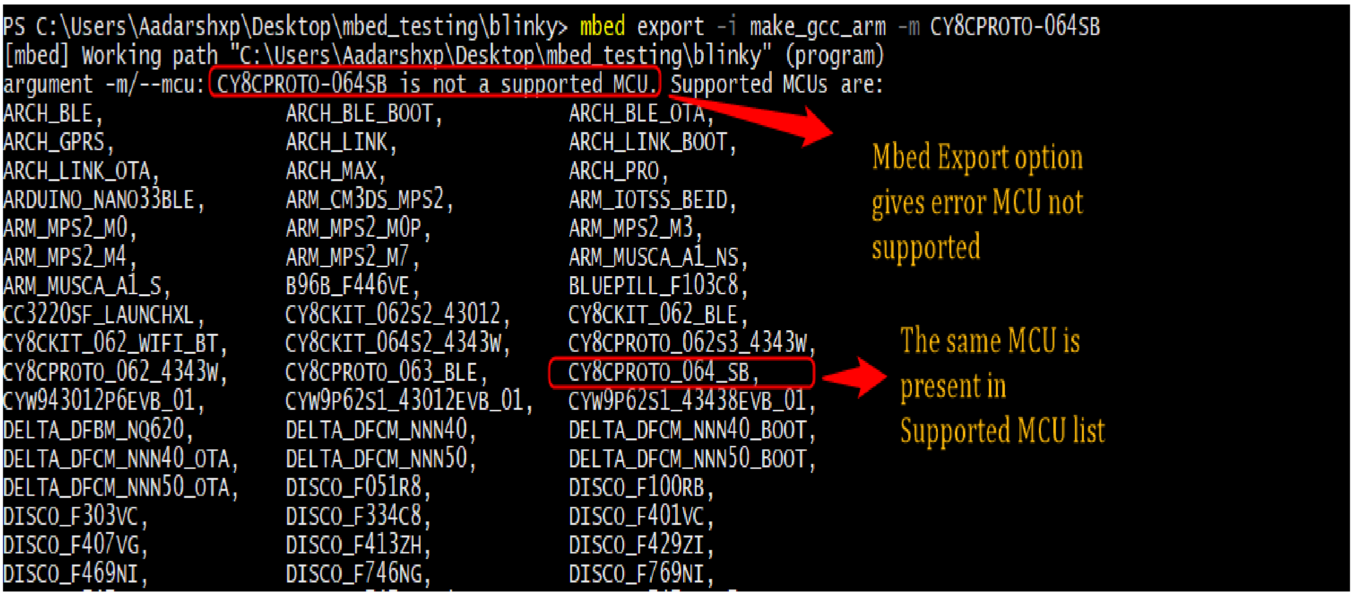
Embedded Security Showcase using PSoC64

Bug Report

Author: Aadarsh Kumar Singh (aadarsh.k.singh@stud.h-da.de)

Title: Exporting Makefile project for PSoC64 MCU (CY8CPROTO-064S1) to Eclipse IDE throws an error: CY8CPROTO-064S1 is not a supported MCU.

Screenshot:



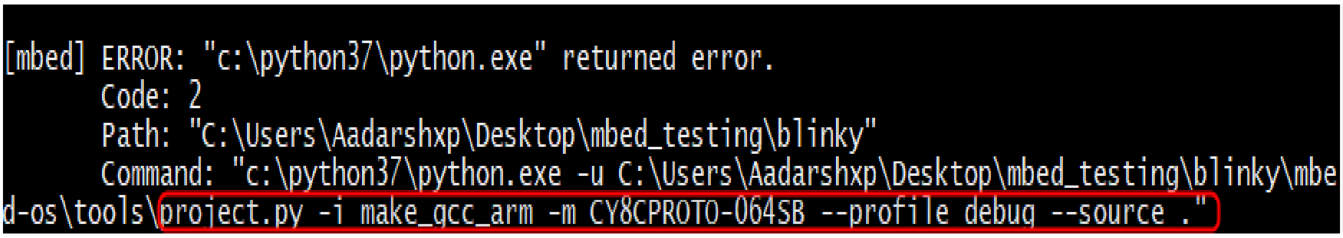
```
PS C:\Users\Aadarshxp\Desktop\mbed_testing\blinky> mbed export -i make_gcc_arm -m CY8CPROTO-064SB
[mbed] Working path "C:\Users\Aadarshxp\Desktop\mbed_testing\blinky" (program)
argument -m/--mcu: CY8CPROTO-064SB is not a supported MCU. Supported MCUs are:
ARCH_BLE,              ARCH_BLE_BOOT,         ARCH_BLE_OTA,
ARCH_GPRS,             ARCH_LINK,             ARCH_LINK_BOOT,
ARCH_LINK_OTA,         ARCH_MAX,              ARCH_PRO,
ARDUINO_NANO33BLE,     ARM_CM3DS_MPS2,        ARM_IOTSS_BEID,
ARM_MPS2_M0,           ARM_MPS2_M0P,          ARM_MPS2_M3,
ARM_MPS2_M4,           ARM_MPS2_M7,           ARM_MUSCA_A1_NS,
ARM_MUSCA_A1_S,        B96B_F446VE,           BLUEPILL_F103C8,
CC3220SF_LAUNCHXL,     CY8CKIT_062S2_43012,   CY8CKIT_062_BLE,
CY8CKIT_062_WIFI_BT,   CY8CKIT_064S2_4343W,   CY8CPROTO_062S3_4343W,
CY8CPROTO_062_4343W,   CY8CPROTO_063_BLE,     CY8CPROTO_064_SB,
CYW943012P6EVB_01,     CYW9P62S1_43012EVB_01, CYW9P62S1_43438EVB_01,
DELTA_DFBM_NQ620,      DELTA_DFCM_NNN40,      DELTA_DFCM_NNN40_BOOT,
DELTA_DFCM_NNN40_OTA,  DELTA_DFCM_NNN50,      DELTA_DFCM_NNN50_BOOT,
DELTA_DFCM_NNN50_OTA,  DISCO_F051R8,          DISCO_F100RB,
DISCO_F303VC,          DISCO_F334C8,          DISCO_F401VC,
DISCO_F407VG,          DISCO_F413ZH,          DISCO_F429ZI,
DISCO_F469NI,          DISCO_F746NG,          DISCO_F769NI,
```

Mbed Export option gives error MCU not supported

The same MCU is present in Supported MCU list

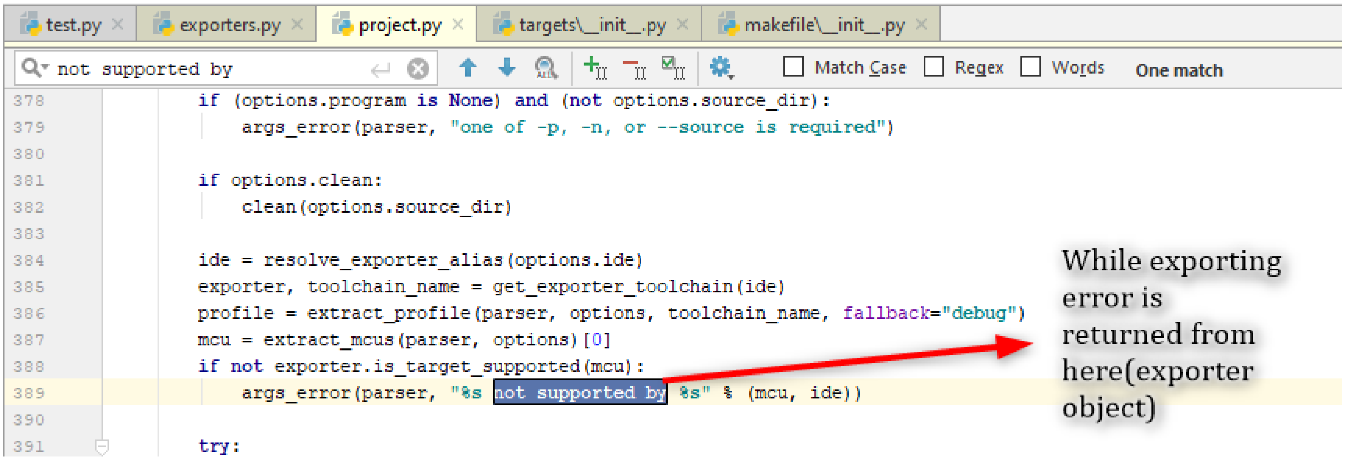
Description and temporary fix:

1- The result of the verbose output is attached below, the error is thrown from the script project.py. The file is present in the path <root>\mbed-os\tools.



```
[mbed] ERROR: "c:\python37\python.exe" returned error.
       Code: 2
       Path: "C:\Users\Aadarshxp\Desktop\mbed_testing\blinky"
       Command: "c:\python37\python.exe -u C:\Users\Aadarshxp\Desktop\mbed_testing\blinky\mbed-os\tools\project.py -i make_gcc_arm -m CY8CPROTO-064SB --profile debug --source ."
```

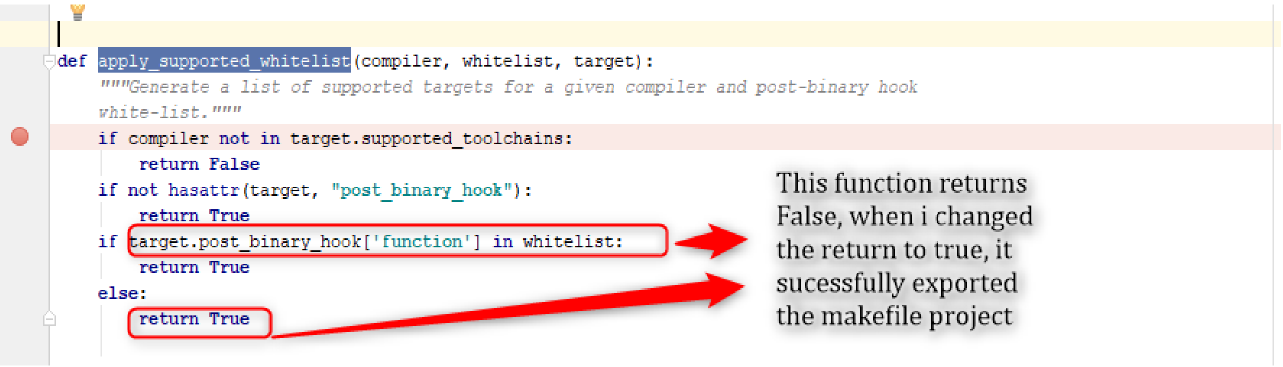
2- On debugging the project.py script found the script throws error while trying to export at this point , In screenshot at 385 line , we see that exporter object is created and contains all the exporter toolchain and MCU list. Further debugging (step into operation) found it returns some error values from exporter.py python script. (<root>\mbed-os\tools\export\exporter.py)



```
378         if (options.program is None) and (not options.source_dir):
379             args_error(parser, "one of -p, -n, or --source is required")
380
381         if options.clean:
382             clean(options.source_dir)
383
384         ide = resolve_exporter_alias(options.ide)
385         exporter, toolchain_name = get_exporter_toolchain(ide)
386         profile = extract_profile(parser, options, toolchain_name, fallback="debug")
387         mcu = extract_mcus(parser, options)[0]
388         if not exporter.is_target_supported(mcu):
389             args_error(parser, "%s not supported by %s" % (mcu, ide))
390
391         try:
```

While exporting error is returned from here(exporter object)

