Contents

[1 General Overview 1](#_Toc422909339)

[2 Installation of SharCC Gen 9.3 3](#_Toc422909340)

[3 Connection to the project in SharCC 5](#_Toc422909341)

[4 Checkout project 7](#_Toc422909342)

[4.1 From a Stream 7](#_Toc422909343)

[4.2 From a Snapshot 17](#_Toc422909344)

[5 Build software 27](#_Toc422909345)

[6 Update ASW sources 28](#_Toc422909346)

[6.1 Update a Tarball 28](#_Toc422909347)

[6.2 Add new ASW component 28](#_Toc422909348)

[6.2.1 Copy the file in workarea 28](#_Toc422909349)

[6.2.2 Add PDM files to file list 29](#_Toc422909350)

[6.2.3 Add the new processes in process list 30](#_Toc422909351)

[6.3 DCM files 31](#_Toc422909352)

[6.3.1 Update DCM files 31](#_Toc422909353)

[6.3.2 Add new DCM files 31](#_Toc422909354)

[7 Checkin updates 32](#_Toc422909355)

[8 Components 36](#_Toc422909356)

[8.1 Create a new component 36](#_Toc422909357)

[8.2 Add files in a component 38](#_Toc422909358)

[9 ASW Simulation Build 52](#_Toc422909359)

[9.1 Update Simulation files 52](#_Toc422909360)

[9.2 Add a file in bcfg for ASW Simulation 53](#_Toc422909361)

# General Overview

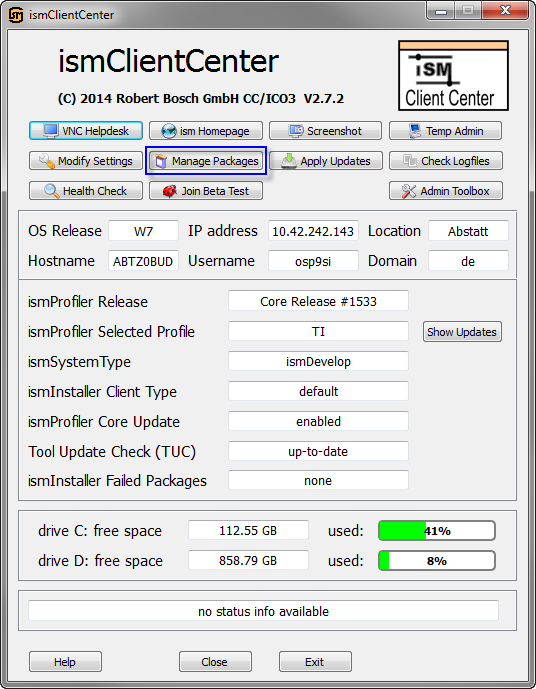
This document will explain how to:

* Install the tools needed to compile software in Gen 9.3
* Run a compilation
* Archive the updates

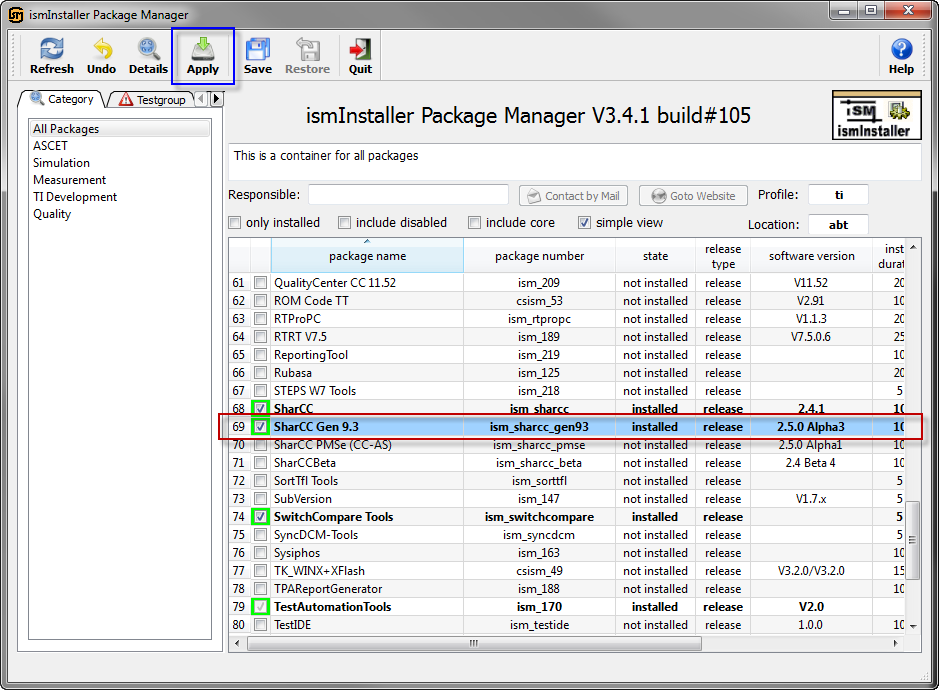
# Installation of SharCC Gen 9.3

Open ismClientCenter

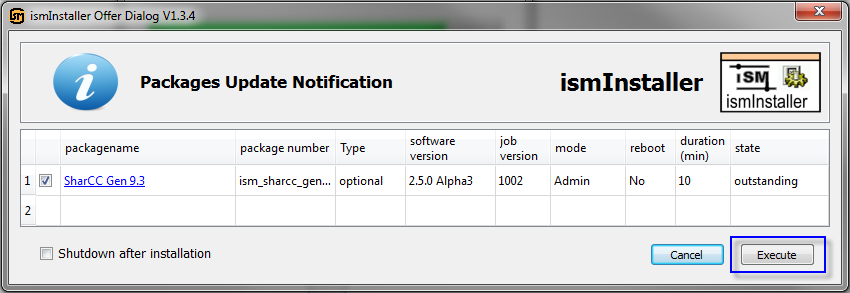
* Click on “**Manage Packages**”:



* Select the package “**SharCC Gen 9.3**”
* Click on “**Apply**”:

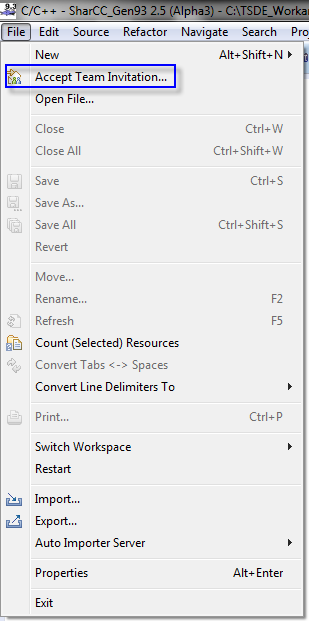


Once the package is downloaded, run the installation of SharCC in ismInstaller:



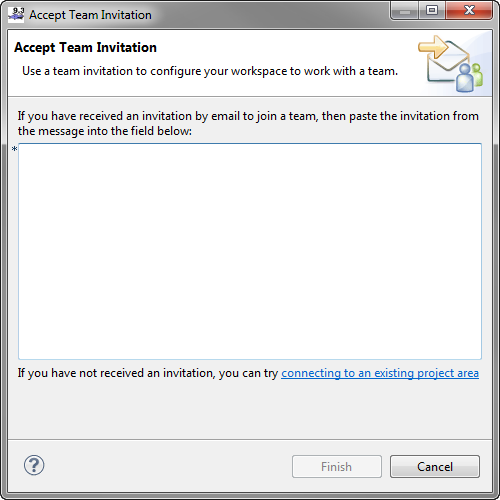
# Connection to the project in SharCC

Open SharCC\_**Gen93** [](file:///C:\tsdct\SharCC_Gen93\sharcc_gen93.exe)



In SharCC, go to File

* Accept Team Invitation...

Copy and paste the invitation you received by email:

teamRepository=https://rb-alm-06-p.de.bosch.com/ccm/

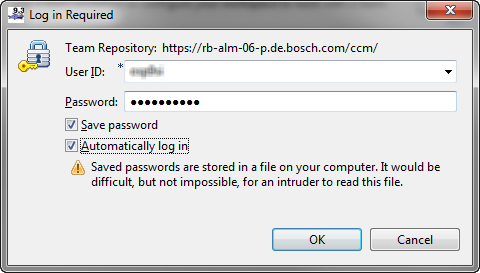
userId=*user Id*

userName=*User Name*

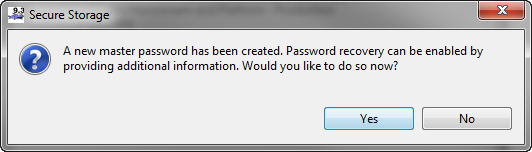
projectAreaName=Mainstream and Platform - Planning

teamAreaPath=/IPB

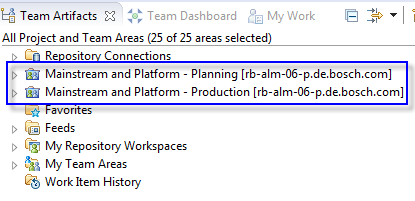
Enter your User ID and Password



Optional: Provide information to recover your password:



The new Team Workspace will be added on the left side with the following icon: 



# Checkout project

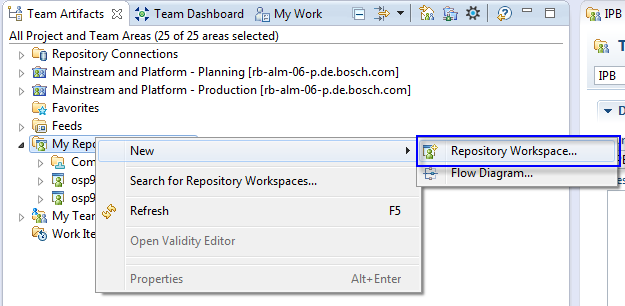
On the upper-right corner, go to “**Work Items**”:



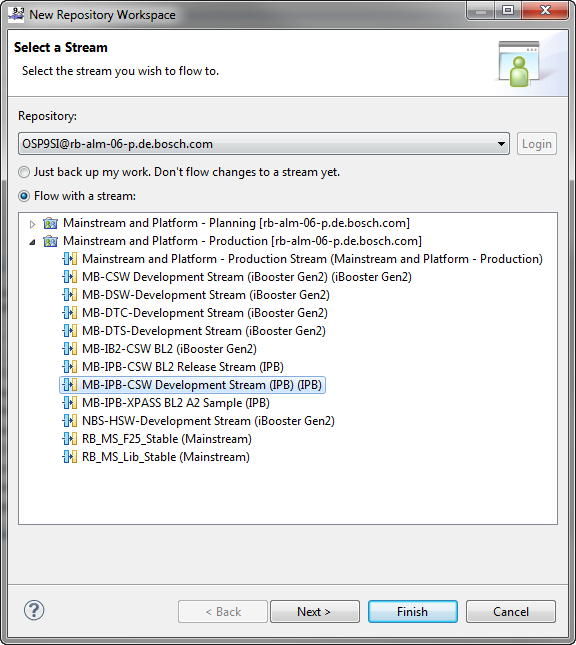
## From a Stream

Right click on:

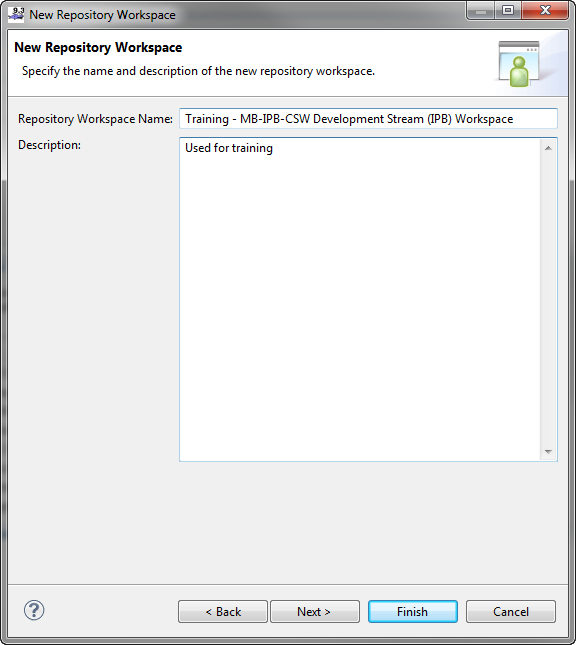
* My Repository Workspace
* New
* Repository Workspace...



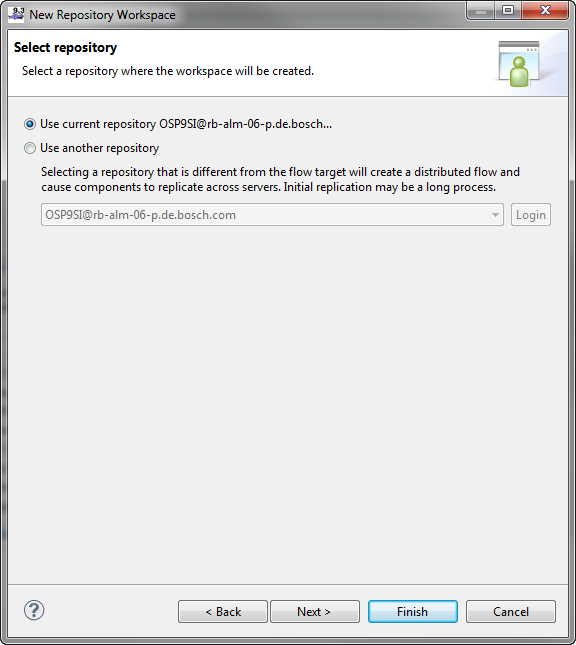
Select the Stream your want to use:



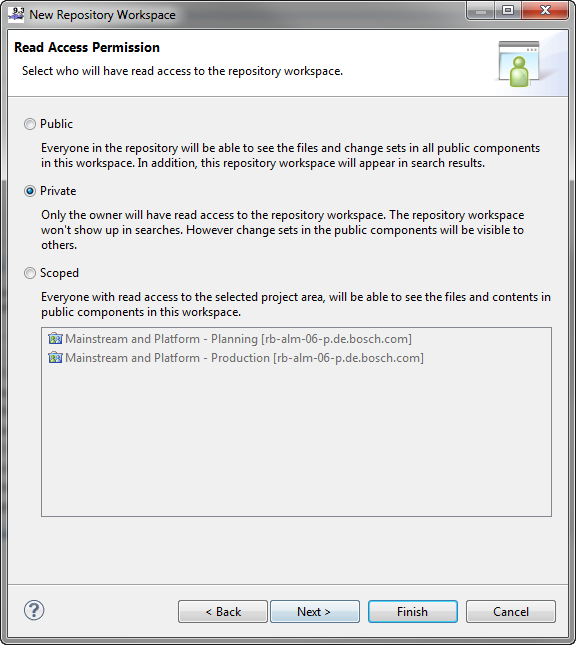
Rename the Repository Workspace



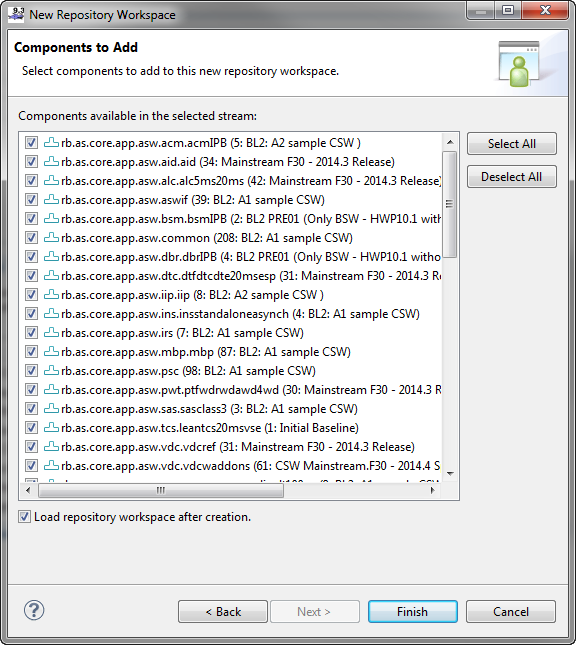
Select “Use current repository...”



Read Access: Usually “**Private**”

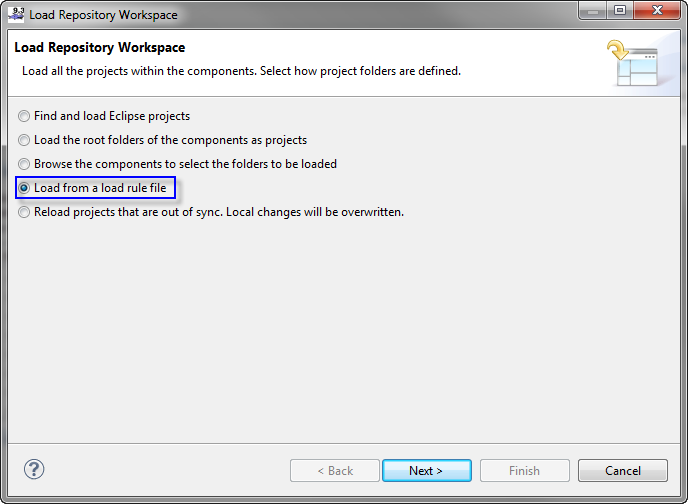


* Check that all components are selected
* Click on “**Finish**”

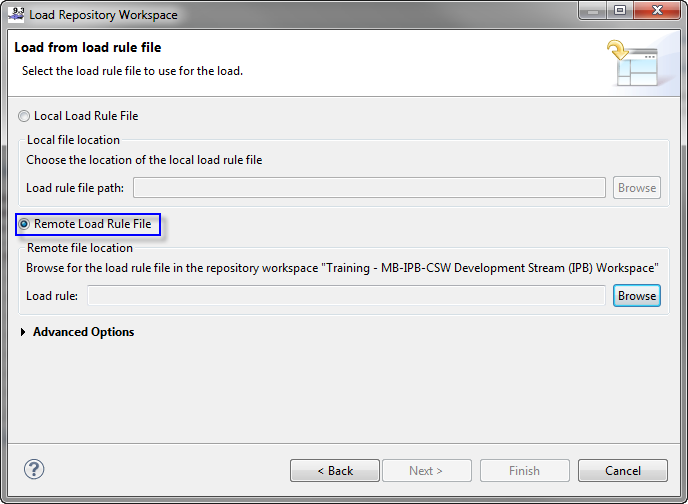


On the new window:

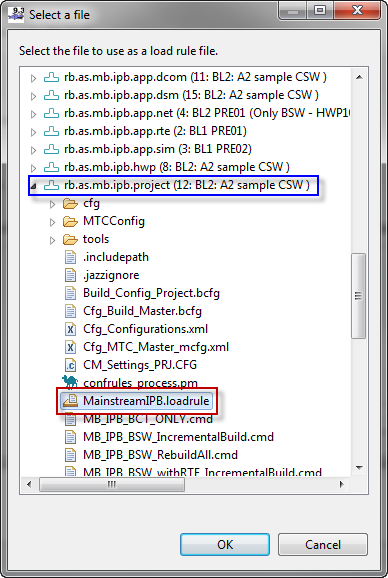
* Select “**Load from a load rule file**”
* Click on “**Next**”



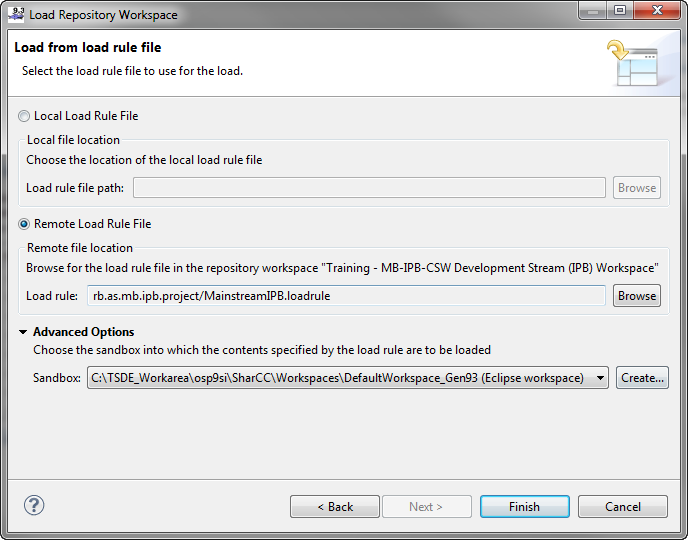
* Select “**Remote Load Rule File**”
* Then “**Browse**”



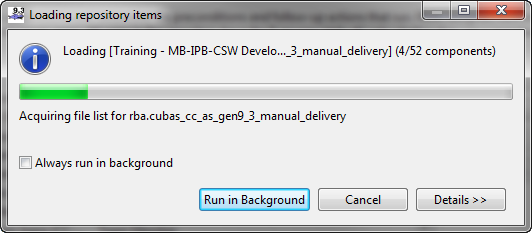
* The Load Rule File should be located in: rb.as.mb.*projectname*.project.
* The name of the file is *xxxx*.loadrulefile:



Optional: it is possible to change the sandbox location in “Advanced Options”:



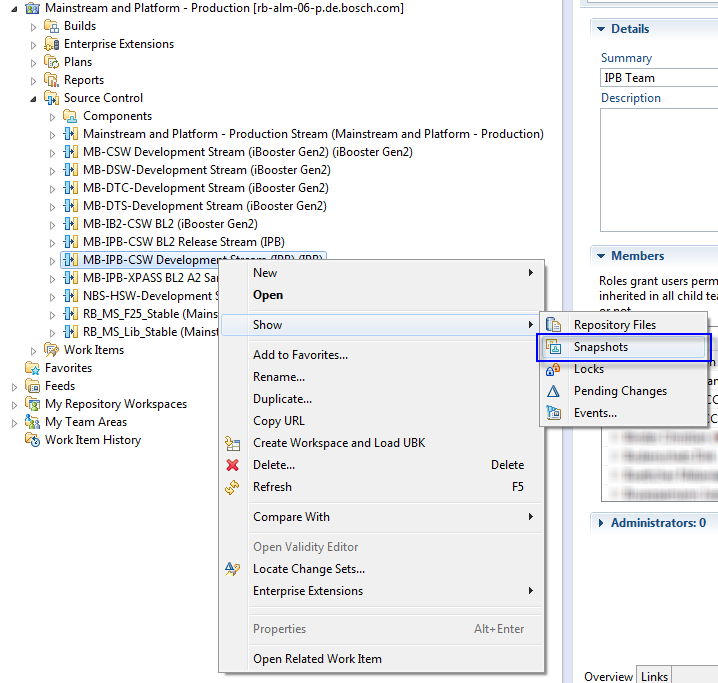
Press “**Finish**”, the download of the files begins:



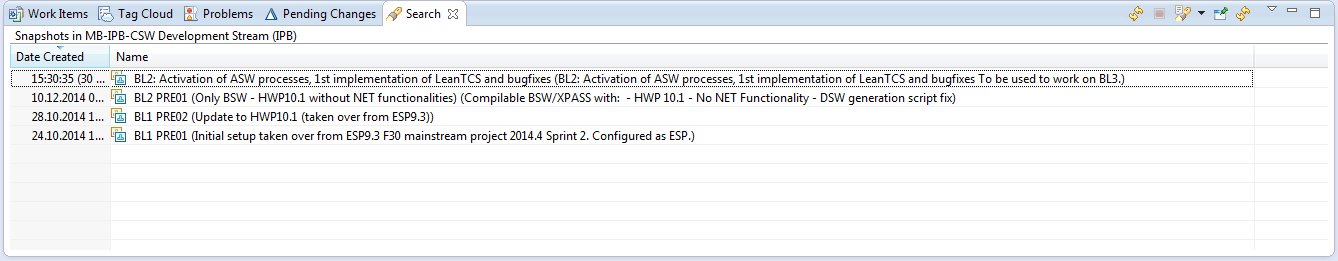
## From a Snapshot

Snapshots are usually compiling software which can be used as base for new development. Here is the procedure to add it in the workarea:

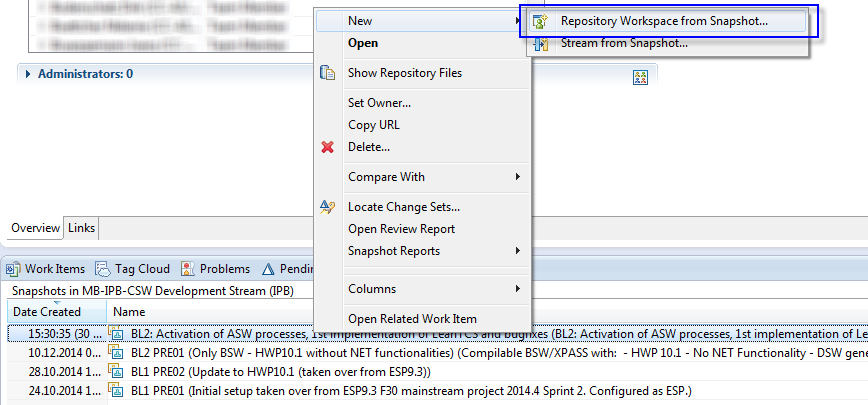
* In the “**Source** **Control**”, right click on the project you need
* “**Show**” 🡪 “**Snapshots**”



The list of the snapshots will be displayed on the bottom of SharCC:

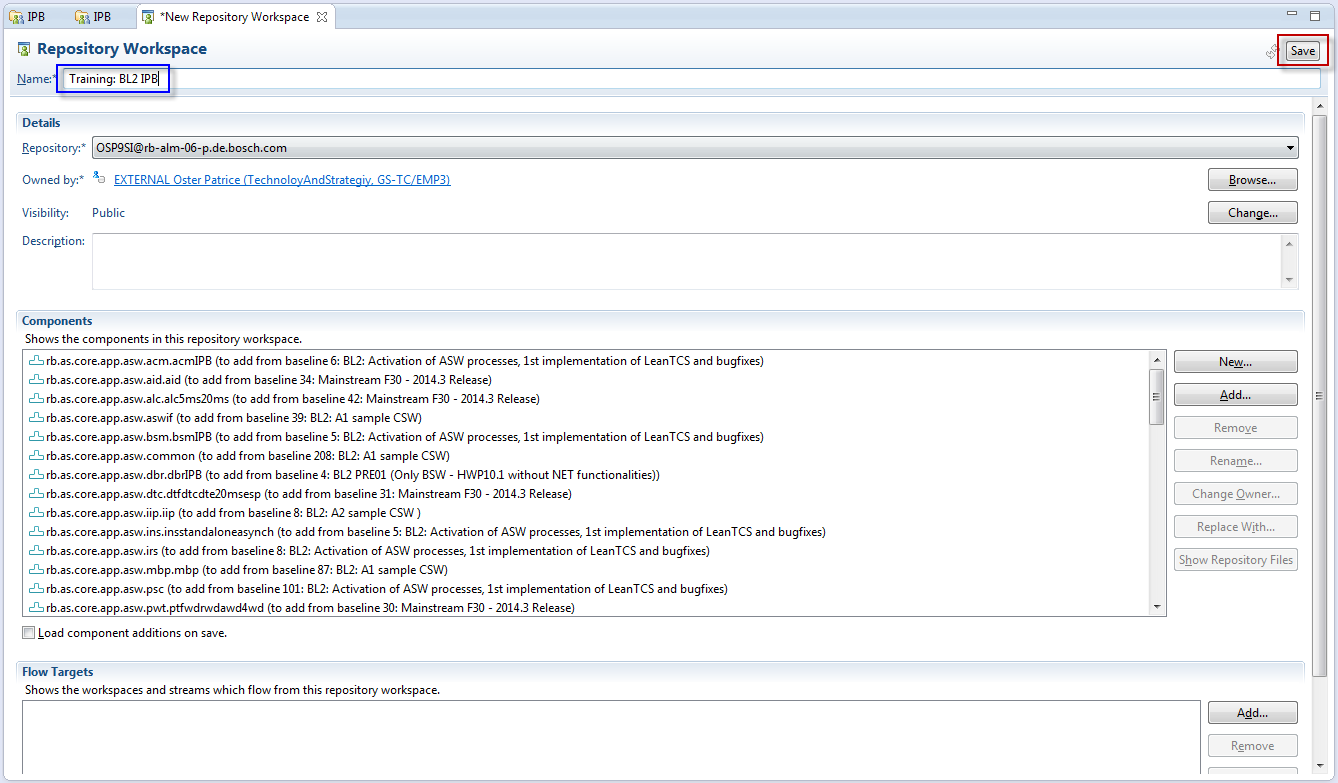


* Right click on the Snapshot to download
* “**New**” 🡪 “**Repository Workspace from Snapshot...**”



On the central window:

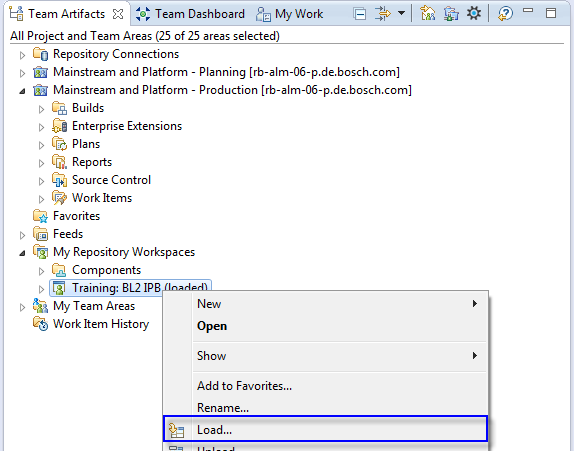
* Rename the Repository Workspace
* Click “**Save**”



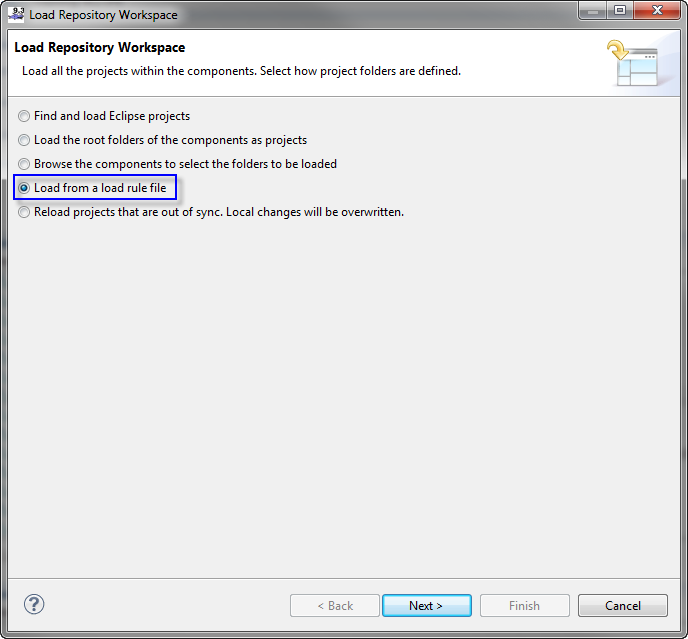
The new workspace has been created, but the source files are missing and the flow target is not defined.

Load the source files:

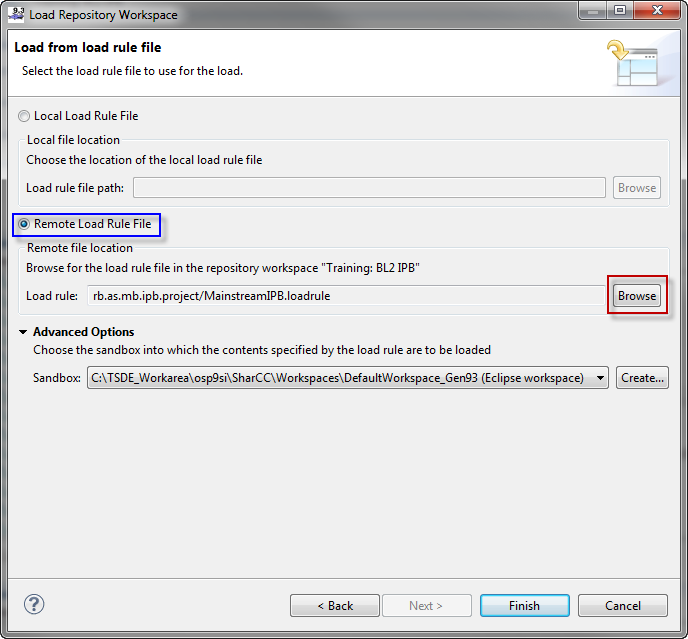
* Right click on the workspace name
* Click on “**Load...**”



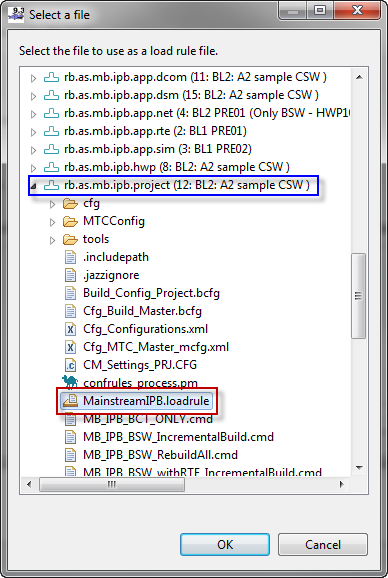
* Select “**Load from a load rule file**”
* Click on “**Next**”



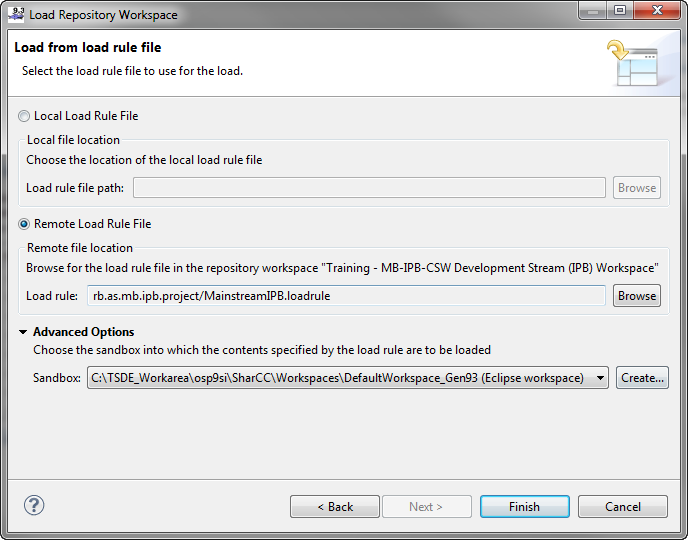
* Select “**Remote Load Rule File**”
* Click on “**Browse**”



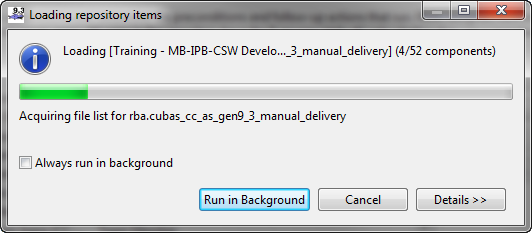
* The Load Rule File should be located in: rb.as.mb.*projectname*.project.
* The name of the file is *xxxx*.loadrulefile:



Optional: it is possible to change the sandbox location in “Advanced Options”:

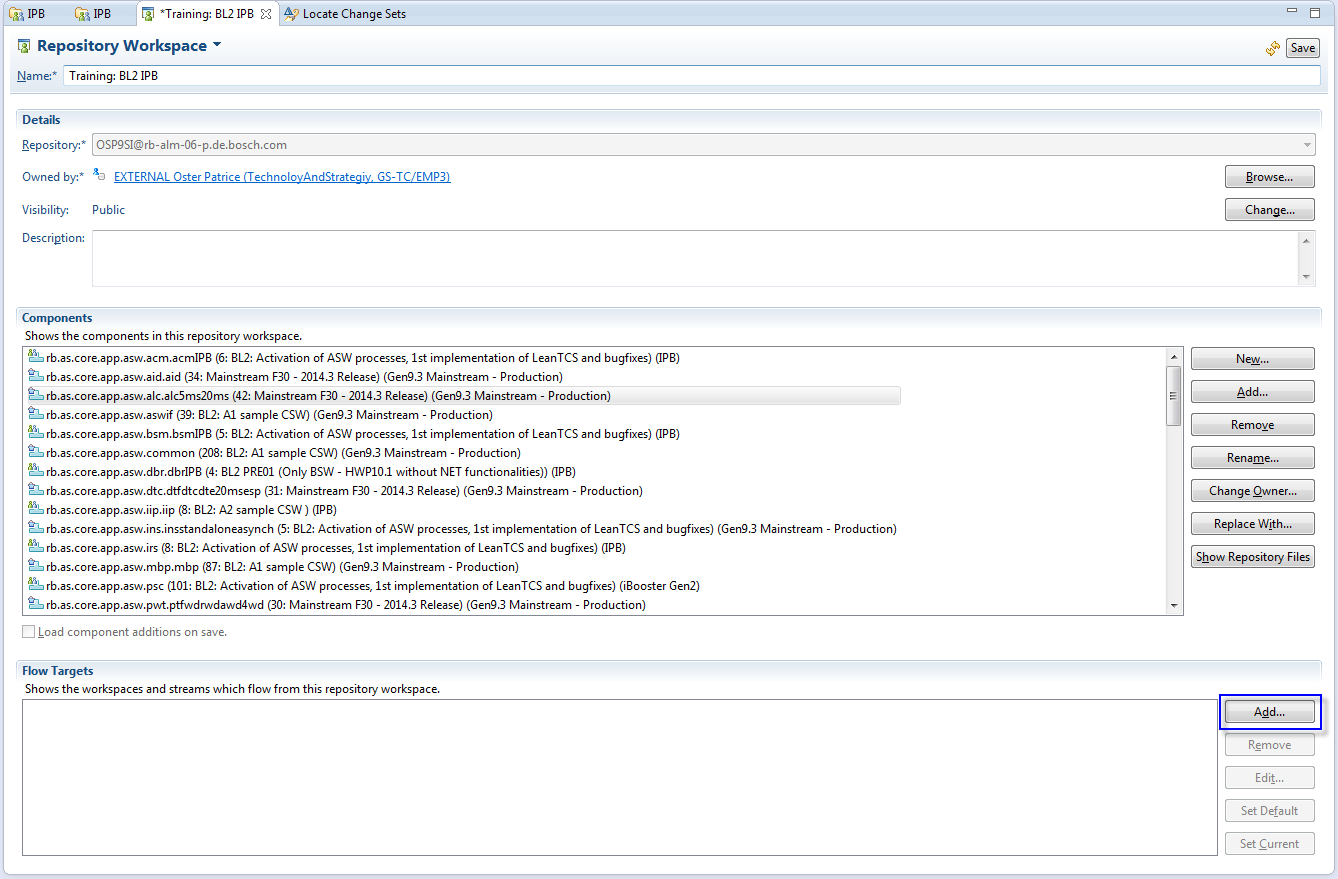


Press “**Finish**”, the download of the files begins:

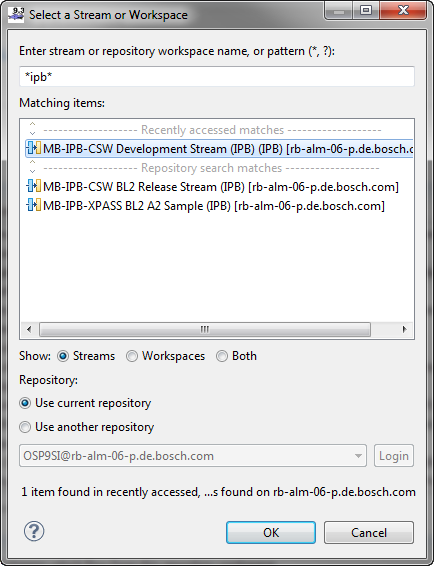


Add a flow target:

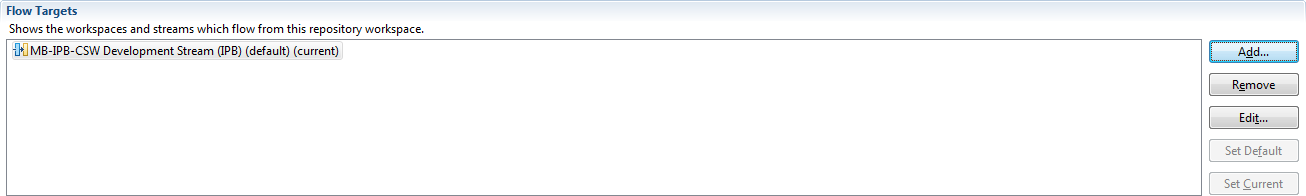
* Click on “Add”



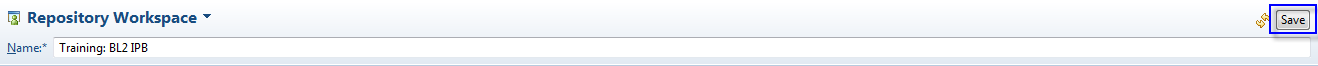
* Select the flow target in the list:
* Click on “**OK**”



* The new target is added as default and current:
  + Default: Flow there the code is archived
  + Current: Flow where the code is downloaded



* Save the project

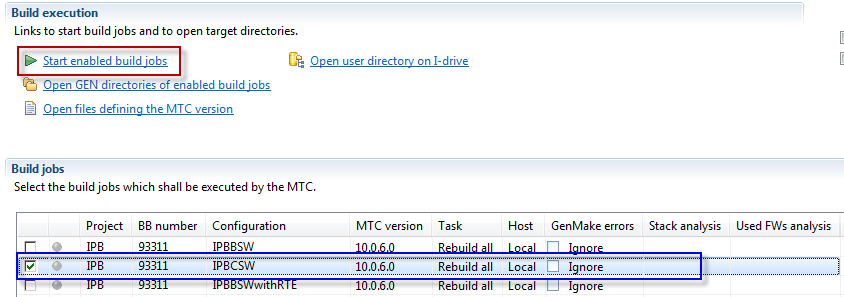


# Build software

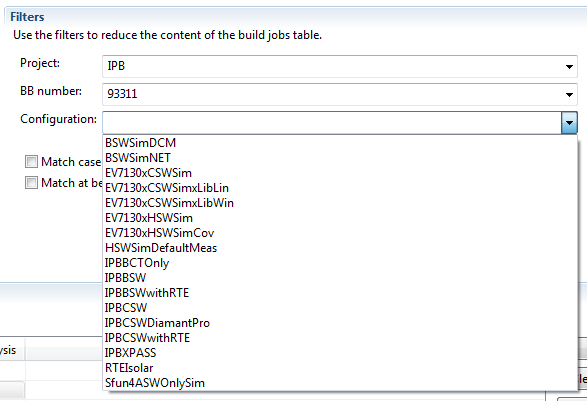
To build software, go to the MTC Build view (on the upper right corner):



* Select the project to build in the list
* Click on “**Start enabled build jobs**” to run the compilation



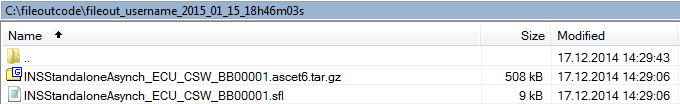
It is possible to add filters on the Project, BB Number and Configuration:

****

# Update ASW sources

## Update a Tarball

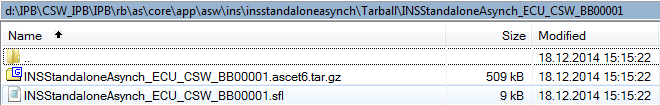
Run the compilation of the tarball in ASCET. Once the files have been generated, they are located in: C:\fileoutcode\fileout\_*username*\_*compilation*\_*date*



Copy the files to the location of the tarballs in the project. The ASW files are in the folder:

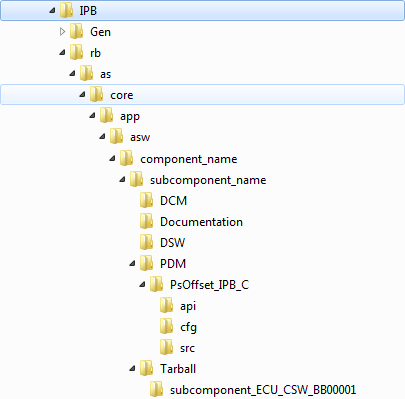
rb\as\core\app\asw\*component\_name\subcomponent\_name\*

Note: if the tarball has been compiled in local mode, remove “\_local” from the name.



## Add new ASW component

### Copy the file in workarea



To add a new component to the project, create a new folder in:

rb\as\core\app\asw

Create the same tree as the screenshot.

Note: PDM and DSW folders are optional.

To compile the new tarballs, they have to be included in the Tarball list:

*ProjectWorkspace*\IPB\rb\as\mb\ipb\app\asw\config\AscetCodeGenerationTarBalls.XML

Open the file and add these informations:

<AscetProjectSettings>

<AscetASW>rb/as/core/app/asw/*Component*/*Subcomponent*/Tarball/*SubComponent\_ECU\_CSW\_BB00001*/*TarballName.ascet6.tar.gz*</AscetASW>

<OptimizerSettings>rb/as/mb/ipb/app/asw/config/Mainstream\_Finalize\_No.xml</OptimizerSettings>

<GenerationOptions>rb/as/core/app/asw/*Component*/*SubComponent*/Tarball/*XmlFileName*.xml</GenerationOptions>

</AscetProjectSettings>

Example:

<AscetProjectSettings>

<AscetASW>rb/as/core/app/asw/*acm*/*acmIPB*/Tarball/*AcmIPB\_ECU\_CSW\_BB00001*/*AcmIPB\_ECU\_CSW\_BB00001.ascet6.tar.gz*</AscetASW>

<OptimizerSettings>rb/as/mb/ipb/app/asw/config/Mainstream\_Finalize\_No.xml</OptimizerSettings>

<GenerationOptions>rb/as/core/app/asw/*acm*/*acmIPB*/Tarball/*ACM\_ECU\_CSW\_BB00001*.xml</GenerationOptions>

</AscetProjectSettings>

Note: The separation character in the folder path is “ / ” and not “ \ ”:

rb**\**as\core\app\asw\*acm*\*acmIPB*\Tarball\*ACM\_ECU\_CSW\_BB00001*.xml http://www.clipartbest.com/cliparts/MiL/Lpy/MiLLpypAT.png

rb/as/core/app/asw/*acm*/*acmIPB*/Tarball/*ACM\_ECU\_CSW\_BB00001*.xml http://www.clipartbest.com/cliparts/eTM/kRa/eTMkRaAAc.png

### Add PDM files to file list

If the component has PDM related files, they have to be added to the ASWCommon bcfg file:

rb\as\mb\ipb\app\asw\config\Cfg\_Build\_ASWCommon.bcfg

In the section “**pdm**”, add the following information:

pdm {

"\*\*/*SubComponentName*/PDM/*SubfolderName*/cfg/\*.pdm";

}

Example:

pdm {

"\*\*/bsmIPB/PDM/PTSOffset\_IPB\_C/cfg/\*.pdm";

"\*\*/acmIPB/PDM/PsOffset\_IPB\_C/cfg/\*.pdm";

}

### Add the new processes in process list

The project has a file which contains the list of all the processes called by the software:

rb\as\mb\ipb\app\asw\config\ProcAswMainstream\_ECU\_CSW\_BB00001.proc

* Add the “**PROC\_SECTION**” of the component’s proc file to the ProcAswMainStream file:
  + Copy the name of each PROC\_SECTION in component’s file:

/\*PROC\_SECTION: RBACM\_PsOffset\_SC\_Subsystem \*/ /\* Header files \*/

RBPDM\_DataItem\_PsOffset\_SC.h

/\*PROC\_SECTION\_END: \*/

/\*PROC\_SECTION: **ASCET\_EP\_\_\_Component\_TaskInit** \*/ /\* Process \*/

ASCET\_Component\_TaskInit

/\*PROC\_SECTION\_END: \*/

* + Add it in the ProcAswMainstream file:

/\*PROC\_EP: **ASCET\_EP\_\_\_Component\_TaskInit** \*/

* If the component contains PDM items:
  + Copy the PROC\_SECTION from each proc file from the PDM subfolders:

/\* PROC\_SECTION: **SPG\_RBACM\_PsOffset\_SC\_NvmWrite\_X24**\*/

RBPDM\_DataItem\_PsOffset\_SC\_AsapWrite

/\* PROC\_SECTION\_END: \*/

* + Add the PROC\_SECTION which includes **header** files in the “**Ascet\_Subsystem**” section as “**PROC\_EP**” of ProcAswMainstream:

/\*PROC\_SECTION: Ascet\_Subsystem \*/

/\***PROC\_EP**: **RBACM\_PsOffset\_SC\_Subsystem** \*/

...

/\*PROC\_SECTION\_END: \*/

* + Add the process sections after the pre-processor condition

“#if (RBFS\_BUILDCONFIG != RBFS\_BUILDCONFIG\_Sfun4ASWOnlySim)”

in the ProcAswMainstream file:

#if (RBFS\_BUILDCONFIG != RBFS\_BUILDCONFIG\_Sfun4ASWOnlySim)

....

/\*PROC\_EP: **SPG\_RBACM\_PsOffset\_SC\_NvmWrite\_X24** \*/

...

#endif

## DCM files

### Update DCM files

The DCM files stored in the component folder (rb\as\core\app\asw) are not compiled.

The files used for compilation are in:

rb\as\mb\ipb\app\asw\config\Parameter

Then, the updated DCM files must be stored at this location, with a different name from the file stored in the component folder.

### Add new DCM files

Copy the new DCM file in the folder:

rb\as\mb\ipb\app\asw\config\Parameter

To compile it, it must be added in the DCM file list:

rb\as\mb\ipb\app\asw\config\VariantDefinition.xml

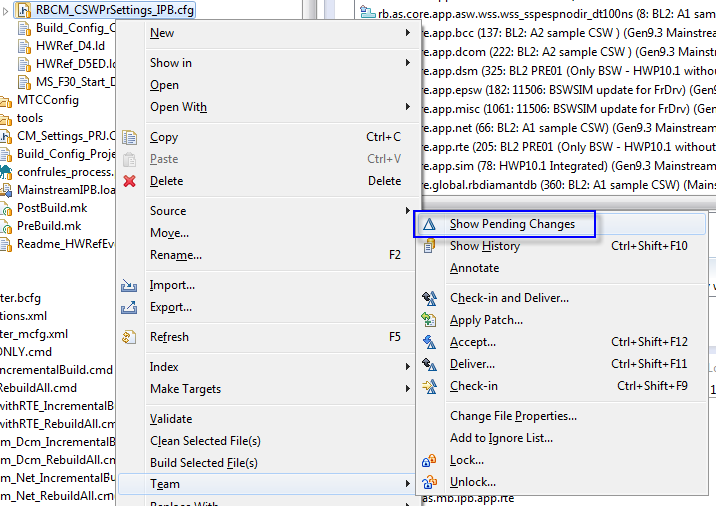
In **DCMs** field, add a new line:

<DCM>*DCM\_FileName.dcm*</DCM>

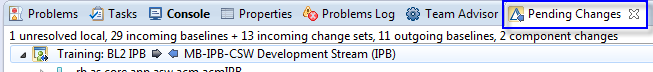
# Checkin updates

The updated files can be deliver to the stream in the “**Pending Changes**” interface:

* Right on a file in the “**Project Explorer**”
* Go to “**Team**”
* Click on “**Show Pending Changes**”



* The “**Pending Changes**” tab will be displayed on the bottom:



* This view contains the list of all the components and shows the differences with the flow target:

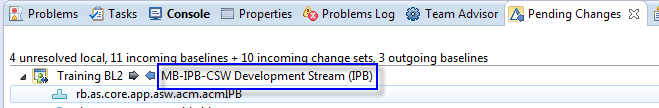
 No difference between local component and the stream

 Contains all the items that are not in your repository workspace because they are either conflicted or not checked-in.

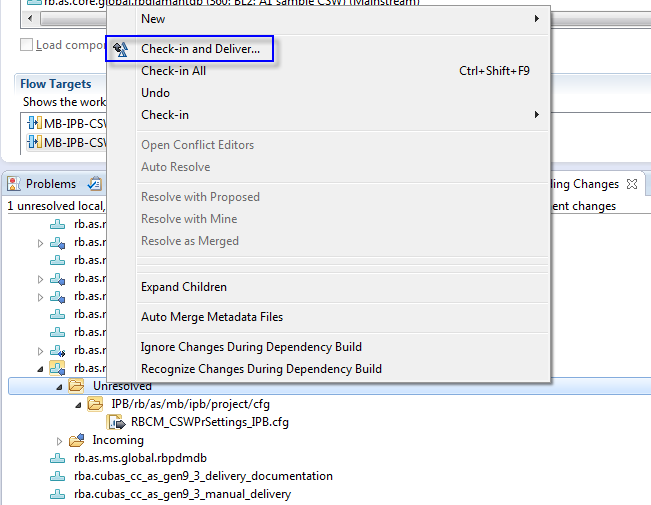
 Baselines or change sets that the flow target has but the repository workspace does not.

 Baselines or change sets that the repository workspace has but the flow target does not have.

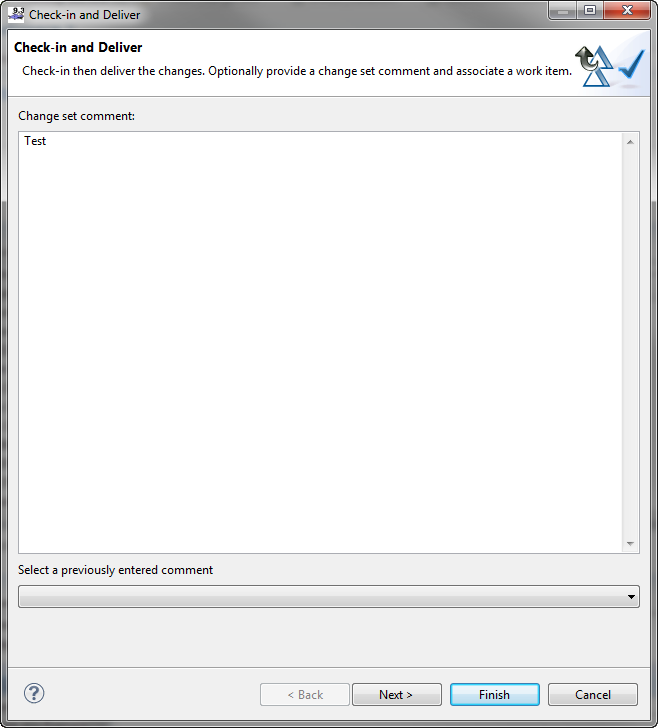
* To deliver the updates to the flow:
  + Make sure the target is correct



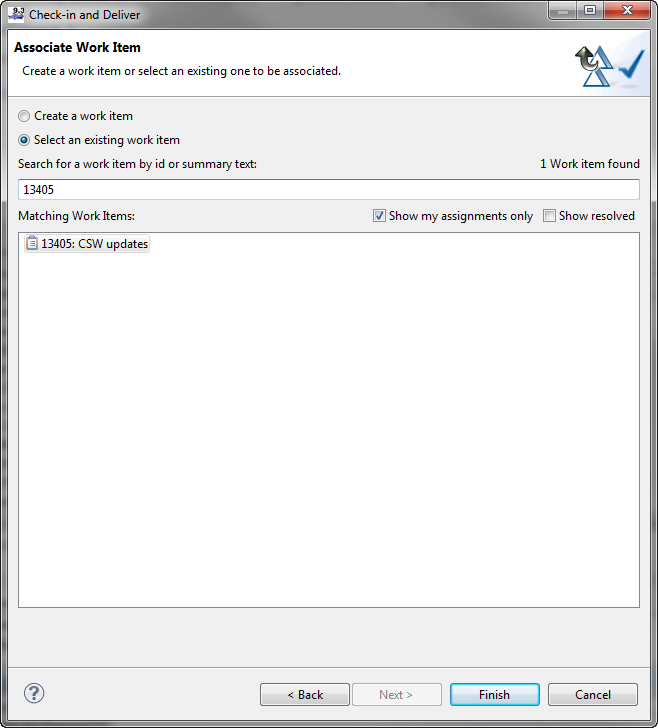
* + Expand the component to archive (with yellow icon)
  + Right click on “**Unresolved**”
  + Click on “**Check-in and Deliver...**”



* Write a comment: explain what have been changed in the component and why?



