**Generic SCons**

**QuickStartGuide**

|  |  |
| --- | --- |
| Document Control: | |
| Project: | ETK: SCT\_Sconstools |
| Revision: | 0.2 |
| Last Change: | 28-Jul-2015 |
| Confidence Level: | 🞏 Public 🗷 Confidential |

CONFIDENTIAL AND PROPRIETARY PROPERTY OF ETK - ALL RIGHTS RESERVED

|  |  |
| --- | --- |
| Document State: | |
| State: | Draft |
| Author: | Hendra Sasmito |
| Reviewed by: | André Fischer |
| Released by: | 31-Jul-2015 |

Revision History

| Version | Date | Change Description |
| --- | --- | --- |
| 0.2 | 28-Jul-2015 | Update outdated information on GenericScons  Change doc to Continental corporate layout |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table of Contents

[Quick Start Guide : GenericScons Build Environment 4](#_Toc426034566)

[1 Folder structure and sharing concept of SCons 4](#_Toc426034567)

[2 Structure 4](#_Toc426034568)

[2.1 Folder Structure of Workspace 4](#_Toc426034569)

[2.2 Configuration Templates 6](#_Toc426034570)

[2.3 Required Tools 7](#_Toc426034571)

[2.4 Build results and their location 8](#_Toc426034572)

[3 How to build with SCons 11](#_Toc426034573)

[3.1 Build commands 11](#_Toc426034574)

[3.2 Help menu 12](#_Toc426034575)

[3.3 General information about the SCons build 13](#_Toc426034576)

[4 Steps of implementing GenericSCons build environment for a new component 14](#_Toc426034577)

[5 Checking for errors in Scons 14](#_Toc426034578)

# 

Quick Start Guide : GenericScons Build Environment

# Folder structure and sharing concept of SCons

The GenericScons build environment contains:

1. Windows batch file (scons.bat which starts scons)
2. Python scripts

The python scripts consist of:

1. SConstruct
2. SConscript.py files and .py scripts
3. .scfg files (not shared, contain configuration/setting for SConscript.py or list of source files and include paths required to build target).

Only .scfg files, scons.bat and SConscript.py in 03\_Workspace\algo\xxx\_sim folder need to be created in the project, because other **SConscript files are copied from 02\_Development\_Tools\scons\_tools**. When there is a new feature added, a new 02\_Development\_Tools\scons\_tools checkpoint is created and the new SConscript files are automatically copied to the project when scons is executed.

In special case, it is possible to use custom SConscript.py when it is needed instead of using shared SConscript but it should be avoided since it makes the maintenance more difficult when there is a new feature/improvement needed. It is preferred to create a new shared SConscript when needed and added it to the current shared SConscripts.

# Structure

## Folder Structure of Workspace

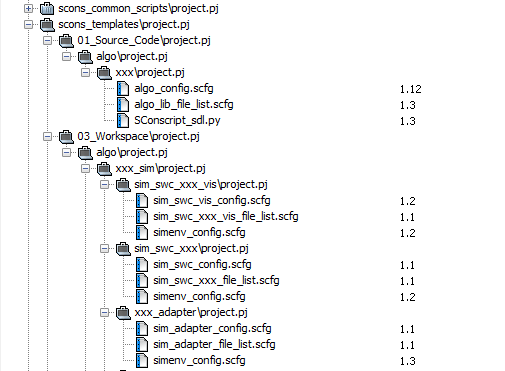
**The following folder structure is required in order to use the GenericScons build, which follows the CM plan (xxx is the component name).**

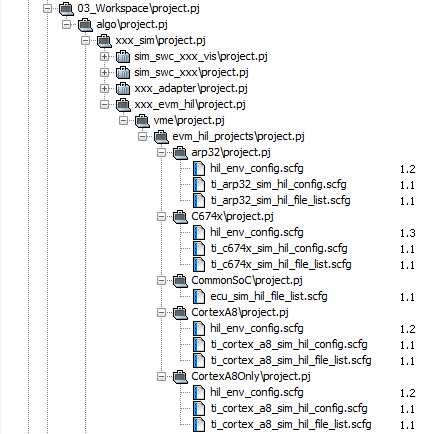
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **04\_Engineering\** | | |  |  |  |  |  |  |  |
|  | **01\_Source\_Code\** | | |  |  |  |  |  |  |
|  |  | algo\ |  |  |  |  |  |  |  |
|  |  |  | xxx\ |  |  |  |  |  |  |
|  | **03\_Workspace\** | | |  |  |  |  |  |  |
|  |  | algo\ |  |  |  |  |  |  |  |
|  |  |  | externals\ | |  |  |  |  |  |
|  |  |  | xxx\ |  |  |  |  |  |  |
|  |  |  |  | ti\_dsp\_c674x\ | | |  |  |  |
|  |  |  |  | ti\_dsp\_c66xx\ | | |  |  |  |
|  |  |  |  | ti\_eve\_arp32\ | | |  |  |  |
|  |  |  |  | ti\_arm\_cortex\_a8\ | | |  |  |  |
|  |  |  |  | ti\_arm\_cortex\_a15\ | | | |  |  |
|  |  |  |  | ti\_arm\_cortex\_m3\ | | | |  |  |
|  |  |  |  | ti\_arm\_cortex\_m4\ | | | |  |  |
|  |  |  | xxx\_sim\ | |  |  |  |  |  |
|  |  |  |  | xxx\_adapter\ | | |  |  |  |
|  |  |  |  | sim\_swc\_xxx\ | | |  |  |  |
|  |  |  |  | sim\_swc\_xxx\_vis\ | | |  |  |  |
|  |  |  |  | xxx\_evm\_hil\ | | |  |  |  |
|  |  |  |  |  | vme\ |  |  |  |  |
|  |  |  |  |  |  | evm\_hil\_projects\ | | |  |
|  |  |  |  |  |  |  | arp32\ | |  |
|  |  |  |  |  |  |  | c674x\ | |  |
|  |  |  |  |  |  |  | CommonSOC\ | | |
|  |  |  |  |  |  |  | CortexA8\ | |  |
|  |  |  |  |  |  |  | CortexA8Only\ | | |
|  |  |  |  |  |  |  | CortexM3 | |  |
|  |  |  |  |  |  | generic\ | |  |  |
|  |  |  |  |  |  |  | evm\_sw\ | |  |
|  |  |  |  |  |  |  |  | hil\_server\ | |
|  |  |  |  |  | vh28\ |  |  |  |  |
|  |  |  |  |  |  | evm\_hil\_projects\ | | |  |
|  |  |  |  |  |  |  | arp32\ | |  |
|  |  |  |  |  |  |  | C66xx\ | |  |
|  |  |  |  |  |  |  | CommonSOC\ | | |
|  |  |  |  |  |  |  | CortexM4\ | |  |
|  |  |  |  |  |  |  | CortexA15\ | |  |
|  |  |  |  |  |  | generic\ | |  |  |
|  |  |  |  |  |  |  | evm\_sw\ | |  |
|  |  |  |  |  |  |  |  | hil\_server\ | |
| **05\_Testing\** | |  |  |  |  |  |  |  |  |
|  | **05\_Test\_Environment\** | | | |  |  |  |  |  |
|  |  | algo\ |  |  |  |  |  |  |  |
|  |  |  | modtests\ | |  |  |  |  |  |
|  |  |  |  | cantata\_tests\ | | |  |  |  |
|  |  |  |  |  | xxx\ |  |  |  |  |
|  |  |  |  | qac\_tests\ | |  |  |  |  |
|  |  |  |  |  | common\ | |  |  |  |
|  |  |  |  |  | xxx\ |  |  |  |  |
|  |  |  |  | qacpp\_tests | |  |  |  |  |
|  |  |  |  |  | common\ | |  |  |  |
|  |  |  |  |  | xxx\ |  |  |  |  |

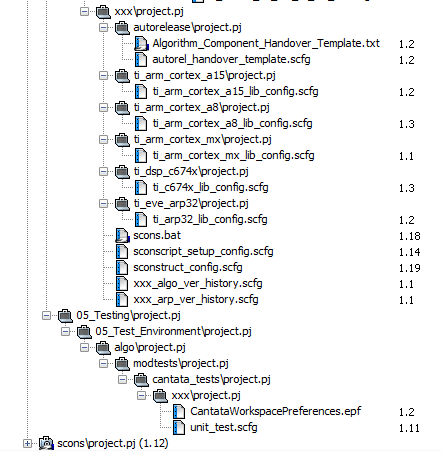
All the SCons files (SConscript, scfg files) can be found in 04\_Engineering\01\_Source\_Code, 04\_Engineering\03\_Workspace and 05\_Testing\05\_Test\_Environment. 04\_Build is used to store the build result. All build results which are checked in to MKS are stored in 05\_Deliverables.

## Configuration Templates

The template for scfg files and non-shared SConscript can be found in 02\_Development\_Tools\scons\_tools\scons\_templates. The template files are arranged according to the folder structure of the algo project.







## Required Tools

The following tools should be available in 02\_Development\_Tools depending on the available build targets:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **02\_Development\_Tools\** | | | | |  |  |
|  |  | doxygen\ | | |  |  |  |
|  |  | movpy\ | | |  |  |  |
|  |  | pdo\_tool\ | | |  |  |  |
|  |  | scons\_tools\ | | | |  |  |
|  |  | sdlcompiler\ | | | |  |  |
|  |  | ti\_tools\ | | |  |  |  |
|  |  |  | *bios\* | |  |  |  |
|  |  |  | *c6x\_simulator\* | | | |  |
|  |  |  | *edma3\_lld\* | | |  |  |
|  |  |  | *evestarterware\* | | | |  |
|  |  |  | *networking\* | | | |  |
|  |  |  | *pdk\* | |  |  |  |
|  |  |  | *starterware\* | | | |  |
|  |  |  | *unzip\_tools\* | | | |  |
|  |  |  | *xdctools\* | | |  |  |
|  |  | visual\_studio\_scripts\ | |  | | | |

Before implementing GenericScons, please make sure to have all required tools to avoid error because some tools are missing. Because shared-SConscripts will be copied from scons\_tools, it is important to have the correct scons\_tools checkpoint.

## Build results and their location

SCons produces static library (.lib), dynamic library (.dll), out file, header file containing algo checksum, SDL file, doxygen documentation and IDE project (Visual C++ project and CCS project).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **04\_Engineering\** | | |  |  |  |  |  |  |  |
|  | **03\_Workspace\** | | |  |  |  |  |  |  |
|  |  | algo\ |  |  |  |  |  |  |  |
|  |  |  | xxx\_sim\ | |  |  |  |  |  |
|  |  |  |  | sim\_swc\_xxx\ | | |  |  |  |
|  |  |  |  |  | doxygen\ | |  |  |  |
|  |  |  |  |  |  | html\ |  | --> Doxygen documentation | |
|  | **04\_Build\** | |  |  |  |  |  |  |  |
|  |  | algo\ |  |  |  |  |  |  |  |
|  |  |  | xxx\ |  |  |  |  |  |  |
|  |  |  |  | pc\ |  |  |  | --> pc algo library, sdl for SIM under <variant>\01\_Source\_Code\algo\xxx | |
|  |  |  |  | ti\_c66xx\ | |  |  | --> C66xx library, C66xx CCS lib project | |
|  |  |  |  | ti\_c674x\ | |  |  | --> C674x library, C674x CCS lib project | |
|  |  |  |  | ti\_arp32\ | |  |  | --> ARP32 library, ARP32 CCS lib project | |
|  |  |  |  | ti\_cortex\_a8\ | | |  | --> Cortex A8 library, Cortex A8 CCS lib project | |
|  |  |  |  | ti\_cortex\_a15\ | | |  | --> Cortex A15 library, Cortex A15 CCS lib project | |
|  |  |  |  | ti\_cortex\_m3\ | | |  | --> Cortex M3 library, Cortex M3 CCS lib project | |
|  |  |  |  | ti\_cortex\_m4\ | | |  | --> Cortex M4 library, Cortex M4 CCS lib project | |
|  |  |  | xxx\_sim\ | |  |  |  |  |  |
|  |  |  |  | pc\ |  |  |  | --> Visual studio solution, dll, pdb, cdl, res | |
|  |  |  |  | ti\_c66xx\ | |  |  | --> C66xx out file for vh28, RAM/ROM allocation, C66xx CCS Hil wrapper CCS project | |
|  |  |  |  | ti\_c674x\ | |  |  | --> C674x out file for vme, RAM/ROM allocation, C674x CCS hil wrapper CCS project | |
|  |  |  |  | ti\_arp32\ | |  |  | --> ARP32 out file for vme, ARP32 CCS hil wrapper CCS project | |
|  |  |  |  | ti\_arp32\_vh28e1\ | | |  | --> ARP32 out file for vh28, ARP32 CCS hil wrapper CCS project | |
|  |  |  |  | ti\_cortex\_a8\ | | |  | --> Cortex A8 out file for vme, RAM/ROM allocation, Cortex A8 CCS hil wrapper CCS project | |
|  |  |  |  | ti\_cortex\_a15\ | | |  | --> Cortex A15 out file for vh28, RAM/ROM allocation, Cortex A15 CCS hil wrapper CCS project | |
|  |  |  |  | ti\_cortex\_m3\ | | |  | --> Cortex M3 out file for vme, RAM/ROM allocation, Cortex M3 CCS hil wrapper CCS project | |
|  |  |  |  | ti\_cortex\_m4\ | | |  | --> Cortex M4 out file for vh28, RAM/ROM allocation, Cortex M4 CCS hil wrapper CCS project | |
|  |  | xxx\_utils\ | |  |  |  |  |  |  |
|  | **05\_Deliverables\** | | |  |  |  |  |  |  |
|  |  | dll\ |  |  |  |  |  |  |  |
|  |  |  | algo\ |  |  |  |  |  |  |
|  |  |  |  | xxx\_sim\ | |  |  | --> Release DLLs for PC simulation | |
|  |  | lib\ |  |  |  |  |  |  |  |
|  |  |  | pc\ |  |  |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> PC algo library (release and/or debug) | |
|  |  |  | ti\_c66xx\ | |  |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> C66xx release algo library | |
|  |  |  | ti\_c674x\ | |  |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> C674x release algo library | |
|  |  |  | ti\_arp32\ | |  |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> ARP32 release algo library | |
|  |  |  | ti\_cortex\_a8\ | | |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> Cortex A8 release algo library | |
|  |  |  | ti\_cortex\_a15\ | | |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> Cortex A15 release algo library | |
|  |  |  | ti\_cortex\_m3\ | | |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> Cortex M3 release algo library | |
|  |  |  | ti\_cortex\_m4\ | | |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> Cortex M4 release algo library | |
|  |  | sdl\ |  |  |  |  |  |  |  |
|  |  |  | ti\_c66xx\ | |  |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> C66xx sdl-file | |
|  |  |  | ti\_c674x\ | |  |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> C674x sdl-file | |
|  |  |  | ti\_cortex\_a8\ | | |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> Cortex A8 sdl-file | |
|  |  |  | ti\_cortex\_a15\ | | |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> Cortex A15 sdl-file | |
| **05\_Testing\** | |  |  |  |  |  |  |  |  |
|  | **02\_Reports\** | |  |  |  |  |  |  |  |
|  |  | algo\ |  |  |  |  |  |  |  |
|  |  |  | modtests\ | |  |  |  |  |  |
|  |  |  |  | cantata\_tests\ | | |  | --> Report |  |
|  |  |  |  | qac\_tests\ | |  |  | --> QAC-Report | |
|  |  |  |  | qacpp\_tests\ | | |  | --> QACPP Report | |
|  | **05\_Test\_Environment\** | | | |  |  |  |  |  |
|  |  | algo\ |  |  |  |  |  |  |  |
|  |  |  | modtests\ | |  |  |  |  |  |
|  |  |  |  | cantata\_tests\ | | |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> cantata project solution | |
|  |  |  |  | qac\_tests\ | |  |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> QAC project files | |
|  |  |  |  | qacpp\_tests\ | | |  |  |  |
|  |  |  |  |  | xxx\ |  |  | --> QACPP project files | |
|  | **06\_Test\_tools\** | | |  |  |  |  |  |  |
|  |  | mts\_measurement\ | | | |  |  |  |  |
|  |  |  | dll\ |  |  |  |  |  |  |
|  |  |  |  | algo\ |  |  |  |  |  |
|  |  |  |  |  | xxx\_sim\ | |  | --> pc simulation dll, pdb | |

The information about the library statistic is displayed at the end of SCons build.

# How to build with SCons

## Build commands

|  |  |
| --- | --- |
| Show help information (available targets, option). | **scons -h** |
| Build all. | **scons** |
| Clean all. | **scons -c** |
| Build all with multi-thread (2 threads). | **scons -j2** |
| Build all with certain C preprocessor define. The C preprocessor define will be used when building libs, DLLs, out files. | **scons define=DEFINE1, DEFINE2** |
| Build all with certain C preprocessor define using 2 threads. | **scons –j2 define=DEFINE1, DEFINE2** |
| Build a target. | **scons (target\_name)**, ex: **scons ped\_algo\_ti\_c674x** |
| Clean a target. | **scons -c (target\_name)**, ex: **scons -c ped\_algo\_ti\_c674x** |
| Build a target with certain variant. | **scons (target\_name)\_(variant)**, ex: **scons ped\_algo\_ti\_c674x\_release** |
| Build all IDEs. | **scons (component/algo)\_ide**, ex: **scons ped\_ide** |
| Build doxygen. | **scons (target\_name)**, ex: **scons ped\_doxy** |

Commands above must be executed from “04\_Engineering\03\_Workspace\algo\xxx” folder

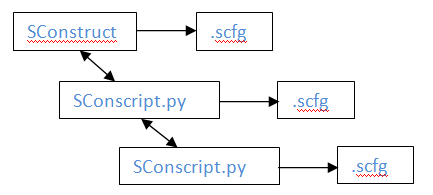
## Help menu

A help menu can be displayed with “scons –h”, see e.g.



## General information about the SCons build

1. Build workflow:
   1. SCons looks for SConstruct which is the main script.
   2. SConstruct reads required .scfg file and calls SConscript.py to build a target.
   3. SConscript.py reads required .scfg files (file list and scons config file).
   4. SConscript.py builds target.



1. SCons creates the same folder structure in build folder as in the source folder.
2. When variant is used, each variant build result must be in different folder. Otherwise SCons gives error message because the build result of one variant overwrites the other variant.
3. When multi-thread build is used, the build log file might look out of order because of multi thread.
4. The SCons build log is stored in 04\_Engineering\03\_Workspace\algo\xxx\sconsbuild.log.
5. GenericScons uses extensions and helper, which can be found in 04\_Engineering\02\_Development\_Tools\scons\_tools\scons\_adas\_extensions
6. The common configuration used by GenericScons can be found in 04\_Engineering\02\_Development\_Tools\scons\_tools\scons\_common\_config\common\_config.scfg
7. GenericScons supports more than one algos in a component. When there are multiple algos, the target\_name starts with algo name instead of component name. For example: in GB component, there are cb and scb algos. Execute **scons cb\_algo** to build cb algo or **scons scb\_algo** to build scb algo. To build both algos, execute **scons gb\_algo**.
8. When 04\_Engineering\03\_Workspace\algo\xxx\.sconsign.dblite file or 04\_Engineering\04\_Build folder is deleted, SCons will do rebuild when it is executed.
9. GenericScons looks for 03\_Workspace\algo\xxx\sconscript\_setup\_config.scfg to know which SConscripts should be copied from 02\_Development\_Tools\scons\_tools\scons\_common\_scripts to project before starts the build process. If the project has non-shared a SConscript, this non-shared SConscript should not be listed in this config file to prevent it being overwritten or the "copy" attribute should be set to False.
10. QAC analysis is integrated in SCons. Fur further information, please refer to “QAC\_manual.docx”.
11. Canata unit test is integrated in SCons. For further information, please refer to “ETK SCT SCons Cantata Integration.docx”.

# Steps of implementing GenericSCons build environment for a new component

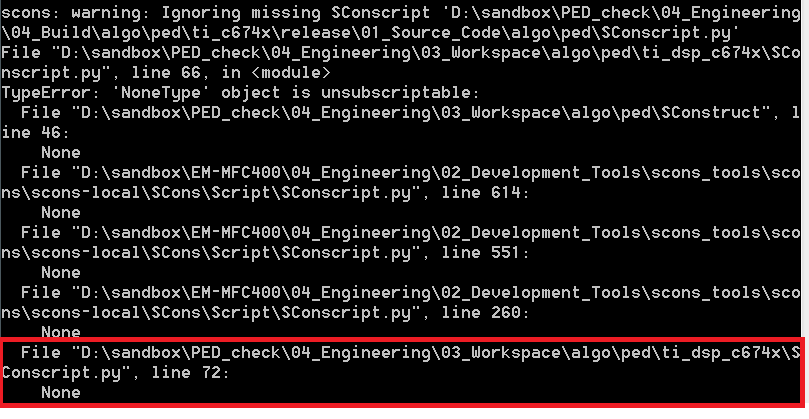
1. Perform necessary structural modification on project in order to meet prerequisites for GenericScons, see above.
2. Add shared subprojects of 02\_Development\_Tools, see above.
3. Set up an empty GenericScons environment, please refer to “generic\_scons\_manual.pptx”.
4. Add GenericScons features according to needs.

**Note:** Step 4 essentially consists in creating scfg-configuration files in the project. Examples for those configuration files are given in scons\_tools\scons\_templates. Those may serve as a reference. For more detailed information on how to set up a new GenericScons environment, please refer to “generic\_scons\_manual.pptx” or “SCons\_build\_environment.docx”.

# Checking for errors in Scons

Before SCons builds target, it reads all SConscript files. When there is an error in the SConscript file, SCons stops and reports the error with the information about script that causes the error, including the scripts in the 02\_Development\_Tools\scons\_tools. When tracing the error, the scripts in 02\_Development\_Tools\scons\_tools can be ignored.

When scons\_tools folder is copied from other component, the path of the scons\_tools might be incorrect as shown below. This is because python uses directly the compiled python file (.pyc) in 02\_Development\_Tools\scons\_tools. To fix this problem, remove all .pyc files.



When there are a lot number of C source files used as PDO source, there is a possibility that the command-line string becomes very long and exceeds the maximum 8191 characters because absolute path is used for each C source file and include paths. This will cause error because Windows only allows maximum 8191 characters. The workaround is to remove unused include paths and have a shorter folder name for the sandbox.