**Generic SCons**

**Testing Guideline and Process**

|  |  |
| --- | --- |
| Document Control: | |
| Project: | ETK: SCT\_Sconstools |
| Revision: | 0.7 |
| Last Change: | 06-Jun-2016 |
| Confidence Level: | 🞏 Public 🗷 Confidential |

CONFIDENTIAL AND PROPRIETARY PROPERTY OF ADAS SIBIU - ALL RIGHTS RESERVED

|  |  |
| --- | --- |
| Document State: | |
| State: | Draft |
| Author: | Lenin Palanisamy |
| Reviewed by: | Vishal Singh |
| Released by: | 28-Jul-2016 |

Revision History

| Version | Date | Change Description | Responsible | Approver |
| --- | --- | --- | --- | --- |
| 0.1 | 27-Apr-2015 | Initial draft version for review | Lenin Palanisamy | Andre Fischer |
| 0.2 | 29-May-2015 | Updated based on review comments | Lenin Palanisamy | Andre Fischer |
| 0.3 | 12-Jun-2015 | Updated to add more information as part of further improvements | Lenin Palanisamy | Andre Fischer |
| 0.4 | 19-Jun-2015 | Add Section 3.3. and update section 3.4 | André Fischer | Lenin Palanisamy |
| 0.5 | 24-Jun-2015 | Update the sections 1.3 and 2.1 based on review comments and Heading 3 with indices and TOC | Lenin Palanisamy | Andre Fischer |
| 0.6 | 22-Jul-15 | Update the section 1.3 and 3.6 for referencing LL document | Lenin Palanisamy | Andre Fischer |
| 0.7 | 06-Jun-16 | Updated as part of continuous improvement:   1. Section 1.3 for IMS location 2. Replace MKS with IMS in all sections | Lenin Palanisamy | Vishal Singh |
| 0.8 | 26-Jul-16 | Added section 3.1.1 | Tridip Bhagawati | Vishal Singh |
|  |  |  |  |  |

Table of Contents

[1 Introduction 4](#_Toc452985850)

[1.1 Purpose 4](#_Toc452985851)

[1.2 Scope 4](#_Toc452985852)

[1.3 Reference documents 4](#_Toc452985853)

[1.4 Overview 5](#_Toc452985854)

[2 Test Strategy 6](#_Toc452985855)

[2.1 Issue Checklist 6](#_Toc452985856)

[3 Testing Guidelines 7](#_Toc452985857)

[3.1 Prerequisite 7](#_Toc452985858)

[3.2 Role and Responsibilities 7](#_Toc452985859)

[3.3 Design and Maintenance Test Frame on Remote Machine 7](#_Toc452985860)

[3.4 Testing Steps 8](#_Toc452985861)

[3.4.1 Testing for Issue Resolution 8](#_Toc452985862)

[3.4.2 Testing for Release 11](#_Toc452985863)

[3.5 Test Results 11](#_Toc452985864)

[3.6 Lessons Learned 11](#_Toc452985865)

# Introduction

## Purpose

This document contains a “How to” guideline for testing the modifications carried out based on the issue/Realization Order (RO) raised in Generic SCons.

## Scope

This document applies for testing the modifications in SCT\_Sconstools of Engineering Tool Kit (ETK) project.

## Reference documents

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Reference Document** | **Location** |
| 1 | Issue Checklist template | <http://ims-adas:7001/si/viewrevision?projectName=/nfs/projekte1/REPOSITORY/Base%5fDevelopment/05%5fAlgorithm/ETK%5fEngineeringToolKit/04%5fEngineering/SCT%5fSconstools/docs/processes/templates/project.pj&selection=Issue%5fChecklist.xlsx> |
| 2 | GenericSCons Release Process | <http://ims-adas:7001/si/viewrevision?projectName=/nfs/projekte1/REPOSITORY/Base%5fDevelopment/05%5fAlgorithm/ETK%5fEngineeringToolKit/04%5fEngineering/SCT%5fSconstools/docs/processes/project.pj&selection=GenericSCons%5fRelease%5fProcess.docx> |
| 3 | All issue checklists for current release | <http://ims-adas:7001/si/viewrevision?projectName=/nfs/projekte1/REPOSITORY/Base%5fDevelopment/05%5fAlgorithm/ETK%5fEngineeringToolKit/04%5fEngineering/SCT%5fSconstools/docs/release%5fnotes/checklists/project.pj&selection=Issue%5fChecklists.xlsx> |
| 4 | GenericSCons Lessons Learned | [http://ims-adas:7001/si/viewrevision?projectName=/nfs/projekte1/REPOSITORY/Base%5fDevelopment/05%5fAlgorithm/ETK%5fEngineeringToolKit/04%5fEngineering/SCT%5fSconstools/docs/generic%5fscons/project.pj&selection=GenericSCons%5fLessons%5fLearned.docx](http://mks-psad:7001/si/viewrevision?projectName=/nfs/projekte1/REPOSITORY/Base%5fDevelopment/05%5fAlgorithm/ETK%5fEngineeringToolKit/04%5fEngineering/SCT%5fSconstools/docs/generic%5fscons/project.pj&selection=GenericSCons%5fLessons%5fLearned.docx) |

