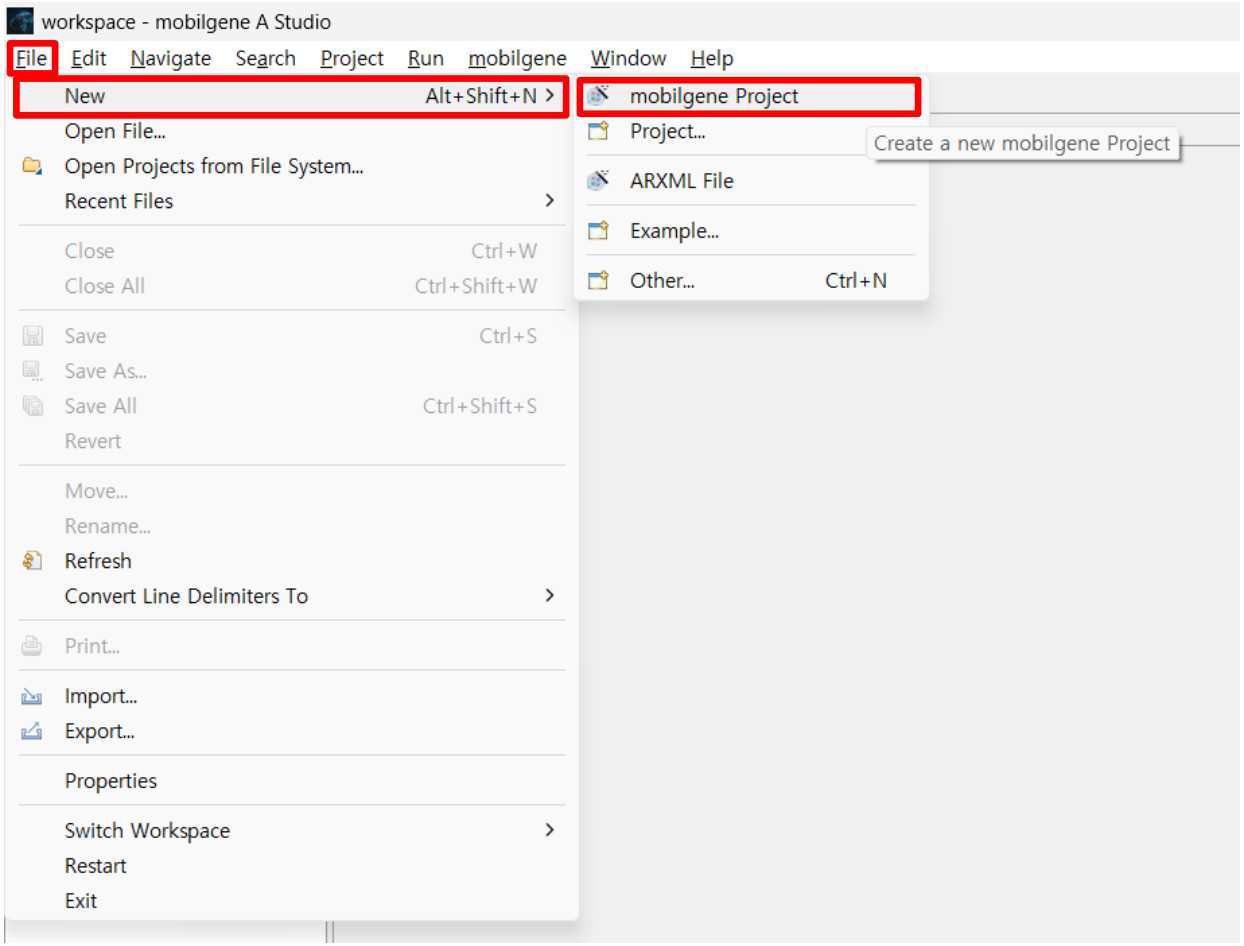
AUTOSAR AP 예제
- EXEC 01 -

Junho Kwak
Architecture and Compiler for Embedded System LAB.
School of Electronics Engineering, KNU, KOREA
2024-07-07

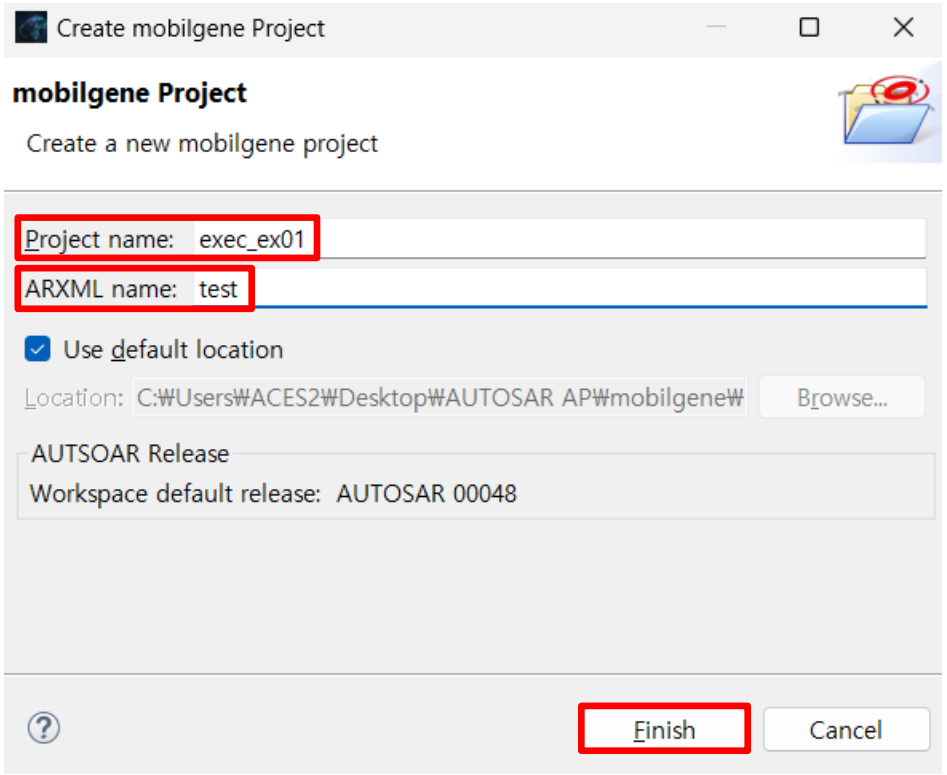
Project 생성

- **새로운 Mobilgene Project 생성 (1)**
  - ✓ 왼쪽 상단의 'File' – 'New' – 'mobilgene Project'를 클릭함



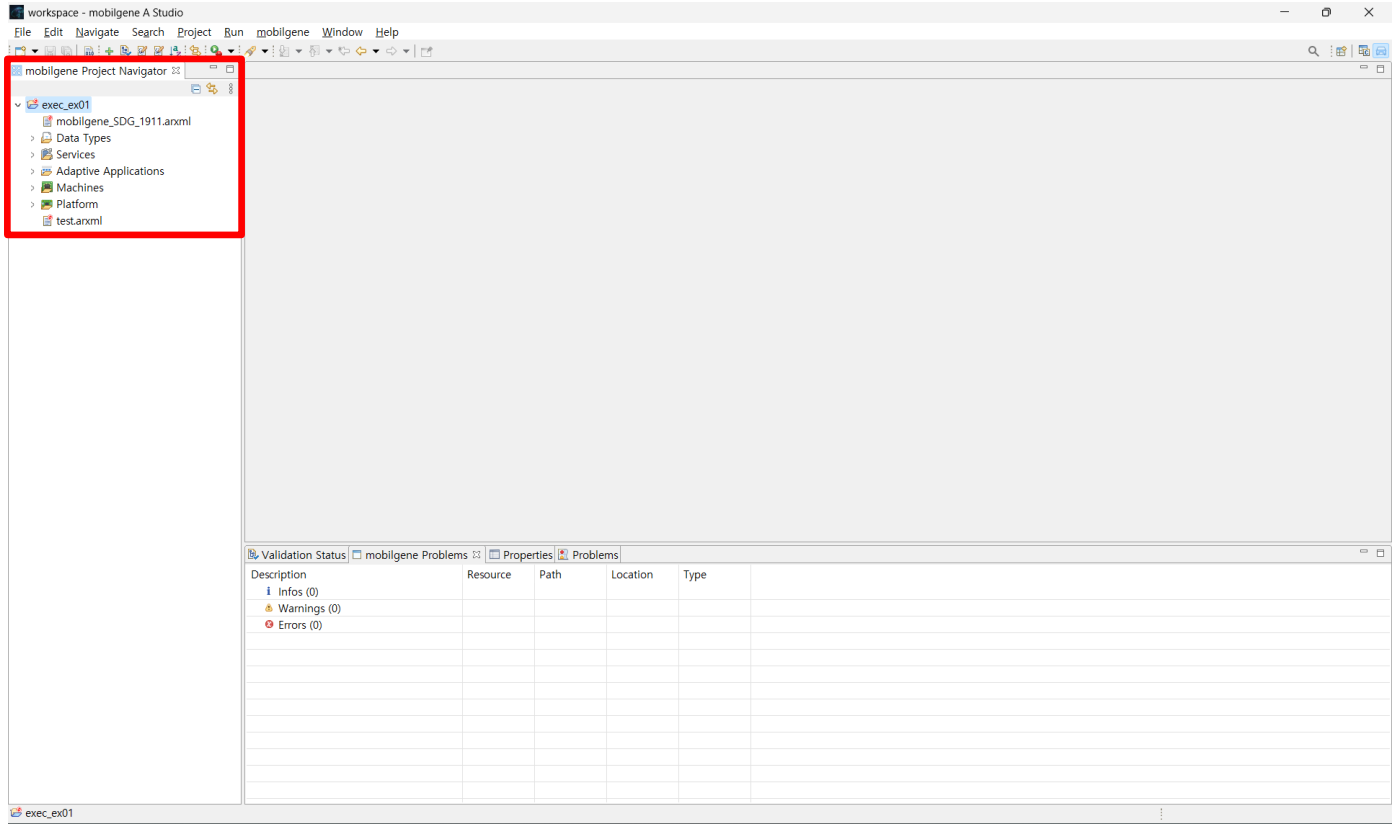
Project 생성

- **새로운 Mobilgene Project 생성 (2)**
  - ✓ 프로젝트 이름과 ARXML 파일 (초기 생성 파일)의 이름을 설정한 후, 'Finish' 버튼을 클릭함
    - ✓ Project name : 'exec_ex01'
    - ✓ ARXML name : 'test'



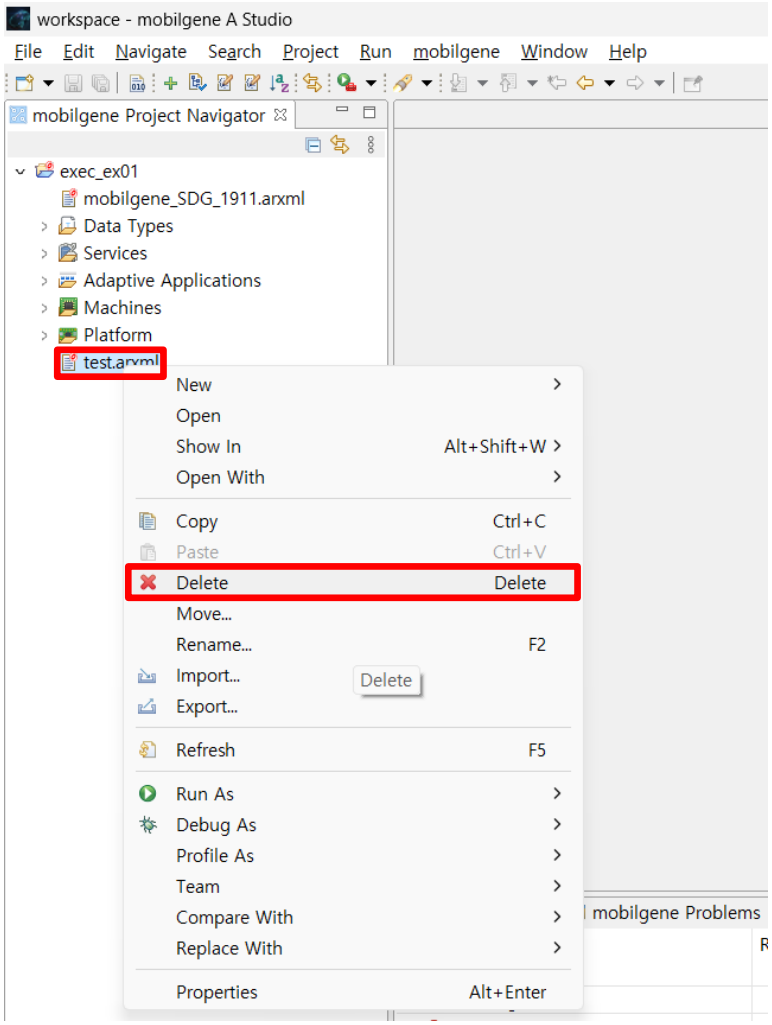
Project 생성

- **Mobilgene Project 생성 확인**
  - ✓ 왼쪽의 'mobilgene Project Navigator' 창에서 생성된 Project를 확인함
    - ✓ ARXML name으로 설정한 초기 생성 파일 (ARXML)이 포함되어 있음



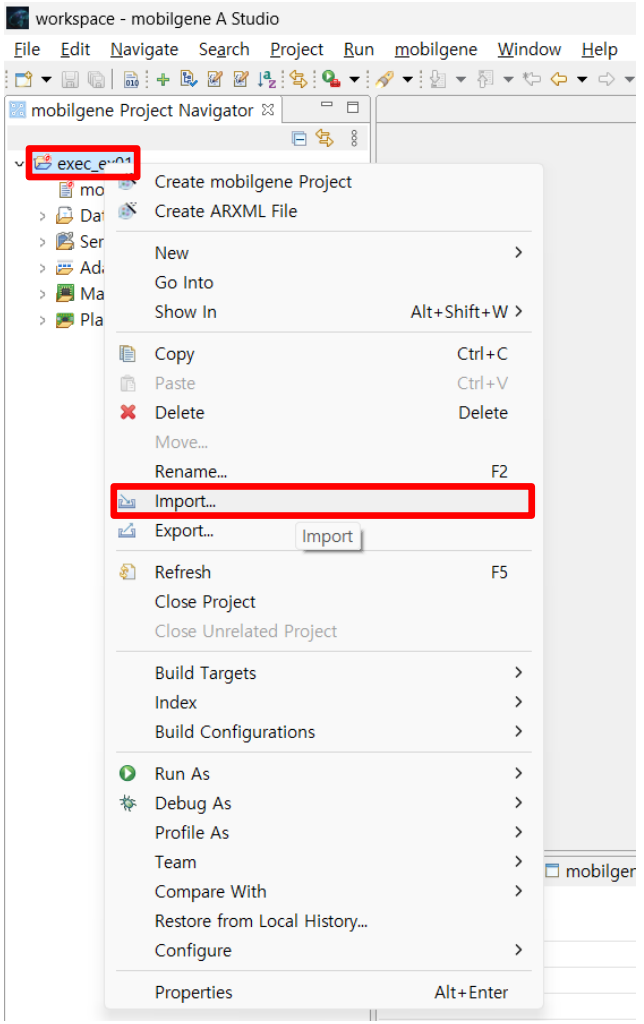
Project 생성

- **초기 생성 파일 삭제**
  - ✓ 기존 파일을 Import 하여 사용할 것이기 때문에 초기 생성 파일을 삭제함
    - ✓ 'test' 파일에서 우클릭 함
    - ✓ 'Delete' 버튼을 클릭함