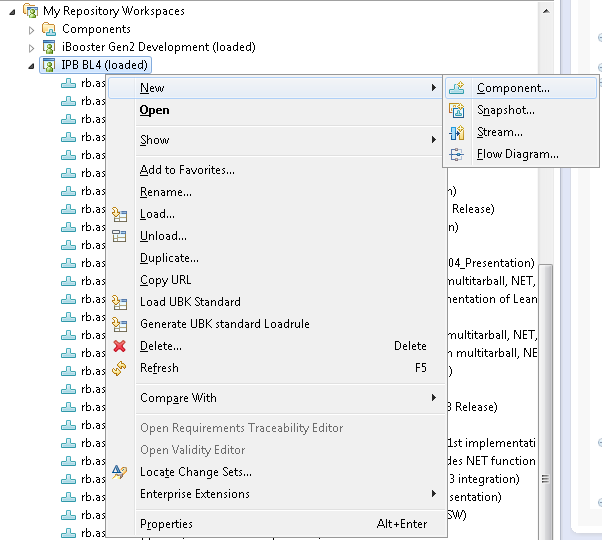
* Attach the delivery to the task:
  + Create new task or write the ID of the task
  + Click on “**Finish**”



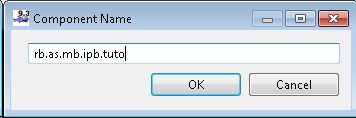
# Components

## Create a new component

* Right click on the Repository Workspace name
  + **New 🡪 Component...**



* The component name is the path to the component, “ \ ” replaced by “ . ”
  + Example: Path: rb\as\mb\ipb\tuto
    - rb.as.mb.ipb.tuto



* Add the information about the new component in the LoadRuleFile.
* The file is located in:
  + ProjectName\rb\as\mb\*projectname*\project\MainstreamProjectName.loadrule
* Add the following information and edit the data in red:

<parentLoadRule>

<component name="**rb.as.mb.ipb.tuto**"/>

<parentFolder repositoryPath="/"/>

<exclude>

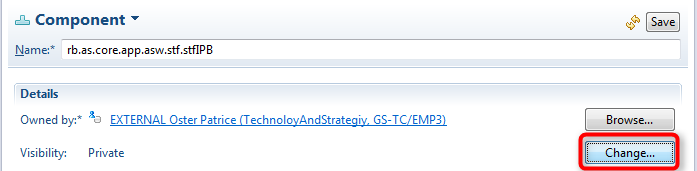
<filter name=".project"/>

</exclude>

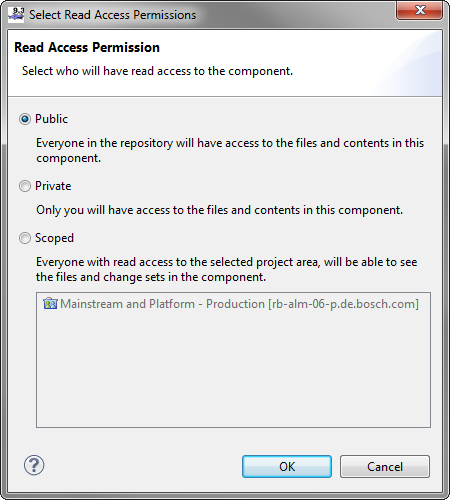
<sandboxRelativePath pathPrefix="**/IPB/rb/as/ipb/tuto**"/>

</parentLoadRule>

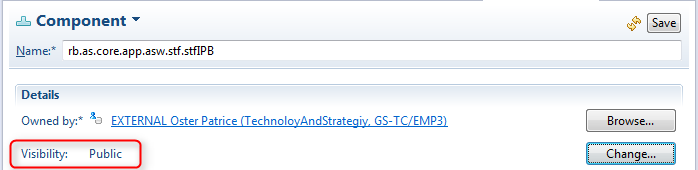
* Change the visibility of the component to **Public**:
  + Open the information regarding the component and click on “**Change**”



* + Set the visibility to “**Public**”



* + Check that it has been updated correctly and “**Save**”.



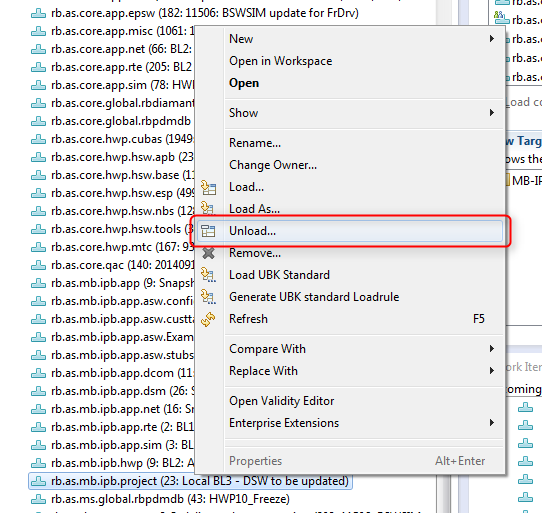
* To add files, follow the steps of the next chapter (Add files in a component).

## Add files in a component

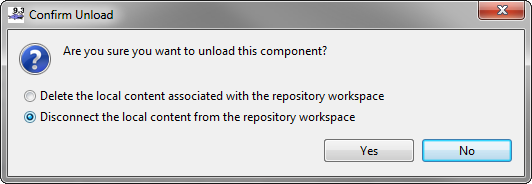
It is not possible to add a new file in SharCC Gen9.3 just by adding the files in a directory.

Here is the procedure to add it:

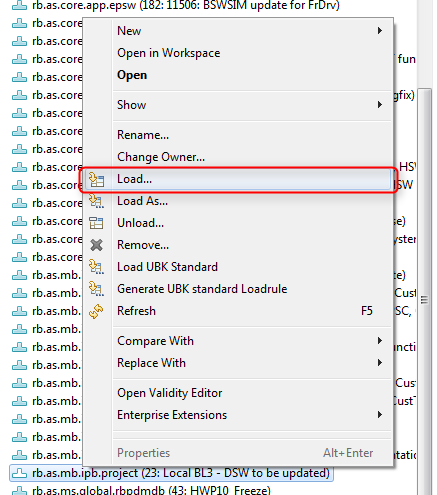
* Unload the component where the file will be added (not needed if creating a new component)



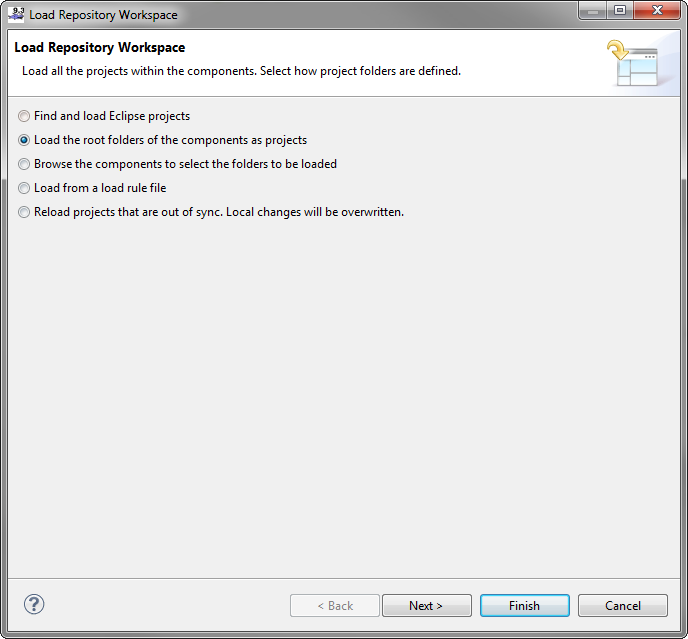
* It is not necessary to remove the files, you can disconnect the local content



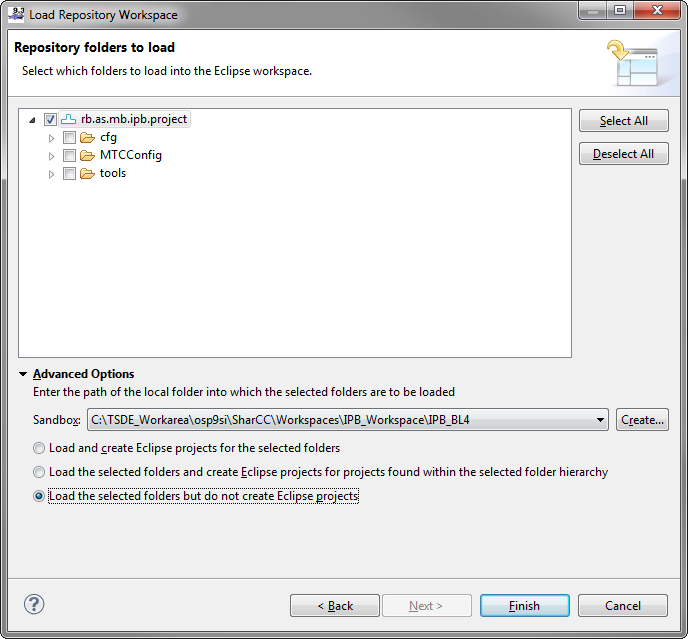
* Click on **Load**



* Select “**Load the root folders of the components as project**”

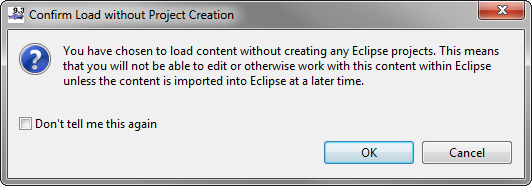


* Select “**Load the selected folders but do not create Eclipse project**” and select the project name in the file list:

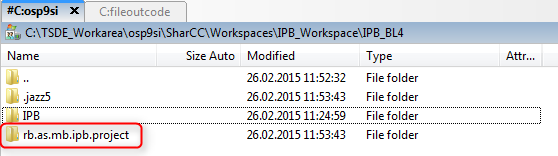


Note: if you create the folder as project, a file named “.project” will be generated. This file has to be removed!

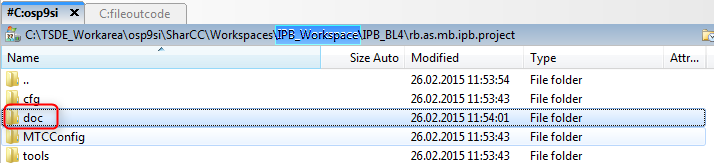
* Click on “**OK**”

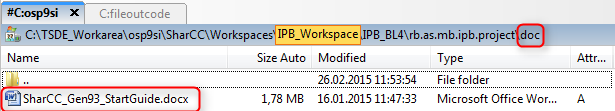


* In the Windows Explorer, you a new folder has been created:

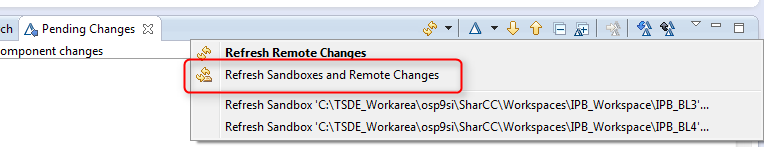


* Create new folders and files:

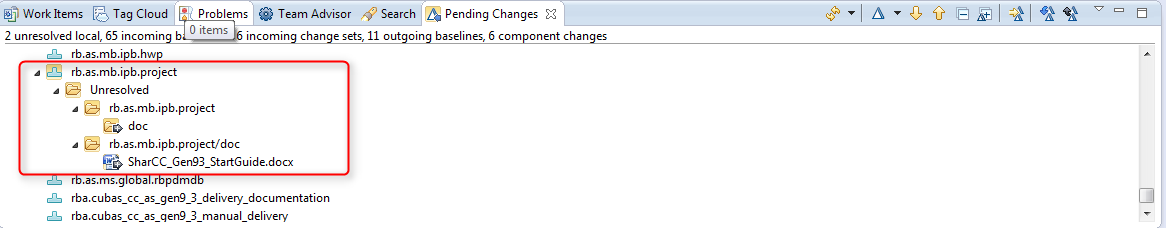




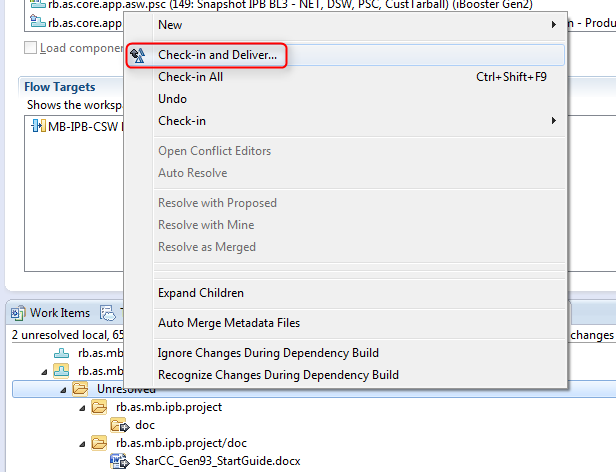
* In SharCC, click on “**Refresh Sandboxes and Remote Changes**”:



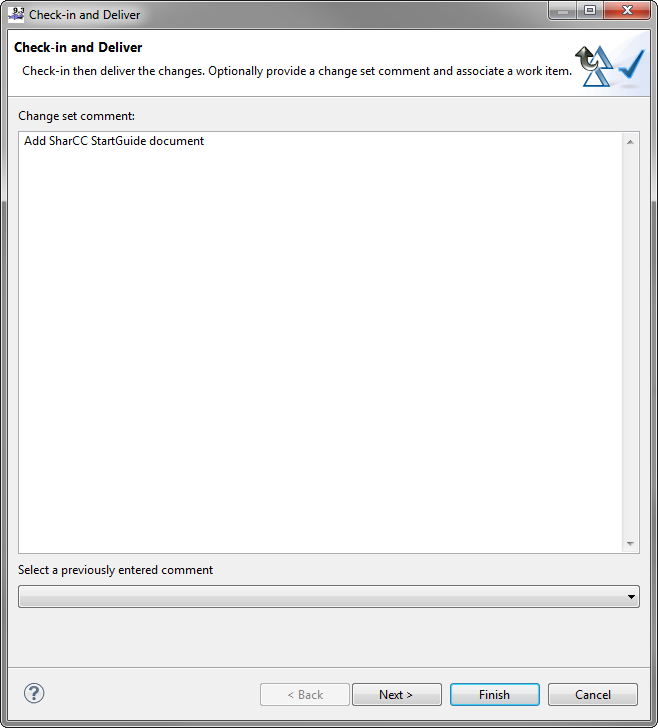
* Now you can see the new folders/files in the pending changes



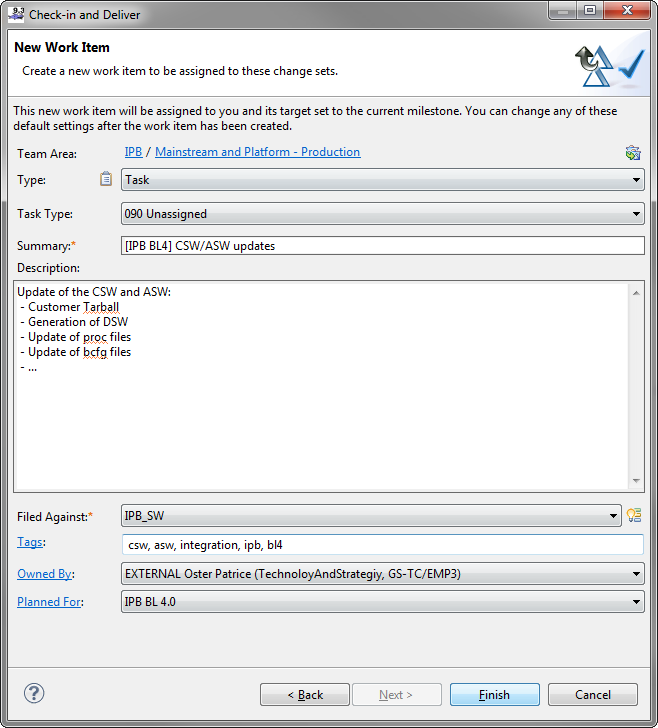
* Archive the new files 🡪 click on “**Check-in and Deliver...**”



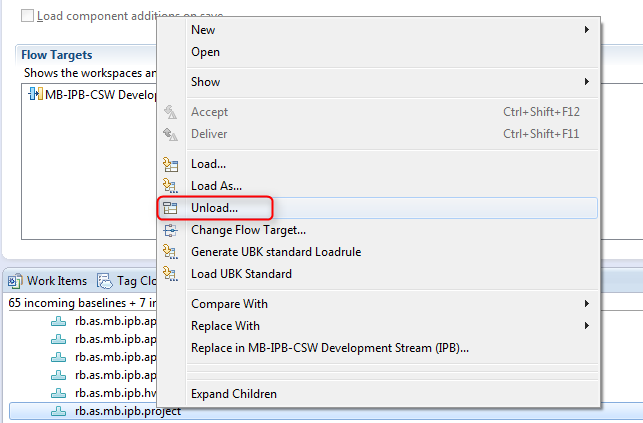
* Add a comment:



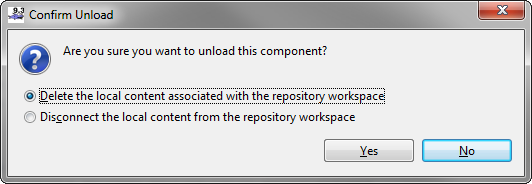
* Create a new task or attach the update to an existing task:



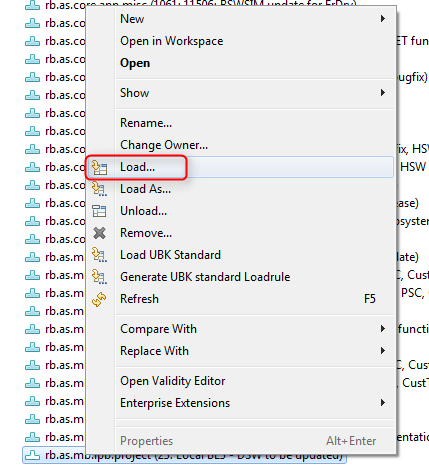
* Once the files are archived, unload the component 🡪 click on “**Unload**”:



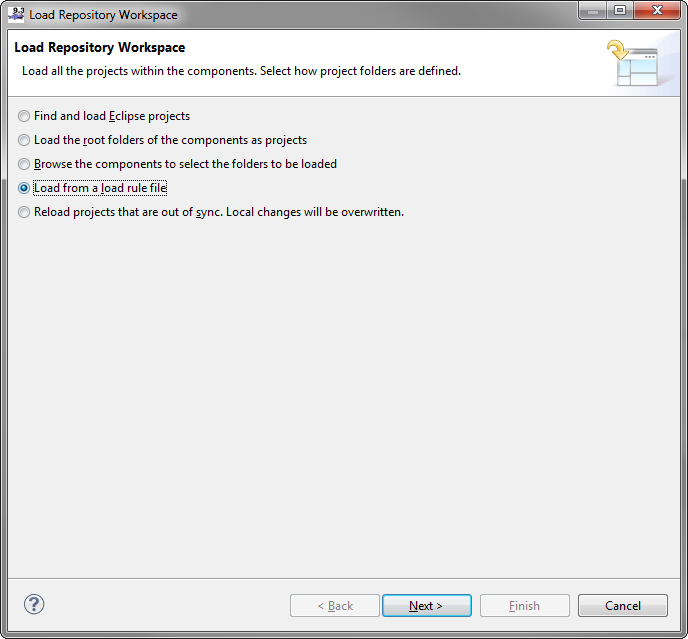
* Delete the files:



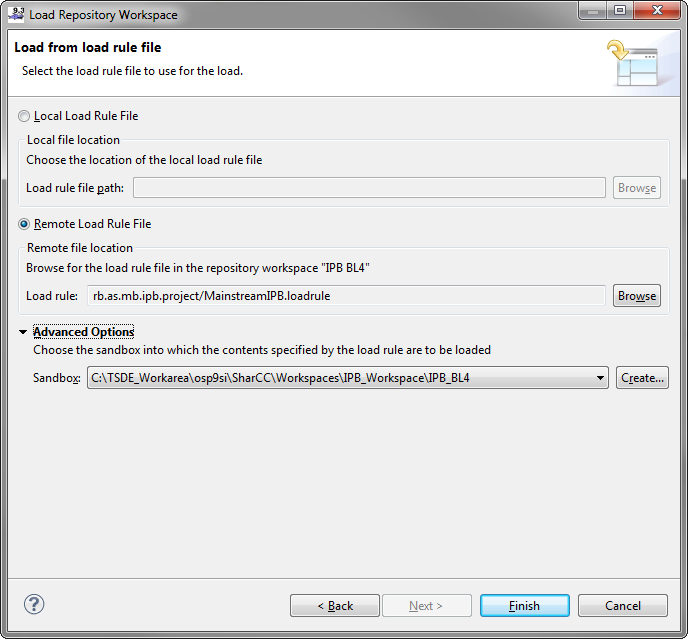
* Reload the component 🡪 click on “**Load**”:



* Select “**Load from a load rule file**”:

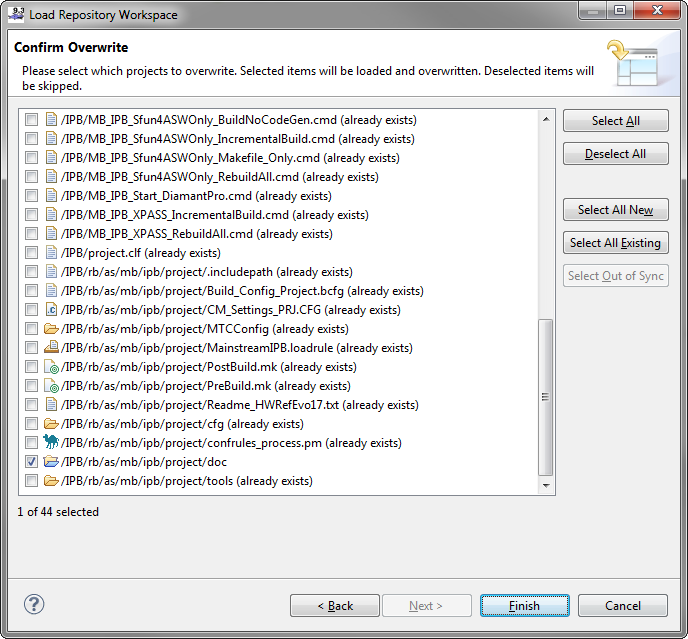


* Select “**Remote Load Rule File**”:

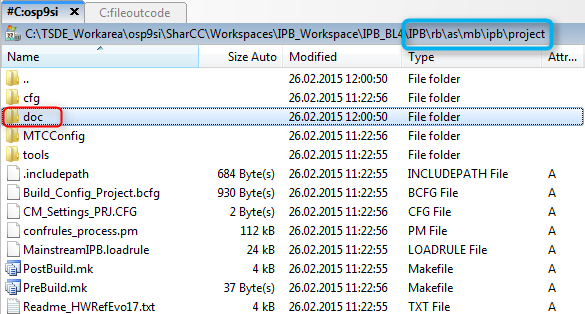


Note: Make sure that the correct Sandbox has been selected.

* Select the files to load:



* Check that the files have been added in your project folder:



# ASW Simulation Build

## Update Simulation files

Before run the Simulation build, some files have to be updated.

To update the PSW ASW interface, copy the files which have been generated during CSW compilation:

*Project\_Folder*\Gen\IPBCSW\out\tmp\_out

Gen9ProDB\_MAIN\_BBxxxxx\_*Project*CSW\_DATA\_INTERFACE\_MAPS

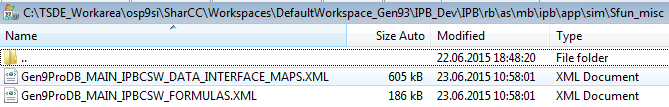
Gen9ProDB\_MAIN\_BBxxxxx\_*Project*CSW\_FORMULAS.XML

To the folder:

*Project\_Folder*\rb\as\mb\*project\_name*\app\sim

Gen9ProDB\_MAIN\_BBxxxxx\_*Project*CSW\_DATA\_INTERFACE\_MAPS

Gen9ProDB\_MAIN\_BBxxxxx\_*Project*CSW\_FORMULAS.XML



If a new class has been defined, like “**ERR\_***COMPONENTNAME\_FUNCTION*”, add a stub definition in the file:

*Project\_Folder*\rb\as\mb\*project\_name*\app\sim\Sim\_Misc\HDR\ASWOnlySim\_cswStubs.h

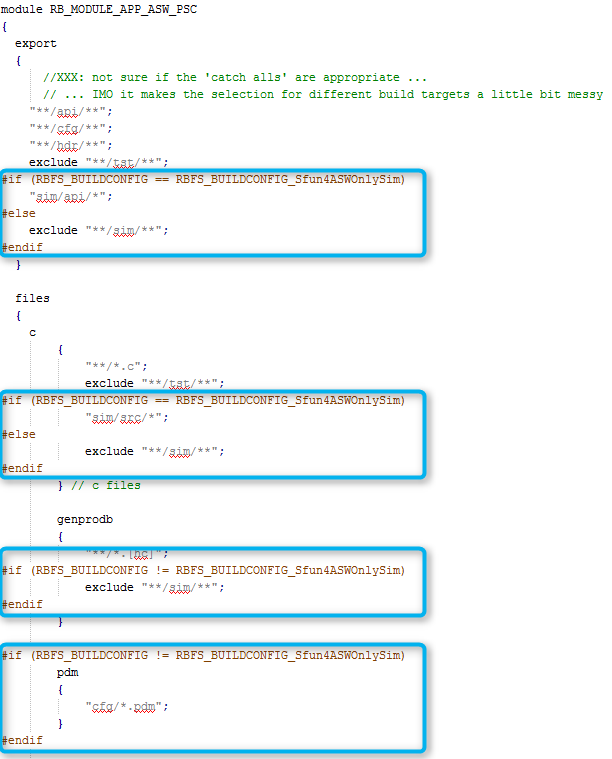
Example:



## Add a file in bcfg for ASW Simulation

To compile a file during the ASW Simulation, add the subfolder which contains the file to the bcfg file.

Example: IPB – PSC



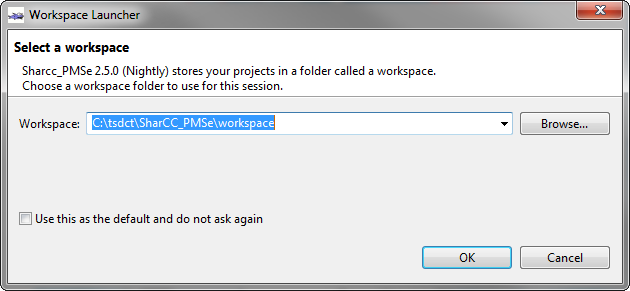
# PMSe

## Load Application Container

Open “SharCC\_PMSe”:

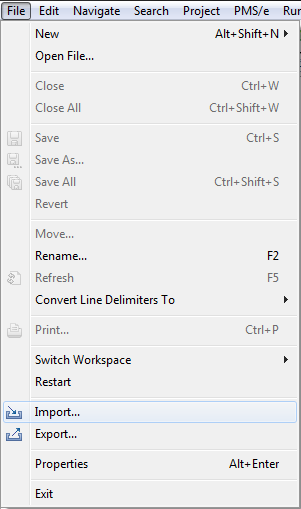


Select a workspace:



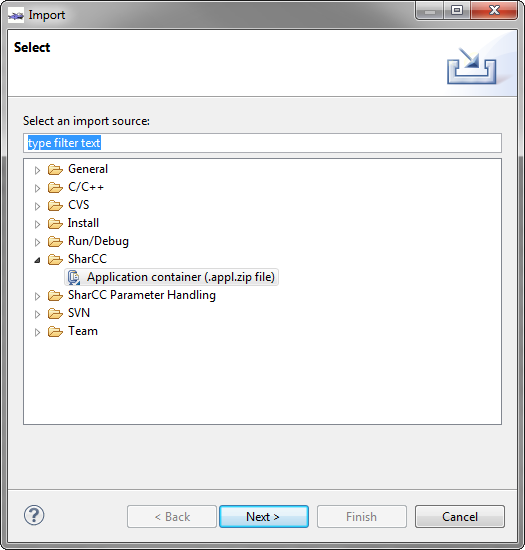
To load the Application Container, go to:

File 🡪 Import

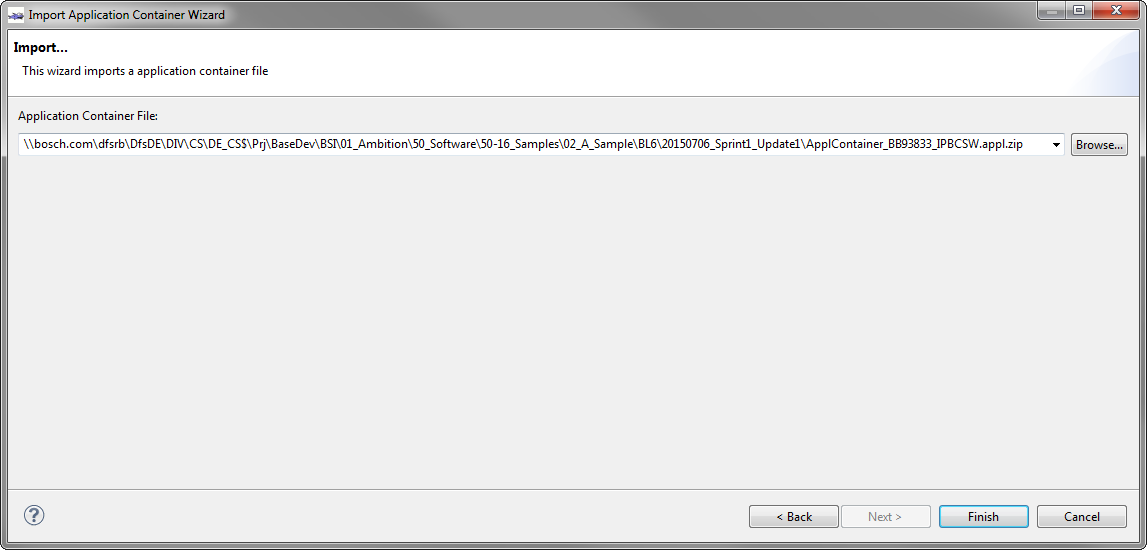


Go to:

SharCC 🡪 Application Container (.appl.zip file)

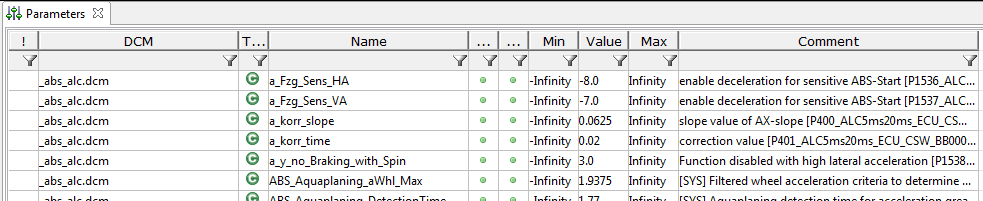


Select the Application Container and press “Finish”:



## Edit and save the Application Container (or Hexfile)

Edit the parameters in the list:



To save the modification, go to:

PMS/e 🡪 Container 🡪 Save (As)

🡪 Hex File 🡪 Save (As)

|  |  |
| --- | --- |
| 2015-07-07_09h12_51 | 2015-07-07_09h19_49 |

Select the path where to save the new file:

