# project.name Title

Produced by **Author**  
<center>#if (“$!{baseline}” == “”)Exported from <b>HEAD</b>#else Exported Baseline <b> $!{baseline.name}</b>#end</center>  
<p><center>${dateTimeFormatted}</center></p>

http://localhost:8080/cb/images/newskin/header/cblogo-xl.png

## Table of contents

Please right-click and choose "Update Field" to initialize this field.

## tracker.name overview

${exporter.transformDescribableToHtml($tracker)}

## 

#\*  
 This is a velocity template  
 Renders the „name” part of the exported entities  
\*#  
#if ("$!{describable.name}" != "")   
 #set($headingLevel = ${headingLevelProvider.apply(${entity})})  
 #if ("$!{headingLevel}" == "") #set($headingLevel = 3) #end  
 ## Word supports max 9 levels  
 #if (${headingLevel} > 9) #set($headingLevel = 9) #end  
  
 ## HTML heading level’s maximum is 6, in HTML <h7> is not present and ignored   
 #set($htmlHeadingLevel = $headingLevel)  
 #if ($htmlHeadingLevel > 6) #set($htmlHeadingLevel = 6) #end  
  
## Note: the “anchor” is used to jump between the exported issues if they refer (contain a link) to each other  
 #set($nameMarkup = “<a name=””$!{anchor}””></a>$!{requirementId.getRelease()} $!{describable.name}”)  
 #if($headingLevel > 6)  
 ## for deeper hierarchies above 4 levels that must be passed to Word document this way  
 #set($nameMarkup = “<p style=’mso-outline-level:${headingLevel};’>$nameMarkup</p>”)  
 #end  
  
 <h${htmlHeadingLevel}>$!{nameMarkup}</h${htmlHeadingLevel}>  
#end

#\*  
 This is a velocity template renders HTML fragment for Properties of entities  
\*#  
  
#if (${exportOptions.exportProperties})  
  
## Notification if issue changed in HEAD revision  
#if ((“$!{baseline}” != “”) && ($changedRequirementIdMap[$describable.id]))  
 <p class=”changed”>This issue has been changed in HEAD revision since $!{baseline.name} was created.<br /><br /></p>  
#end  
  
## load Properties specific macros:  
#parse("/html/mhtml-properties-macros.vm")  
  
#set($requirement = $describable)  
 ## render associations table  
 #if (${incomingAssociationsByIssue.containsKey($requirement.getId())} || ${outgoingAssociationsByIssue.containsKey($requirement.getId())})  
 <b>Associations:</b><br/>  
 <table class="requirement-props">  
 <tr>  
 <td class="field">From</td>  
 <td class="field">Association Type</td>  
 <td class="field">To</td>  
 </tr>  
 #foreach($entry in $outgoingAssociationsByIssue.get($requirement.getId()).entrySet())  
 <tr>  
 <td class="value"><i>&lt;this&gt;</i></td>  
 <td class="value">${entry.getKey()}</td>  
 <td class="value">  
 #foreach($item in $entry.getValue())  
 #if(${item.getKeyAndId()})[${item.getKeyAndId()}] <b>${item.getName()}</b><br/>  
 #else[${item.id}] <b>${item.getName()}</b><br/>#end   
 #end  
 </td>  
 </tr>  
 #end  
 #foreach($entry in $incomingAssociationsByIssue.get($requirement.getId()).entrySet())  
 <tr>  
 <td class="value">  
 #foreach($item in $entry.getValue())  
 #if(${item.getKeyAndId()})[${item.getKeyAndId()}] <b>${item.getName()}</b><br/>  
 #else[${item.id}] <b>${item.getName()}</b><br/>#end  
 #end  
 </td>  
 <td class="value">${entry.getKey()}</td>  
 <td class="value"><i>&lt;this&gt;</i></td>  
 </tr>  
 #end  
 </table>  
 <br/>  
 #end  
  
 ## render references table  
 #set($incomingRefs = $issueReferences.getIncomingReferences($requirement))   
 #set($outgoingRefs = $issueReferences.getOutgoingReferences($requirement))  
 #if (! ($incomingRefs.isEmpty() && $outgoingRefs.isEmpty()))  
 <b>References:</b><br/>  
 <table class="requirement-props">  
 <tr>  
 <td class="field">From</td>  
 <td class="field">Field Name</td>  
 <td class="field">To</td>  
 </tr>  
 #foreach($ref in ${outgoingRefs.entrySet()})  
 #set($dto = ${ref.getValue()})  
 <tr>  
 <td class="value"><i>&lt;this&gt;</i></td>  
 <td class="value">${ref.key}</td>  
 <td class="value">  
 #foreach($version in $ref.value)  
 #set($dto = $version.dto)  
 #if ("${dto.id}" != "")[$!{dto.id}]#end <b>$!{dto.name}</b>  
 #end  
 </td>  
 </tr>  
 #end  
 #foreach($ref in $incomingRefs)  
 #set($dto = ${ref.dto})  
 <tr>  
 <td class=”value”>#if („${dto.id}” != „”)[$!{dto.id}]#end <b>$!{dto.name}</b></td>  
## Note: if you want a link then replace the content above:  
## #set ($url = „${cbBaseUrl}/${dto.urlLink}”)  
## <a href=”${url}”>${dto.name</a>  
 <td class="value">#set($sep=””) #foreach($referer in ${dto.referers}) ${sep} ${referer.label} #set($sep=”, ”) #end</td>  
 <td class="value"><i>&lt;this&gt;</i></td>  
 </tr>  
 #end  
 </table>  
<br/>  
#end  
## Render item/issue/requirement’s history if selected on the export GUI  
#if (${exportOptions.exportItemHistory})  
<div class=”exportItemHistory”><b>History:</b><br/>  
#set($historyRenderer = $springDispatcherServletContext.getBean(“trackerItemHistoryTableRenderer”))  
$!{historyRenderer.renderTrackerItemHistory($request, $user, $requirement)}  
</div>  
#end  
  
## clear variables may have set previously  
#set($description = "")  
#set($preAction = "")  
#set($postAction = "")  
  
## access custom field values  
#set($description = ${requirement.getDescription()})  
#set($preAction = ${requirement.getCustomField(0)})  
#set($postAction = ${requirement.getCustomField(1)})  
  
## <p class=”props”>#renderWiki($!{description})</p>  
<p class=”props”><b>Pre Action:</b></p>  
<p class=”props”>#renderWiki($!{preAction})</p>  
<p class=”props”><b>Post Action:</b></p>  
<p class=”props”>#renderWiki($!{postAction})</p>  
  
#end  
  
#if (${exportOptions.exportTestCasesWithTestSteps})  
#set($requirement = $describable)  
## Render the Test-steps of the Test Case if there is any  
#set($testCase = $testCaseObject.cast($requirement, $user, $applicationContext))  
#set($testSteps = ${testCase.getTestSteps()})  
#set($numSteps = ${testSteps.size()})  
<p class=”props”><b>Number of Test Steps: ${numSteps}</b></p>  
#if ($numSteps > 0)  
 <table class=”testSteps”>  
 <tr>  
 <th></th>  
 <th><b>Action</b></th>  
 <th><b>Expected result</b></th>  
 <th><b>Critical</b></th>  
 </tr>  
 #foreach($testStep in $testSteps)  
 <tr>  
 #set($stepIndex = ${foreach.index} + 1)  
 <td><b>${stepIndex}.</b></td>  
 <td>#renderStepWiki(${testStep.getAction()} ${testStep})</td>  
 <td>#renderStepWiki(${testStep.getExpectedResult()} ${testStep})</td>  
 <td>${testStep.isCritical()}</td>  
 </tr>  
 #end  
 </table>  
#end  
#end  
  
  
#macro (renderWiki $wiki)  
 #if(“$!{wiki}” == “”)  
 --  
 #else   
 $wikiMarkupProcessor.transformToHtml(${wiki}, “W”, false, false, ${wikiContext})  
 #end  
#end  
  
#macro (renderStepWiki $stepWiki $step)  
 #if(“$!{stepWiki}” == “”)  
 --  
 #else   
 ## create a wiki-context that exports the images in the reused steps properly  
 #set($effectiveWikiOwner = ${step.getEffectiveWikiOwner($wikiContext.owner)})  
 #set($wikiContextForStep = $exporter.createWikiContext($wikiContext.httpRequest, $effectiveWikiOwner))  
 $wikiMarkupProcessor.transformToHtml(${stepWiki}, “W”, false, false, ${wikiContextForStep})  
 #end  
#end

#\*  
 This is a velocity template  
 Renders the „description” part of the exported entities  
\*#  
$wikiMarkupProcessor.transformToHtml(${describable.description}, ${describable.descriptionFormat}, false, false, $wikiContext)  
  
#\*   
 ## sample anchor for jumping to top of the current issue  
 #if ("$!{anchor}" != "") <div><a href="#$!{anchor}">Jump to top</a></div>#end  
\*#

#\*   
 This is a velocity template  
 Allows customization of decoration around HTML parts  
\*#  
${defaultTemplate}

#\*  
 This is a velocity template  
 Allows customization of the CSS used in HTML parts  
\*#  
${defaultTemplate}  
  
#\*   
 An example: uncomment this below customizing color and font in tables and lists  
\*#  
p.changed {  
 color: red !important;  
 font-weight: bold !important;  
 padding: 0 !important;  
 margin: 0 !important;  
}  
p.props {  
 padding: 10px 0px !important;  
}  
table.testSteps th {  
 text-align: left !important;  
 font-size: 11px !important;  
 padding: 2px !important;  
 border: 1px solid gray !important;  
 border-collapse: collapse !important;  
 background-color: lightgray !important;  
}  
table.testSteps td {  
 font-size: 11px !important;  
 padding: 5px !important;  
 border: 1px solid gray !important;  
 border-collapse: collapse !important;  
}