**Automatic Version Release**

User Guide

November 2016

# Objective and motivation

The objective is to automatically build a version, main inputs are version name and SVN Paths for each CSCI (TAG\BRANCH\TRUNK).

Motivation is to save time and reduce risk for errors during manual build (Human errors).

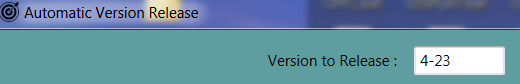
The tool purpose is to perform the same steps which have been performed before by SW Eng. , and by doing so save time, and reduce the risk of mistakes during the version release process.

# Inputs

* **Execute COPY\_FILES\_TO\_C.bat before using the tool for the first time.**

(for creating C:\AutoRelease\ includes all the inputs**)**

1. Version name :



1. XML File which includes general configurations and CSCIs necessary information.

Should be located on C:\AutoRelease\AutoReleaseToolConfigFile.xml

<**GENERAL\_CONFIGURATIONS**>

<LOGGER\_MODE>ERROR</LOGGER\_MODE>

<MAPPING\_SCRIPT\_PATH>C:\AutoRelease\Map.bat</MAPPING\_SCRIPT\_PATH>

<ENV\_PATH>C:/AutoRelease/EFVS/</ENV\_PATH>

<INPUTS\_PATH>C:/AutoRelease/</INPUTS\_PATH>

<OUTPUTS\_PATH>C:/AutoRelease/OUTPUT/</OUTPUTS\_PATH>

</**GENERAL\_CONFIGURATIONS**>

<**CSCI**>

<Name>OFP\_SR1</Name>

<Svn\_Url\_Path><http://svn-subs:18080/svn/efvs/Software/Projects/Dassault/>

Implementation/Source/SR1/OFP\_SR1/trunk</Svn\_Url\_Path>

<Local\_Path\_To\_Checkout>C:\AutoRelease\EFVS\Software\Projects\Dassault\

Implementation\Source\SR1\OFP\_SR1</Local\_Path\_To\_Checkout>

<Scade\_Script\_Path>J:\Implementation\Source\SR1\OFP\_SR1\

SCADE\_KCG\_OFP\_SR1.bat</Scade\_Script\_Path>

<Build\_Script\_Path>C:\AutoRelease\BUILD\_OFP\_SR1.bat</Build\_Script\_Path> <Dependents\_CSCIs>COMMON,COMMON\_SR1,COMMON\_SR1\_SR2

</Dependents\_CSCIs>

<Is\_Sysint>0</Is\_Sysint>

<Is\_Checkout\_Even\_If\_Already\_Exists>1</Is\_Checkout\_Even\_If\_Already\_Exists>

</**CSCI**>

1. XML File which includes CSCIs version data (used for LUH version file and versions acti)

Should be located on C:\AutoRelease\VersionsLibConfigFile.xml

# Process Overview

Process of automatic build :

1. **Checkout** environment into <ENV\_PATH>

For each CSCI : From <Svn\_Url\_Path> to <Local\_Path\_To\_Checkout>.

1. **MAP** environment – execute script <MAPPING\_SCRIPT\_PATH>

* **For each CSCI** from <CSCI\_LIST

Check if CSCI should be **built** - >

If configured to Trunk\Branch and CSCI (or Dependent CSCI\*) Changed from last TAG.

(Compare revision of TRUNK to revision of TAG according to version file):

\*dependent CSCIs will be build before. <Dependents\_CSCIs>

In case CSCI should be build :

* + 1. Generate **SCADE** – execute <Scade\_Script\_Path>.
    2. Increase CSCI **version**.
    3. In case <Is\_Sysint> - Align LUH versions file.
    4. **Build** CSCI – execute script <Build\_Script\_Path>.
    5. **Commit** changesand **TAG** CSCI **\***
    6. In case <Is\_Sysint> - ZIP BIN folder to <OUTPUTS\_PATH>.

1. Wrap-up :
   1. **ZIP environment** into <OUTPUTS\_PATH>.
   2. Create **VDD** in <OUTPUTS\_PATH>.
   3. Copy <OUTPUTS\_PATH> to SW Deliveries folder on network . **\*\***
   4. **Notify** to relevant people of version release – send **email**. **\*\***

**\*** Will be implemented gradually (for reducing risk of causing unwanted SVN operations) :

1.**First step** : Commit and Tag will be done manually – user will be announced that CSCI is ready

for commit and tag.

e.g:

2. **Second step** : Commit and Tag automatically after user confirmation.

**\*\*** Future implementation (upon built using tool will be used regularly).

# Outputs

1. Products of SYSINT (zipped BIN folder)
2. Zipped full environment for released version.
3. VDD includes all CSCIs versions on released version.

# Required

1. .NET framework 4.5.
2. SVN installed.
3. SCADE installed.
4. Windriver installed.

