Contents

[Operator Instruction 2](#_Toc501372498)

[Perquisites 2](#_Toc501372499)

[Steps to launch the script 2](#_Toc501372500)

[Argument Descriptions 2](#_Toc501372501)

[Syntax Examples 2](#_Toc501372502)

[Developer 3](#_Toc501372503)

[Theory 3](#_Toc501372504)

[Main Definition 3](#_Toc501372505)

[Upgrade Class 3](#_Toc501372506)

[Logging 3](#_Toc501372507)

[Errata 4](#_Toc501372508)

[Change Log 5](#_Toc501372509)

# Operator Instruction

This section of the documentation provides instructions on how to launch the script and the options the user has access to. For information on editing functionality or fixing bugs please see the developer section of the instructions.

## Perquisites

The user first needs to open an interactive qconsole before launching the script. A minimum of 8GBs of memory is recommended. If the user is upgrading larger projects for Stratix or Arria more memory may be required. **NOTE:** If Quartus pro is used only pro users will be able to open the project.

### Steps to launch the script

1. arc shell python\_altera/2.7.3
2. arc shell acds/<Quartus Version> (or whichever version is preferred \*\*\*see the note above\*\*\*)
3. Navigate to the directory storing the python script.
4. python dustinRewrite.py --<argument>=<string>

**NOTE:** Steps 1 and 2 only needed to be completed before the initial launch of the script. Steps 3 and 4 can be repeated as many times as needed after first time steps 1 and 2 are completed.

## Argument Descriptions

--single\_upgrade

The input for this argument is a string. The string needs to contain the path to the par file that needs to be upgraded. Only one par file can is allowed to be contained by this directory.   
Example: --single\_upgrade=/path/to/dir

## Syntax Examples

*python* dustinRewrite*.py --single\_upgrade=/data/dustinhe/autoUpgrade/project*

-will update the project in the folder specified

# Developer

## Theory

The Quartus projects are stored over multiple source and settings files (qpf, qsf, sdc, qip, ect). The majority of the settings files use the tcl syntax and reference the design files used in the project. The script takes advantage of this structure and uses it to both build a file list of all the source and settings files in addition to upgrading any IP used in the design.

The script starts with the Quartus settings file (.qsf). This file contains all the settings in addition to any source, qip, and qsys files used by the project. It uses a list of tags to find all relevant files. These tags are stored in the initial of the upgrade class initial definition. From this file the script then moves on the any Quartus IP (.qip) or Platform Designer files (.qsys). The script will pars the qip files for any design files using the same tags for the qsf file.

After the qsf file and all qip files are parsed the script will upgrade an IP cores.

## Main Definition

## Upgrade Class

## Logging

# Errata

# Change Log

This log is used to tack major changes in the document and who completed the changes.

|  |  |  |
| --- | --- | --- |
| **Name** | **Date** | **Notes** |
| Unknown | ??? | Initial Creation. |
| D. Henderson | 12-1-17 | Ported documentation to rewrite of the script. |
| D. Henderson | 12-18-17 | Added developer and errata sections to aid in fixing bugs. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |