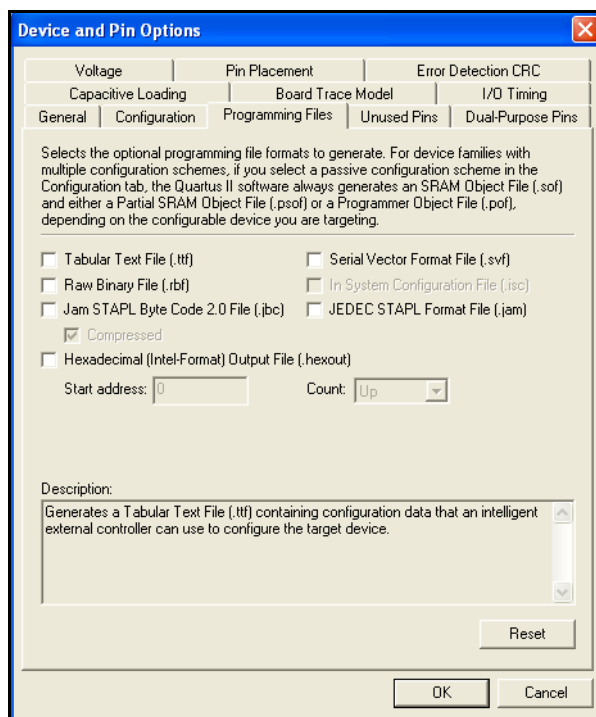


Altera's Quartus® II and MAX+PLUS® II development tools can create one or more configuration and programming files to support the configuration schemes discussed in Volume I. When you compile a design in the Quartus II and MAX+PLUS II software for a device that has programming file support, the software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

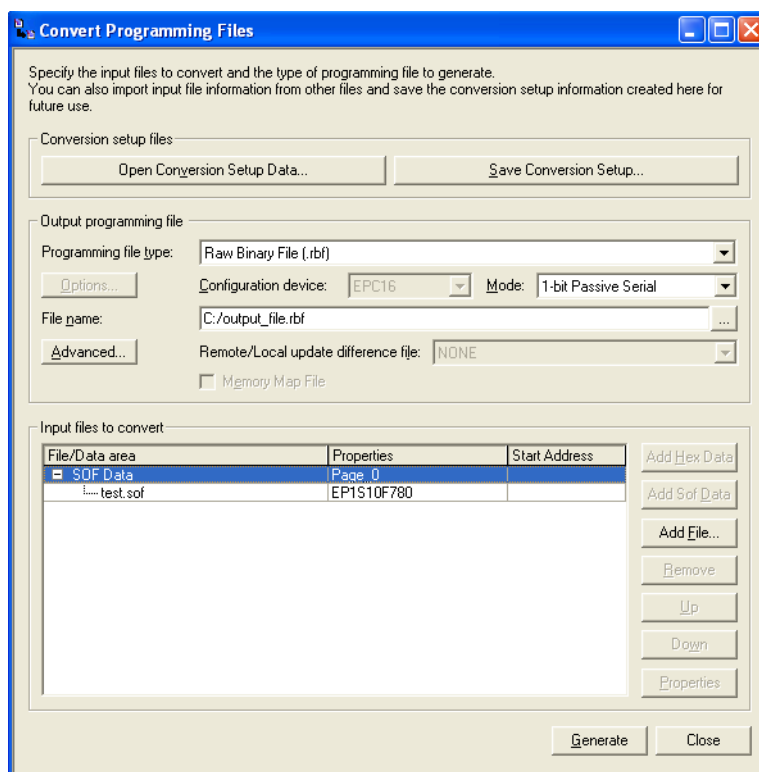
## Generating Configuration Files

To instruct Quartus II to generate other configuration file formats during compilation, go to **Programming Files** tab of the **Device & Pin Options** dialog box (refer to [Figure 6-1](#)).

**Figure 6-1.** Programming Files Dialog Box



You can also convert .sof and .pof through the **Convert Programming Files** window (File menu). [Figure 6-2](#) shows an example of the **Convert Programming Files** dialog box set-up to convert .sof to a Raw Binary File (.rbf).

**Figure 6–2.** 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 **.sof** using the Convert Programming Files dialog box in the Quartus II software. The following steps explain how to combine multiple **.sof** into a **.pof**.

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...** 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** to be converted. This step can be repeated to combine multiple **.sof** into a single **.pof**. The order of the **.sof** should match the order of the devices in the chain.

9. Click **OK**.
10. When generating multiple **.pof** for EPC1 or EPC2 devices, the first device's **.pof** name will be as specified, while the second device's **.pof** 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 **.sof** using the **Convert Programming Files** dialog box in the Quartus II software. The following steps explain how to combine multiple **.sof** 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** to be converted. This step can be repeated to combine multiple **.sof** into one configuration file. The order of the **.sof** (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...** button to specify which **.sof** to convert. This step can be repeated to combine multiple **.sof** into one configuration file. The order of the **.sof** (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 **.sof** 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 about creating **.rbf**, search for “RBF” in Quartus II or MAX+PLUS II Help.

## Raw Programming Data File (.rpd)


The **.rpd** file is a binary file for EPCS devices containing a binary bitstream of configuration data for FPGAs that support Active Serial configuration. This file is stored in the serial configuration devices in an embedded environment outside the Quartus II software. The FPGA can then be configured with the Active Serial (AS) configuration scheme where the 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**, even for a single device configuration file. In multi-device chains, the **.rpd** file is not the concatenation of the corresponding **.rbf**. The LSB of each byte in the **.rpd** file should be written to the serial configuration device first.

-  For more information about creating **.rpd** files, search for “RPD” in Quartus II Help or refer to the *AN 418: SRunner: An embedded Solution for Serial Configuration Device Programming*.


## 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 about creating **.hex** files, search for “Hex File” in Quartus II or MAX+PLUS II Help.

## Tabular Text File (.tff)


The **.tff** 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 **.tff** 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 **.tff** can be imported into nearly any assembly language or high-level language compiler.

-  For more information about creating **.tff**, search for “TTF” in Quartus II or MAX+PLUS II Help.

## Serial Bitstream File (.sbf)


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

-  The BitBlaster is obsolete. **.sbf** is supported by the MAX+PLUS II software only.

-  For more information about creating **.sbf**, 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 about 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 about creating **.jbc** files, search for “JBC” in Quartus II or MAX+PLUS II Help.

## Chapter Revision History

Table 6–1 lists the revision history for this chapter.

**Table 6–1.** Chapter Revision History

Date	Version	Changes Made
December 2009	2.4	<ul style="list-style-type: none"> <li>Updated Figure 6–1 and Figure 6–1.</li> <li>Removed “Referenced Documents” section.</li> </ul>
October 2008	2.3	<ul style="list-style-type: none"> <li>Updated “Raw Programming Data File (.rpd)” section.</li> <li>Added “Referenced Documents” section.</li> <li>Updated new document format.</li> </ul>
April 2007	2.2	<ul style="list-style-type: none"> <li>Added document revision history.</li> </ul>
August 2005	2.1	<ul style="list-style-type: none"> <li>Removed active cross references referring to document outside Chapter 7.</li> </ul>
July 2004	2.0	<ul style="list-style-type: none"> <li>Added paragraph regarding difference of <b>.rpd</b> from <b>.rbf</b> in the “Raw Programming Data File (.rpd)” section.</li> </ul>
September 2003	1.0	<ul style="list-style-type: none"> <li>Initial Release.</li> </ul>