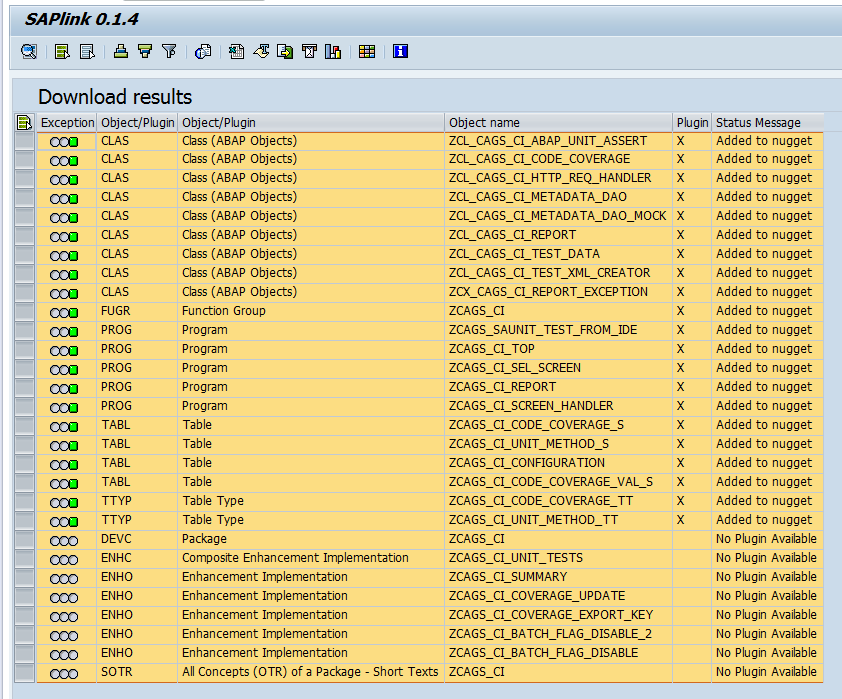
ZCAGS\_CI package content added to NUGG file (later in addition ZCL\_UTIL\_STRING has been added):

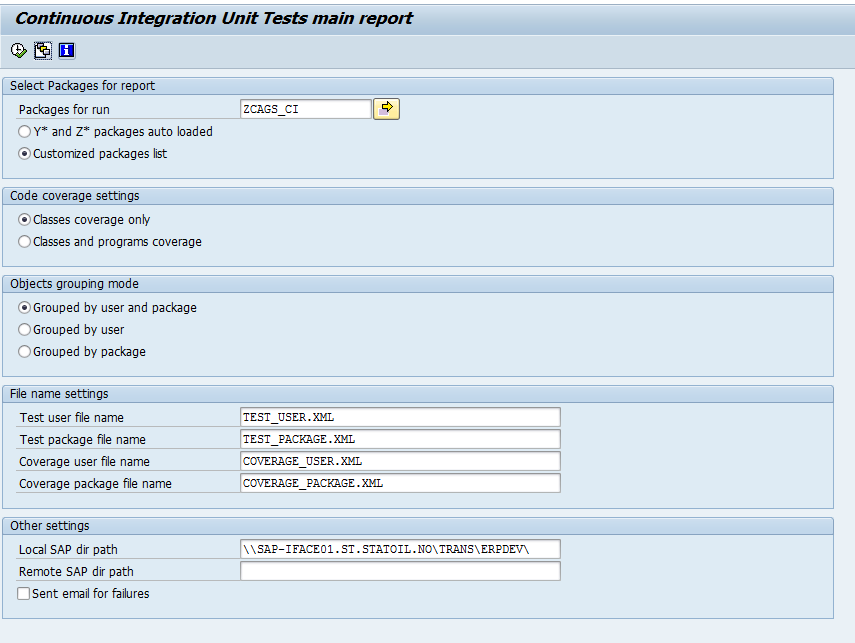


Implicit enhancements were not added so I attach them separately as text files. Implicit enhancements are very important – they enable framework to run during standard SAP unit tests execution.

