

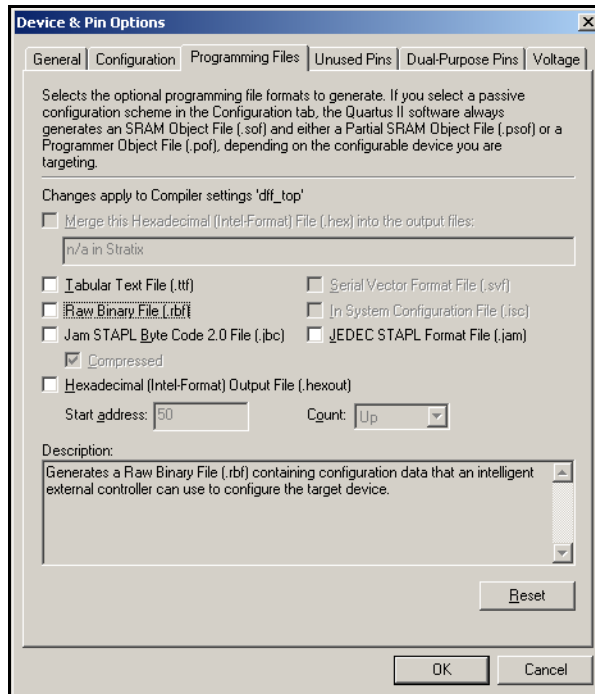
Introduction

Altera's Quartus® II and MAX+PLUS® II development tools can create one or more configuration and programming files to support the configuration schemes discussed in Volume I. When you compile a design in the Quartus II and MAX+PLUS II software for a device that has programming file support, the software will automatically generate a SRAM Object File (.sof) and a Programmer Object File (.pof) for a configuration device.

Generating Configuration Files

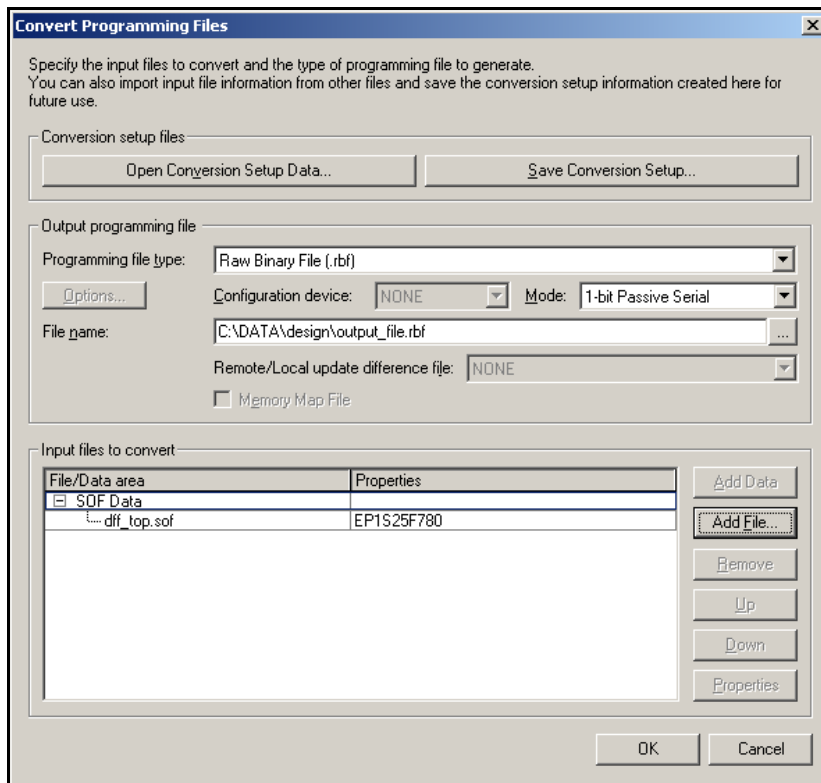
To instruct Quartus II to generate other configuration file formats during compilation, go to **Programming Files** tab of the **Device & Pin Options** dialog box (see [Figure 7-1](#)).

Figure 7-1. Programming Files Dialog Box



You can also convert SOF and POF files through the **Convert Programming Files** window (File menu). **Figure 7-2** shows an example of the **Convert Programming Files** dialog box set-up to convert an SOF to a Raw Binary File (.rbf).

Figure 7-2. Convert Programming Files Dialog Box



When performing multi-device configuration using a configuration device, you must generate the configuration device's POF from each project's SOF. You can combine multiple SOFs using the Convert Programming Files dialog box in the Quartus II software. The following steps explain how to combine multiple SOF files into a POF file(s).

1. Choose **Convert Programming Files...** command (File menu).
2. In the **Programming file type** list, choose **Programming Object File (.pof)**.

3. In the **Configuration device** list, choose the appropriate configuration device.
4. In the **Mode** list, choose the appropriate configuration scheme.
5. You can set configuration devices options by selecting the **Options...** radio button.
6. Specify the name of the output file in the **File name** box.
7. In the **Input files to convert** box, click on **SOF Data**, so that the **Add File...** button becomes active.
8. Click on the **Add File...** button and select the SOF file to be converted. This step can be repeated to combine multiple SOF files into a POF file(s). The order of the SOF files should match the order of the devices in the chain.
9. Click **OK**.
10. When generating multiple POFs for EPC2 or EPC1 devices, the first device's POF file name will be as specified, while the second device's POF file name will have a "_1" extension (e.g., **top_1.pof**)

When performing multi-device configuration using an external host, such as a microprocessor or CPLD, you should generate one combined configuration file from each project's SOF. You can combine multiple SOFs using the **Convert Programming Files** dialog box in the Quartus II software. The following steps explain how to combine multiple SOF files into one configuration file.

1. Choose the **Convert Programming Files...** command (File menu).
2. In the **Programming file** type list, choose the appropriate file format (Hexadecimal (Intel-Format) Output File for SRAM (**.hexout**), RBF, or Tabular Text File (**.tff**)).
3. In the **Mode** list, choose the appropriate configuration scheme.
4. Specify the name of the output file in the **File name** box.
5. In the **Input files to convert** box, click on **SOF Data**, so that the **Add File...** button becomes active.

6. Click on the **Add File...** button and select the SOF file to be converted. This step can be repeated to combine multiple SOF files into one configuration file. The order of the SOF files (from top to bottom) should match the order of the devices in the chain.
7. Click **OK**.

The following steps explain how to convert a SOF for ACEX® 1K, FLEX® 10K or FLEX 6000 devices using the MAX+PLUS II software.

1. In the MAX+PLUS II Compiler or Programmer, choose the **Convert SRAM Object Files** command (File menu.)
2. In the **Convert SRAM Object Files** dialog box, click on the **Select Programming File...** radio button to specify which SOF file to convert. This step can be repeated to combine multiple SOF files into one configuration file. The order of the SOF files (from top to bottom) should match the order of the devices in the chain.
3. Specify the name of the output file in the **File Name** box.
4. Choose the appropriate configuration file format through the **File Format** pull-down list.
5. Click **OK**.

The following sections give a description of the supported configuration file formats.

SRAM Object File (.sof)

You should use a SOF during PS configuration when the configuration data is downloaded directly to the FPGA using the Quartus II or MAX+PLUS II software with a USB Blaster, MasterBlaster™, ByteBlaster™ II, EthernetBlaster™ or ByteBlasterMV™ cable. The Quartus II and MAX+PLUS II compiler automatically generates the SOF for your design. When using a SOF, the Quartus II or MAX+PLUS II software controls the configuration sequence and automatically inserts the appropriate headers into the configuration data stream. All other configuration files are created from the SOF.

Programmer Object File (.pof)

A POF is used by the Altera® programming hardware to program a configuration device. The Quartus II and MAX+PLUS II compiler automatically generate a POF for your design. For smaller devices (e.g., EPF10K20 devices), multiple SOFs can fit into one configuration device; for larger devices (e.g., APEX 20K devices), multiple configuration devices may be required to hold the configuration data.