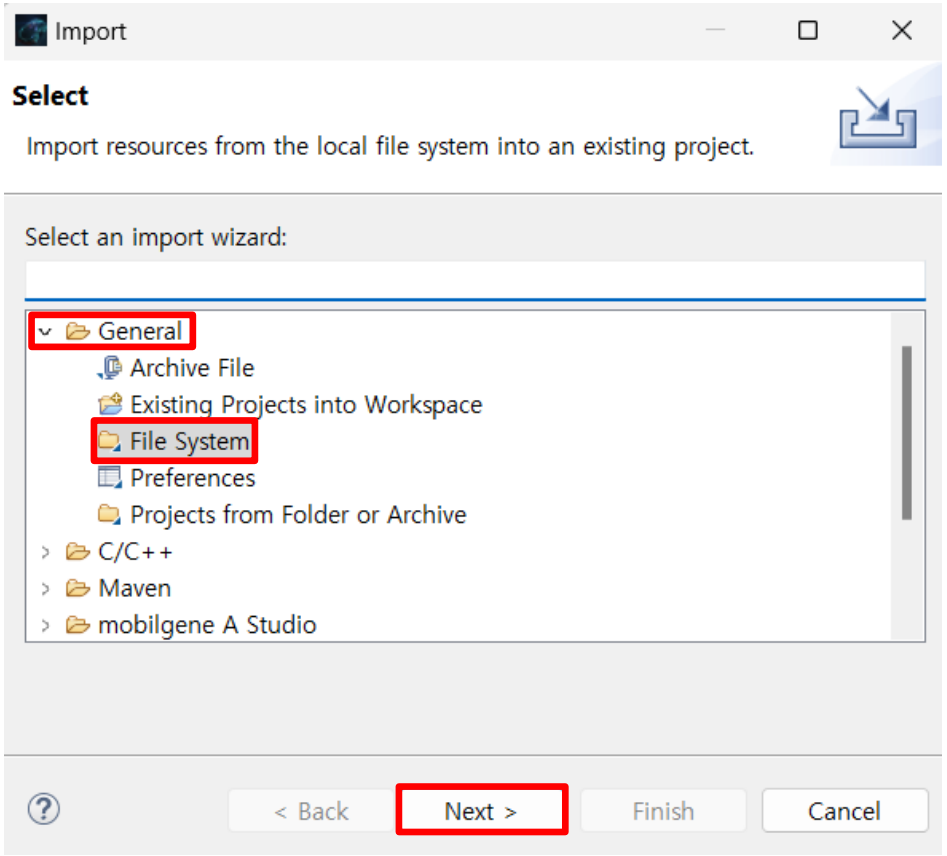
Import

- **파일 Import (1)**
  - ✓ 파일을 Import 할 Project에서 우클릭 함
  - ✓ 'Import…' 버튼을 클릭함



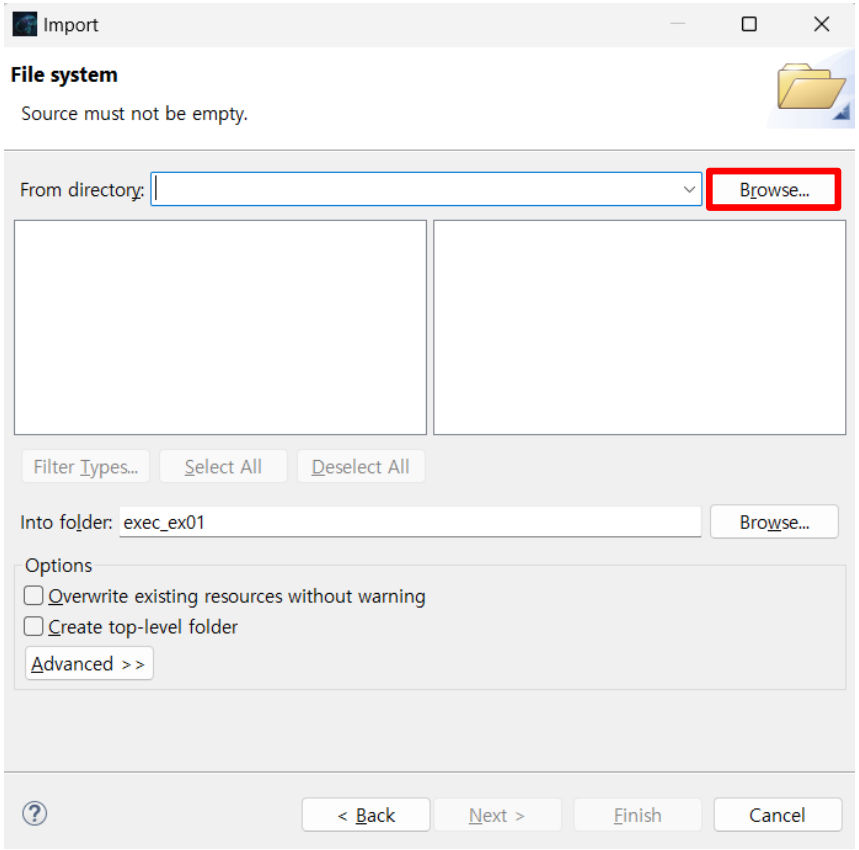
Import

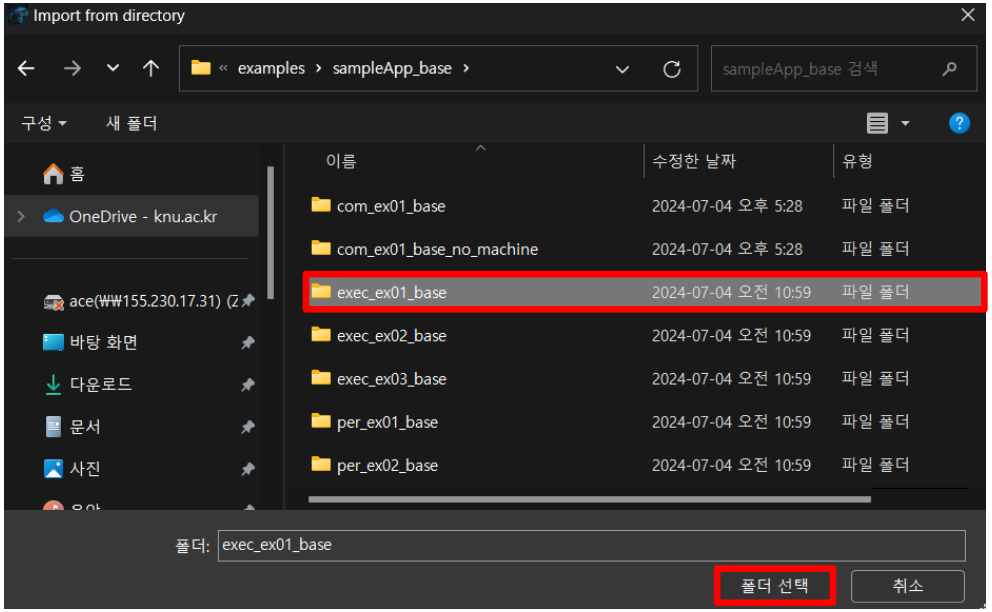
- **파일 Import (2)**
  - ✓ 'General' – 'File System'을 선택하고 'Next' 버튼을 클릭함



Import

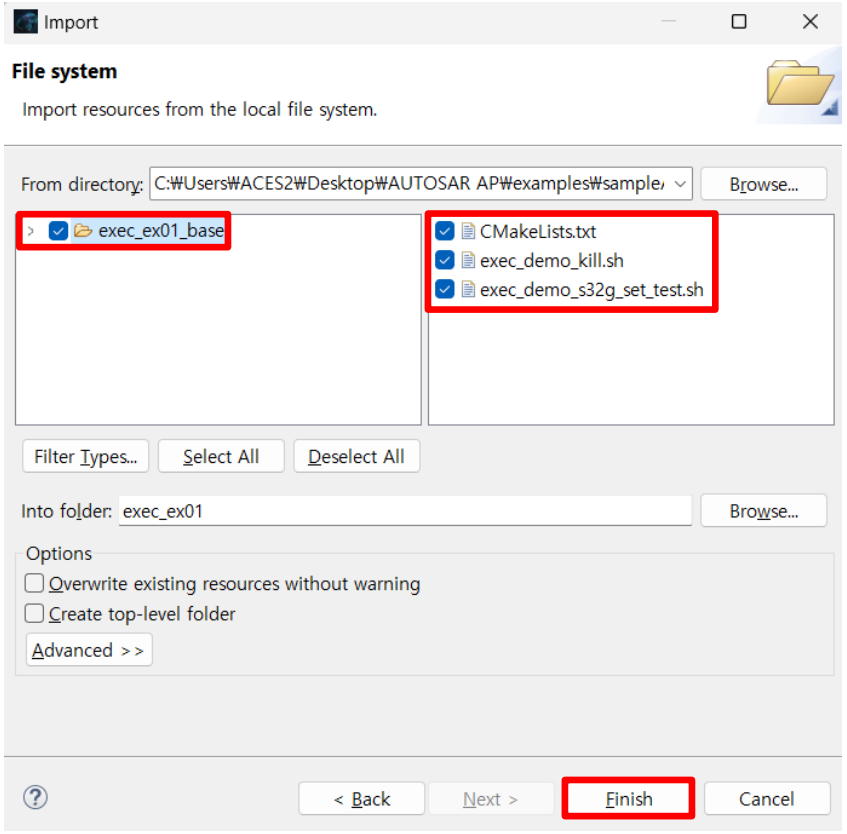
- **파일 Import (3)**
  - ✓ ‘Browse…’ 버튼을 클릭하고, Import 할 파일이 있는 폴더를 선택함





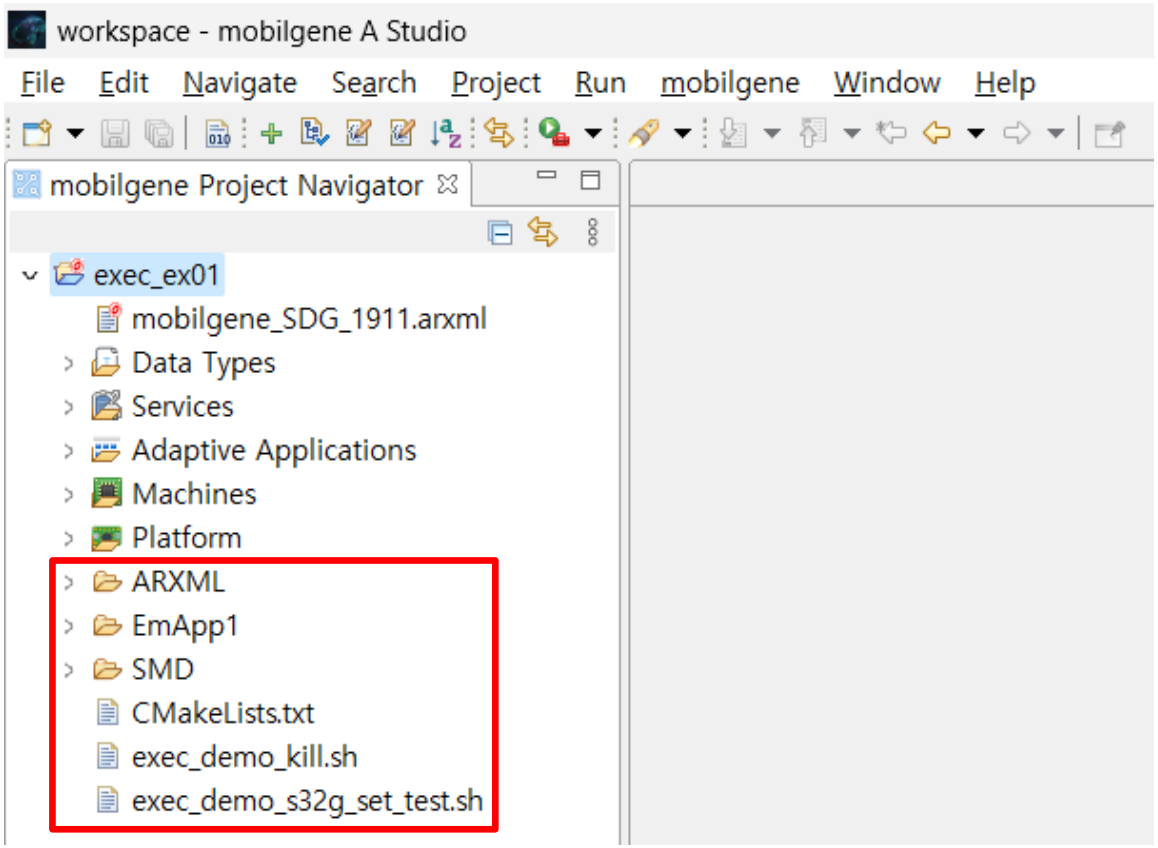
Import

- **파일 Import (4)**
  - ✓ 왼쪽 창의 폴더 옆 체크박스를 클릭하여 해당 폴더 내의 모든 파일을 선택함
  - ✓ 'Finish' 버튼을 클릭하여 Import를 수행함



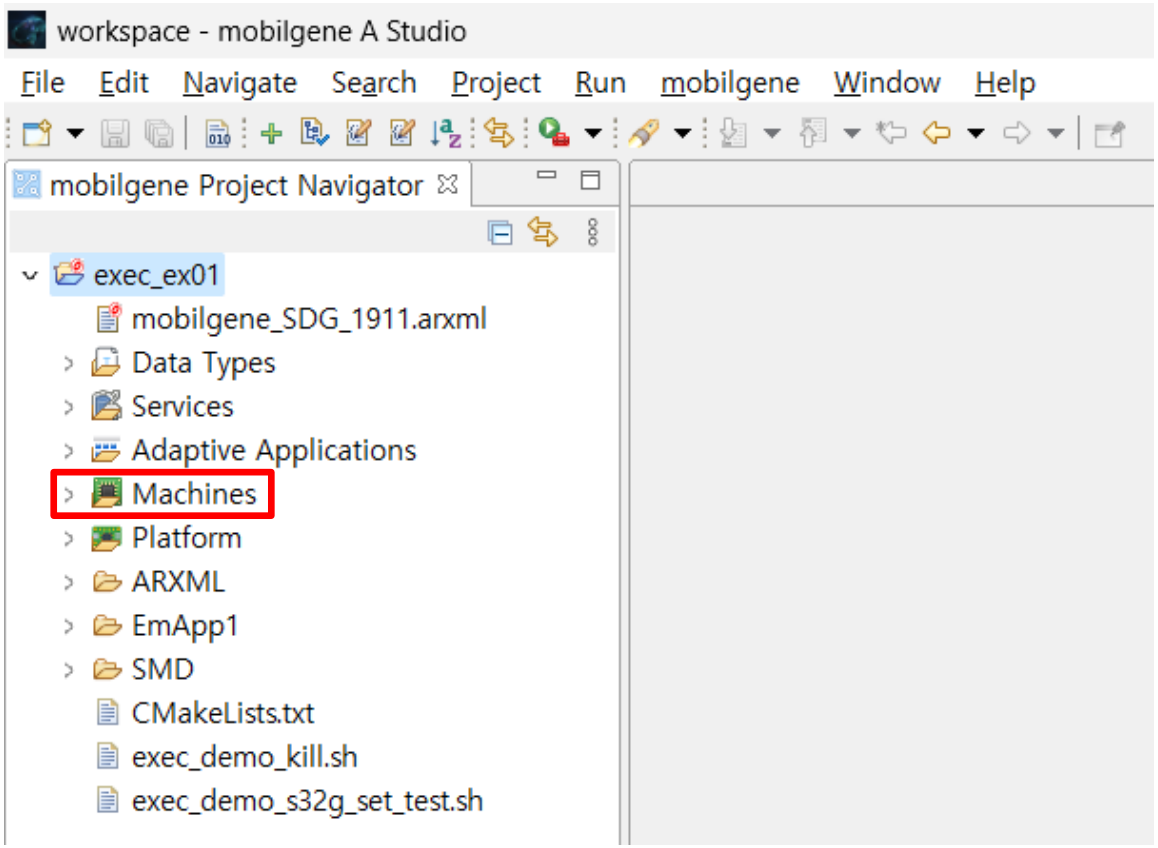
Import

- **파일 Import 확인**
  - ✓ 왼쪽의 'mobilgene Project Navigator' 창에서 Import 된 파일을 확인함



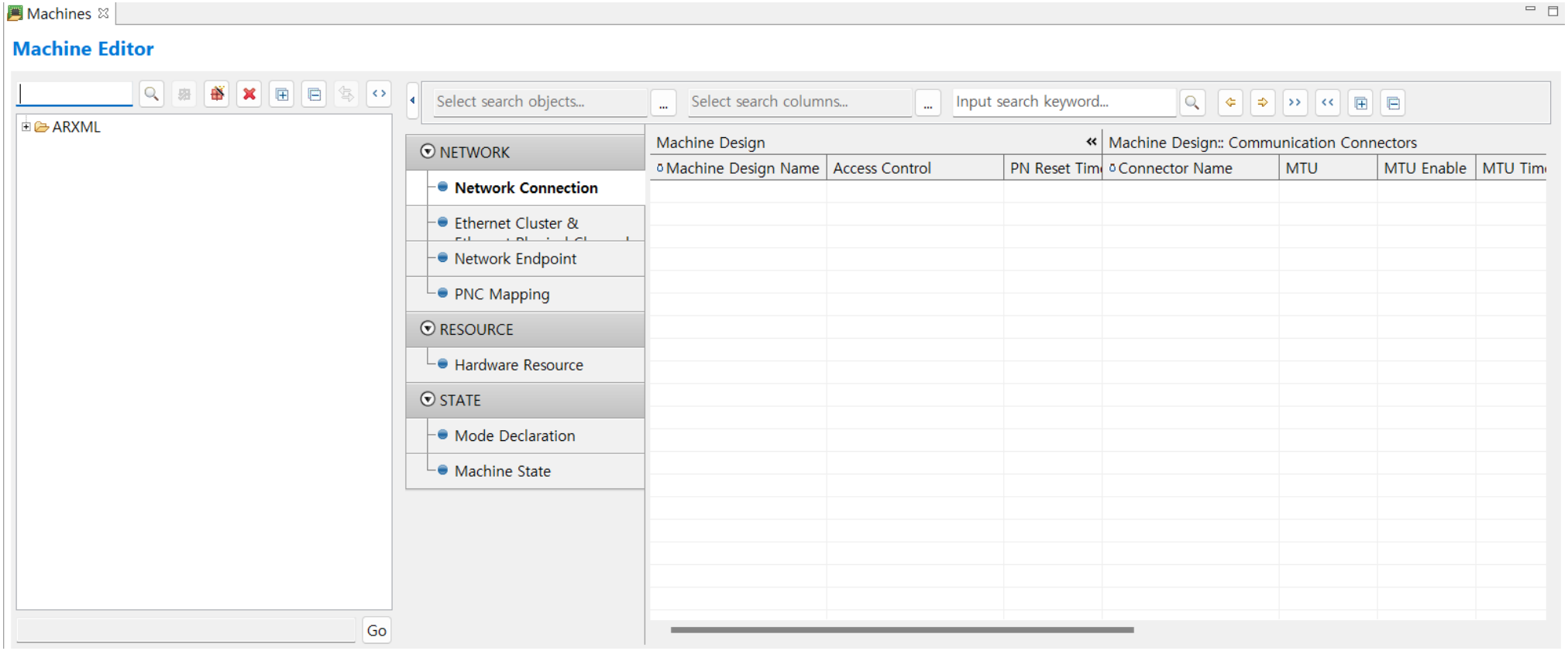
EXEC: Machines 설정

- **Machine Editor 활성화**
  - ✓ 왼쪽의 'mobilgene Project Navigator' 창에서 해당 Project의 'Machines'를 더블 클릭함



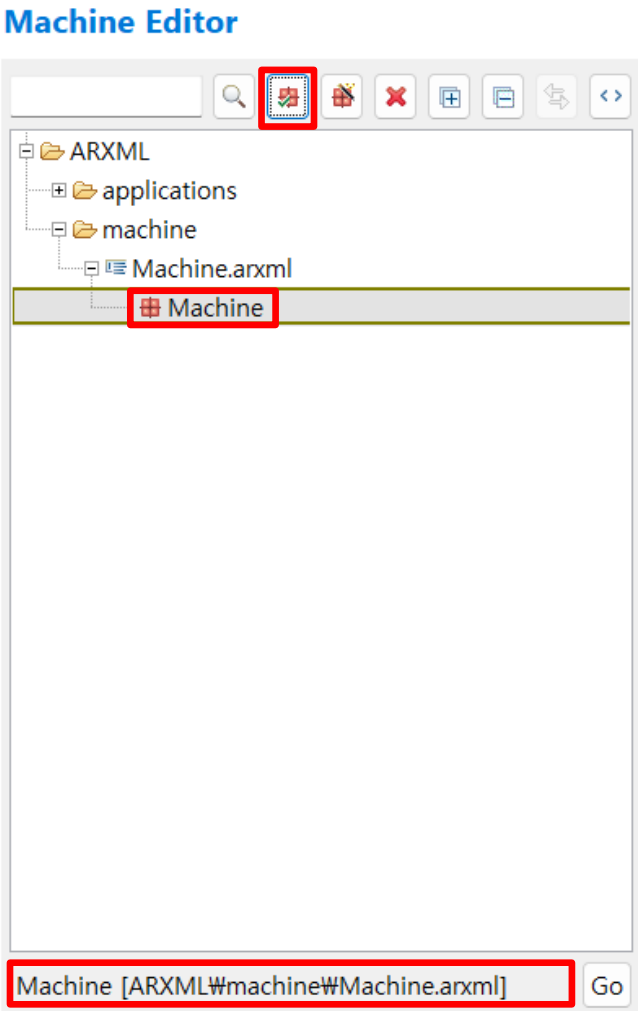
EXEC: Machines 설정

- **Machine Editor 활성화 확인**
  - ✓ 활성화 된 Machine Editor를 확인함



EXEC: Machines 설정

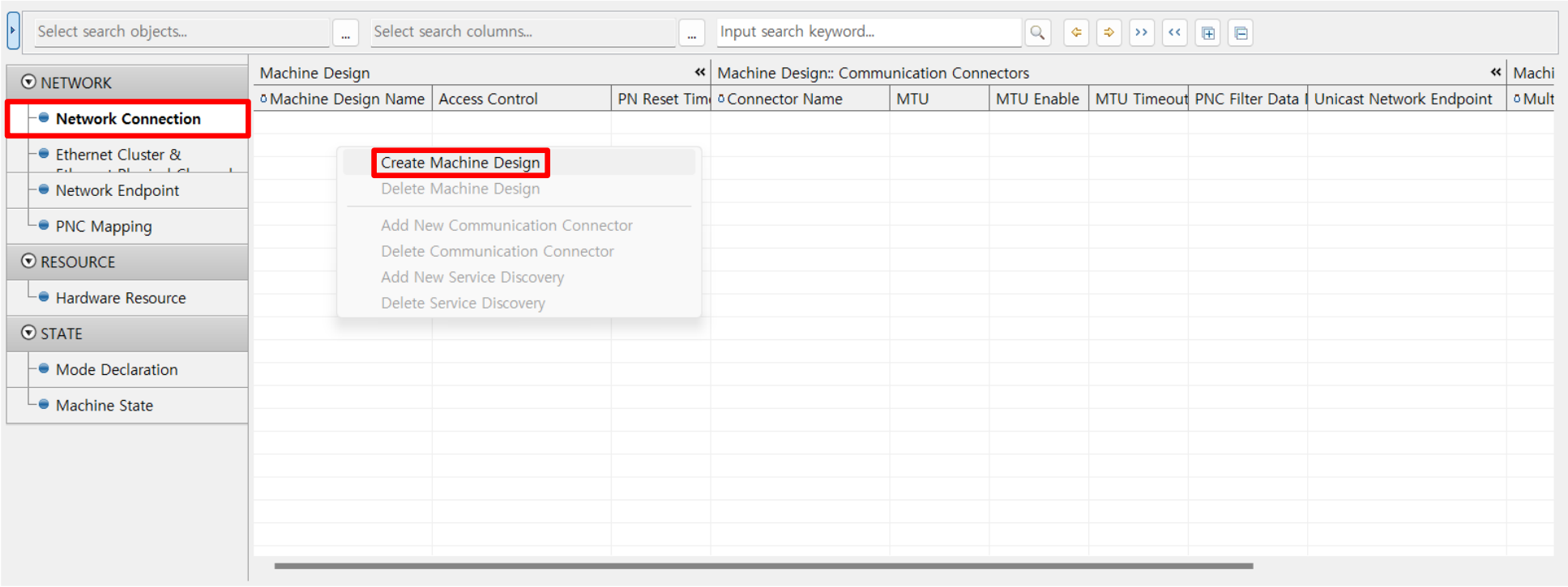
- **Default Package 설정**
  - ✓ 설정이 저장될 Default Package를 설정함
    - ✓ Default Package로 설정할 'Machine' Package를 선택함
    - ✓ 우측 상단의 'Set Default Package' 버튼을 클릭함
    - ✓ 하단에 설정된 Default Package를 확인함



EXEC: Machines 설정

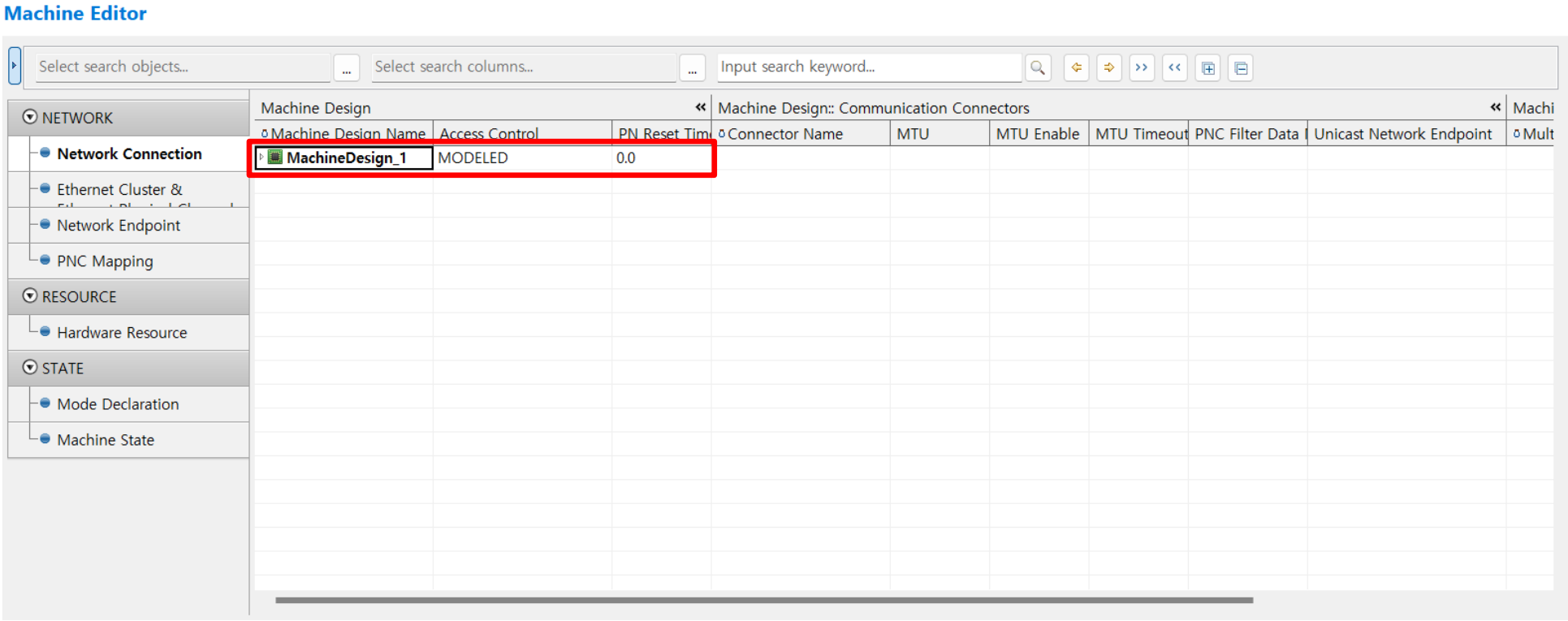
- **Machine Design 추가**
  - ✓ Machine Design 추가를 위해 좌측의 'Network Connection' 탭으로 이동함
  - ✓ 빈 곳에서 우클릭을 하여 'Create Machine Design'을 클릭함

Machine Editor



EXEC: Machines 설정

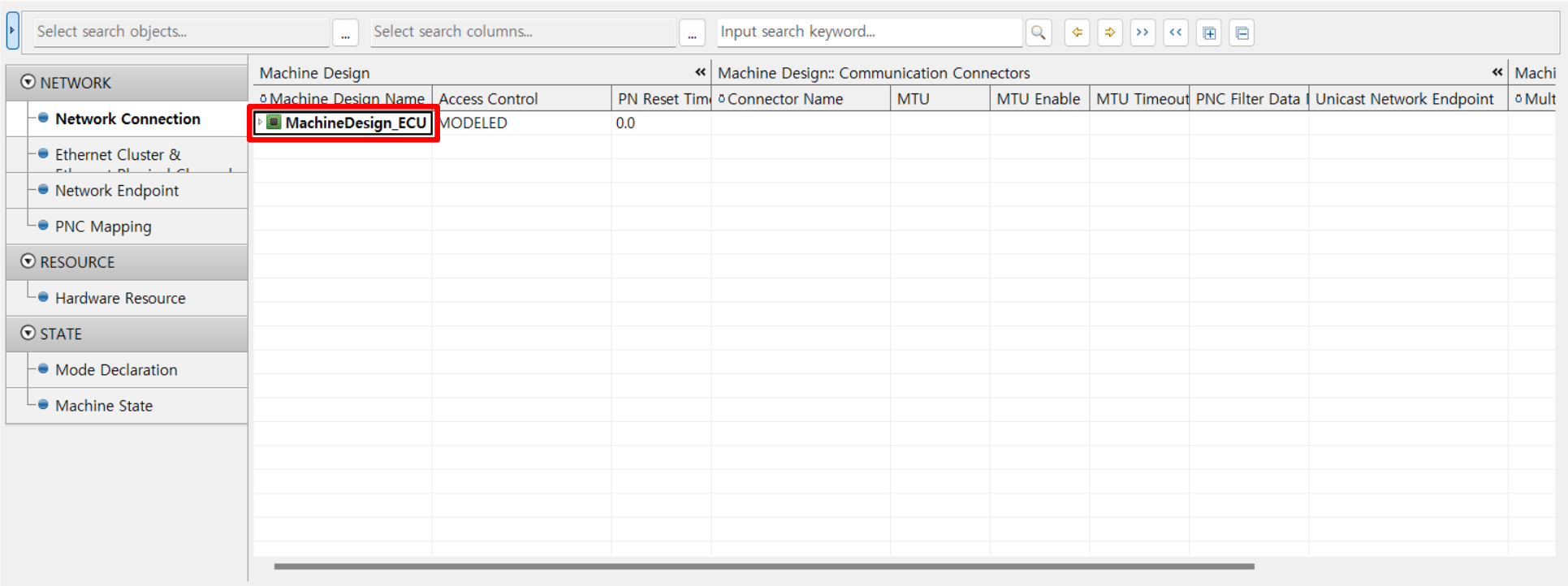
- **Machine Design 추가 확인**
  - ✓ 생성된 새로운 Machine Design을 확인함



EXEC: Machines 설정

- **Machine Design 설정**
  - ✓ Machine Design Name을 더블 클릭하여 다음과 같이 수정함
    - ✓ Machine Design Name : 'MachineDesign_ECU'

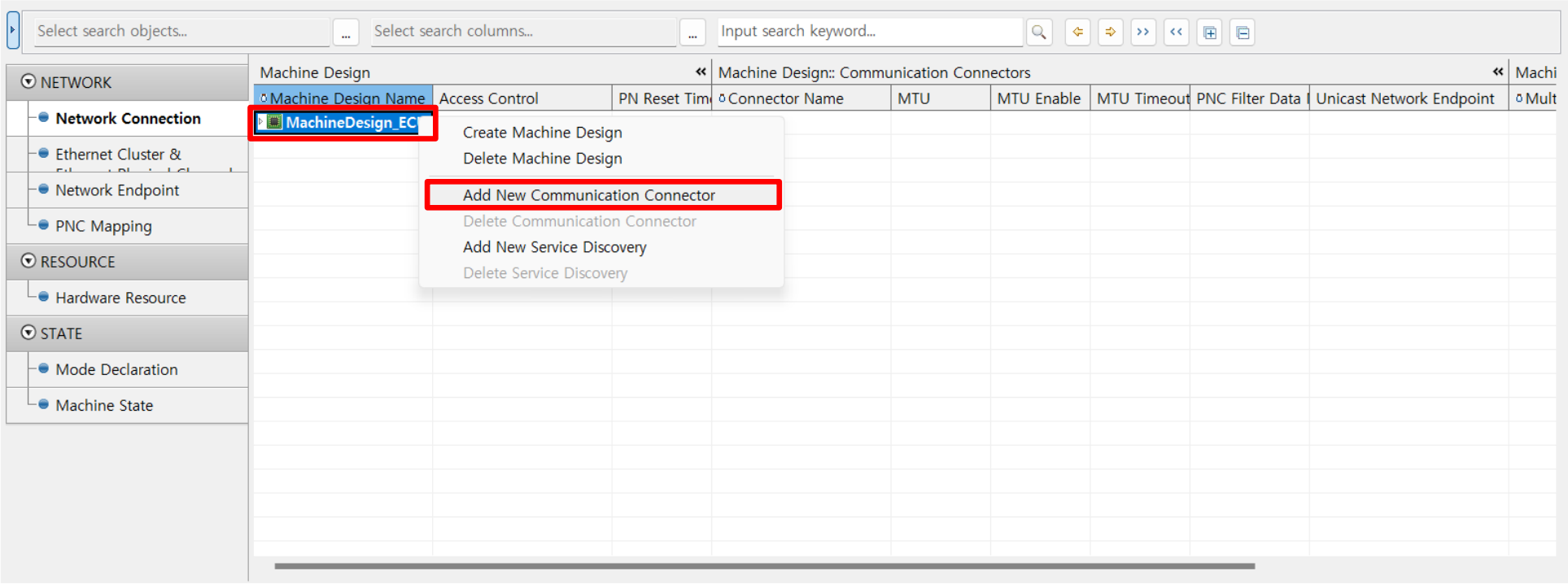
Machine Editor



EXEC: Machines 설정

- **Communication Connector 추가**
  - ✓ 생성한 Machine Design에서 우클릭하여 'Add New Communication Connector'를 클릭함

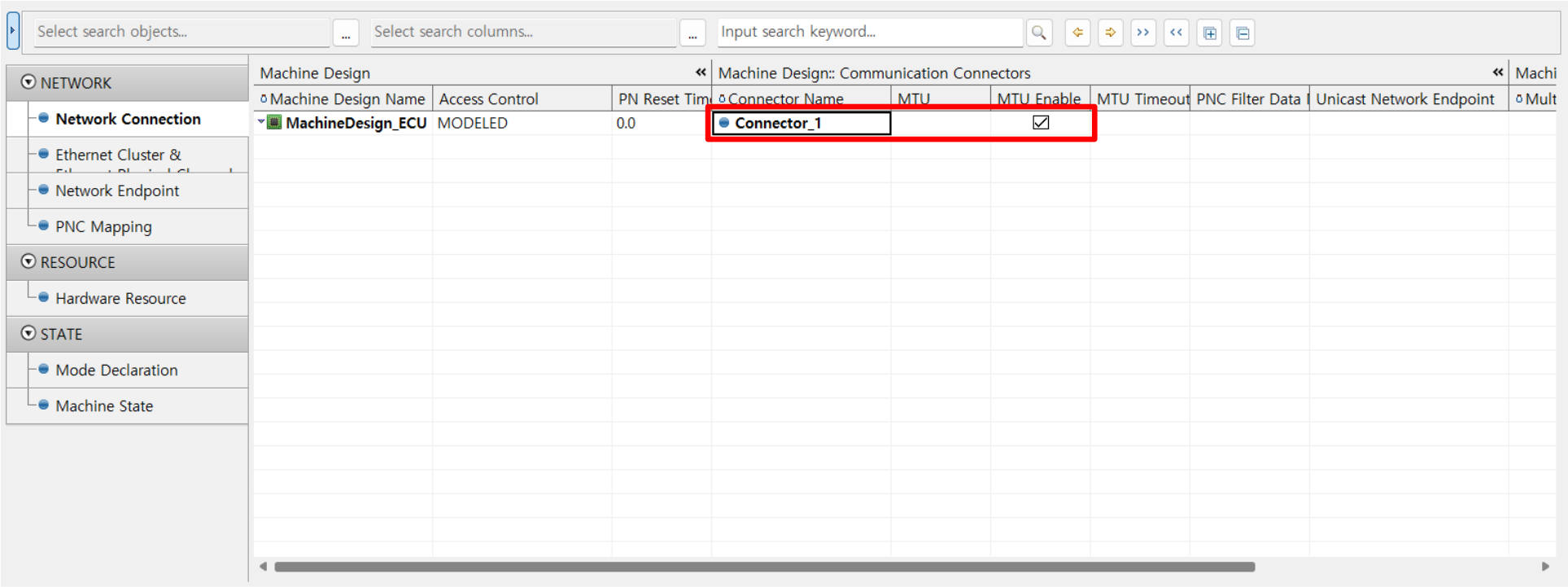
Machine Editor



EXEC: Machines 설정

- **Communication Connector 추가 확인**
  - ✓ 생성된 새로운 Communication Connector를 확인함

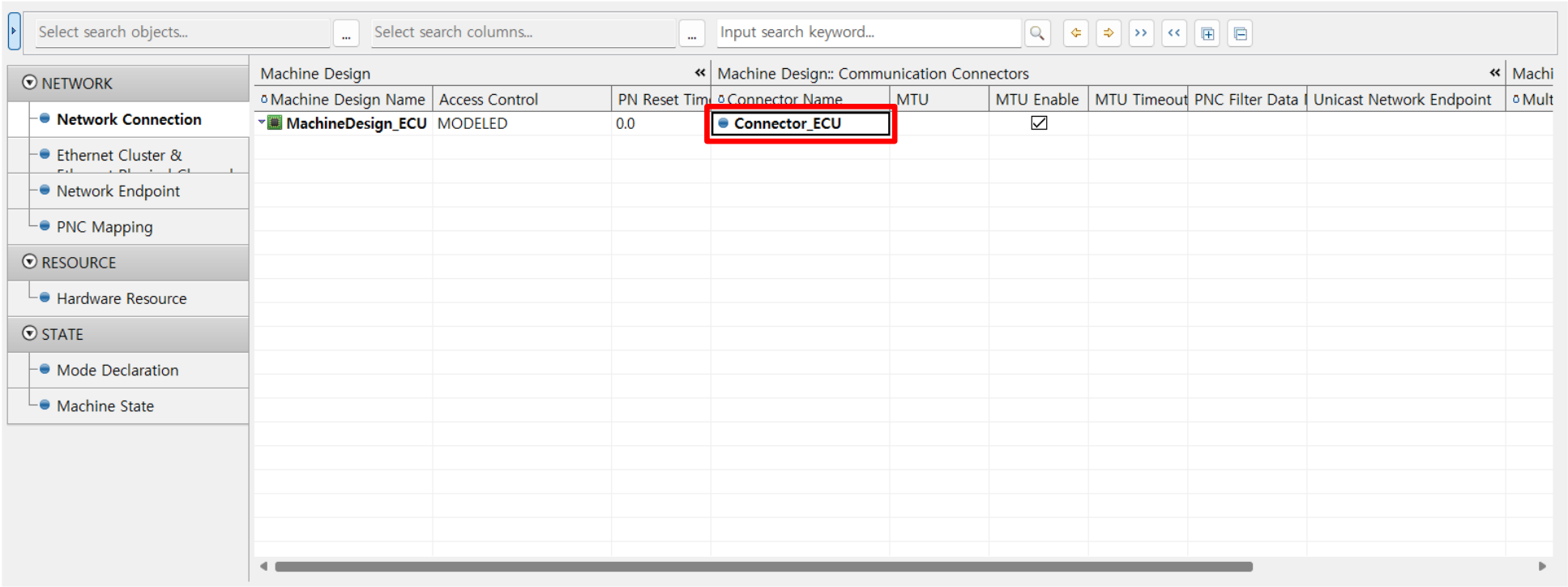
Machine Editor



EXEC: Machines 설정

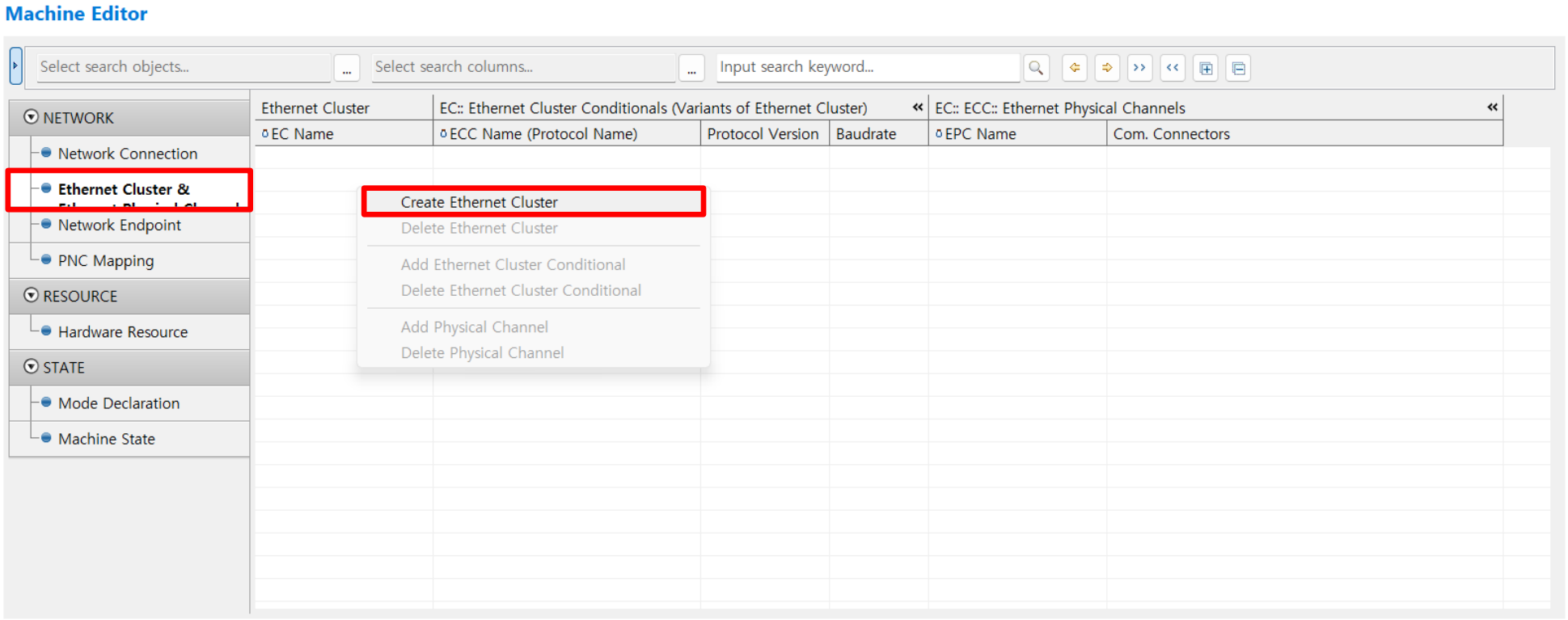
- **Communication Connector 설정**
  - ✓ Connector Name을 더블 클릭하여 다음과 같이 수정함
    - ✓ Connector Name : 'Connector_ECU'

Machine Editor



EXEC: Machines 설정

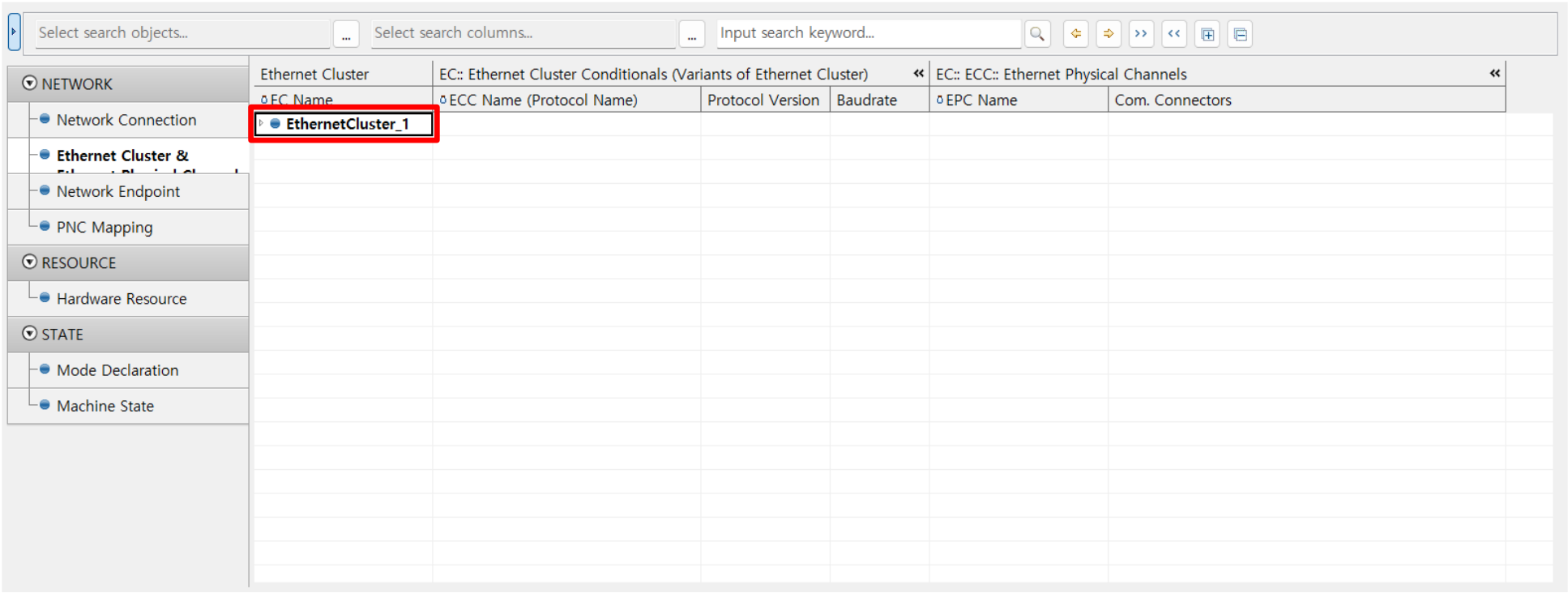
- **Ethernet Cluster 추가**
  - ✓ Ethernet Cluster 추가를 위해 좌측의 'Ethernet Cluster & ...' 탭으로 이동함
  - ✓ 빈 곳에서 우클릭을 하여 'Create Ethernet Cluster'를 클릭함



EXEC: Machines 설정

- **Ethernet Cluster 추가 확인**
  - ✓ 생성된 새로운 Ethernet Cluster를 확인함

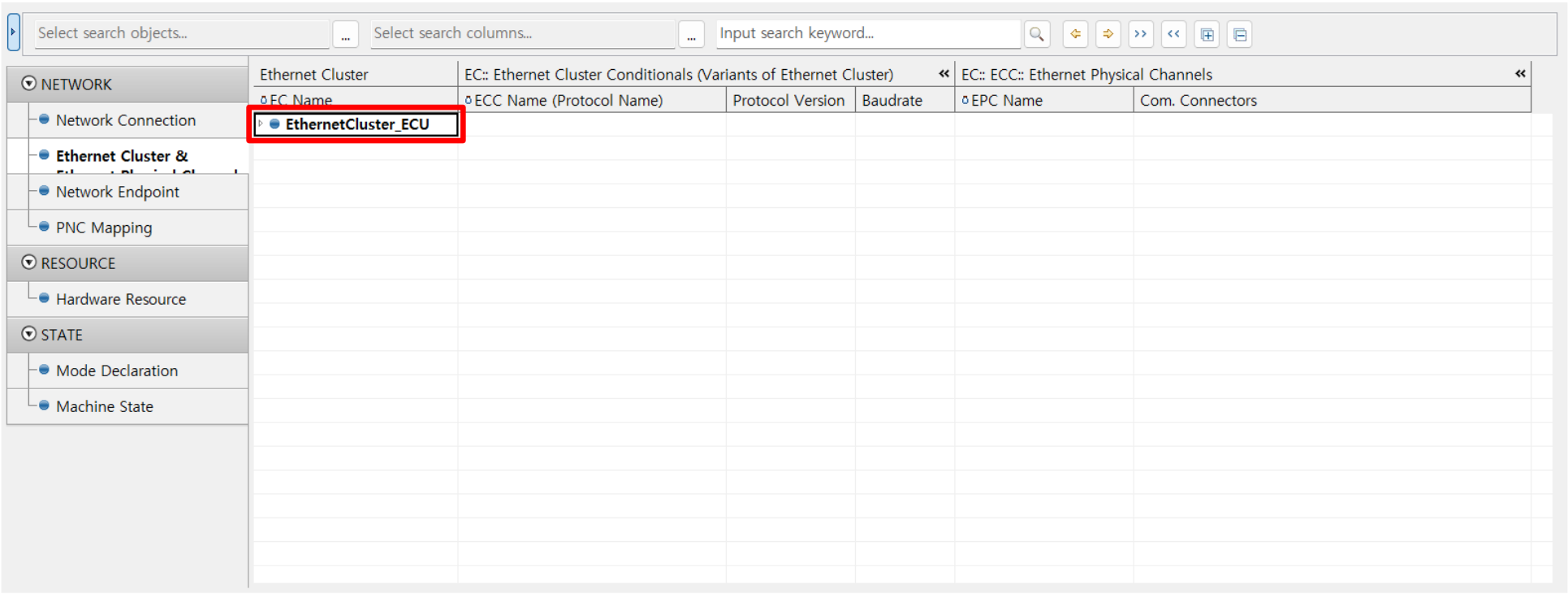
Machine Editor



EXEC: Machines 설정

- **Ethernet Cluster 설정**
  - ✓ EC Name을 더블 클릭하여 다음과 같이 수정함
    - ✓ EC Name : 'EthernetCluster_ECU'

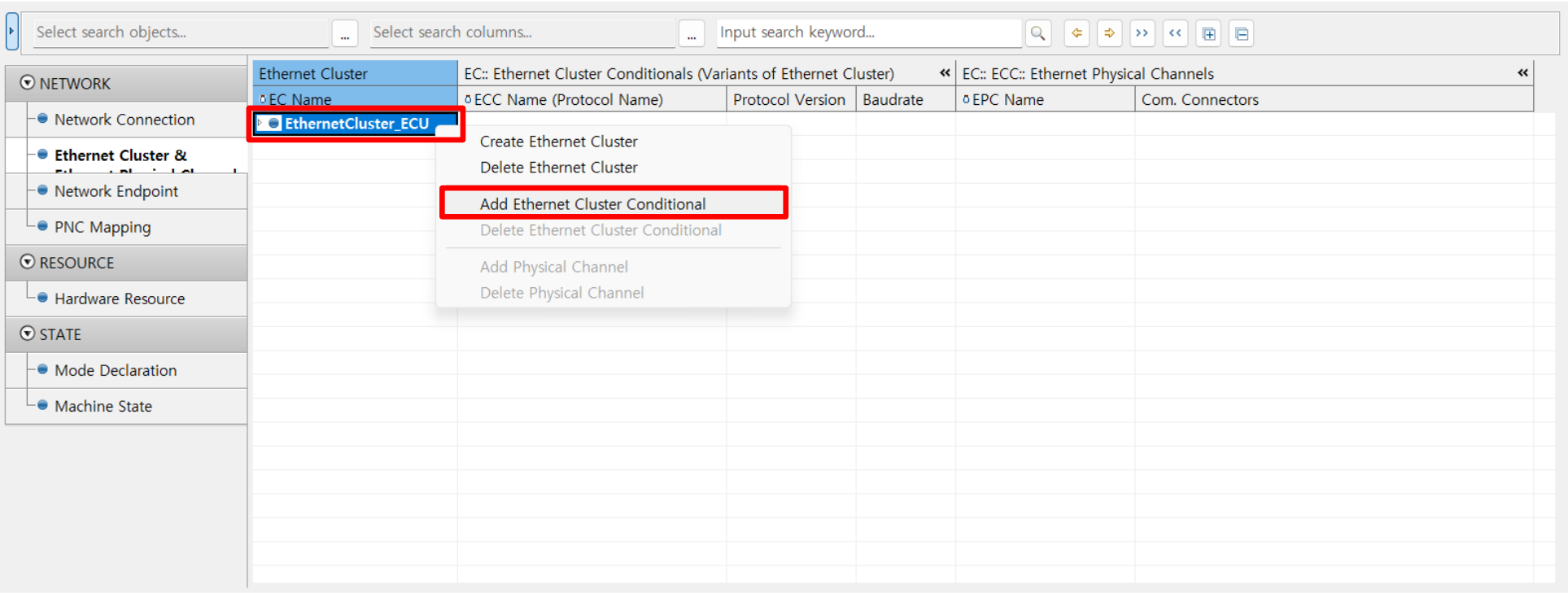
Machine Editor



EXEC: Machines 설정

- **Ethernet Cluster Conditional 추가**
  - ✓ 생성한 Ethernet Cluster에서 우클릭하여 'Add Ethernet Cluster Conditional'을 클릭함

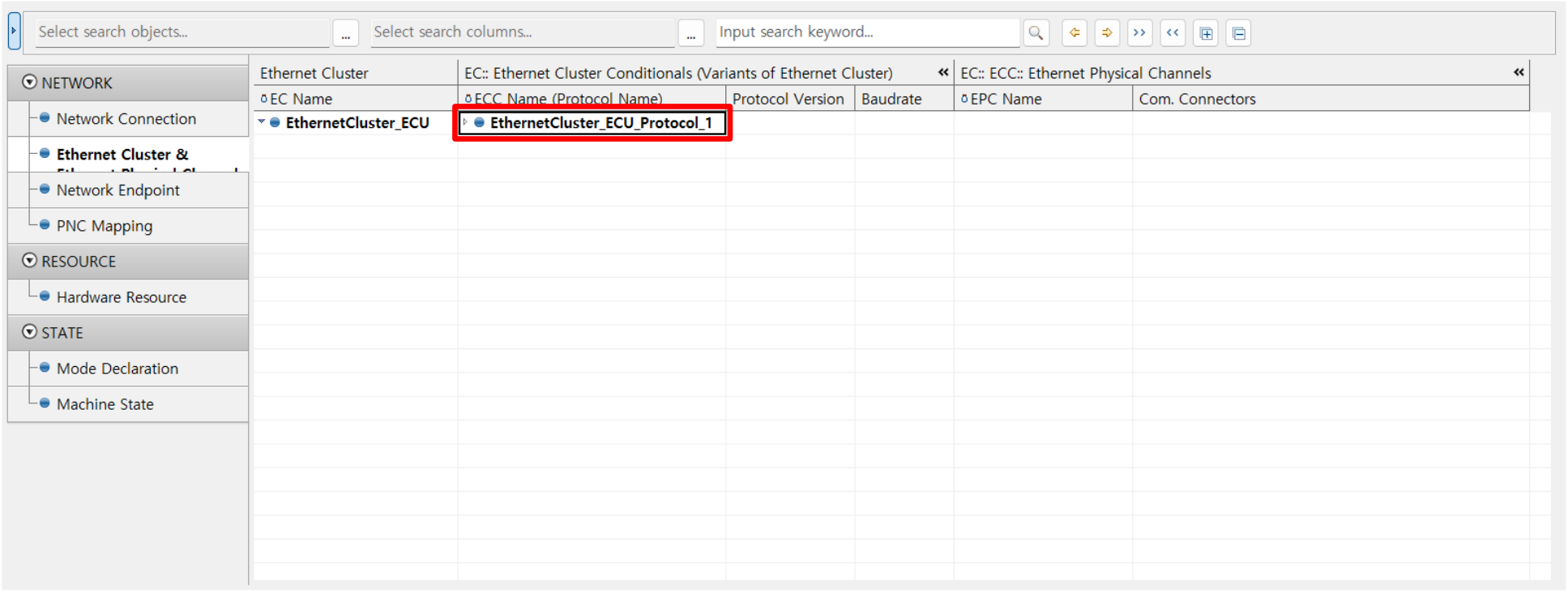
Machine Editor



EXEC: Machines 설정

- **Ethernet Cluster Conditional 추가 확인**
  - ✓ 생성된 새로운 Ethernet Cluster Conditional을 확인함

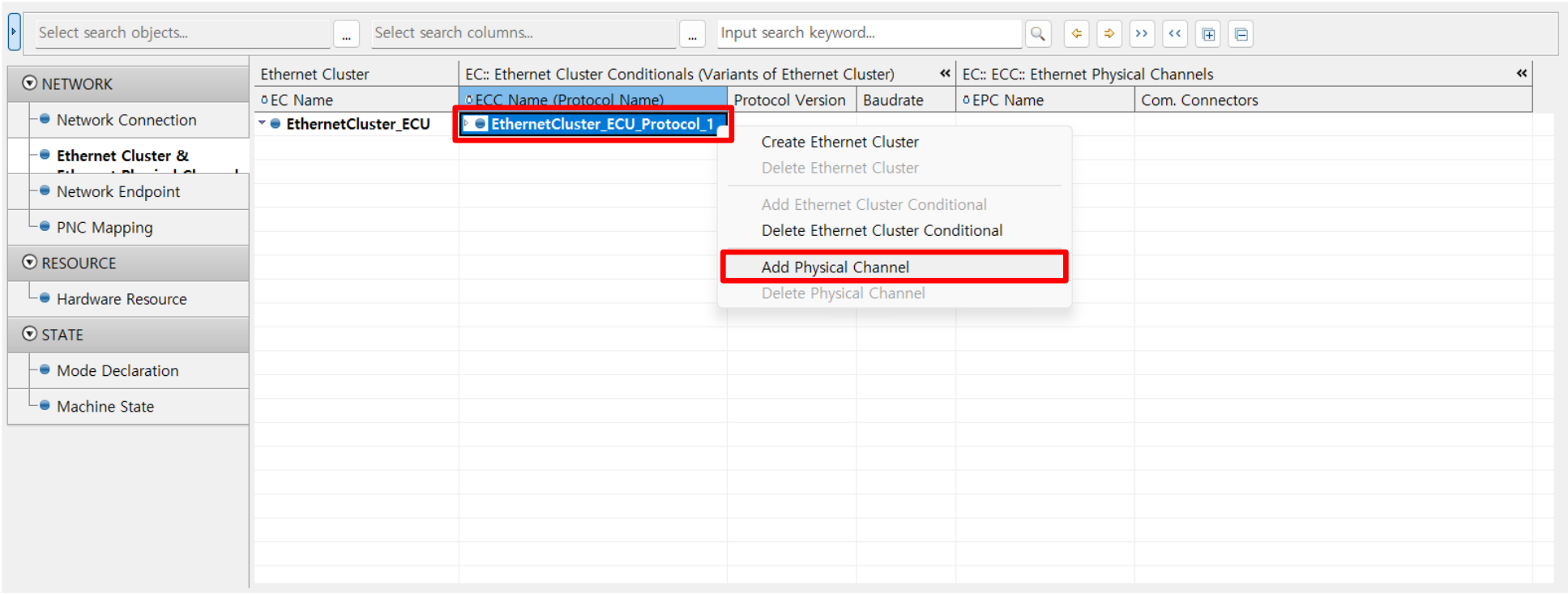
Machine Editor



EXEC: Machines 설정

- **Physical Channel 추가**
  - ✓ 생성한 Ethernet Cluster Conditional에서 우클릭하여 'Add Physical Channel'을 클릭함

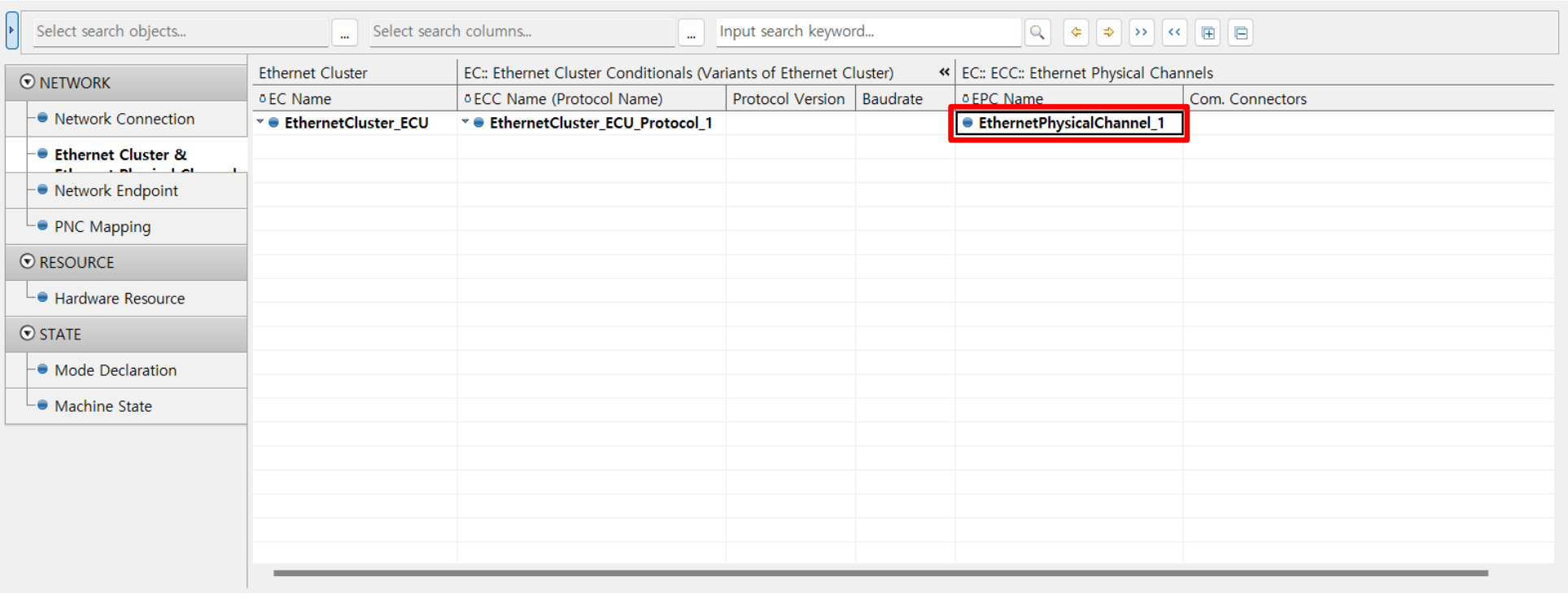
Machine Editor



EXEC: Machines 설정

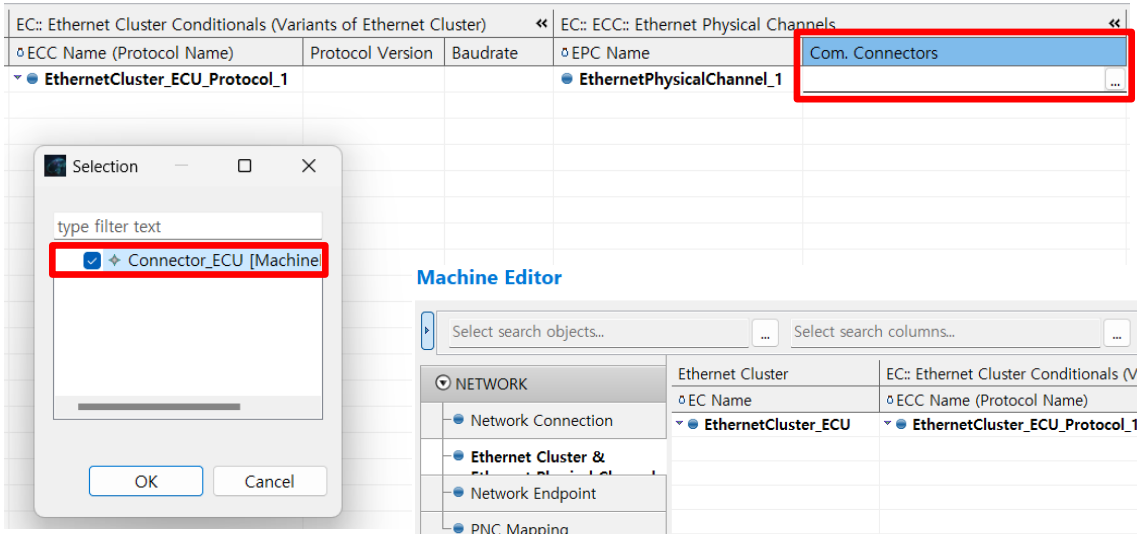
- **Physical Channel 추가 확인**
  - ✓ 생성된 새로운 Physical Channel을 확인함

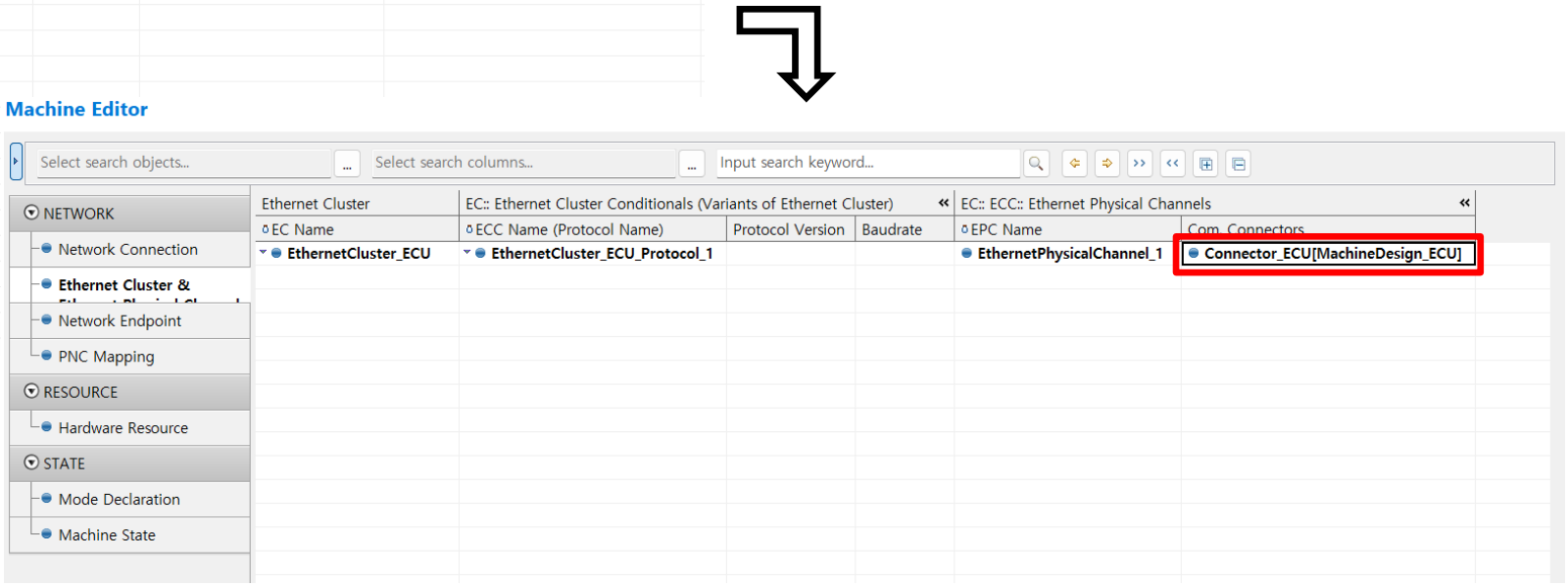
Machine Editor



EXEC: Machines 설정

- **Physical Channel 설정**
  - ✓ Com. Connectors를 더블 클릭하고, '…' 버튼을 클릭하여 연결될 Connector를 설정함
    - ✓ Com. Connectors : 'Connector_ECU'

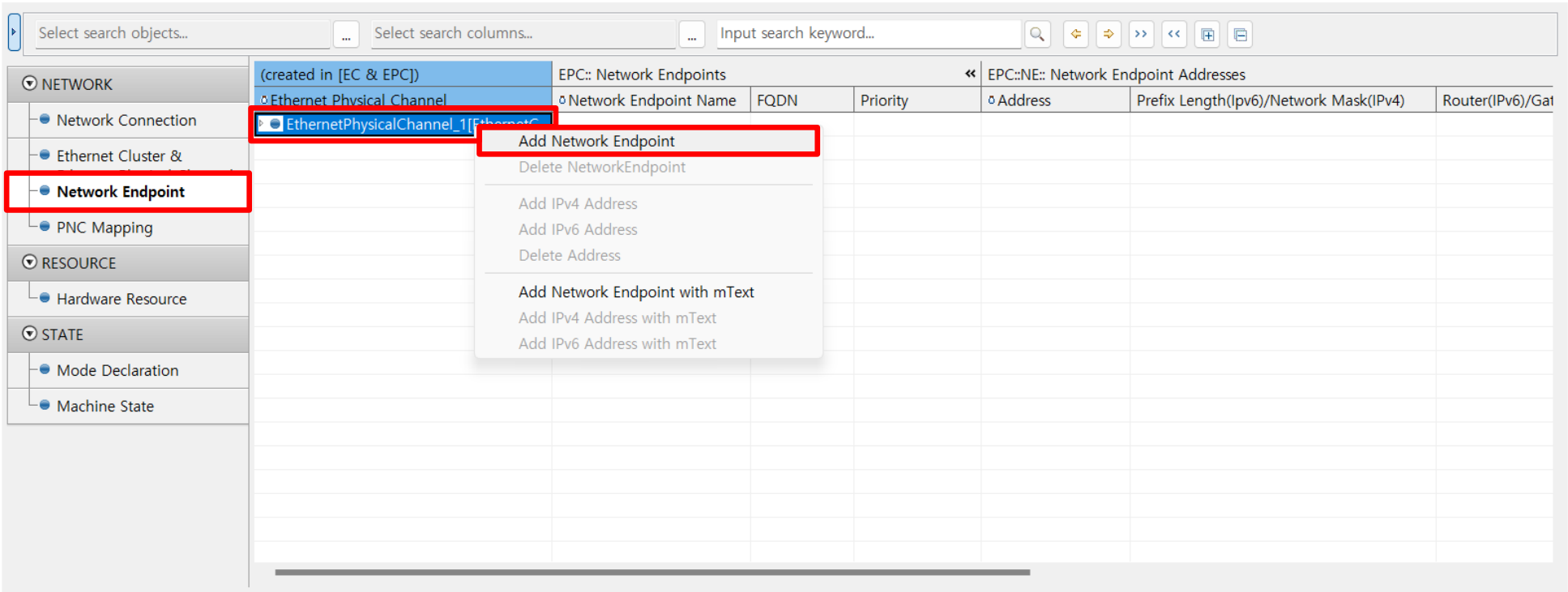




EXEC: Machines 설정

- **Network Endpoint 추가**
  - ✓ Network Endpoint 추가를 위해 좌측의 'Network Endpoint' 탭으로 이동함
  - ✓ 이전에 생성한 Ethernet Physical Channel에서 우클릭하여 'Add Network Endpoint'를 클릭함

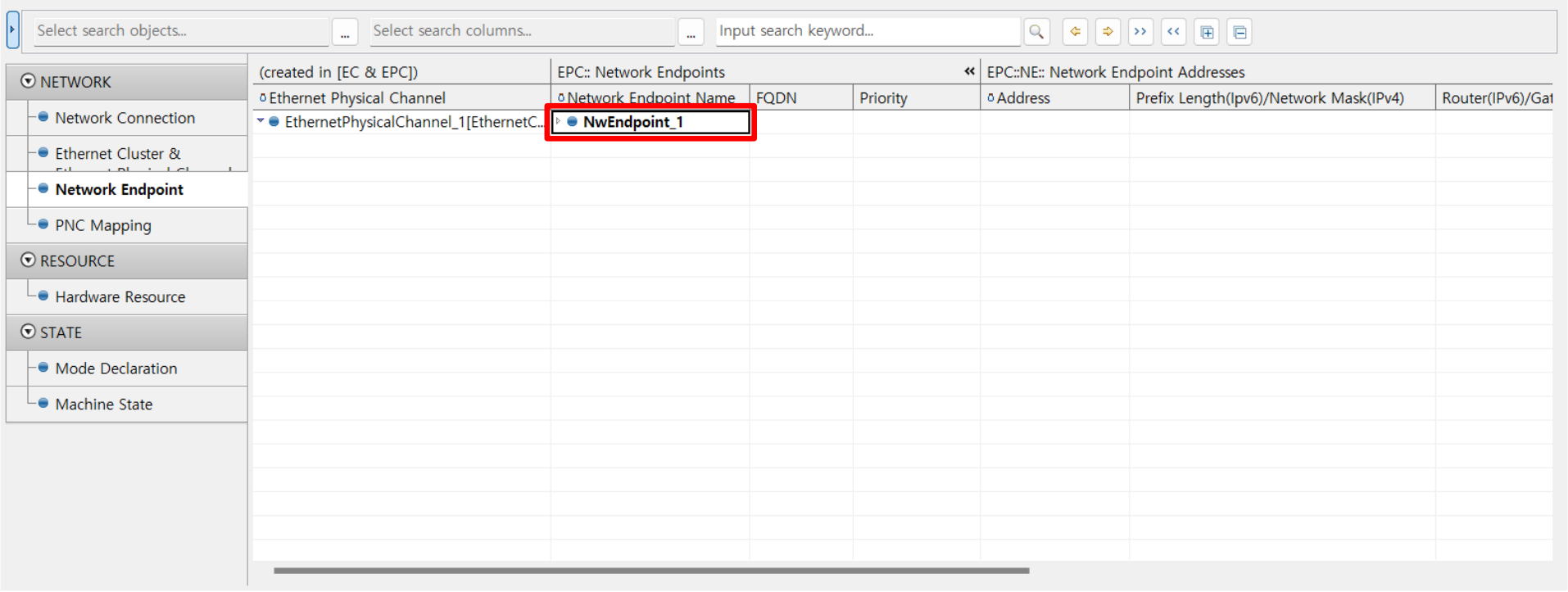
Machine Editor



EXEC: Machines 설정

- **Network Endpoint 추가 확인**
  - ✓ 생성된 새로운 Network Endpoint를 확인함

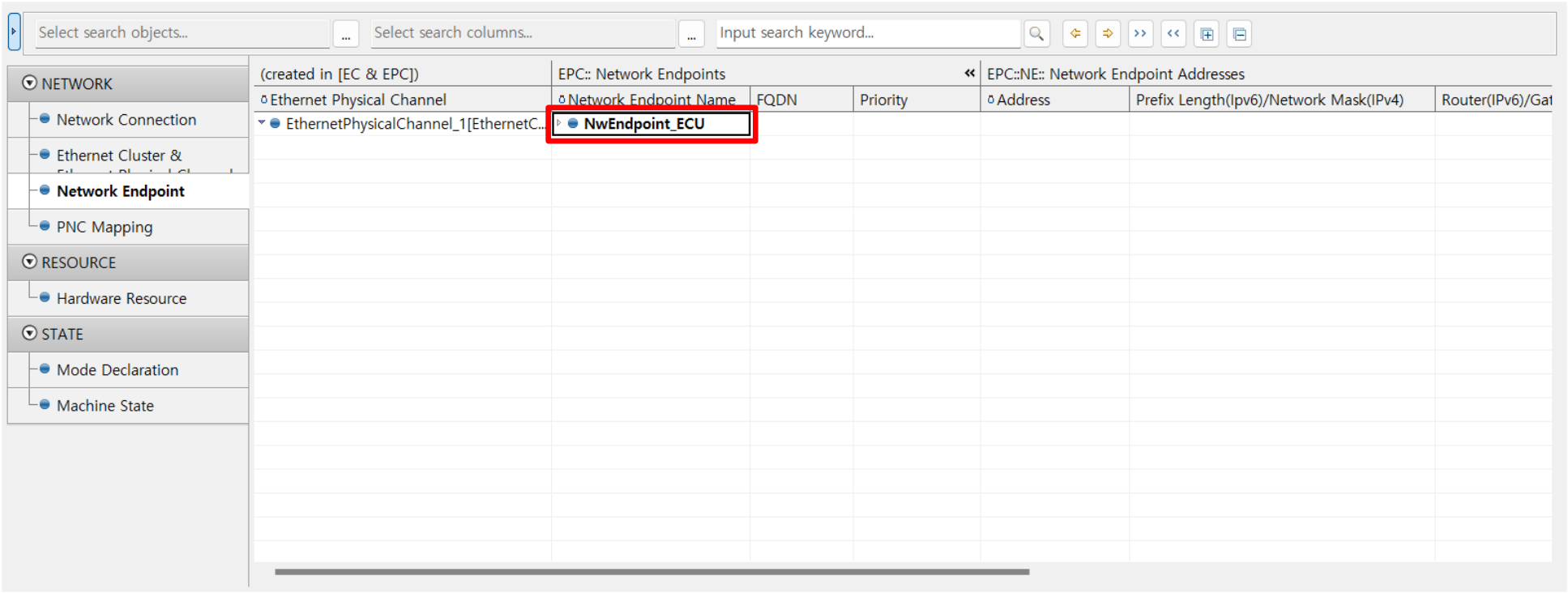
Machine Editor



EXEC: Machines 설정

- **Network Endpoint 설정**
  - ✓ Network Endpoint Name을 더블 클릭하여 다음과 같이 수정함
    - ✓ Network Endpoint Name : 'NwEndpoint_ECU'

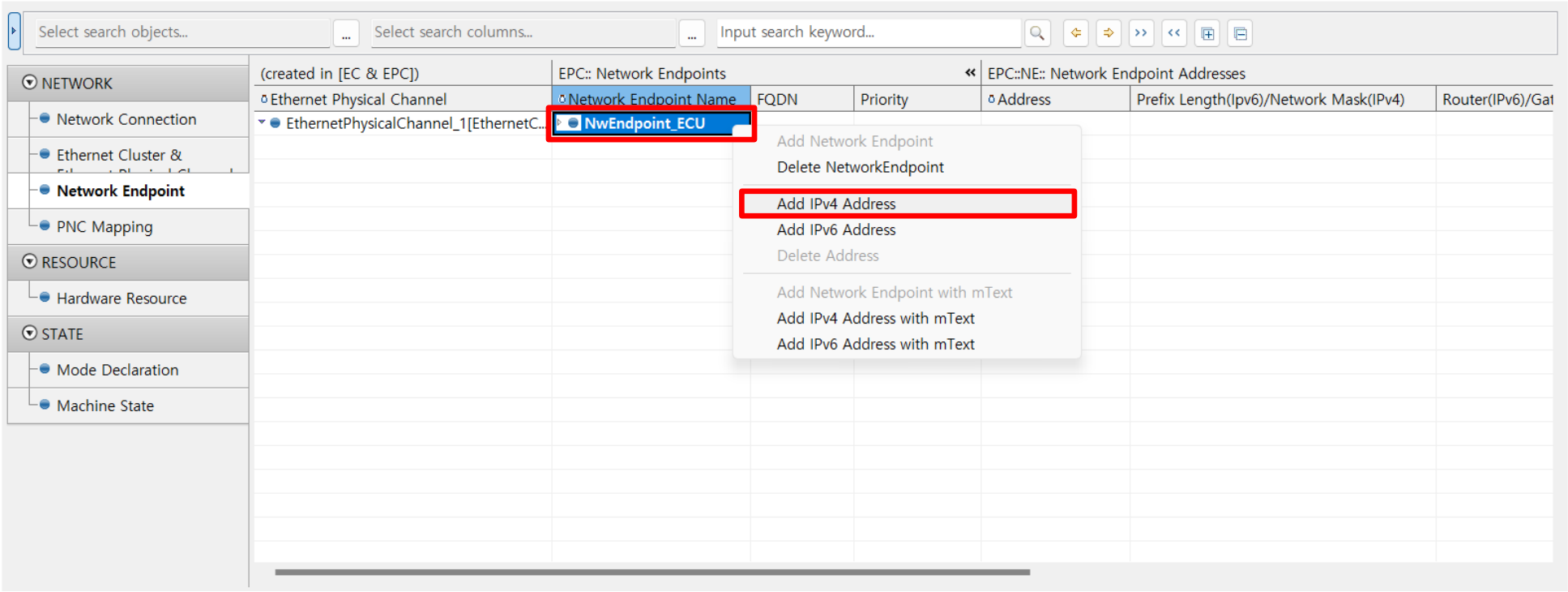
Machine Editor



EXEC: Machines 설정

- **IPv4 Address 추가**
  - ✓ 생성한 Network Endpoint에서 우클릭하여 'Add IPv4 Address'를 클릭함

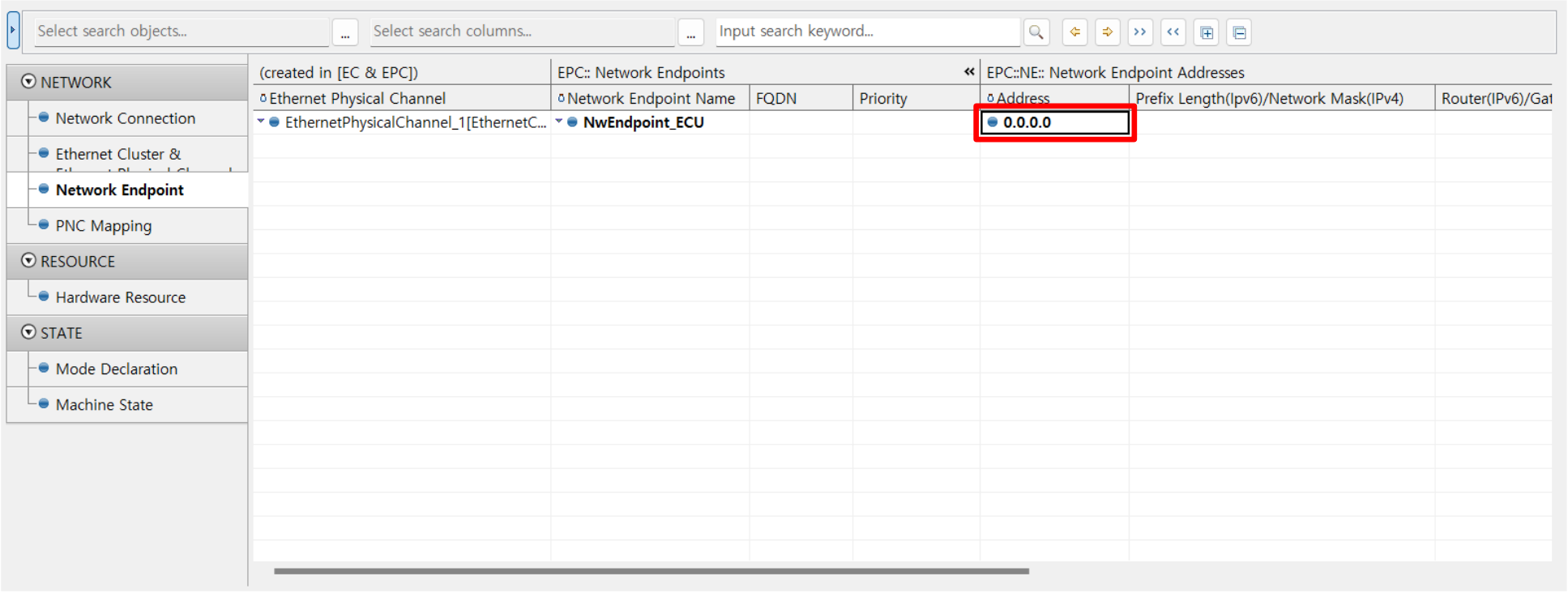
Machine Editor



EXEC: Machines 설정

- **IPv4 Address 추가 확인**
  - ✓ 생성된 새로운 IPv4 Address를 확인함

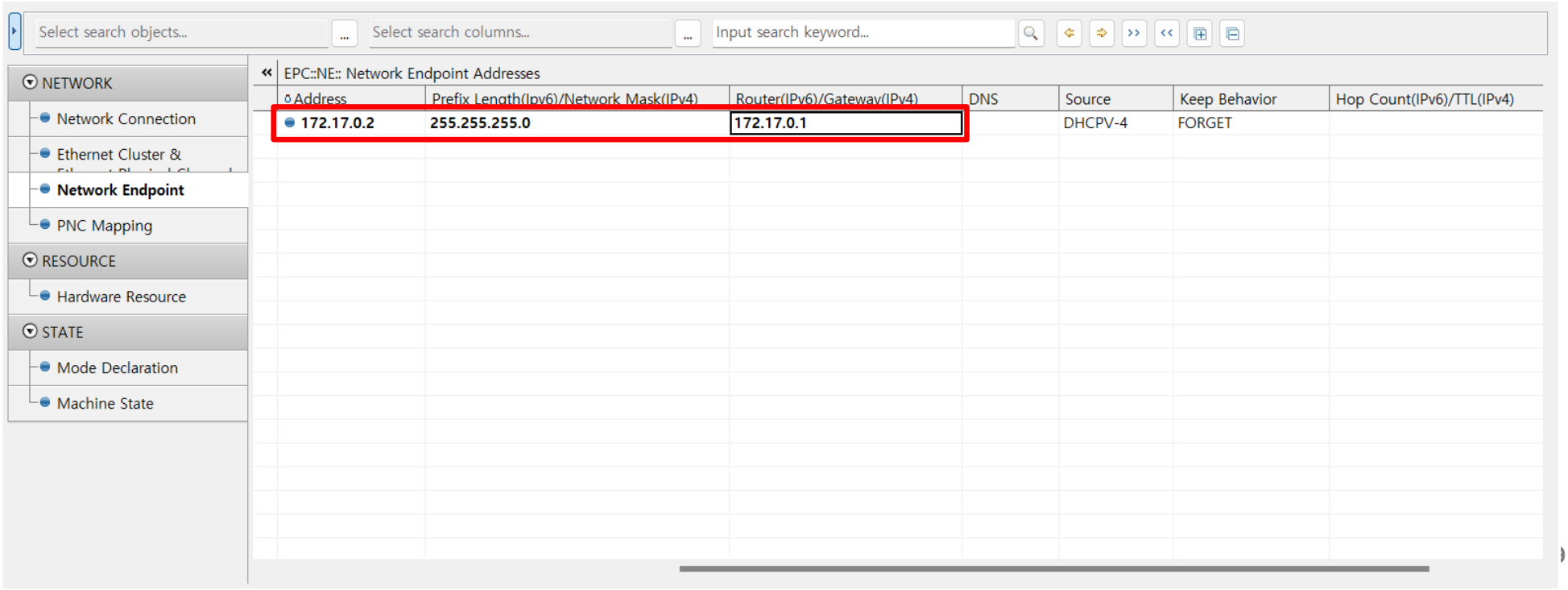
Machine Editor



EXEC: Machines 설정

- **IPv4 Address 설정**
  - ✓ 생성한 IPv4 Address를 다음과 같이 수정함
    - ✓ Address : '172.17.0.2'
    - ✓ Network Mask (IPv4) : '255.255.255.0'
    - ✓ Gateway (IPv4) : '172.17.0.1'

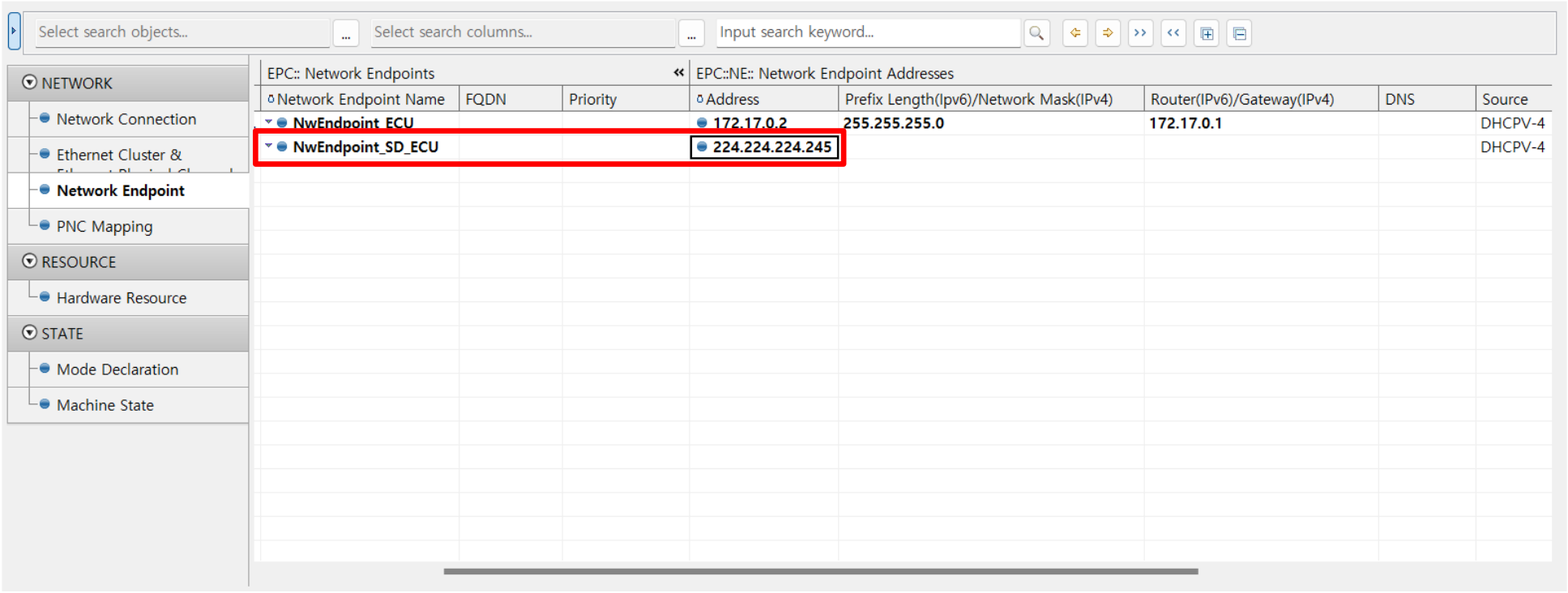
Machine Editor



EXEC: Machines 설정

- **IPv4 Address 추가 설정**
  - ✓ 동일한 과정을 통해 Service Discovery (SD)를 위한 Network Endpoint 및 IPv4를 추가하고 설정함

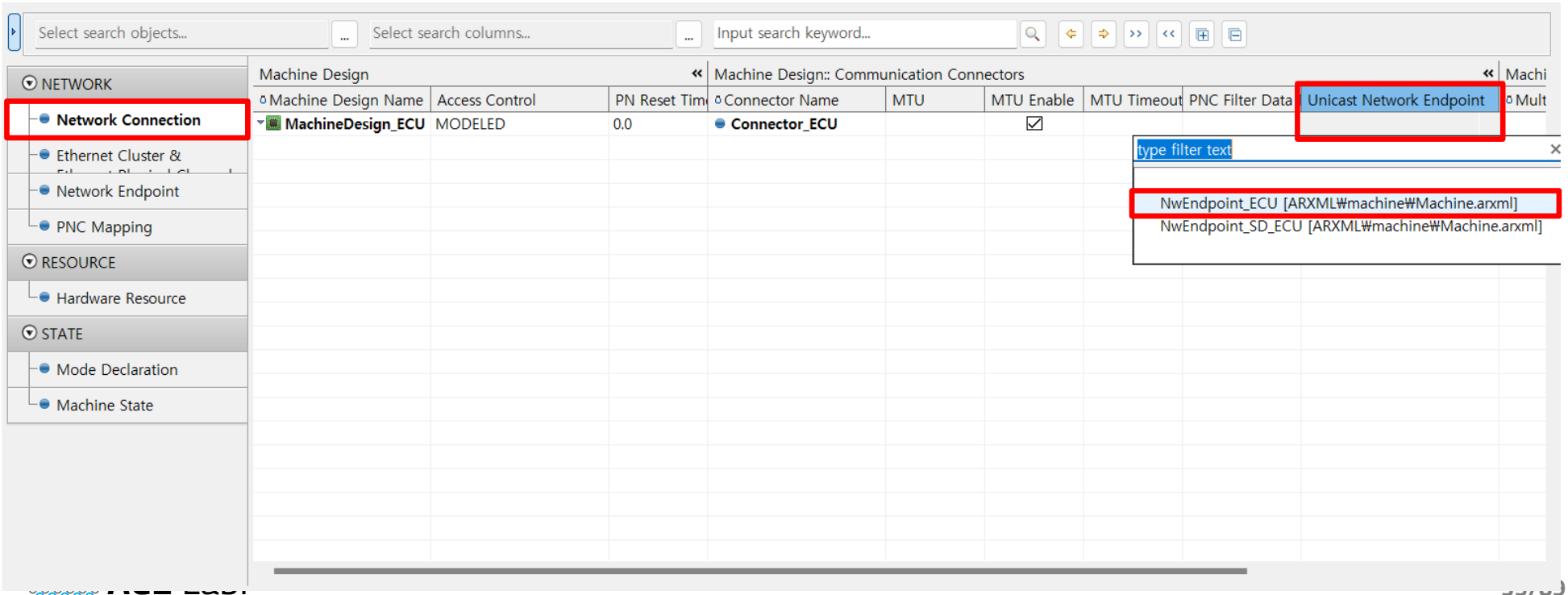
Machine Editor



EXEC: Machines 설정

- **Communication Connector – Network Endpoint 연결**
  - ✓ Network Endpoint를 Communication Connector에 연결하기 위해 좌측의 'Network Connection' 탭으로 이동함
  - ✓ Unicast Network Endpoint를 더블 클릭하여 'NwEndpoint_ECU'를 클릭함

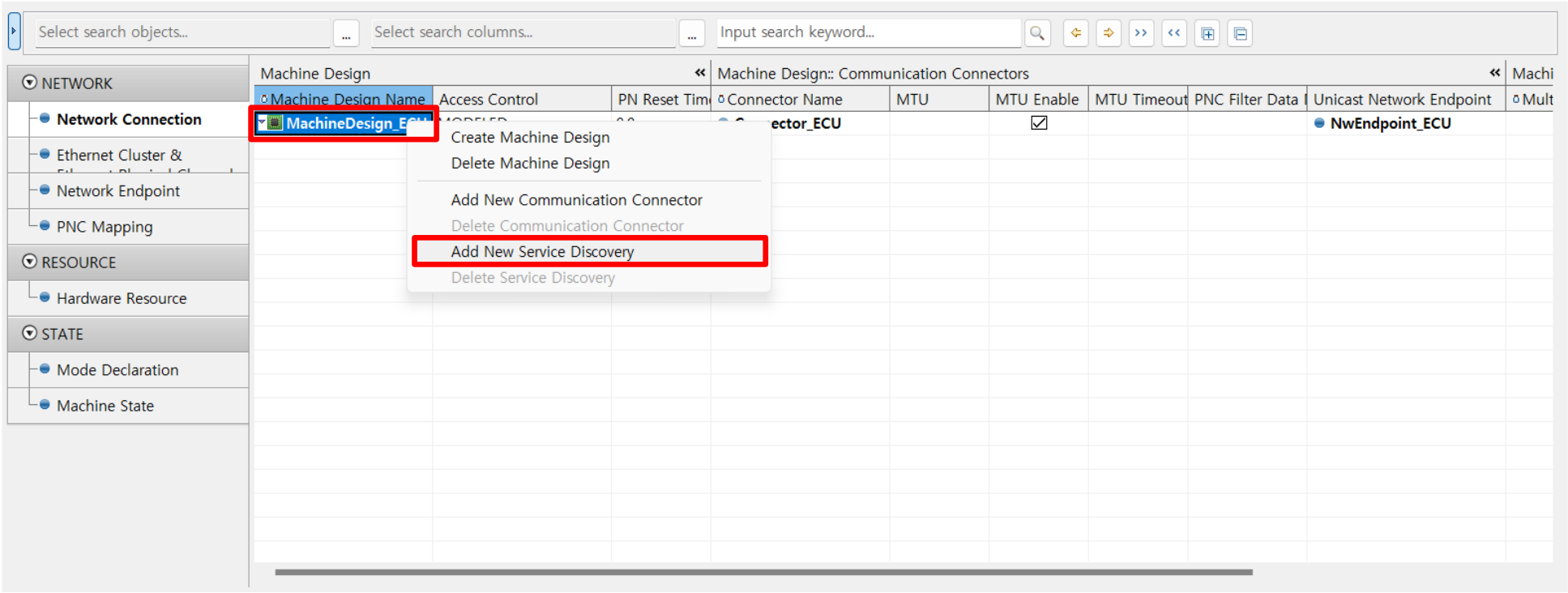
Machine Editor



EXEC: Machines 설정

- **Service Discovery 추가**
  - ✓ Machine Design에서 우클릭하여 'Add New Service Discovery'를 클릭함