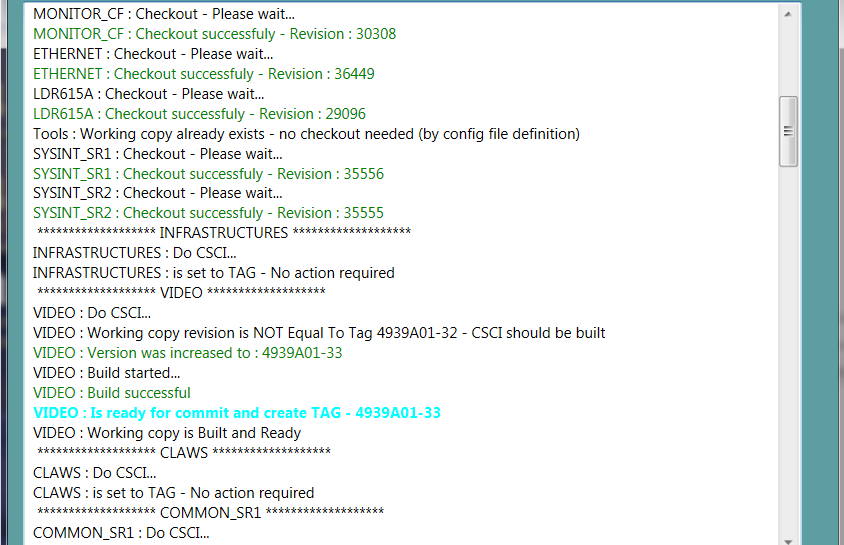
# Example

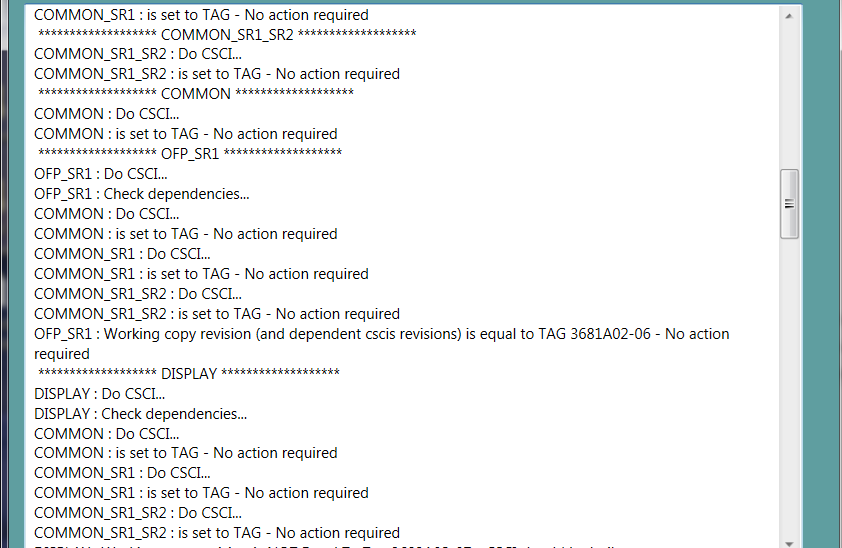
## Input configuration file

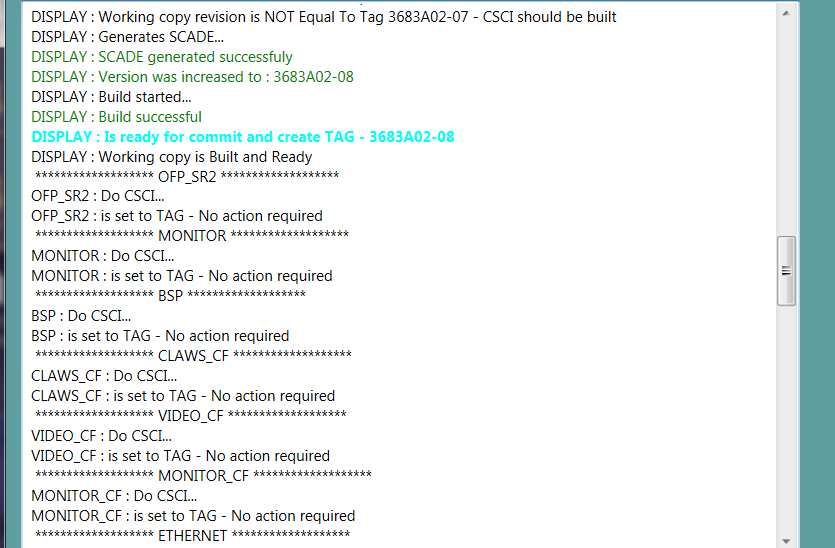


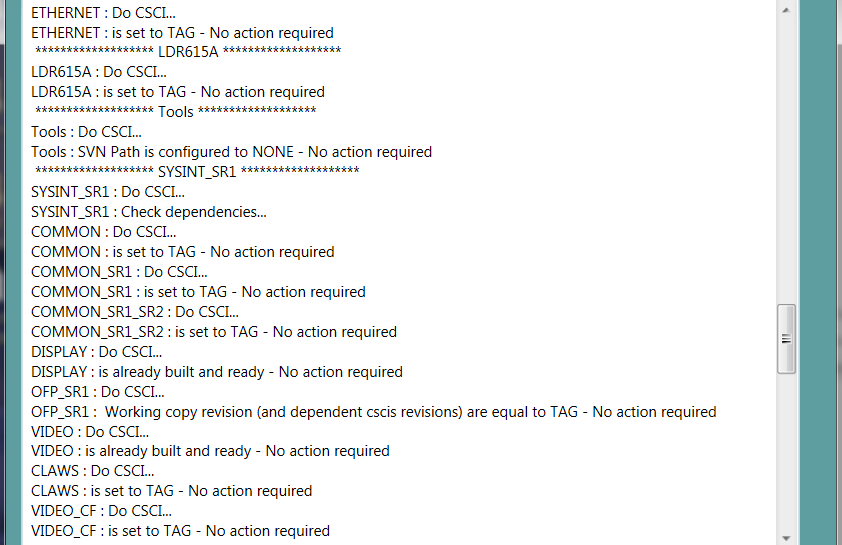