Raw Binary File (.rbf)

The RBF is a binary file containing the configuration data. The RBF does not contain byte separators (e.g. commas or carriage returns); it is literally a raw binary file that contains a binary bitstream of configuration data. For example, one byte of RBF data is 8 configured bits 10000101 (85 Hex). Data must be stored so that the least significant bit (LSB) of each data byte is loaded first. The converted image can be stored on a mass storage device. The microprocessor can then read data from the binary file and load it into the FPGA. You can also use the microprocessor to perform real-time conversion during configuration. In the PS configuration schemes, each byte of data should be sent with LSB first. In the FPP, PPS, and PPA configuration schemes, the target device receives its information in parallel from the data bus, a data port on the microprocessor, or some other byte-wide channel.



For more information on creating RBFs, search for “RBF” in Quartus II or MAX+PLUS II Help.

Raw Programming Data File (.rpd)

The RPD File is a binary file containing a binary bitstream of Cyclone® configuration data. This file is stored in the serial configuration devices in an embedded environment outside the Quartus II software. The Cyclone FPGA can then be configured by using the Active Serial (AS) configuration scheme where the Cyclone FPGA loads the RPD file stored in the serial configuration device. The RPD file size is equal to the memory size of the targeted serial configuration device. A RPD file can only be generated from a POF in the **Convert Programming Files** dialog box (File menu).

The RPD file is different from the RBF file, even for a single device configuration file. In multi-device chains, the RPD file is not the concatenation of the corresponding RBF files. The LSB of each byte in the RPD file should be written to the serial configuration device first.



For more information on creating RPDs, search for “RPD” in Quartus II Help or refer to the *SRunner: An embedded Solution for Serial Configuration Device Programming White Paper*.

Hexadecimal (Intel-Format) File (.hex) or (.hexout)

A HEX File is an ASCII file in the Intel HEX format. Microprocessors or external hosts can use the HEX file to store and transmit configuration data using the configuration schemes supported by microprocessors. This file can also be used by third-party programmers to program Altera’s configuration devices.



For more information on creating Hex Files, search for “Hex File” in Quartus II or MAX+PLUS II Help.

Tabular Text File (.tff)

The TTF is a tabular ASCII file that provides a comma-separated version of the configuration data for the FPP, PPS, PPA, and bit-wide PS configuration schemes. In some applications, the storage device containing the configuration data is neither dedicated to nor connected directly to the target device. For example, a configuration device can also contain executable code for a system (e.g., BIOS routines) and other data. The TTF allows you to include the configuration data as part of the microprocessor's source code using the include or source commands. The microprocessor can access this data from a configuration device or mass-storage device and load it into the target device. A TTF can be imported into nearly any assembly language or high-level language compiler.



For more information on creating TTFs, search for “TTF” in Quartus II or MAX+PLUS II Help.

Serial Bitstream File (.sbf)

An SBF is used in PS schemes to configure FLEX 10K and FLEX 6000 devices in-system with the BitBlaster™ cable.



The BitBlaster is obsolete. SBFs are supported by the MAX+PLUS II software only.



For more information on creating SBFs, search for “SBF” in MAX+PLUS II Help.

Jam File (.jam)

A Jam™ File is an ASCII text file in the Jam device programming language that stores device programming information. These files are used to program, verify, and blank-check one or more devices in the Quartus II or MAX+PLUS II Programmer or in an embedded processor environment.



For more information on creating Jam Files, search for “Jam” in Quartus II or MAX+PLUS II Help.

Jam Byte-Code File (.jbc)

A JBC File is a binary file of a Jam File in a byte-code representation. JBC files store device programming information used to program, verify, and blank-check one or more devices in the Quartus II or MAX+PLUS II Programmer or in an embedded processor environment.



For more information on creating JBC Files, search for “JBC” in Quartus II or MAX+PLUS II Help.

Document Revision History

Table 7–1 shows the revision history for this document.

Date & Document Version	Changes Made	Summary of Changes
April 2007 v2.2	Added document revision history.	
August 2005 v2.1	Removed active cross references referring to document outside Chapter 7.	
July 2004 v2.0	Added paragraph regarding difference of <code>.rpd</code> from <code>.rbf</code> in the “Raw Programming Data File (.rpd)” section.	
September 2003 v1.0	Initial Release.	