## Overview

Generic SCons modules are modified or added based on the requirements (New Feature/Change/Problem Requests) described in IMS issues and ROs. These issues/ROs are planned, executed and delivered. (Release Process Ref. 1.3). Testing is a mandatory activity for any modification in Generic SCons to ensure the following:

* New Feature/Change/Problem request has been implemented
* There is no scripting error in Generic SCons modules
* Targets which are related to modifications are still buildable
* There is no side effect (i.e., no new issue introduced)
* There is no failure in Jenkins Server test for all components

# Test Strategy

Test definition and cases would be different for each issue/RO. Hence the test approach for an issue is important to assure the quality.

Test cases should be defined to ensure that the modifications in Generic SCons modules against an issue or RO are correct (Refer the points mentioned in the section 1.3). Hence, the test cases would be developed in two parts:

* Issues' specific test cases
* Regression test cases

## Issue Checklist

For an issue/RO, an issue check list needs to be filled up to confirm that all general and test check points are verified. Test cases will be briefly described in the issue checklist in “Remarks” column. All issue checklists only related to current release should be inserted into a single workbook “Issue\_Checklists.xlsx” and checked in to IMS at the location (Ref. point 3 in section 1.3) before closing the Release Item in IMS.

Note: The latest issue checklist template Issue\_NNNNNN\_Checklist should be used for each issue and template configured at IMS location (Ref. 1.3)

# Testing Guidelines

## Prerequisite

System and test environment requirements and Access rights need to be checked prior to any testing activity after updating Generic SCons modules. Check for the following software, tools and access privileges are in place:

* Code Composer Studio 5.3.0, 5.5
* Visual Studio 2005
* QAC/QACPP(Refer QAC\_manual.docx)
* Cantata 6.2
* PTC Integrity Client 2010
* Login credentials to remote PC “lud4b3dg.cw01.contiwan.com**”**
* Tester should have admin rights on the remote machine in order to execute certain batch files
* Jenkins server login credentials

**3.1.1 Prerequisite steps to install QAC/QACPP**

To test QAC/QACPP for all the components of camera in Jenkins sever, requires a few pre requisite steps to be followed

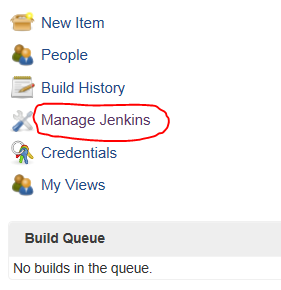
**Step 1:**

1. Install QAC in the remote machine (lud4b3dg)  
   <https://workspace1.conti.de/content/00002191/Lists/QAC/DispForm.aspx?ID=1&Source=https%3A%2F%2Fworkspace1%2Econti%2Ede%2Fcontent%2F00002191%2FLists%2FQAC%2FAllItems%2Easpx>
2. Install QACPP in remote machine (lud4b3dg)  
   <https://workspace1.conti.de/content/00002191/Lists/QAC/DispForm.aspx?ID=22&Source=https%3A%2F%2Fworkspace1%2Econti%2Ede%2Fcontent%2F00002191%2FLists%2FQAC%2FAllItems%2Easpx>
3. To know all QAC/QACPP related information follow link  
   <https://workspace1.conti.de/content/00002191/Lists/QAC/AllItems.aspx>
4. Configure license server with QAC/QACPP

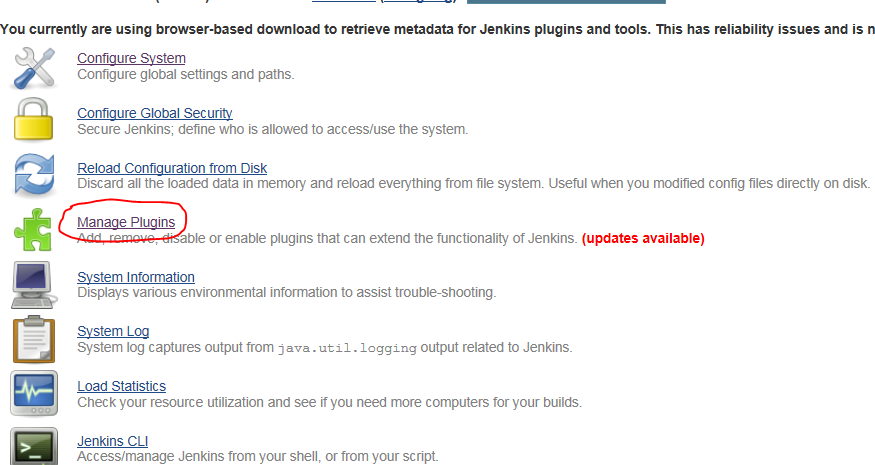
**Step 2:**

At first, install PRQA Plugin by following the steps below:

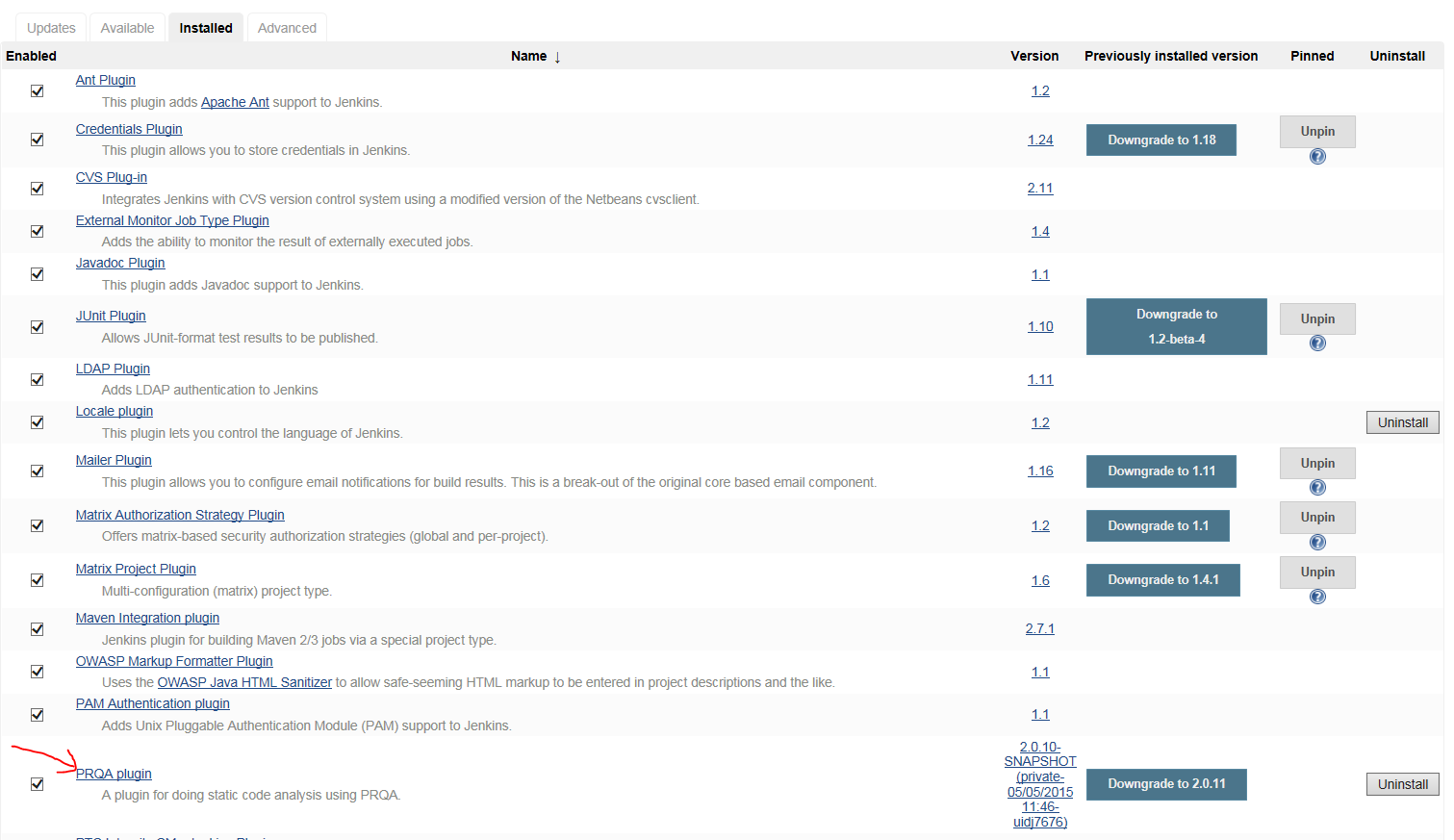
1. Click on Manage Jenkins in the left as shown below



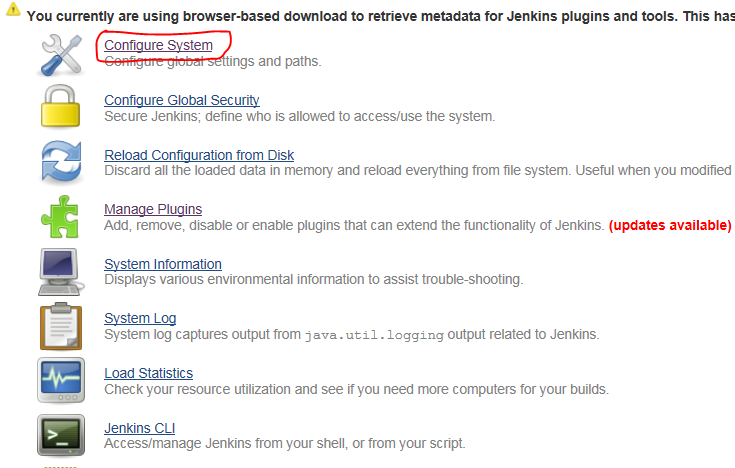
1. In the next screen, click on Manage Plugins as shown below



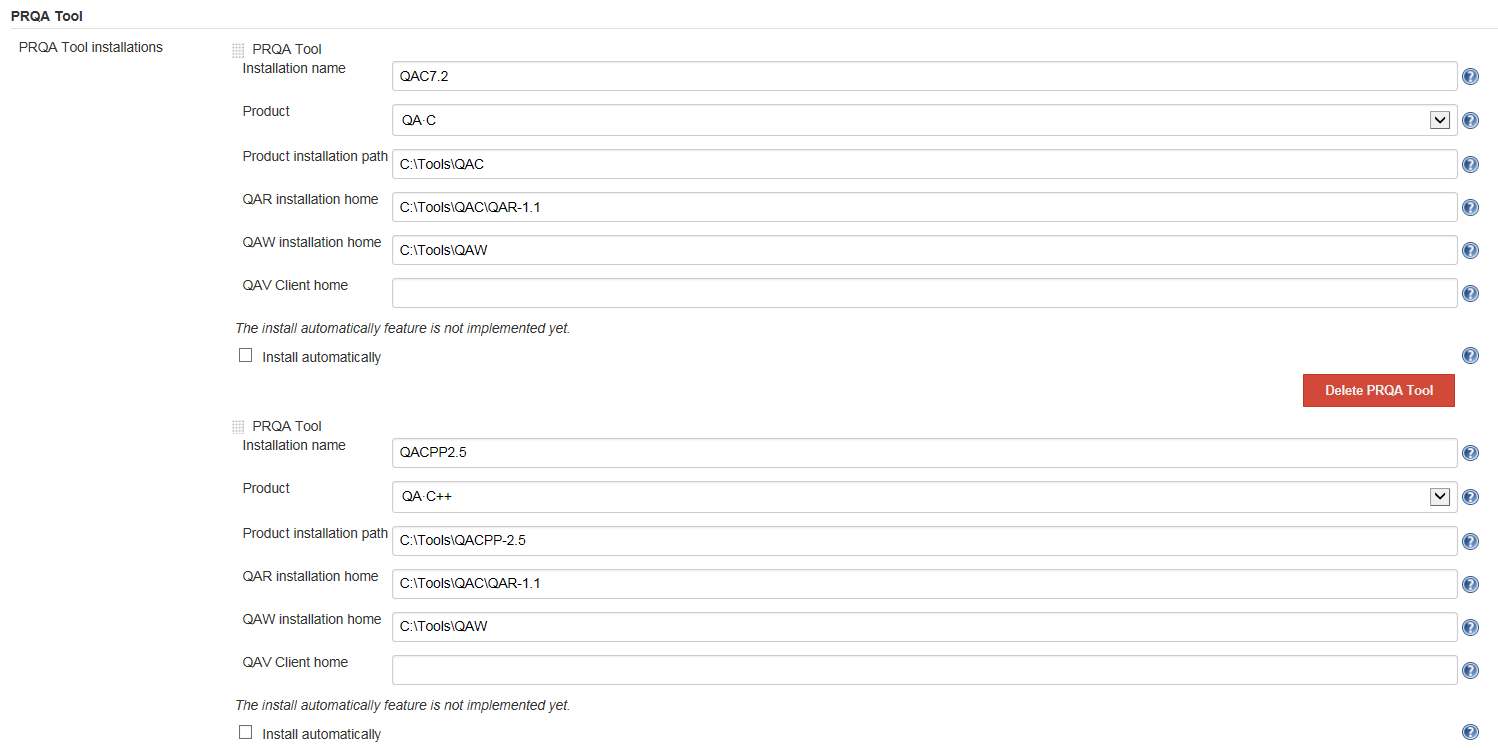
1. In the next screen, install PRQA plugin and after installed you can see as shown below



1. After PRQA plugin is installed, go back to Manage Jenkins page and click on Configure System as shown below.



1. In this page go down to PRQA tool and fill as shown below



## Role and Responsibilities

Testing is a responsibility of developer whoever doing the modifications in Generic SCons modules. (Refer the points mentioned in the section 1.4)

## Design and Maintenance Test Frame on Remote Machine

Generic SCons developers are advised to keep reasonably up-to-date versions of regular/variant/build sandboxes of some components on the system for implementation and testing.

The remote machine “lud4b3dg.cw01.contiwan.com” contains sandboxes of all components running on Generic SCons. More precisely, the server contains

* Trunk sandboxes of all those components under **D:\Sandboxes\Regular**
* Build Sandboxes of all those components under **D:\Sandboxes\Build\Components**

Since components’ projects and Generic SCons itself will change over the course of time, it makes sense to replace the build sandboxes by more recent ones on a regular base. As a guideline, the build sandboxes should be configured on the server within the first two weeks of each quarter of the year.

## Testing Steps

### Testing for Issue Resolution

Testing can be done in two parts:

1. **Testing in local machine**:

Steps for testing in local machine are as follow:

* 1. Create/resynchronize sandbox of the trunk version of SCT\_SCONS project from IMS.
  2. Replace all the modified file/files of Generic SCons in your sandbox of SCT\_SCONS project.
  3. Copy and paste all the folders of SCT\_SCONS in the path 04\_Engineering\02\_Development\_Tools\scons\_tools of the working component.
  4. Test for the desired build target and also all relevant the build targets of the working component.

Note: It might not be necessary to build the whole component, only those targets being related to changes since a trunk version of a component often does not build completely successfully!!

1. **Testing in Jenkins server**:
   1. **Below steps need to be followed before testing in Jenkins server:**
      1. Check in all the necessary changed file/files in IMS. Remember do not make “update member revision” in IMS.
      2. Connect to remote machine " lud4b3dg.cw01.contiwan.com" through remote desktop connection and sign in with your windows user ID and password.

Note: Remote machine "lud4b3dg" is located in Lindau, Germany.

* + 1. Make sure all the required tools are present in the folder location "D:\Sandboxes\DevTools\DevTools\_trunk".

Note: if not - In the location "D:\Sandboxes\DevTools" check the file “link\_generate.bat” present and see if all the linked steps are done.

This batch file serves the purpose to set up new "DevTools"-versions, like e.g. Dev\_Tools\_1\_55, a version of 02\_Development\_Tools containing scons\_tools\_1\_55.

So in order to create a new such version, create the folder DevTools\_1\_xx, copy this batch file into that folder, modify it according to the needs and then execute it with admin rights.

* + 1. Import all the projects present in the location "D:\Sandboxes\Build\Components" and "D:\Sandboxes\Regular" in IMS in Build and Regular respectively.
    2. Go to the location "D:\Sandboxes\Build\Components" and run the batch file “resync.bat”

Note: This batch file will resynchronize all the files of all the components of the build version.

* + 1. Go to the location "D:\Sandboxes\Build\Components" and run the batch file setup\_sandboxes.bat.

Note:

1) This batch file will link all the folders in the path "D:\sandboxes\DevTools\DevTools\_trunk" to “02\_Development\_Tools” folder of all the components present in the path "D:\Sandboxes\Build\Components\04\_Engineering\02\_Development\_Tools".

2) This batch file can also be used to link other versions of 02\_Development\_Tools to the sandboxes, this can be needed, e.g., for reproducing errors, etc.

* + 1. Go to the location "D:\Sandboxes\Regular" and run the batch file “resync.bat”.
    2. Go to the location "D:\Sandboxes\Regular" and run the batch file “setup\_sandboxes.bat”.
    3. Go to the location "D:\Sandboxes\Regular\Batch\_Files" modify the batch file “build.bat” content with the target name you want to build for all the components.

Note: Uncomment or add the necessary lines as per the target you want to build, for all the components.

Since the build status depends on using the correct version of scons.bat, otherwise errors can be missed (therefore there is a call to copy the correct version of the scons.bat). Hence a minimal version of “build.bat” given below:

|  |
| --- |
| set project=%1  set SCONS\_ERROR=0  set ERRORMSG=  for %%I in ( %project% ) do set component=%%~nI  del /F scons.bat  xcopy ..\..\..\02\_Development\_Tools\scons\_tools\scons\_templates\03\_Workspace\algo\xxx\scons.bat .[[1]](#footnote-1)  xcopy /Y ..\..\..\02\_Development\_Tools\scons\_tools\scons\_common\_scripts\doxygen\convert\_xml.py ..\%component%\_sim\sim\_swc\_%component%\  call %project%\scons.bat %component%\_<target>  if not %ERRORLEVEL%==0 (  set SCONS\_ERROR=1  set ERRORMSG=Clean  )  echo %ERRORMSG%  exit /b %SCONS\_ERROR% |

* 1. **Testing steps in Jenkins server are as follows:**
     1. Go to the link <https://lud4b3dg:8443/>

Note: Sign in with Windows login credentials provided.

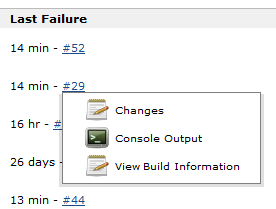
* + 1. Press the button in the extreme right for all the components.

|  |  |
| --- | --- |
| |  | | --- | | You will build the target/targets provided in the **build.bat** file for all the components | |

* + 1. See the result of the build in the left corner.



* + 1. If build fails for any component check the Console Output for respective component and correct and take necessary action to complete testing.



|  |
| --- |
|  |

### Testing for Release

After setting the Checkpoint for a release it may be required to be tested in conclusion. In order to do so, all components’ sandboxes on the Jenkins Server, which are fully buildable, shall be fully built with this new Checkpoint. In order to do so, the batch script build.bat is to be adapted such that the standard target “all” is executed for all those components, so in its core it may look as follows:

set project=%1

set SCONS\_ERROR=0

set ERRORMSG=

for %%I in ( %project% ) do set component=%%~nI

call %project%\scons.bat all

if not %ERRORLEVEL%==0 (

set SCONS\_ERROR=1

set ERRORMSG=Clean

)

echo %ERRORMSG%

exit /b %SCONS\_ERROR%

So with these settings the steps in 2.2 may be re-executed.

## Test Results

All test results would be attached/configured wherever possible. For example, the final "sconsbuild.log" will be attached as an evidence of the testing.

## Lessons Learned

While implementing and testing an issue, all lessons learned points need to be recorded in the document for continuous improvement for an efficient implementation and testing.

Lessons Learned document would be found at the IMS location (Ref. point 4 in section 1.3)

1. **Remark:** For correct reporting of the error level of the GenericScons build, it is eminently important that the scons.bat returns the correct value. Quite a few components seem to use an outdated/buggy version of scons.bat in this respect. This causes GenericScons to report “Build succeeded.” at times while it was not. Therefore it is a well-established method to use the generic version of scons.bat under *scons\_tools\scons\_templates\03\_Workspace\algo\xxx\scons.bat.*

   With this version it is assured that the errorlevel returned is correct. [↑](#footnote-ref-1)