Software Component Support Usage

TL109A\_SwcSuprt

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 1 | 01-Dec-2016 | Owen Tosh | Initial version |
| 2 | 07-Dec-2016 | Owen Tosh | Added Polyspace report generation |
| 3 | 30-Mar-2017 | Matt Leser | Added Sandbox Project Creation |
| 4 | 18-May-2017 | Matt Leser | Expanded on descriptions, Updated Graph to show where Sandbox Project is located |

Table of Contents

[Introduction 3](#_Toc466623812)

[Running the Tool 4](#_Toc466623813)

[The Main Menu 5](#_Toc466623814)

[The Component Menu 6](#_Toc466623815)

[File and Directory Structure 8](#_Toc466623816)

# Introduction

SwcSuprt is a component support script for EA4 component development. It automates many redundant tasks, and is useful to easily update a component to the latest standards.

Both GUI and command-line interfaces are supported. If this script is run with both a path to a component and a command, the script continues running in the command line. Otherwise, the GUI is initialized.

## Dependencies

This tool requires the TL112A\_Python component. This component contains the Nexteer Python library nxtr.py, which is required by this tool (a vanilla Python environment will not contain this library).

# Running the Tool

## Launch.bat

In the tools directory, running Launch.bat will load the tool directly with no arguments. The tool prompts the user with the main menu, which can be used to create a new component or select an existing component. Once this is done, the user is prompted with the component menu.

## SWCSupport.bat

Once the tool is integrated with a component, it can be launched with SWCSupport.bat, located in the component’s tools directory. This will immediately launch the component menu.

## Command Line Usage

The expected command line format is as follows:

swcsuprt.py ["/path/to/component" [command]]

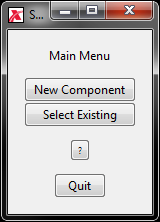
If no arguments are provided, the tool will present the main menu. If run with a single argument, the tool will open the component menu for the component at the path provided. The path may be relative or absolute.

If run with two arguments, the tool will run in a non-interactive command line format. command may be one of the following:

* initialize – create basic directories if they don't already exist
* migrate – perform component migration to latest directory format. Also runs:
  + generate\_greenhills
  + generate\_davinci
  + generate\_integration\_files
  + generate\_generation\_script
* generate\_greenhills – generate Green Hills project file
* generate\_davinci – generate DaVinci component files
* generate\_integration\_files – generate integration files
* generate\_generation\_script – generate generation batch script, or remove it if not necessary
* generate\_sandbox – generate sandbox project
* generate\_local\_headers – generate local header files from Configurator
* generate\_polyspace\_files – generate Polyspace project files and stubs
* unzip\_poly space – unzip Polyspace archives
* open\_bugfinder – open Polyspace Bug Finder project
* open\_codeprover – open Polyspace Code Prover project
* zip\_polyspace – zip current Polyspace results into archives

See the description of the functions in the component menu for more details on each command.

# The Main Menu



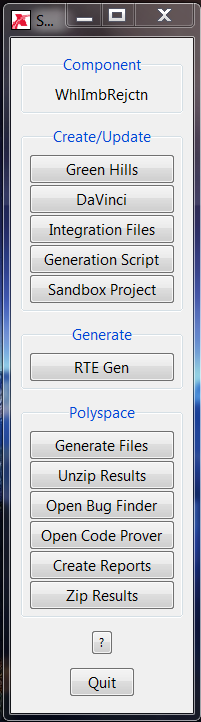
Click **New Component** to create a folder template for a new component. The tool will prompt for a full component name and parent directory, and then present the component menu for the newly created component. The new component will only contain the basic folder structure, and the SWCSupport.bat script.

Click **Select Existing** to open the component menu for an existing component. The tool will prompt for the main component directory. If the selected directory is missing the standard component subfolders (e.g. autosar, src, include), the tool will display a warning.

Click **?** to open this usage document.

Click **Quit** to close the tool.

# The Component Menu



The main menu contains five sections:

## Component

The component short name is shown here.

If the component appears to be of an older format (e.g. contains a tools/contract directory), this section will also contain an additional button. Click **Migrate** to automatically update this component to the latest format, moving files as needed. Note that this will also run all commands in the Create/Update section.

## Create/Update

Click **Green Hills** to generate the Green Hills project file (.gpj).

Click **DaVinci** to generate the DaVinci Configurator project files. This includes the DaVinci Project file (.dpa) and some files in the tools/local/autosar directory. Note that if a .bswmd.arxml file is added to the autosar folder, this function must be run in order to include it in the component configuration.

Click **Integration Files** to generate files for integration. This feature checks for any .bswmd.arxml files in the autosar directory, then creates an integration script in the tools/integrate folder. If no .bswmd.arxml files are found, no integration script is needed and it is removed.

Click **Generation Script** to generate the batch script used by Configurator when generating configuration files from ARTT text template files. This feature checks for any template files (.tt) in the tools/template directory, and creates a generation script if found. Otherwise, the generation script is removed.

Click **Sandbox Project** to generate the sandbox project used from the Green Hills project file. (.gpj). The project file will open as well even if no update/creation took place. The sandbox is an area used to test if component level source codes compile. This will also generate the .d, .dbo, and .o files in a folder called SandboxOutput that is located on the same level as the components.

## Davinci

Click **Generate** to generate the RTE files, contract header files, and any configurable files of the component for local compilation testing and static analysis. If an error occurs, log files may exist in the tools directory, as well as the tools/local/generate directory.

## Polyspace

Click **Generate Files** to generate all project and related files to the tools/Polyspace directory. This function checks for all source files and include directories, adding them to the Polyspace projects. It also looks for a corresponding design component in the component’s parent directory. If not automatically found, the user is prompted to locate it manually or proceed without using a data dictionary. Note that not using a data dictionary causes many orange errors in Code Prover. It is recommended to run this function before analysis is performed.

Click **Unzip Results** to decompress any saved results from previous analyses. Previous results are stored in the tools/Polyspace/Saved directory.

Click **Open Bug Finder** to open the Bug Finder Polyspace project.

Click **Open Code Prover** to open the Code Prover Polyspace project.

Click **Create Reports** to generate XML reports from the Bug Finder and Code Prover results. These reports may be used in the future to gather information at the request of the customer. Currently, they will be stored in Synergy to prepare for potential use.

Click **Zip Results** to delete backup comment folders and archive the current results to the tools/Polyspace/Saved directory.

## Other Buttons

Click **?** to open this usage document.

Click **Quit** to close the tool.

# File and Directory Structure

The following directories exist in every software component:

autosar – contains all Developer/Configurator files

doc – documentation

generate – all generated source and header files (used in the integration project)

include – component header files

src – component source files

tools – contains the component DaVinci project file, the Green Hills Project file, and a script to launch the SwcSuprt tool

tools/integrate – all integration-related files

tools/local – all files used in local component development, but not used at the integration project level. This contains any stubs required for static analysis, as well as configurator files used to create a local environment. Each directory in this path resembles the base directory structure (e.g. autosar, generate, include, src).

tools/Polyspace – Polyspace Bug Finder and Code Prover project files, along with project-related files. Also contains the archived results from previous analyses.

tools/template – any files involved in generation from text templates at both the component and integration project level