If you have different version of SAP that mine, try to adjust places where implicit enhancements are put (I remember that after some upgrade names have changed so I had to adjust enhancements to new local classes).

How to perform first run and see generated XML files?

Run the starting point report zcags\_ci\_report from GUI for selected package. Then check if XML file with results (coverage and unit tests) is generated in directory. Make sure to put valid existing parameters. Example configuration:



Note: Remote SAP dir path is actually not needed. Local SAP dir path must exist on SAP server and user must have right to save file there.

After report was run, check AL11 transaction to see if files are generated inside. If you see generated files it means that framework works. Now it is just the matter of integrating all together:

* How to run this report from Hudson? I used wget tool which calls URL that runs SAP code logic and generate batch job? That is convenient solution as I can run job directly from Hudson by “Run” button. Optionally instead of scheduling from Hudson you can schedule batch job in development system that runs unit tests for example every day at 22:00, then you skip the integration of job running from Hudson but just collect results to Hudson PC server.
* How to get files to local PC where Hudson server is installed? I used web service and wget run from Hudson PC, which downloads file content to local directory by GET protocol.
* Define projects in Hudson which take generated files and converts them to result graphs (see D83\_Unit\_Tests\_Package example at the end of document).
* Integrate all steps together.

How I run report from Hudson?

I defined Webservice which can be called from Hudson. Hudson calls this code by wget tool and URL.

ZIF\_RESOURCE\_HANDLER~HANDLE\_GET

Depending on what is the parameter in the URL different actions can be done:

* Unit tests run remotely in system – option JOBRUN
* Lock file can be deleted (which blocks parallel unit tests execution) – option DELLOCK
* Files with tests can be cleaned – option DELFILES
* File content will be returned as response to GET query: any other string (file name actually)

In addition, wget application is installed so that it is possible to call URL and redirect output to file. Hudson uses this mechanism to check tests execution status and finally read results from XML file.

Example call from Hudson:

This call will start unit tests in batch job on the server and results will be saved to runjob.log file:

C:\appl\ci\d83\_results>C:\appl\WorkTools\wget\wget --secure-protocol=auto --http-user=BATCH\_TEST --http-password=Password123 <http://sapq152.statoil.no:8014/sap/bc/resources/ci_results/JOBRUN_D83_774888> -O C:\appl\ci\d83\_results\runjob.log

This command will try to read file D83\_CI\_ABAP\_UNIT\_USER.XML in SAP server. This file is generated after unit tests are finished. If file will be read, it will be transferred to local file “C:\appl\ci\d83\_results/D83\_CI\_ABAP\_UNIT\_USER.XML”. Then batch job in Hudson checks file content if it is not empty it means that results are provided and tests finished.

C:\appl\ci\d83\_results>C:\appl\WorkTools\wget\wget --secure-protocol=auto --http-user=BATCH\_TEST --http-password=Password123<http://sapq152.statoil.no:8014/sap/bc/resources/ci_results/D83_CI_ABAP_UNIT_USER.XML> -O C:\appl\ci\d83\_results/D83\_CI\_ABAP\_UNIT\_USER.XML

Instead of wget I could put that full HTTP URL into my browser, provide SAP user and password when prompt and see result of file filled in my browser. WGET tool is just used to do it in the background by Hudson project and save results to file.

