export (ProASIC3/E)

Saves your design to a file in the bts_stp file format.

```
export -format bts_stp [-feature value][-secured_device value][-signature
value][-pass_key value][-aes_key value][-from_config_file value][-
number_of_devices value][-from_progfile_type value][-target_programmer value]
[-custom_security value][-fpga_security_level value][-from_security_level
value][-security_permanent value]{filename}
```

Arguments

`-feature value`
Select the silicon feature(s) you want to program. The acceptable values for this option are all, setup_security, prog_from, or prog_fpga.

`-secured_device value`
Specifies whether the device you are programming is secured. The acceptable values for this option are yes or no.

`-signature value`
Optional for afm, dio, and fus file types.

`-pass_key value`
Protects all the security settings for FPGA Array and FlashROM. the maximum length of this value is 32 characters. You must use hexadecimal characters for the pass key value.

`-aes_key value`
Decrypts FPGA Array and/or FlashROM programming file content.
Max length is 32 HEX characters.

`-from_config_file value`
Specifies the location of the FlashROM configuration file.

`-number_of_devices value`
Specifies the number of devices you want to program.

`-from_progfile_type value`
The following table shows the acceptable values for this option:

Value	Description
single	Generates one programming file with all the generated incremental value(s) in the external source file
multiple	Generates one individual programming file for each generated incremental value(s) in the external source file

`-target_programmer value`
The following table shows the acceptable values for this option:

Value	Description
specific	Silicon Sculptor, BP Auto Programmer, or FlashPro
generic	Generic STAPL player

`-custom_security value`

The following table shows the acceptable values for this option:

Value	Description
yes	Custom security level
no	Standard security level

`-fpga_security_level` *value*

The following table shows the acceptable values for this option:

Value	Description
write_verify_protect	The security level is considered medium (standard). No one will be able to write to or verify this file. (custom FPGA).
write_protect	The security level is considered write protected. No one will be able to write to this file, but they will be able to verify it (custom FPGA).
encrypted	The security level is considered high (standard). It uses 128-bit AES encryption (custom FPGA).

`-from_security_level` *value*

The following table shows the acceptable values for this option:

Value	Description
write_verify_protect	The security level is considered medium (standard). No one will be able to read, write to, or verify this file (custom FlashROM).
write_protect	The security level is considered write protected. No one will be able to write to this file, but they will be able to read and verify it (custom FPGA).
encrypted	The security level is considered high (standard). It uses 128-bit AES encryption (custom FlashROM).
none	The file has no security.

`-security_permanent` *value*

Specifies whether the security settings for this file are permanent or not. The following table shows the acceptable values values for this option:

Value	Description
yes	Permanently disable future modification of security settings for FPGA Array and FlashROM
no	Enable future modifications for FPGA Array and FlashROM

{*filename*}

Specifies the path and name of the file you are exporting.

Supported Families

ProASIC3/E

Notes

- None

Exceptions

- None

Example

```
export -format "bts_stp"
   -feature "all"
   -secured_device “no”
   -signature "123"
   -pass_key "FB318707864EC889AE2ED8904B8EB30D"
   -custom_security "no"
   -fpga_security_level "write_verify_protect"
   -from_security_level "write_verify_protect"
   -from_config_file {.\g3_test\from.ufc}
   -number_of_devices "1"
   -from_progfile_type "single"
   -target_programmer "specific" \
 {.\flp4.stp}
```

See Also

export (ProASIC^PLUS, Axcelerator, ProASIC, MX, eX, and SX/SX-A)
Exporting files
Importing files
Tcl documentation conventions