**Features Document for TeaLeaves**

**Document History**

|  |  |  |
| --- | --- | --- |
| **Version,**  **State,**  **Date** | **Author** | **Modifications** |
| 0.1,Draft,  18/11/19 | RBEI/ETB4 Tools Team | Initial version (Includes updates from Adler Max) |
| 0.2  20/11/19 |  | Included changes from Hai and Adler Max |
| 0.3  22/11/19 |  | Added points from ETB4 tools team with refinement |
| 0.4  25/11/2019 |  | Updated for points from Partha,Senthil and Adler Max |
| 0.5  27/11/2019 |  | Included Prediction mode and updated review comments from PR’s(Kanchana and Brevina) |

[1 Introduction 2](#_Toc25923251)

[1.1 Scope 3](#_Toc25923252)

[2 Requirements for Auto-Stubbing Mode 3](#_Toc25923253)

[2.1 Missing Process Not added in OS\_Sched file 3](#_Toc25923254)

[2.2 Stubbed Missing Proc not in same order as SC pavast(Dysched) 3](#_Toc25923255)

[2.3 Stubbed Calibration should have user provided value 4](#_Toc25923256)

[2.4 Changes in solver report for variables 4](#_Toc25923257)

[2.5 TeaLeaves should provide 2 seperate option for ADDITON and DELETION of interfaces(system constants..) along with C(\*.h and \*.c) files updation only for SWAdp component 5](#_Toc25923258)

[2.6 LADB support for Auto-Stubbing – Interface Handling 6](#_Toc25923259)

[2.7 Auto-stubbing, how to select the right properties of interfaces, how to define new interfaces into tealeaves – Interface Handling 6](#_Toc25923260)

[2.8 Option to select multiple container (FCL/FCP) for different interfaces 7](#_Toc25923261)

[2.9 List Swadp module for multiple definition (2FC's + 1 Swadp) 7](#_Toc25923262)

[2.10 Removal of defnitions from .C/.H files 8](#_Toc25923263)

[2.11 TeaLeaves should support on PCFG component (PVER structure would change) 8](#_Toc25923264)

[2.12 Multiple defntion of Elements (compu methods/message/system constant/cal-params) in CEL and SWadap 8](#_Toc25923265)

[2.13 Removal of Multiple definition in ConfData is not supported currently in solver mode 8](#_Toc25923266)

[2.14 Extensibility of TeaLeaves support of automated integration of hints/Cheat Sheets (PST Patching) 9](#_Toc25923267)

[2.15 Support of BIT Variables in Auto-Stubbing 9](#_Toc25923268)

[3 Improvments under Prediction Mode 10](#_Toc25923269)

[3.1 Pre-BCT -> BCT -> Pre-MDGB->MDGB (Patching) 10](#_Toc25923270)

[3.2 Handling of improper XML entries in artifact(ArXML) 11](#_Toc25923271)

[3.3 BCT Validations Not Handled 11](#_Toc25923272)

[3.4 Differences in generated pavast files 12](#_Toc25923273)

[3.5 TeaLeaves doesn’t validate \*.C files for Validation 13](#_Toc25923274)

[4 General Refinements 14](#_Toc25923275)

[4.1 Evaluating SW Label 14](#_Toc25923276)

[4.2 Tracking of usage of TeaLeaves 14](#_Toc25923277)

[4.3 Detect Common Errors (CORE and DATA steps) 14](#_Toc25923278)

[4.4 Ignore template (test case rule) 14](#_Toc25923279)

[5 Additional points 15](#_Toc25923280)

[5.1 Register stubbed file for Build Execution 15](#_Toc25923281)

[5.2 System Constants- Input File are not stubbed in Solver Mode 15](#_Toc25923282)

# Introduction

This document contains the features refinement for TeaLeaves tool on various usecases.

## Scope

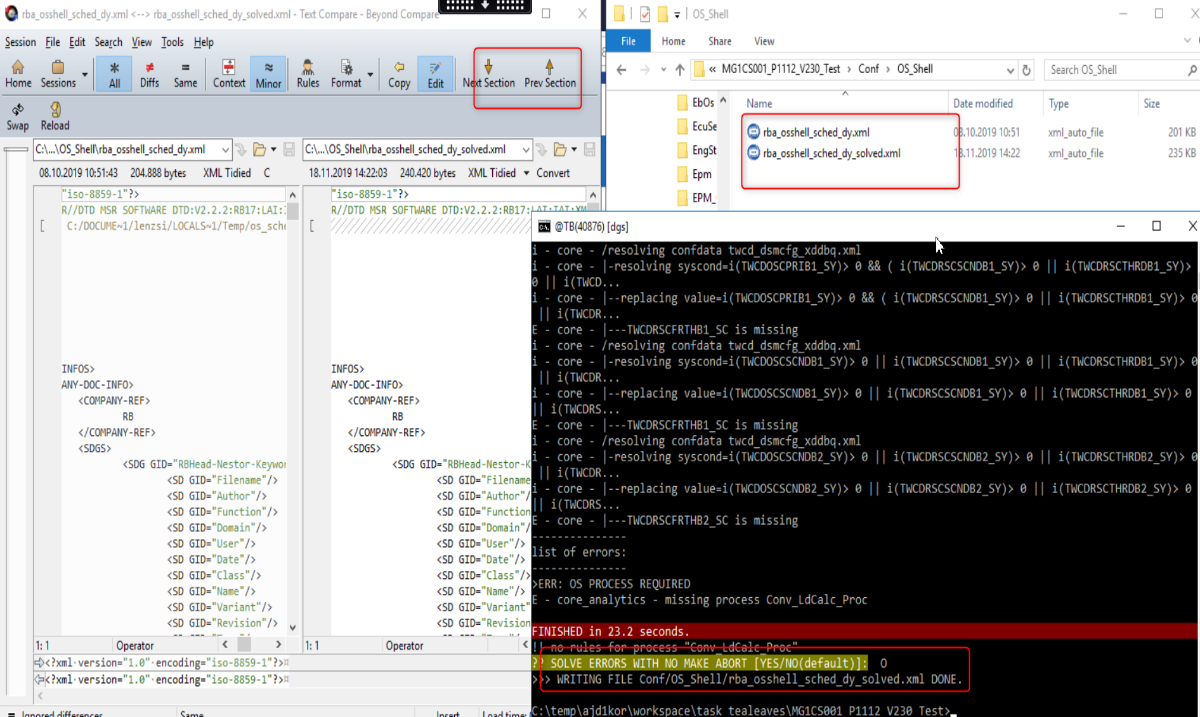
The purpose of this document is to specify new requirements and changes to existing tool. This document is derived by running the tool on different usecases and deriving the expectations of tool from Function developers, Integrators and package responsbiles. The document is

# Requirements for Auto-Stubbing Mode

The below are the requirements derived when tealeaves is run in solver mode

## Missing Process Not added in OS\_Sched file

Although new rba\_osshell\_sched\_dy\_solved.xml file is created but the missing, proc is not added to the file.



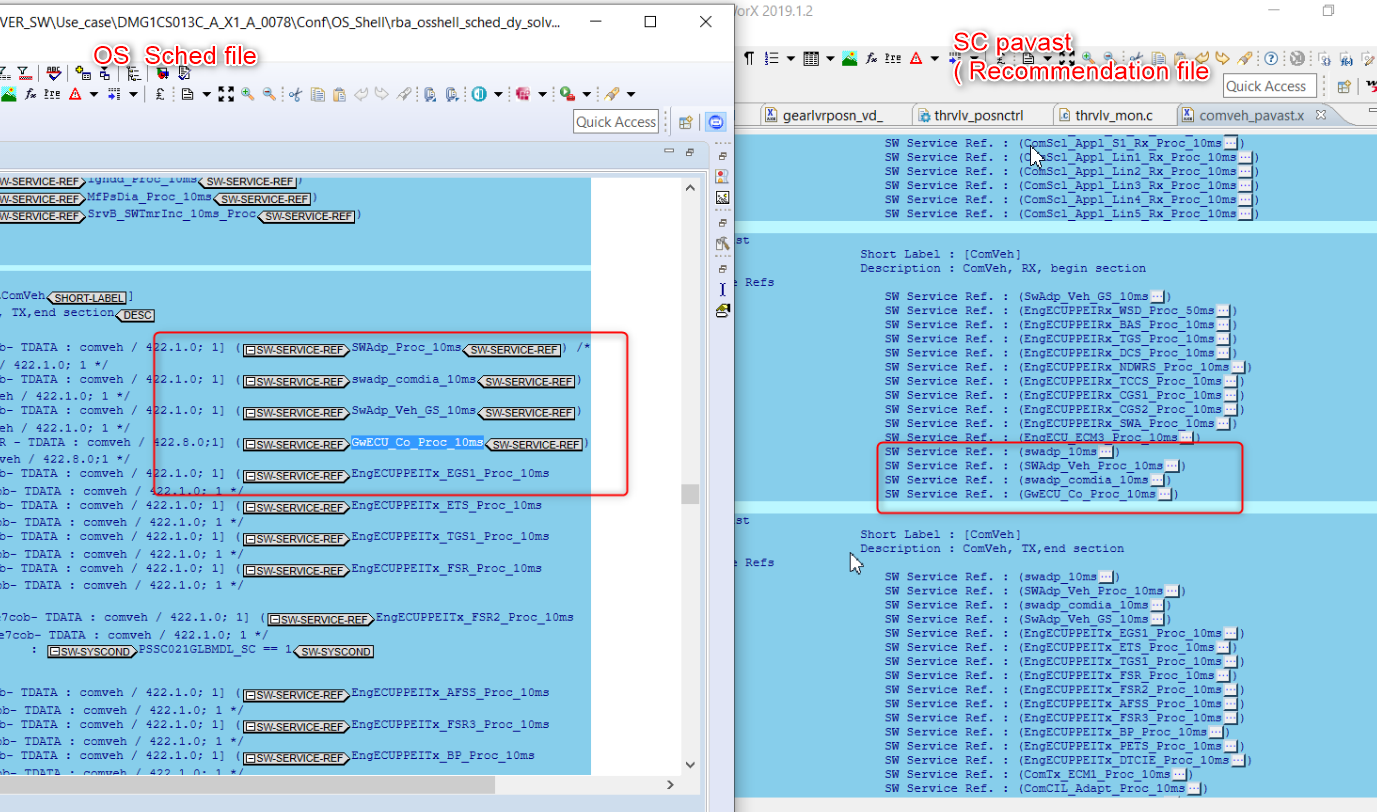
NOTE : Ideally SC(BC:SC:TDATA ) pavast should be checked and in case of missing , then Read-Write order should be followed.

Additionally tealeaves should throw warning CCP17(Procman –Error : Reffered by CE teams)

## Stubbed Missing Proc not in same order as SC pavast(Dysched)

As per process Scheduler file and recommendation file should follow same order.

In the below picture the procs are not ordered as in scheduling file.



NOTE : BC:SC:TDATA is the process recommended order to follown up

## Stubbed Calibration should have user provided value

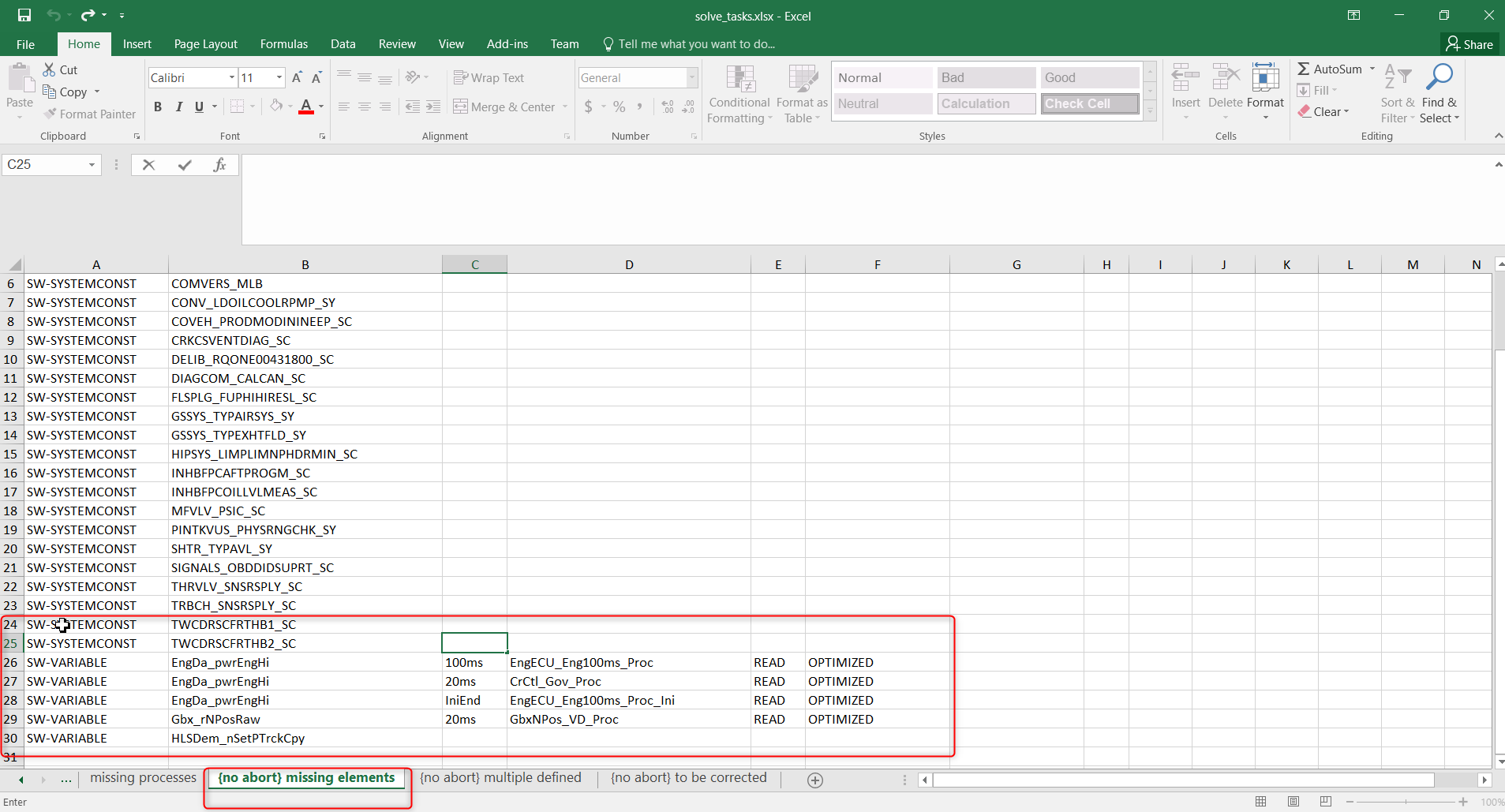
Stubbed Calibration should have option in tool for the user to provide safe value. Currently its 0 by default



Note : Interfacing should be updated with this point

## Changes in solver report for variables

sometimes tool is giving the variable list in no abort(missing elements) page but those variables needed for build and might cause build abort.

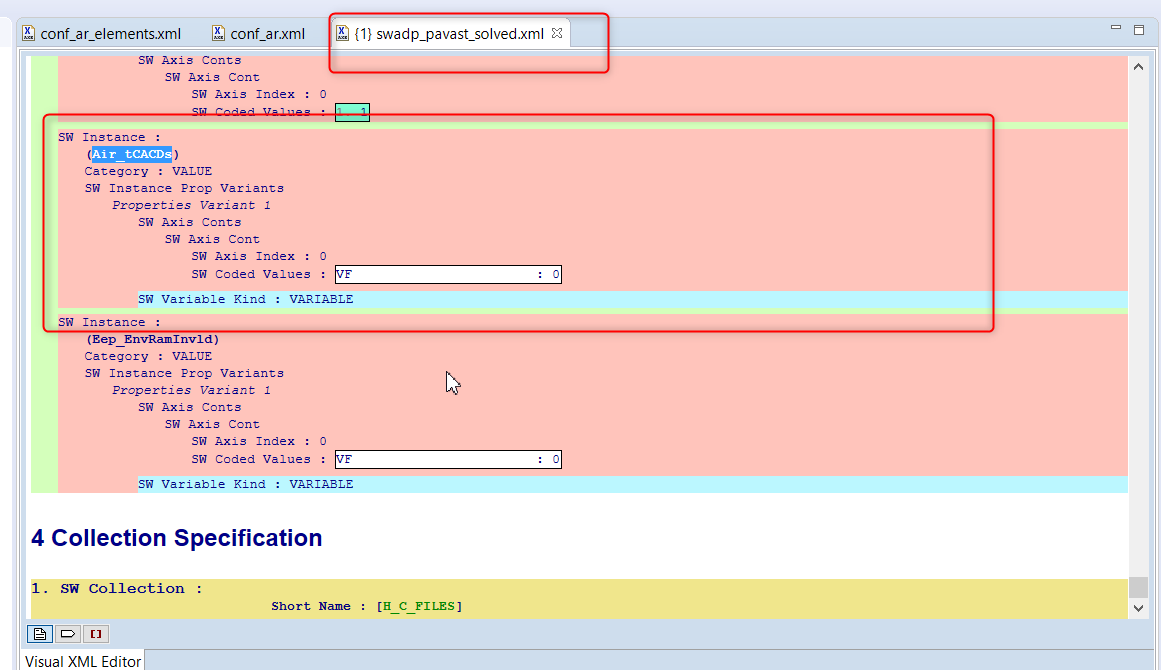


## TeaLeaves should provide 2 seperate option for ADDITON and DELETION of interfaces(system constants..) along with C(\*.h and \*.c) files updation only for SWAdp component

TeaLeaves should provide 2 seperate option for ADDITON and DELETION of interfaces(system constants,compu methods..) along with C(\*.h and \*.c) files updation only for SWAdp component

**Eg**: variable = variable\_C in C/header files

Instead, initialize variables in the code along with fixing of open interfaces in pavast as shown in above example(calibration stubbing).



## LADB support for Auto-Stubbing – Interface Handling

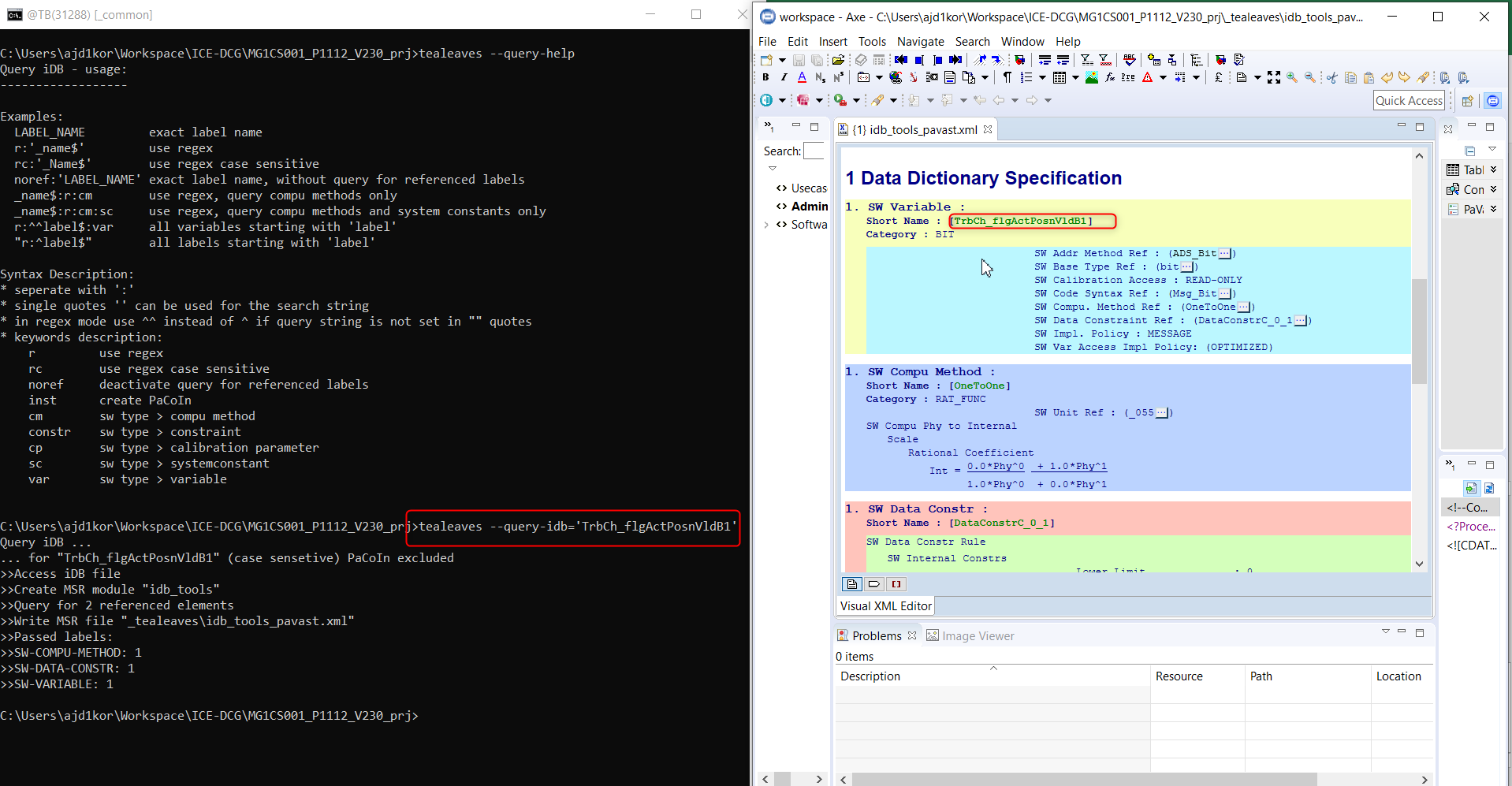
LADB interface for auto-stubbing (Tool only available E2019) should be supported .Tealeaves will query interface directly from LADB.

\*\*Already Supported : Below commands used to querry defination.

|  |
| --- |
| tealeaves --query-idb=QUERY\_IDB |
| tealeaves --query-idb-file=QUERY\_IDB\_FILE |
| tealeaves--query-help |

. Need to verify the Network path provided in code is linked to the LADB.

\\si42361\tme17\_appl$\Programme\STU\DGS\iDB\iDB-DGS\_mini.idb



**Note** : Need to check the difference between idb and LADB

## Auto-stubbing– Interface Handling

Tealeaves should provide option to define and select the right properties of interfaces

Currently Tealeaves provides

-- Choose from idb file //Refer to 2.6 for details

-- PVER selection(local)

Additionally the below possibilities can be supported (order wise)

-- Interfacing of LADB support

-- Appropriate File entry where user can update the interfaces . eg: CSV file

-- Check for Import Defns and update the relevant export interfaces 🡪 Refer 4.1

-- Option in Command Prompt for user inputs . eg: Adding new compu-method

-- Option to select BC/FC in existing PVER

-- Complete PVER

-- GUI option (**more relevant**)

**NOTE**: Interfacing option should also be supporting for adding system conditions in export interface

## Option to select multiple container (FCL/FCP) for different interfaces

Current single containter interface is being provided. Instead, tool should provide options for user to add the interfaces in desired container.

Eg: User may intent to add pwrtrn specific interface in swadp and vehicle specific interface in swadp\_veh

**Note** : CEL : CELPrj / container can be avoided and option should be for FCL/FCP.

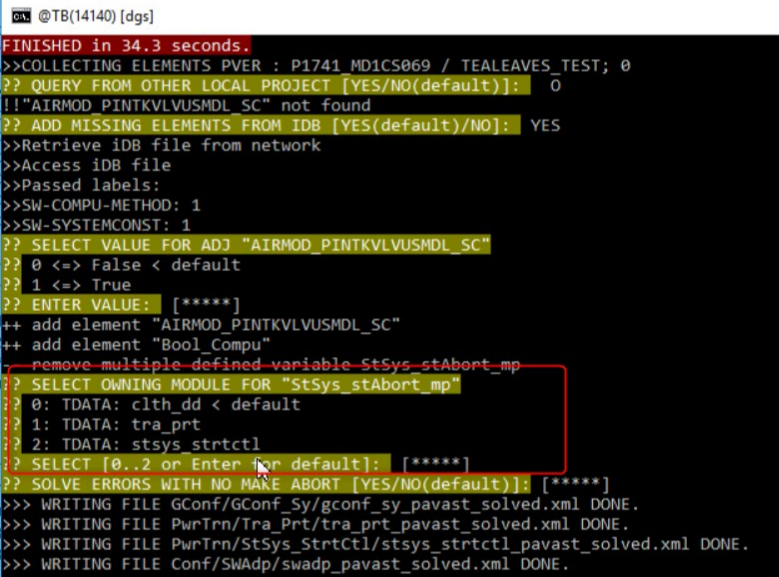
This can also be part of the above requirement(2.7: interface handling)

## List Swadp module for multiple definition (2FC's + 1 Swadp)

Swadp module /entry should be listed by tealeaves which is useful for function delvelopment and integration testing.

Eg: during Integration input was prevously exported from swadp now exported from BC. Integrator may miss on the testing the interface who logic has changed.

Note: SWADP files should be part of finding open interfaces



## Removal of defnitions from .C/.H files

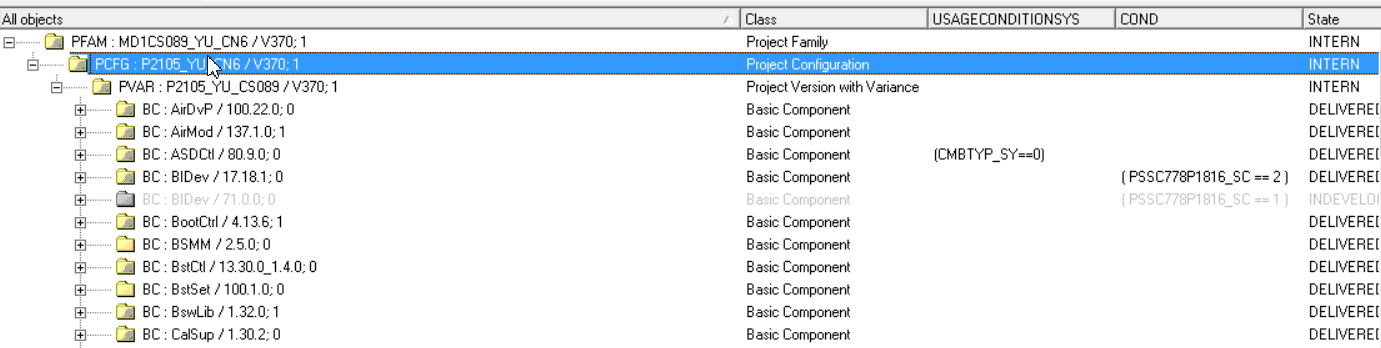
Double definition of interface removed from pavast but not from .c /.h file, which will lead to compiler error

Similar to 2.5 requirement.

Note : Impact of Prediction Mode

## TeaLeaves should support on PCFG component (PVER structure would change)

Presently TeaLeaves supports at PVER level. Support on PCFG component would be benefication for variant handling.

NOTE : Support PCFG might be a complex feature. Need to check if CI is capable in handling. Currently PVER and PVAR is supported 

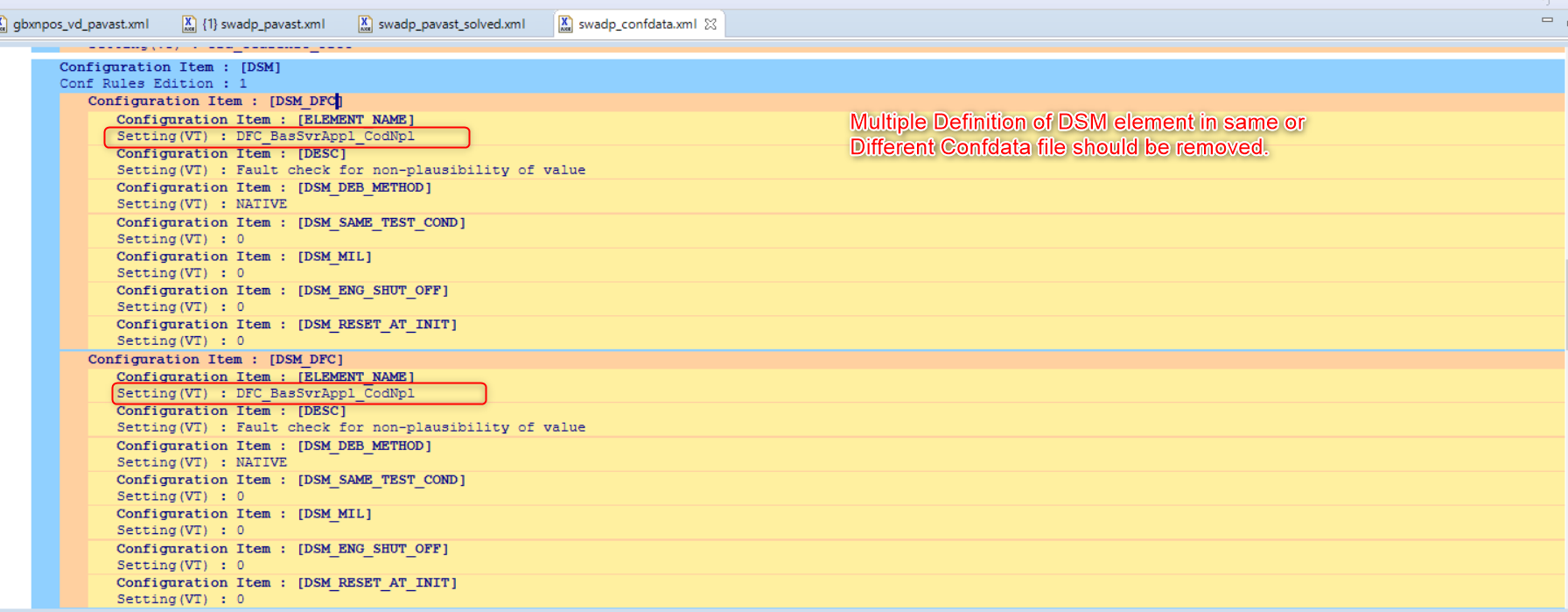
## Multiple defntion of Elements (compu methods/message..) in CEL and SWadap

Listing of CEL and SWadp should be supported by tool in case of multiple definition of Elements(compu methods/message/system constant/cal-params)

**Note: SWADP files should be part of finding open interfaces. Refer to 2.9**

## Removal of Multiple definition in ConfData is not supported

Multiple definition of ConfData(eg:DSM) should be removed in same OR different confdata file .This suppot is mssing in solver mode.

Currently tool is able to Detect in checker mode. 

## Extensibility of TeaLeaves support of automated integration of hints/Cheat Sheets (PST Patching)

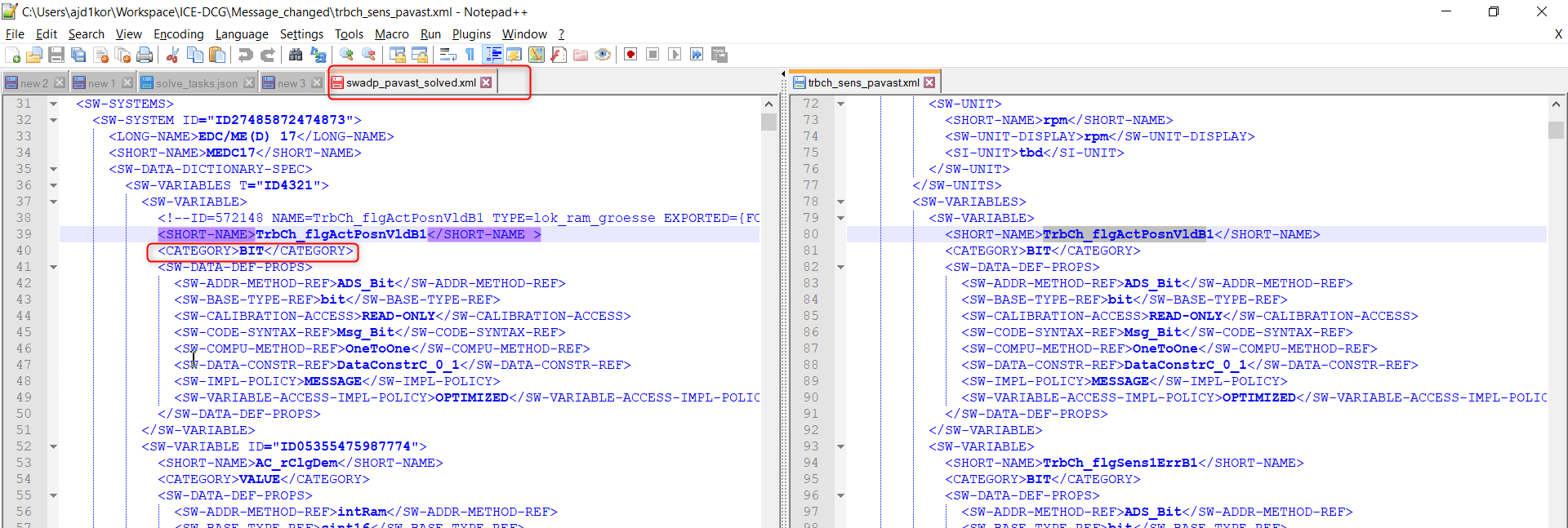
FD's would provide integration hints (raster updates). With this info tealeaves should update the PVER in Sovler Mode. Higher relevance for as part of integration machine (inma).

Note: Redundant data is available during integration hints and might not be feasbile as per Process side

## Support of BIT Variables in Auto-Stubbing

Currently general Variables are supported in stubbing with suitable entries available in swadp\_pavast\_solve.xml file. How ever variables of BIT type are ignored which should be supported

Currently its being supported in latest version



# Improvments under Prediction Mode

## Pre-BCT -> BCT -> Pre-MDGB->MDGB (Patching)

TeaLeaves should predict if there are patches (\*.bat scripts) available in PVER and inform the user on patching files --> two Steps expectations: Inform the user and Execute the path before Tealeaves runs in Checker mode

**Usecase**: Patching/batch scripts are being considered by build and these are invoked before actual build is executed. This is also applicable in CI Dashboard.

Patch files usuallystart with “Pre-BCT” usually present in Misc/MAK folder

**NOTE**: PVER with patches are not applicable for Series Release

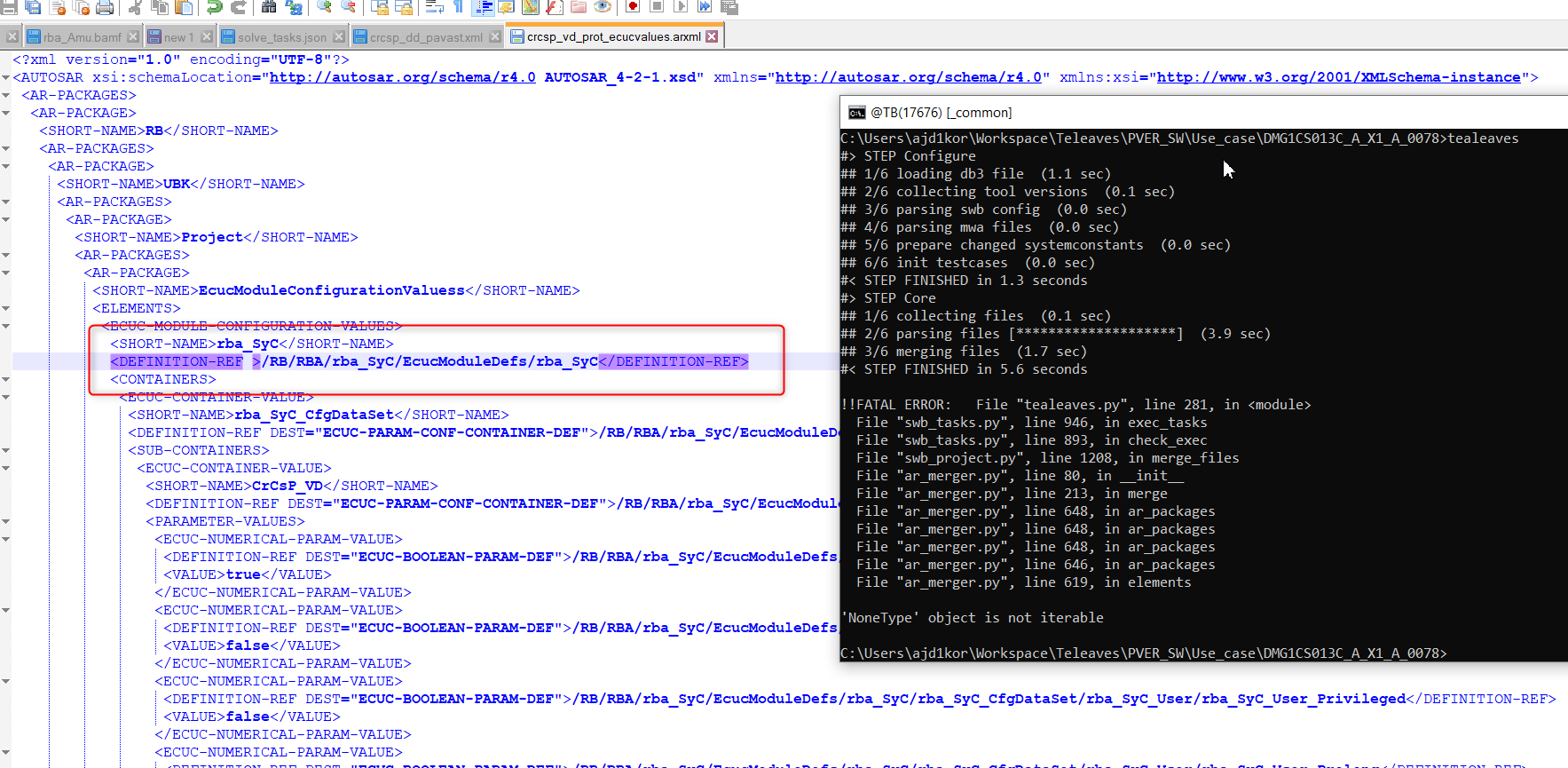
## Handling of improper XML entries in artifact(ArXML)

TeaLeaves should specify if there are missing/mandatory tags in arxml file by specifying the name and location of file in error message.

Current usecase: Missing/Mandator tag information are thrown as Tool errors with code exception stacktrace.

Eg: In the below snapshot the missing tag info DEST="ECUC-MODULE-DEF" for which tool error is thrown

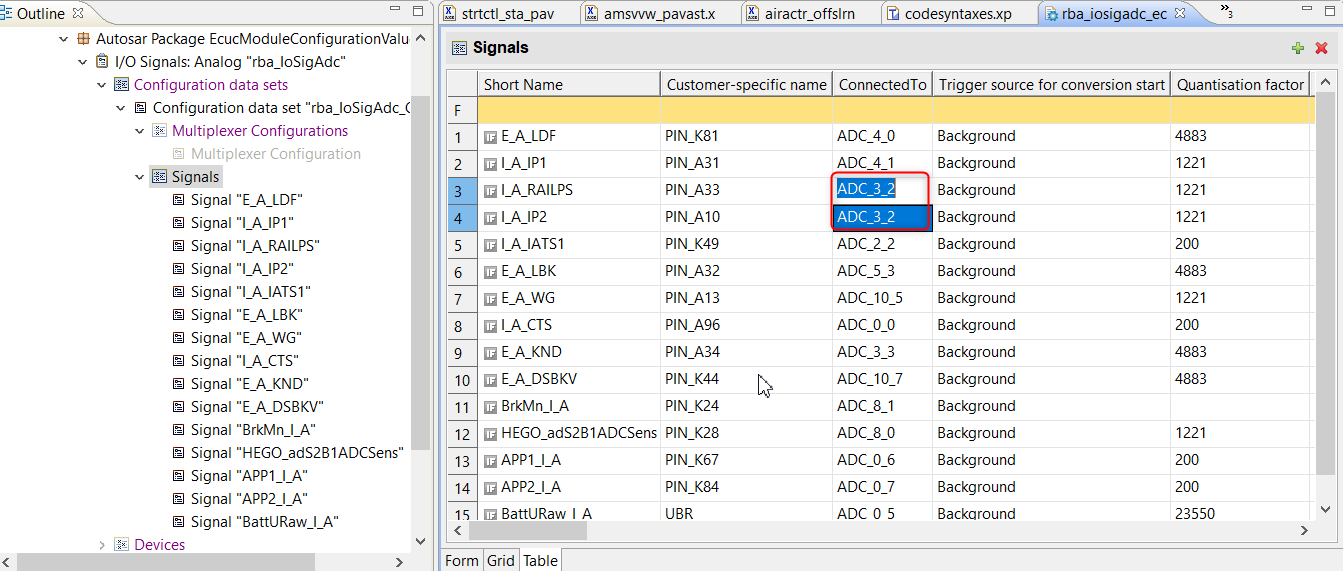
Acutal tag content is <DEFINITION-REF DEST="ECUC-MODULE-DEF"> if DEST is not present then it will show the error as shown in the snapshot

Merged failed since DEST="ECUC-MODULE-DEF" attribute missing in crccsp\_vd\_prot\_ecucvalues.arxml file   
correct tag def <DEFINITION-REF DEST="ECUC-MODULE-DEF"> “

## BCT Validations Not Handled

TeaLeaves doesn’t Handle /Call BCT validations which are required to validate port mappings for ArXML files

Eg: 2 Pins assigned to same port ADC\_3\_2 will through BCT error but will not be caught by tealeaves

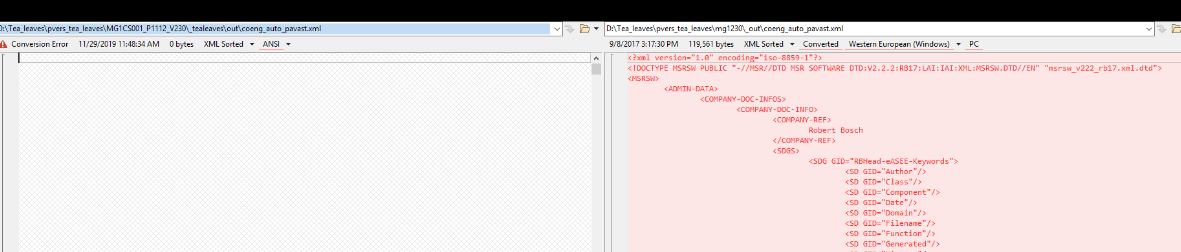


## Differences in generated pavast files

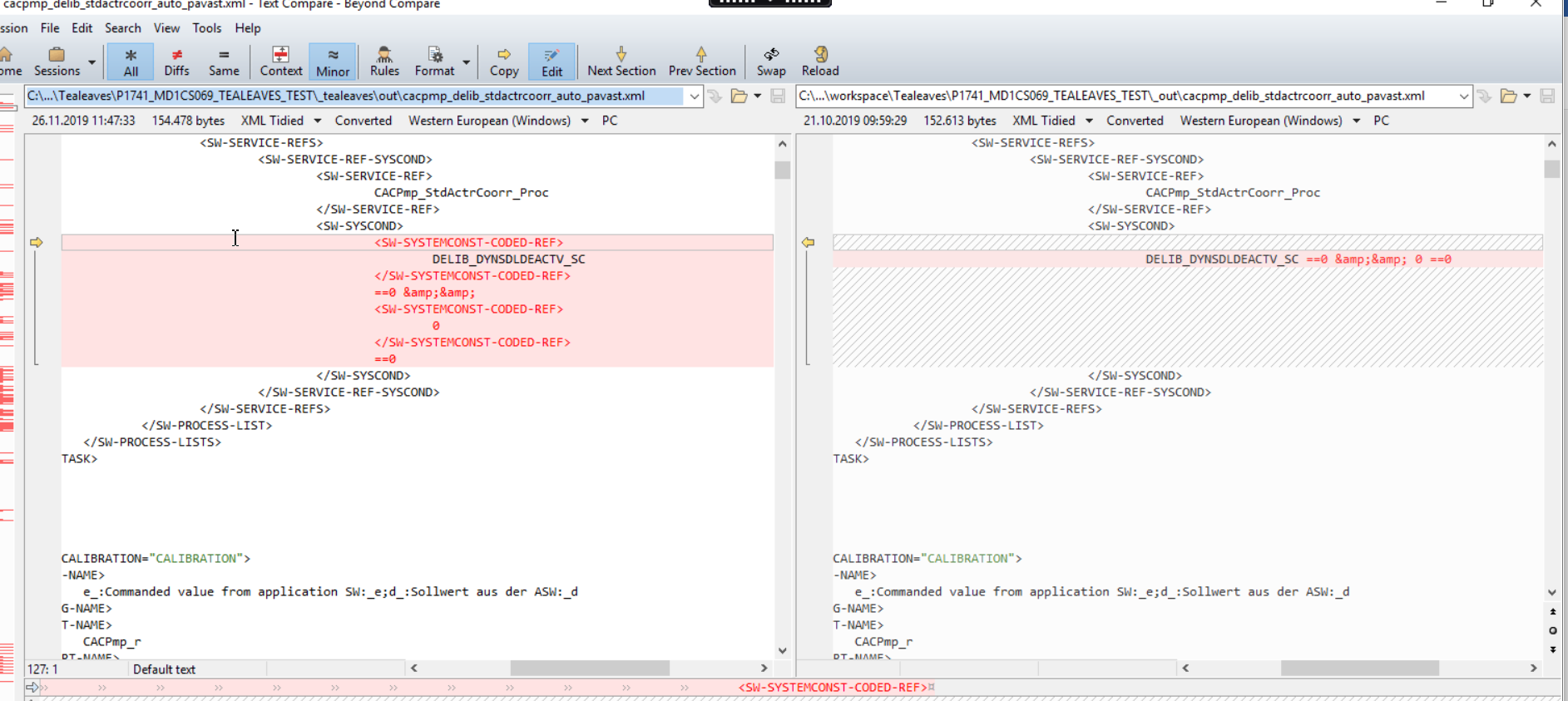
TeaLeaves generates required pavast files which are generated during BCT Build.

eg : (delib\_elecactr\_auto\_pavast.xml).  
Some common differences and observations found between the files are listed below

🡪Empty files generated as shown below

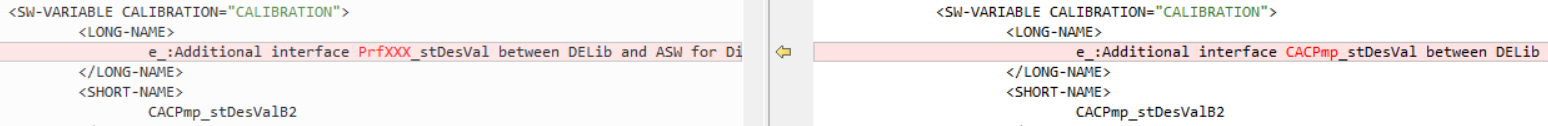


🡪<SW-SYSTEMCONST-CODED-REF> Tag Missing as shown below.