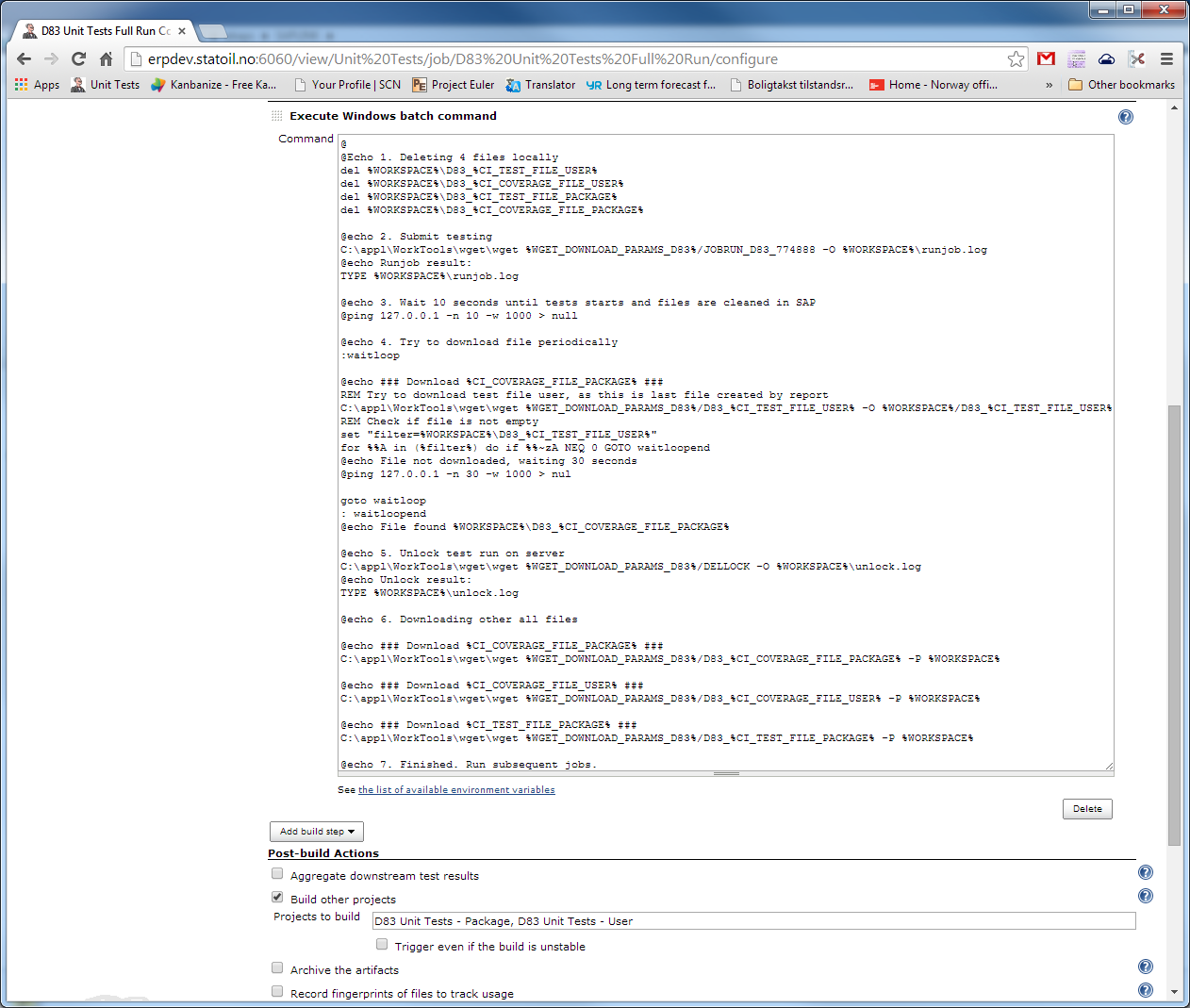
This is how Hudson project is configured:



Example output from Hudson project run:



This is screenshot of how the main Hudson job is configured. The job D83\_Unit\_Tests\_Full\_Run is responsible for scheduling unit tests (through remote call to Hudson server) and checking if file with results is downloaded successfully. If file contains tests result, then subsequent jobs that visualize results in Hudson will be called.



Job which processes and visualizes unit tests results is shown below. This job is run as subsequent job of main job which runs the test. The job must refresh files so they have new creation date. Hudson is checking this before unit tests and coverage is visualized by project. This is actually the job which will show results (D83\_Unit\_Tests\_Package).

