Ways to run scripts in the “Utilities” folder:

- **ModelAdvisorAction.m** : This is a MATLAB® class file which takes the model name as an input argument to its constructor. It has a method called “run()” to execute ModelAdvisor Checks on the model provided to the constructor.

```
ma = modelAdvisorAction('TargetSpeedThrottle');
v = ver('MATLAB');
if v.Release == "(R2021a)"
    ma.configFile = './tools/utilities/config_data/do178ChecksR21a.json';
else
    ma.configFile = './tools/utilities/config_data/do178ChecksR20b.json';
end
ma = ma.run();
ma.generateReport();
```

- **designVerifierAction.m:** This is a MATLAB class file which takes the model name as an input argument to its constructor. It has a method called “verify()” to perform design verification on the model provided to the constructor.

```
mdv = designVerifierAction('TargetSpeedThrottle');
mdv.verify();
mdv.generateReport();
```

- **modelBuildAction.m**: This is a MATLAB class file which takes the model name as an input argument to its constructor. It has a method called “build()” which builds the model provided to the constructor.

```
mb = modelBuildAction('TargetSpeedThrottle');
mb.build();
```

- **ModelTestsAction.m:** This is a MATLAB class file which takes the model name and other (optional)parameters to generate various types of results.

```
tests = modelTestsAction('ModelName', 'TargetSpeedThrottle','TestFile', 'TargetSpeedThrottleTestManager.mldatx', ...
    'PDFReport', 'yes', 'SimulinkTestMgrResults', 'yes', 'JUnitFormatResults', 'yes');
tests.runTestsAndGenerateResults();
```

This file supports optional name-value parameters. The default values for these parameters are ‘no’. They are:

  - **‘PDFReport’ :** Generates a PDF report of the tests that were run.
  - ‘**TAPResults’** : Generates TAP test results of the tests that were run.
  - **‘JUnitFormatResults’ :** Generates JUnit-style test results of the tests that were run.
  - **‘SimulinkTestMgrResults’:** Generates Simulink® Test™ Manager results of the tests that were run.
  - **‘CoberturaModelCoverage’:** Generates Cobertura Model Coverage results of the tests that were run on the model.

- **generateXMLFromLogs.m:** This script generates a single XML file which has dynamic data from all stages of a pipeline on a model

```
generateXMLFromLogs('TargetSpeedThrottle');
```

- **generateHTMLReport.m:** This script generates the Summary Report at the end of the pipeline using ‘report.xsl’ and XML file(generated in above step) for a model.

```
generateHTMLReport('TargetSpeedThrottle');
```

- **cleanUp.m, openModels.m and runAllTestsLocally.m :** These three scripts can be run via the MATLAB editor or in the Command Window.

- **startUp.m :** This file will be automatically run by Simulink Project.