🡪Some special character skipped in generated file. eg Ã¼, Ã¤ , ä

🡪Some Interface name not replaced in <LONG-NAME> as shown below



## TeaLeaves doesn’t validate \*.C files for Validation

Compiler warnings are ignored as the tool doesn’t process \*.C/H files

# General Refinements

## Evaluating SW Label

Specific to FC/BC level function development: .If process information (process name ex: 10ms, 20ms) available, extract module name from process info and open corresponding pavast file and look for properties of the SW label. If not, is there idb to refer?. Else if, need to check in the pver pavast files? Else dummy export should be created finally.

NOTE: The above requirement is part of Interface Handling (2.7)

## Tracking of usage of TeaLeaves

Usage of TeaLeaves should be tracked using TUL

## Detect Common Errors (CORE and DATA steps)

Tool specific (BCT/compiler) can update OR Provide Hint to Tealeaves in case if there are already known errors specific to their release version.

Usecase: Whenever there is a common and known errors from specific tool version then it can be updated in a file format which could be provided to tealeaves and tool can examine before it processes the data which saves time and effort in upfront.

Requestor: Steffen Zoyer

**Eg : Hightec Compiler\_v4.6.6.1-bosch-1.2 Contains bug HDP-1346, HDP-1515, HDP-1568, tool should be given this in file format so that the CE team can check weather they are impacted by the Bug**

## Ignore template (test case rule)

Ignore scenarios can be replicated in test template so that teaLeaves can skip those scenarios and proceed with build prediction.

Hint : Test templates can be reifined

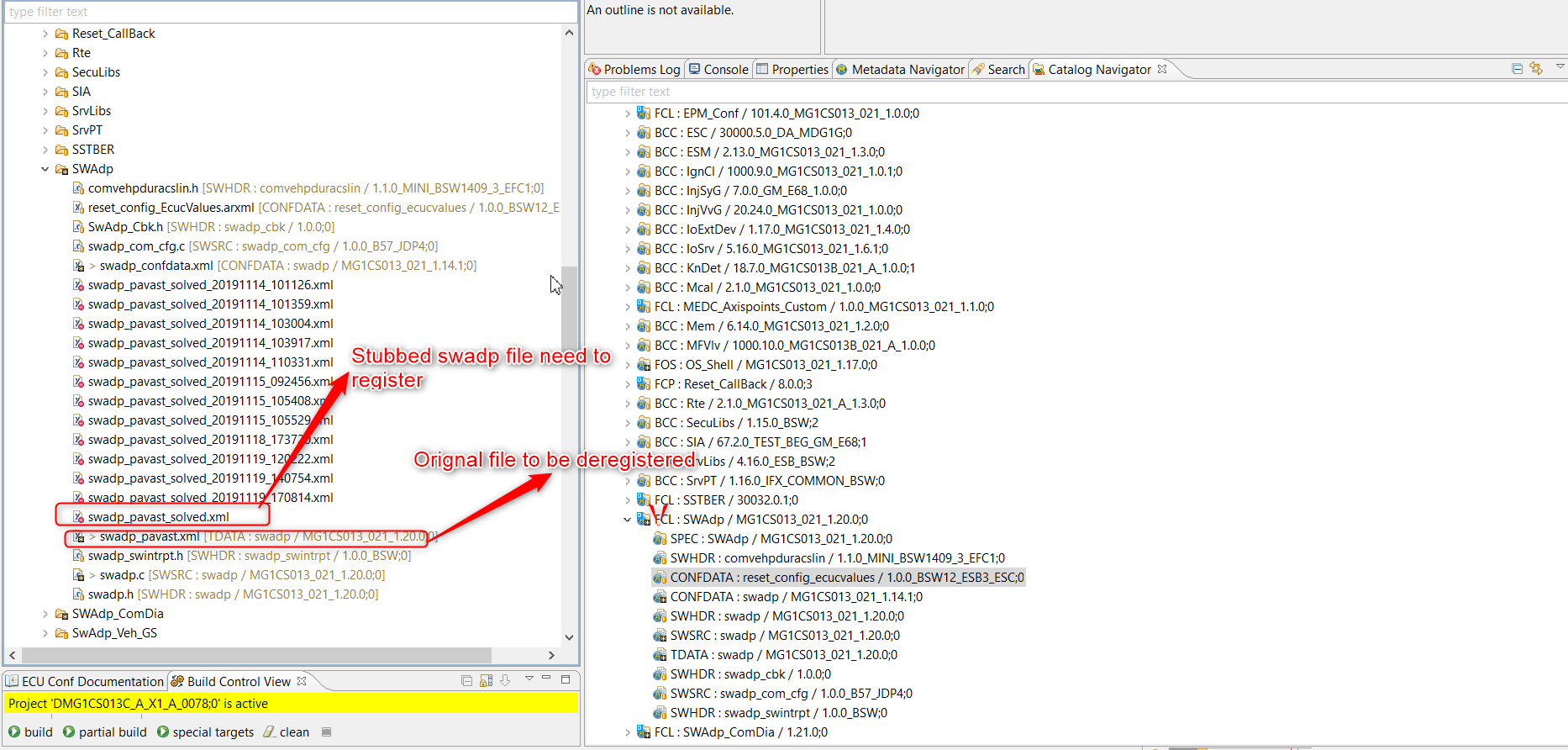
\*not clear about ignore templates.-->Need to check with Hai

# Additional points

## Register stubbed file for Build Execution

Tealeaves should register the stubbed container and deregister the Base container so that build can go through.otherwise stubbed file will not be considered during build. In addition, CI should have gui options or set default option.

Registry of artifacts Option to be provided as its nice to have feature .



NOTE: Tealeaves should ensure naming conventions while registering.

Also SDOM interface should be supported by TeaLeaves

## System Constants- Input File are not stubbed in Solver Mode

In Tealeaves Solver mode, System constants added in Inputfile(sysconst\_defines.txt) or added by command (--sc=systConst\_sy=0) should be stubbed in PVER (gconf\_sy\_pavast\_solved.xml).

The below fig shows how this has been handled till date.

