Документация к файлу: cls_native.py

Сгенерировано: 2025-04-04 18:36

```
#/*
# * Licensed to the OpenAirInterface (OAI) Software Alliance unde
# * contributor license agreements.  See the NOTICE file distribu
# * this work for additional information regarding copyright owne
# * The OpenAirInterface Software Alliance licenses this file to
# * the OAI Public License, Version 1.1  (the "License"); you may
# * except in compliance with the License.
# * You may obtain a copy of the License at
# *
# *      http://www.openairinterface.org/?page_id=698
# *
# * Unless required by applicable law or agreed to in writing, so
# * distributed under the License is distributed on an "AS IS" BA
# * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
# * See the License for the specific language governing permissio
# * limitations under the License.
# *-------------------------------------------------------------
# * For more information about the OpenAirInterface (OAI) Softwar
# *      contact@openairinterface.org
# */
#---------------------------------------------------------------

import logging
import re
import os

import cls_cmd
import cls_oai_html
import cls_analysis
import constants as CONST

LOG_PATH_PHYSIM = 'phy_sim_logs'

class Native():

        def Build(test_case, HTML, host, directory, options):
                logging.debug(f'Building on server: {host}')
                HTML.testCase_id = test_case

                with cls_cmd.getConnection(host) as ssh:
                        base = f"{directory}/cmake_targets"
```

```
                        ret = ssh.run(f"{base}/build_oai {options
                        success = ret.returncode == 0
                        logs = ssh.run(f"cat {base}/log/build_oai
                        logging.debug(f"build finished with code

                        # create log directory, and copy build lo
                        target = f"{base}/build_log_{test_case}/"
                        ssh.run(f"mkdir -p {target}")
                        ssh.run(f"mv {base}/log/* {target}")

                        # check if build artifacts are there
                        # NOTE: build_oai should fail with exit c
                        build_dir = f"{base}/ran_build/build"
                        build_gnb = re.search('--gNB', options) i
                        if build_gnb:
                                success = success and ssh.run(f"
                        build_enb = re.search('--eNB', options) i
                        if build_enb:
                                success = success and ssh.run(f"

                if success:
                        logging.info('\u001B[1m Building OAI Pass
                        HTML.CreateHtmlTestRow(options, 'OK', CON
                else:
                        logging.error('\u001B[1m Building OAI Fai
                        HTML.CreateHtmlTestRow(options, 'KO', CON
                return success

        def Run_Physim(HTML, host, directory, options, physim_tes
                logging.debug(f'Runnin {physim_test} on server: {
                workSpacePath = f'{directory}/cmake_targets'
                os.system(f'mkdir -p ./{LOG_PATH_PHYSIM}')
                runLogFile=f'physim_{HTML.testCase_id}.log'
                with cls_cmd.getConnection(host) as cmd:
                        cmd.run(f'sudo {workSpacePath}/ran_build/
                        cmd.copyin(src=f'{workSpacePath}/{runLogF
                success, msg = cls_analysis.Analysis.analyze_phys
                if success:
                        HTML.CreateHtmlTestRowQueue(options, 'OK
                else:
                        logging.error(msg)
                        HTML.CreateHtmlTestRowQueue(options, 'KO
                return success
```