## Execution results

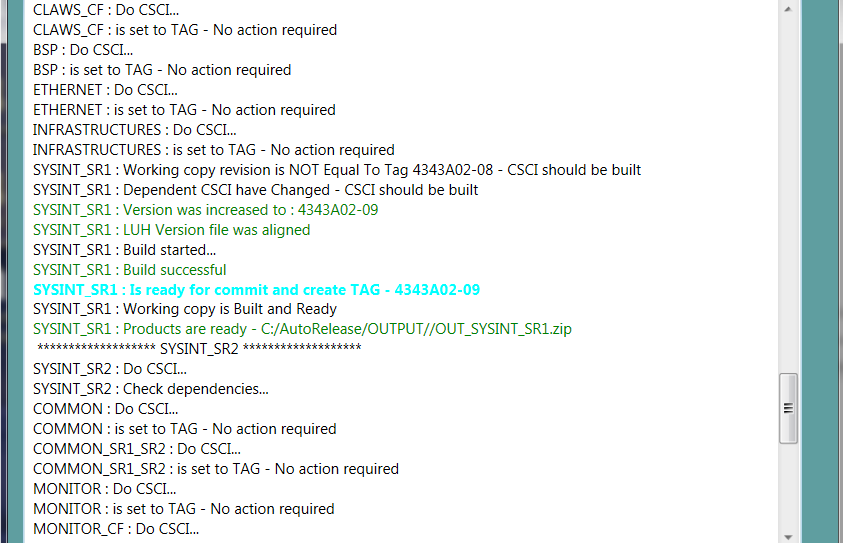


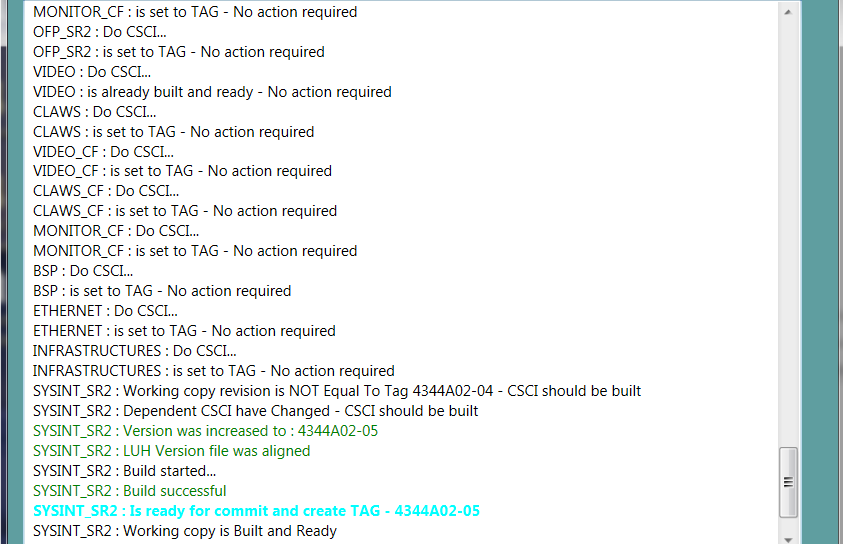


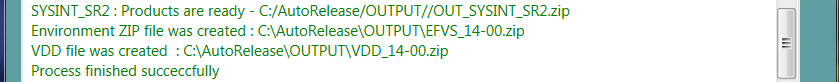




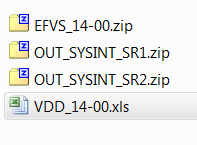








## Outputs



VDD : 