Qsys Pro Command Line Usage

File Structures of Qsys Pro

Here is the hierarchy of the files generated from Qsys Pro:

- <Project Directory>
  - <your_ip>.ip – Top level IP variation file
  - <your_ip> - IP core variation files
    - <your_ip>.qgsimc - Simulation caching file
    - <your_ip>.qgsynthc - Synthesis caching file
  - sim - IP simulation files
    - <your_ip>.v or vhd - Top-level simulation file
    - <simulator vendor> - Simulator setup scripts
      - <simulator_setup_scripts>
  - synth - IP synthesis files
    - <your_ip>.v or .vhd - Top-level IP synthesis file
  - <IP Submodule>_<version> - IP Submodule Library
    - Sim - IP submodule 1 simulation files
      - <HDL files>
    - Synth - IP submodule 1 synthesis files
      - <HDL files>
  - <your_ip>_tb - IP testbench system
    - <your_testbench>_tb.qsys - testbench system file
    - <your_ip>_tb - IP testbench files
      - your_testbench> _tb.csv or .spd - testbench file
      - sim- IP testbench simulation files

File Name	Description
<your_ip>.ip	Top-level IP variation file that contains the parameterization of an IP core in your project. If the IP variation is part of a Qsys Pro system, the parameter editor also generates a .qsys file.
<your_ip>.qgsimc	Simulation caching file that compares the .qsys and .ip files with the current parameterization of the Qsys Pro system and IP core. This comparison determines if Qsys Pro can skip regeneration of the HDL.

<your_ip>.qgsynth	Synthesis caching file that compares the .qsys and .ip files with the current parameterization of the Qsys Pro system and IP core. This comparison determines if Qsys Pro can skip regeneration of the HDL.
<your_ip>.v <your_ip>.vhd	HDL files that instantiate each submodule or child IP core for synthesis or simulation.
/submodules	Contains HDL files for the IP core submodule.
<IP submodule>/	For each generated IP submodule directory, Qsys Pro generates /synth and /sim sub-directories.

IP Core Generation Qsys Pro

Use the qsys-script and qsys-generate utilities to define and generate an IP core variation outside of the Quartus Prime GUI.

To execute a TCL script that instantiates the IP and sets desired parameters:

- qsys-script –script=<script_file>.tcl

Option	**Usage**	**Description**
--cmd=<value>	Optional	A string containing Tcl scripting commands for creating or manipulating Qsys sytems. If a Tcl is specified, commands in the script will be run before the contents of the script file.
--script=>file>	Optional	A file contain Tcl script commands for creating or manipulating Qsys systems. If Tcl command string is specified, commands strings will be run before the commands in the script file.
--package-version=<value>	Optional	Specify which version of the system scripting Tcl API to use. This version determines the availability and behavior of the Tcl commands. The minimum supported version is 12.0. If not specified, the Tcl script must request the system scripting API directly using the “package require -exact qsys <version>” command in the script.

--system-file=<file>	Optional	Specifies the path to a .qsys system file. This system will be loaded before running any scripting commands.
--search-path=<value>	Optional	If omitted, a standard default path will be used. If provided, a comma-separated list of paths will be searched. To include the standard path in your replacement, use "$", like "/extra/dir,$".
--jvm-max-heap-size=<value>	Optional	The maximum memory size to be used for allocations when running this tools. This value is specified as <size><unit> where unit can be m (or M) for multiples of megabyutes or g (or G) for multiples of gigabytes. The default value is 512m.
--help	Optional	Display help for this tool.

To generate the IP core variation:

- qsys-generate <IP variation file>.qsys

Options:

Option	**Usage**	**Description**
<1st arg file>	Required	Specifies the name of the .qsys system file to generate.
--synthesis=<VERILOG|VHDL>	Optional	Creates synthesis HDL files that Qsys uses to compile the system in a Quartus Prime project. Specify the preferred generation language for the top-level RTL file for the generated Qsys system. The default value is VERILOG.
--block-symbol-file	Optional	Creates a Block Symbol File (.bsf) for the Qsys system.
--greybox	Optional	If you are synthesizing your design with a third-party EDA synthesis tool, generate a netlist for the synthesis tool to estimate timing and resource usage for this design.
--ipxact	Optional	If you set this option to true, Qsys renders the post generation system as an IPXACT-compatible component description.
--simulation=<VERILOG|VHDL>	Optional	Creates a simulation model for the Qsys system. The simulation model contains generated HDL files for the simulator, and may include simulation-only features. Specify the preferred simulation language. The default value is VERILOG.
--testbench=<SIMPLE|STANDARD>	Optional	Creates a testbench system that instantiates the original system, adding bus functional models (BFMs) to drive the top-level interfaces. When you generate the system, the

		BFMs interact with the system in the simulator. The default value is STANDARD.
--testbenchsimulation =<VERILOG|VHDL>	Optional	After you create the testbench system, create a simulation model for the testbench system. The default value is VERILOG.
--example-design=<value>	Optional	Creates example design files. For example, --example-design or --example-design=all. The default is All, which generates example designs for all instances. Alternatively, choose specific filesets based on instance name and fileset name. For example --exampledesign=instance0.example_design1, instance1.example_design 2. Specify an output directory for the example design files creation.
--output-directory=<value>	Optional	Sets the output directory. Qsys creates each generation target in a sub-directory of the output directory. If you do not specify the output directory, Qsys uses a sub-directory of the current working directory matching the name of the system.
--search-path=<value>	Optional	If you omit this command, Qsys uses a standard default path. If you provide this command, Qsys searches a comma-separated list of paths. To include the standard path in your replacement, use "$", for example, "/extra/dir,$".
--family=<value>	Optional	Sets the device family name.
--part=<value>	Optional	Sets the device part number. If set, this option overrides the --family option.
--upgrade-variation-file	Optional	If you set this option to true, the file argument for this command accepts a .v file, which contains a IP variant. This file parameterizes a corresponding instance in a Qsys system of the same name.
--upgrade-ip-cores	Optional	Enables upgrading all the IP cores that support upgrade in the Qsys system.
--clear-output-directory	Optional	Clears the output directory corresponding to the selected target, that is, simulation or synthesis.
--export-qsys-script	Optional	If you set this option to true, Qsys exports the postgeneration system as a Qsys script file with the extension .tcl.
--jvm-max-heap-size=<value>	Optional	The maximum memory size that Qsys uses when running qsys-generate. You specify the value as , where unit is m (or M) for multiples of megabytes or g (or G) for multiples of gigabytes. The default value is 512m.
--help	Optional	Displays help for --qsys-generate

Creating a Tcl script from a .qsys file

You will use the system script File:Save script.tcl to save a Qsys system to a Tcl script. Save this script to your working directory with your .qsys file. Make sure the script is named "save_script.tcl" when you save it.

To create a Tcl script based on a system:

- qsys-script --cmd="set system_name <*systemname*>" --script=save_script.tcl --system-file=<*system*>.qsys

To create the .qsys file from the saved script:

- qsys-script --script=*my_system*.tcl