration File ( <b>.isc</b> )	✓	—

**Table 18–3. File Types Generated by the Quartus II Software and Supported by the Quartus II Programmer (Part 2 of 2)**

File Type	Generated by the Quartus II Software	Supported by the Quartus II Programmer
Hexadecimal (Intel-Format) Output File (.hexout)	✓	—
Raw Binary File (.rbf)	✓	—
Tabular Text File (.ttf)	✓	—
Raw Programming Data File (.rpd)	✓	—


 For more information, refer to *Generating Secondary Programming Files* in Quartus II Help.

## Convert Programming Files Dialog Box

The **Convert Programming Files** dialog box in the Programmer allows you to convert programming files from one file format to another. For example, to store the FPGA data in configuration devices, you can convert the .sof data to another format, such as .pof, .hexout, .rbf, .rpd, or .jic, and then program the configuration device.

On the Quartus II main menu, click **File**, and then click **Convert Programming Files** to access the **Convert Programming Files** dialog box. You can then perform the following tasks:

- Configure multiple devices, such as combining multiple .sof files into one .pof.
- Configure multiple devices with an external host, such as a microprocessor or CPLD. For example, you can combine multiple .sof files into one configuration file.

 For more information about converting programming files with the Quartus II software, refer to the *Configuration File Formats* chapter of the *Configuration Handbook*.

You can use the **Advanced** option in the **Convert Programming Files** dialog box to debug your configuration. You must choose the advanced settings that apply to your Altera device. You can direct the Quartus II software to enable or disable an advanced option by turning the option on or off in the **Advanced Options** dialog box.


 When you change settings in the **Advanced Options** dialog box, the change affects .pof, .jic, .rpd, and .rbf files.

Table 18-4 lists the **Advanced Options** settings in more detail.

**Table 18-4. Advanced Options Settings**

Option Setting	Description
Disable EPCS ID check	<ul style="list-style-type: none"> <li>■ FPGA skips the EPCS silicon ID verification.</li> <li>■ Default setting is unavailable (EPCS ID check is enabled).</li> <li>■ Applies to the single- and multi-device AS configuration modes on all FPGA devices.</li> </ul>
Disable AS mode CONF_DONE error check	<ul style="list-style-type: none"> <li>■ FPGA skips the CONF_DONE error check.</li> <li>■ Default setting is unavailable (AS mode CONF_DONE error check is enabled).</li> <li>■ Applies to single- and multi-device (AS) configuration modes on all FPGA devices.</li> </ul>
Program Length Count adjustment	<ul style="list-style-type: none"> <li>■ Specifies the offset you can apply to the computed PLC of the entire bitstream.</li> <li>■ Default setting is 0. The value should be an integer.</li> <li>■ Applies to single- and multi-device (AS) configuration modes on all FPGA devices.</li> </ul>
Post-chain bitstream pad bytes	<ul style="list-style-type: none"> <li>■ Specifies the number of pad bytes appended to the end of an entire bitstream.</li> <li>■ Default value is set to 0 if the bitstream of the last device is uncompressed. Set to 2 if the bitstream of the last device is compressed.</li> </ul>
Post-device bitstream pad bytes	<ul style="list-style-type: none"> <li>■ Specifies the number of pad bytes appended to the end of the bitstream of a device.</li> <li>■ Default value is 0. No negative integer.</li> <li>■ Applies to all single-device configuration modes on all FPGA devices.</li> </ul>
Bitslice padding value	<ul style="list-style-type: none"> <li>■ Specifies the padding value used to prepare bitslice configuration bitstreams, such that all bitslice configuration chains simultaneously receive their final configuration data bit.</li> <li>■ Default value is 1. Valid setting is 0 or 1.</li> <li>■ Use only in 2, 4, and 8-bit PS configuration mode, when you use an EPC device with the decompression feature enabled.</li> <li>■ Applies to all FPGA devices that support enhanced configuration devices.</li> </ul>

Table 18-5 lists symptoms you may encounter if a configuration fails, and describes the advanced options you must use to debug your configuration.

**Table 18-5. Failure Symptoms and Options Settings (Part 1 of 2)**

Failure Symptoms	Disable EPCS ID Check	Disable AS Mode CONF_DONE Error Check	PLC Settings	Post-Chain Bitstream Pad Bytes	Post-Device Bitstream Pad Bytes	Bitslice Padding Value
Configuration failure occurs after a configuration cycle. Decompression feature is enabled. Encryption feature is enabled.	—	—	—	Use only for multi-device chain.	Use only for single-device chain.	—
CONF_DONE stays low after a configuration cycle.	—	—	Start with positive offset to the PLC settings.	Use only for multi-device chain.	Use only for single-device chain.	—

**Table 18-5. Failure Symptoms and Options Settings (Part 2 of 2)**

Failure Symptoms	Disable EPCS ID Check	Disable AS Mode CONF_DONE Error Check	PLC Settings	Post-Chain Bitstream Pad Bytes	Post-Device Bitstream Pad Bytes	Bitslice Padding Value
CONF_DONE goes high momentarily after a configuration cycle.	—	—	Start with negative offset to the PLC settings.	—	—	—
FPGA does not enter user mode even though CONF_DONE goes high.	—	—	—	Use only for multi-device chain.	Use only for single-device chain.	—
Configuration failure occurs at the beginning of a configuration cycle.	—	—	—	—	—	—
Newly introduced EPCS, such as EPCS128.	—	—	—	—	—	—
Failure in .pof generation for EPC device using Quartus II Convert Programming File Utility when the decompression feature is enabled.	—	—	—	—	—	—

- For more information about the **Convert Programming Files** dialog box, refer to *Convert Programming Files Dialog Box* in Quartus II Help.

## Flash Loaders

Parallel and serial configuration devices do not support the JTAG interface. However, you can use a flash loader to program configuration devices in-system via the JTAG interface. You can use an FPGA as a bridge between the JTAG interface and the configuration device. The Quartus II software supports parallel and serial flash loaders.

- For more information, refer to *About Flash Loaders* in Quartus II Help.

## Scripting Support

In addition to the Quartus II Programmer GUI, you can use the Quartus II command-line executable `quartus_pgm.exe` to access programmer functionality from the command line and from scripts. The programmer accepts `.pof`, `.sof`, and `.jic` programming or configuration files and Chain Description Files (`.cdf`).

[Example 18-1](#) shows a command that programs a device:

### Example 18-1. Programming a Device

```
quartus_pgm -c byteblasterII -m jtag -o bpv;design.pof ↵
```

Where:

- `-c byteblasterII` specifies the ByteBlaster™ II download cable
- `-m jtag` specifies the JTAG programming mode
- `-o bpv` represents the blank-check, program, and verify operations
- `design.pof` represents the `.pof` used for the programming

The Programmer automatically executes the erase operation before programming the device.

- ❓ For more information about scripting command options, refer to [About Quartus II Scripting](#) in Quartus II Help.

## The jtagconfig Debugging Tool

You can use the `jtagconfig` command-line utility (which is similar to the auto detect operation in the Quartus II Programmer) to check the devices in a JTAG chain and the user-defined devices.

For more information about the `jtagconfig` utility, type one of the following commands at the command prompt:

### Example 18-2.

```
jtagconfig -h ↵  
jtagconfig --help ↵
```

- 👉 The help switch does not reference the `-n` switch. The `jtagconfig -n` command shows each node for each jtag device.
- 🔗 For more information about command-line scripting, refer to the [Command-Line Scripting](#) chapter in volume 2 of the *Quartus II Handbook*.

## Conclusion


The Quartus II Programmer offers you a wide variety of options to program and configure your Altera devices. With the Quartus II Programmer, the Quartus II software provides you with a complete solution for your FPGA or CPLD design prototyping, which you can also use in the production environment.

## Document Revision History

Table 18-6 lists the revision history for this chapter.

**Table 18-6. Document Revision History**

Date	Version	Changes
November 2011	11.1.0	<ul style="list-style-type: none"><li>■ Updated “Configuration Modes” on page 18-5.</li><li>■ Added “Optional Programming or Configuration Files” on page 18-6.</li><li>■ Updated Table 18-2 on page 18-5.</li></ul>
May 2011	11.0.0	<ul style="list-style-type: none"><li>■ Added links to Quartus II Help.</li><li>■ Updated “Hardware Setup” on page 21-4 and “JTAG Chain Debugger Tool” on page 21-4.</li></ul>
December 2010	10.1.0	<ul style="list-style-type: none"><li>■ Changed to new document template.</li><li>■ Updated “JTAG Chain Debugger Example” on page 20-4.</li><li>■ Added links to Quartus II Help.</li><li>■ Reorganized chapter.</li></ul>
July 2010	10.0.0	<ul style="list-style-type: none"><li>■ Added links to Quartus II Help.</li><li>■ Deleted screen shots.</li></ul>
November 2009	9.1.0	No change to content.
March 2009	9.0.0	<ul style="list-style-type: none"><li>■ Added a row to Table 21-4.</li><li>■ Changed references from “JTAG Chain Debug” to “JTAG Chain Debugger”.</li><li>■ Updated figures.</li></ul>

 For previous versions of the *Quartus II Handbook*, refer to the [Quartus II Handbook Archive](#).

 Take an [online survey](#) to provide feedback about this handbook chapter.