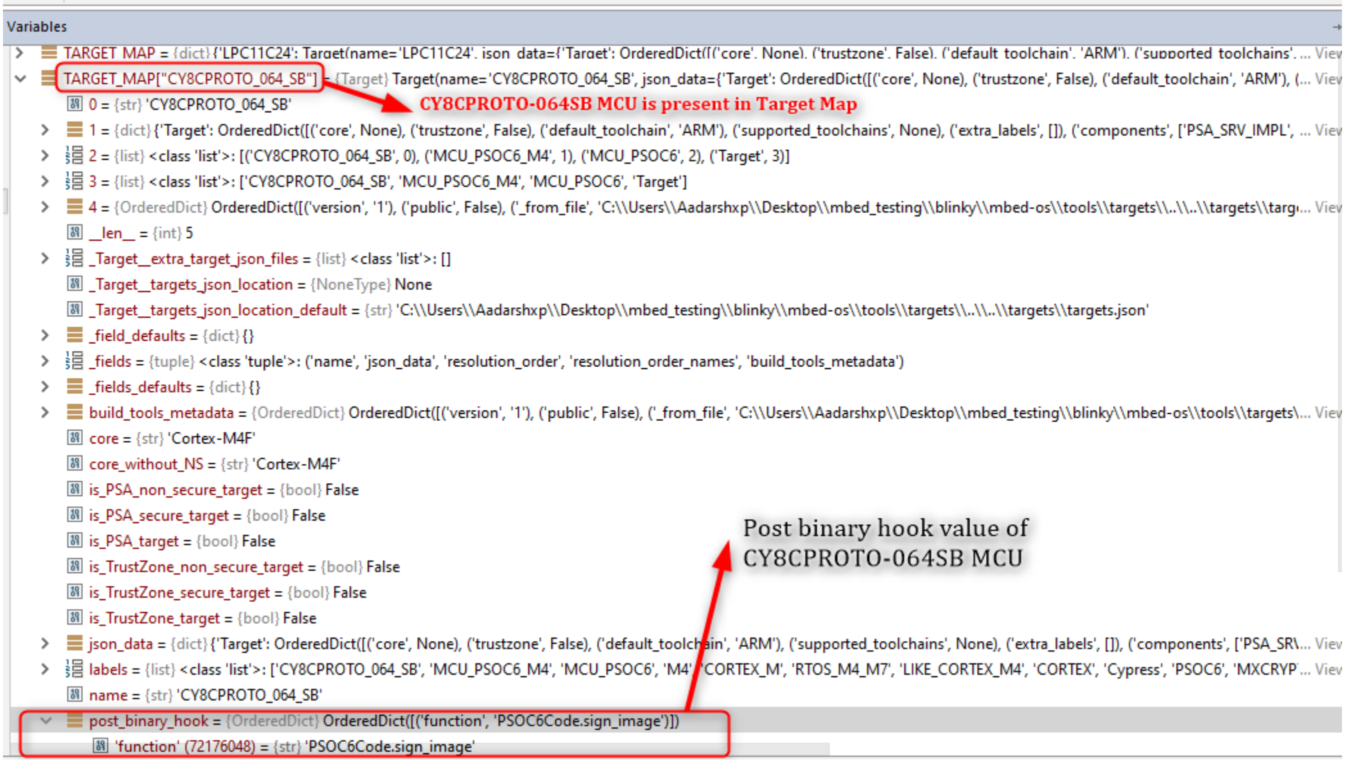
3- On debugging the exporter.py found that in the API apply_supported_whitelist the toolchain was supported , and the attribute for the MCU (CY8CPROTO-064SB) was present but in the last API binary hook returned False, on changing this value to true , i was successfully able to import the makefile project for CY8CPROTO-064SB board in Eclipse IDE.



```
def apply_supported_whitelist(compiler, whitelist, target):
    """Generate a list of supported targets for a given compiler and post-binary hook
    white-list."""
    if compiler not in target.supported_toolchains:
        return False
    if not hasattr(target, "post_binary_hook"):
        return True
    if target.post_binary_hook['function'] in whitelist:
        return True
    else:
        return True
```

This function returns False, when i changed the return to true, it sucessfully exported the makefile project

4- The **expected return and Actual return value for the post binary hook API** is attached below :

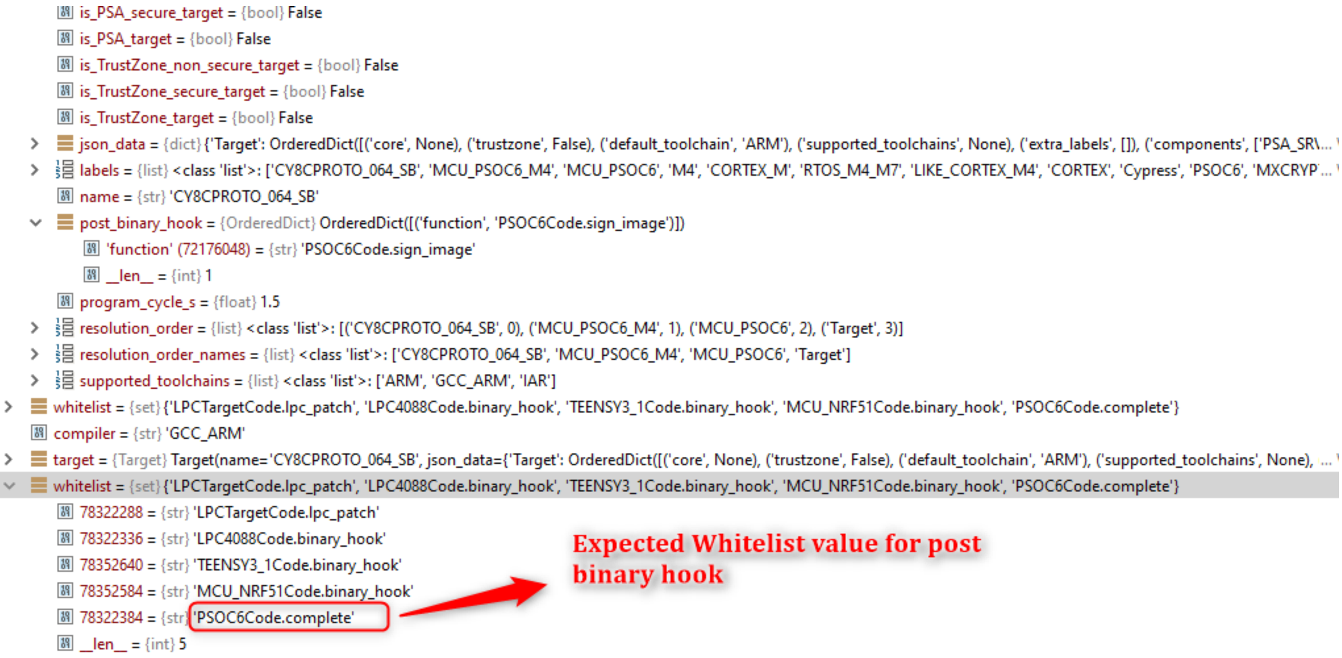


```
Variables
> TARGET_MAP = {dict} {'LPC11C24': Target(name='LPC11C24', json_data={'Target': OrderedDict([('core', None), ('trustzone', False), ('default_toolchain', 'ARM'), ('supported_toolchains', ... View
v TARGET_MAP["CY8CPROTO_064_SB"] = {Target} Target(name='CY8CPROTO_064_SB', json_data={'Target': OrderedDict([('core', None), ('trustzone', False), ('default_toolchain', 'ARM'), (... View
    0 = {str} 'CY8CPROTO_064_SB'
  > 1 = {dict} {'Target': OrderedDict([('core', None), ('trustzone', False), ('default_toolchain', 'ARM'), ('supported_toolchains', None), ('extra_labels', []), ('components', ['PSA_SRV_IMPL', ... View
  > 2 = {list} <class 'list'>: [('CY8CPROTO_064_SB', 0), ('MCU_PSOC6_M4', 1), ('MCU_PSOC6', 2), ('Target', 3)]
  > 3 = {list} <class 'list'>: ['CY8CPROTO_064_SB', 'MCU_PSOC6_M4', 'MCU_PSOC6', 'Target']
  > 4 = {OrderedDict} OrderedDict([('version', '1'), ('public', False), ('_from_file', 'C:\\Users\\Aadarshxp\\Desktop\\mbed_testing\\blinky\\mbed-os\\tools\\targets\\..\\..\\targets\\targ... View
    __len__ = {int} 5
  > _Target__extra_target_json_files = {list} <class 'list'>: []
    _Target__targets_json_location = {NoneType} None
    _Target__targets_json_location_default = {str} 'C:\\Users\\Aadarshxp\\Desktop\\mbed_testing\\blinky\\mbed-os\\tools\\targets\\..\\..\\targets\\targets.json'
  > _field_defaults = {dict} {}
  > _fields = {tuple} <class 'tuple'>: ('name', 'json_data', 'resolution_order', 'resolution_order_names', 'build_tools_metadata')
  > _fields_defaults = {dict} {}
  > build_tools_metadata = {OrderedDict} OrderedDict([('version', '1'), ('public', False), ('_from_file', 'C:\\Users\\Aadarshxp\\Desktop\\mbed_testing\\blinky\\mbed-os\\tools\\targets\... View
    core = {str} 'Cortex-M4F'
    core_without_NS = {str} 'Cortex-M4F'
    is_PSA_non_secure_target = {bool} False
    is_PSA_secure_target = {bool} False
    is_PSA_target = {bool} False
    is_TrustZone_non_secure_target = {bool} False
    is_TrustZone_secure_target = {bool} False
    is_TrustZone_target = {bool} False
  > json_data = {dict} {'Target': OrderedDict([('core', None), ('trustzone', False), ('default_toolchain', 'ARM'), ('supported_toolchains', None), ('extra_labels', []), ('components', ['PSA_SR\... View
  > labels = {list} <class 'list'>: ['CY8CPROTO_064_SB', 'MCU_PSOC6_M4', 'MCU_PSOC6', 'M4', 'CORTEX_M', 'RTOS_M4_M7', 'LIKE_CORTEX_M4', 'CORTEX', 'Cypress', 'PSOC6', 'MXCRYP'... View
    name = {str} 'CY8CPROTO_064_SB'
  v post_binary_hook = {OrderedDict} OrderedDict([('function', 'PSOC6Code.sign_image')])
      'function' (72176048) = {str} 'PSOC6Code.sign_image'
```

CY8CPROTO-064SB MCU is present in Target Map

Post binary hook value of CY8CPROTO-064SB MCU

Expected Post binary hook value in whitelist :



```
    is_PSA_secure_target = {bool} False
    is_PSA_target = {bool} False
    is_TrustZone_non_secure_target = {bool} False
    is_TrustZone_secure_target = {bool} False
    is_TrustZone_target = {bool} False
  > json_data = {dict} {'Target': OrderedDict([('core', None), ('trustzone', False), ('default_toolchain', 'ARM'), ('supported_toolchains', None), ('extra_labels', []), ('components', ['PSA_SR\...
  > labels = {list} <class 'list'>: ['CY8CPROTO_064_SB', 'MCU_PSOC6_M4', 'MCU_PSOC6', 'M4', 'CORTEX_M', 'RTOS_M4_M7', 'LIKE_CORTEX_M4', 'CORTEX', 'Cypress', 'PSOC6', 'MXCRYP'...
    name = {str} 'CY8CPROTO_064_SB'
  v post_binary_hook = {OrderedDict} OrderedDict([('function', 'PSOC6Code.sign_image')])
      'function' (72176048) = {str} 'PSOC6Code.sign_image'
      __len__ = {int} 1
    program_cycle_s = {float} 1.5
  > resolution_order = {list} <class 'list'>: [('CY8CPROTO_064_SB', 0), ('MCU_PSOC6_M4', 1), ('MCU_PSOC6', 2), ('Target', 3)]
  > resolution_order_names = {list} <class 'list'>: ['CY8CPROTO_064_SB', 'MCU_PSOC6_M4', 'MCU_PSOC6', 'Target']
  > supported_toolchains = {list} <class 'list'>: ['ARM', 'GCC_ARM', 'IAR']
> whitelist = {set} {'LPCTargetCode.lpc_patch', 'LPC4088Code.binary_hook', 'TEENSY3_1Code.binary_hook', 'MCU_NRF51Code.binary_hook', 'PSOC6Code.complete'}
  compiler = {str} 'GCC_ARM'
> target = {Target} Target(name='CY8CPROTO_064_SB', json_data={'Target': OrderedDict([('core', None), ('trustzone', False), ('default_toolchain', 'ARM'), ('supported_toolchains', None), ...
v whitelist = {set} {'LPCTargetCode.lpc_patch', 'LPC4088Code.binary_hook', 'TEENSY3_1Code.binary_hook', 'MCU_NRF51Code.binary_hook', 'PSOC6Code.complete'}
    78322288 = {str} 'LPCTargetCode.lpc_patch'
    78322336 = {str} 'LPC4088Code.binary_hook'
    78352640 = {str} 'TEENSY3_1Code.binary_hook'
    78352584 = {str} 'MCU_NRF51Code.binary_hook'
    78322384 = {str} 'PSOC6Code.complete'
    __len__ = {int} 5
```

Expected Whitelist value for post binary hook