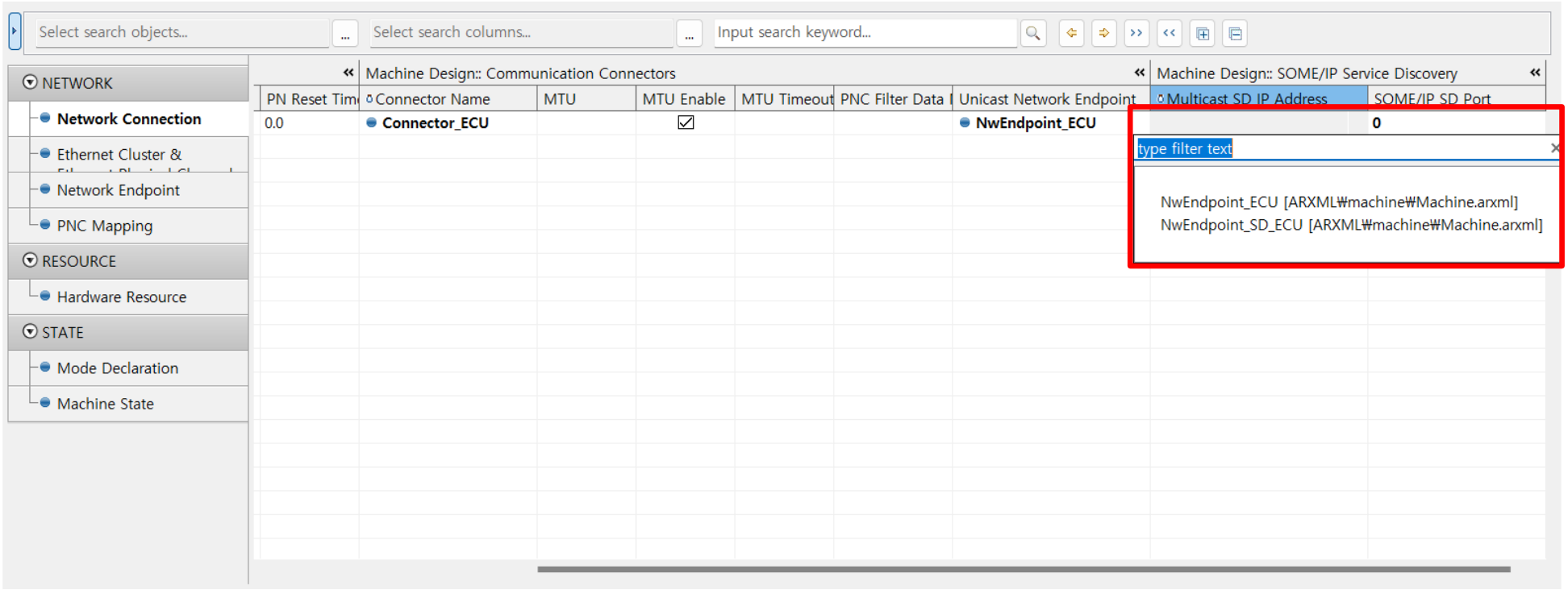
Machine Editor



EXEC: Machines 설정

- **Service Discovery 추가 확인**
  - ✓ 생성된 새로운 Service Discovery를 확인함

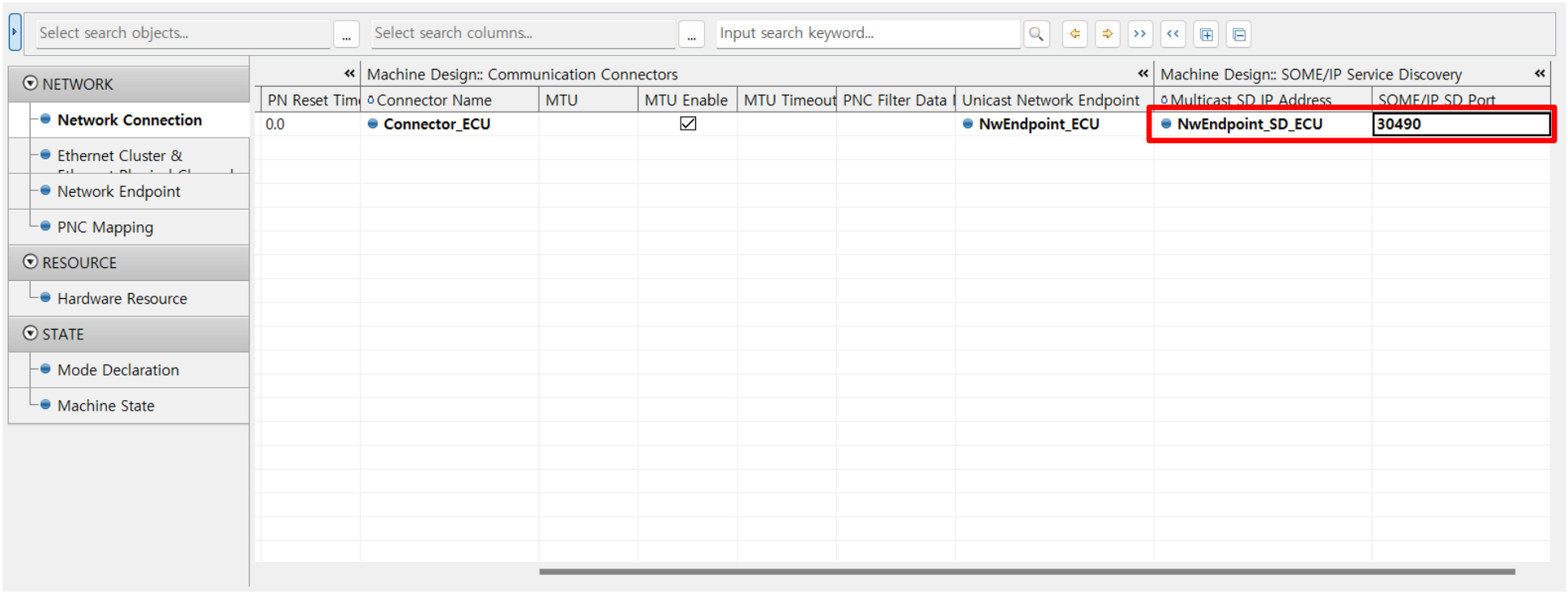
Machine Editor



EXEC: Machines 설정

- **Service Discovery 설정**
  - ✓ 생성한 Service Discovery를 다음과 같이 수정함
    - ✓ Multicast SD IP Address : 'NwEndpoint_SD_ECU'
    - ✓ SOME/IP SD Port : '30490'

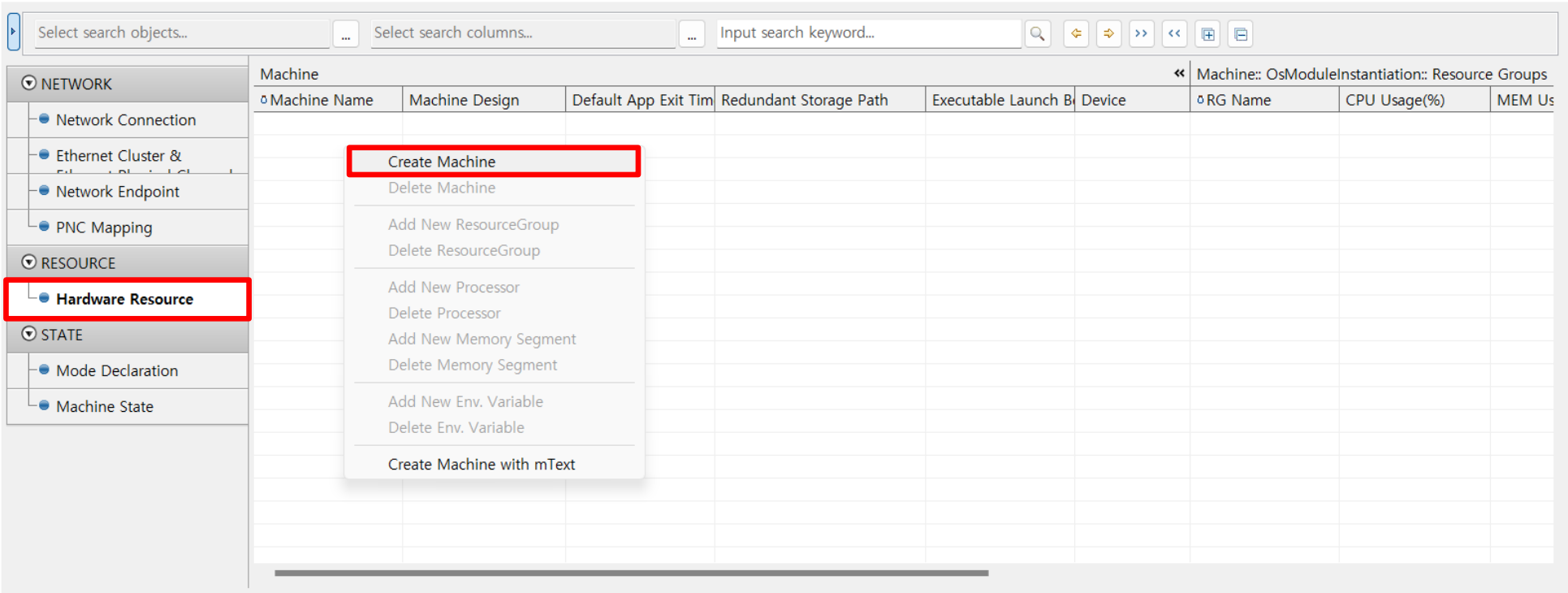
Machine Editor



EXEC: Machines 설정

- **Machine 추가**
  - ✓ Machine Design으로부터 Machine을 추가하기 위해 좌측의 'Hardware Resource' 탭으로 이동함
  - ✓ 빈 곳에서 우클릭을 하여 'Create Machine'을 클릭함

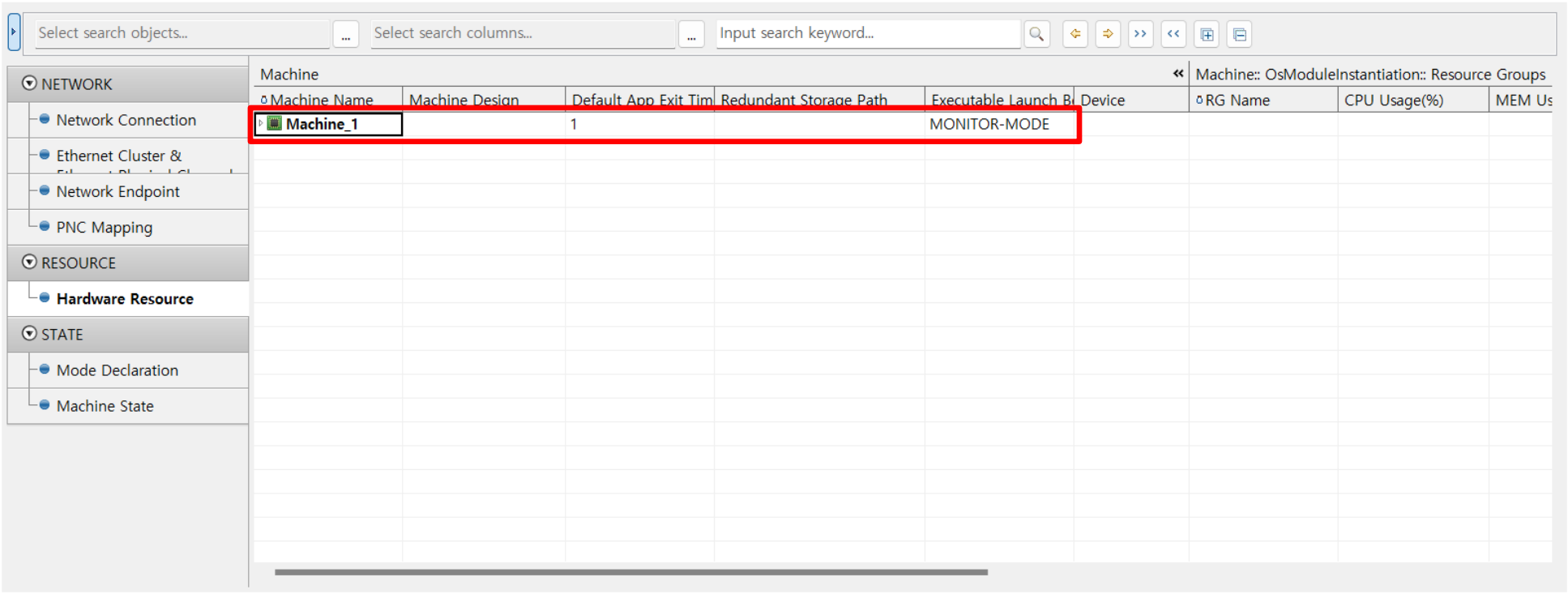
Machine Editor



EXEC: Machines 설정

- **Machine 추가 확인**
  - ✓ 생성된 새로운 Machine을 확인함

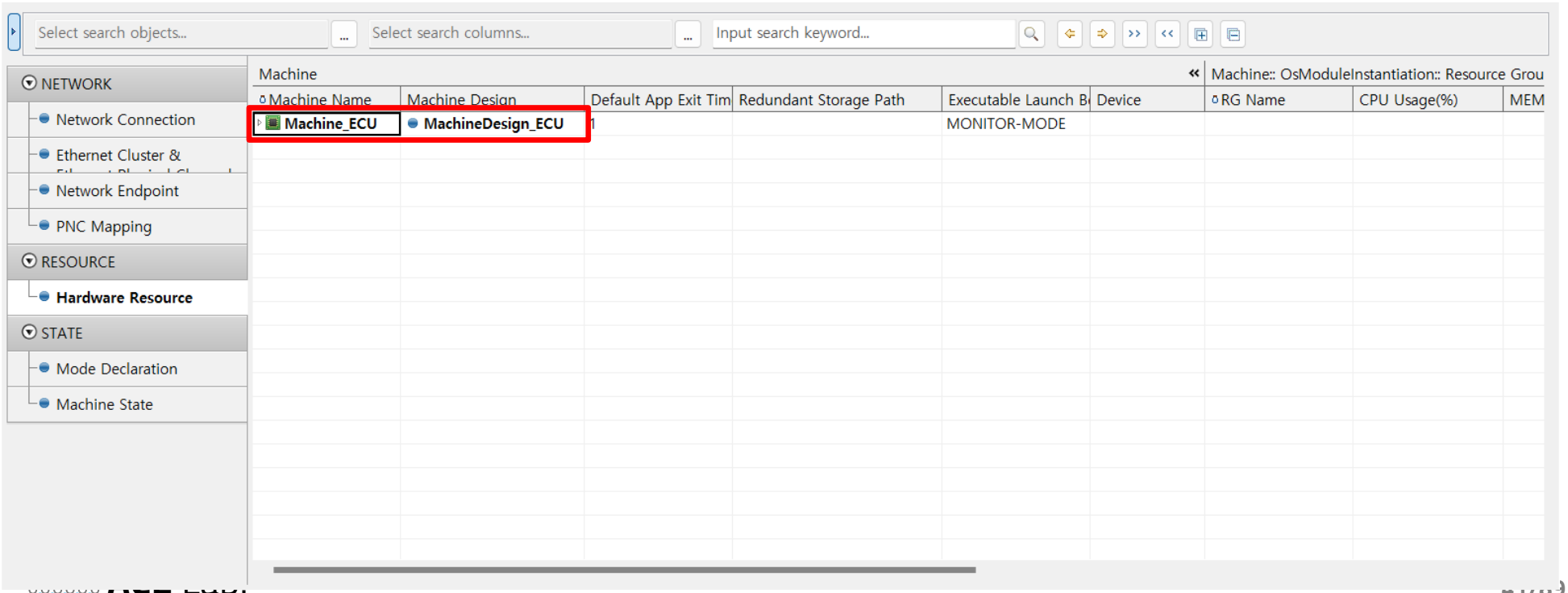
Machine Editor



EXEC: Machines 설정

- **Machine 설정**
  - ✓ 생성한 Machine을 다음과 같이 수정함
    - ✓ Machine Name : 'Machine_ECU'
    - ✓ Machine Design : 'MachineDesign_ECU'

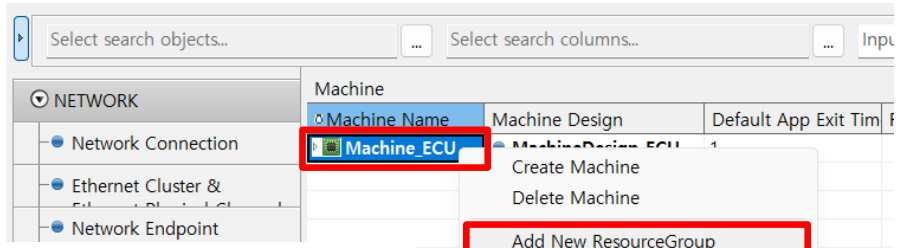
Machine Editor



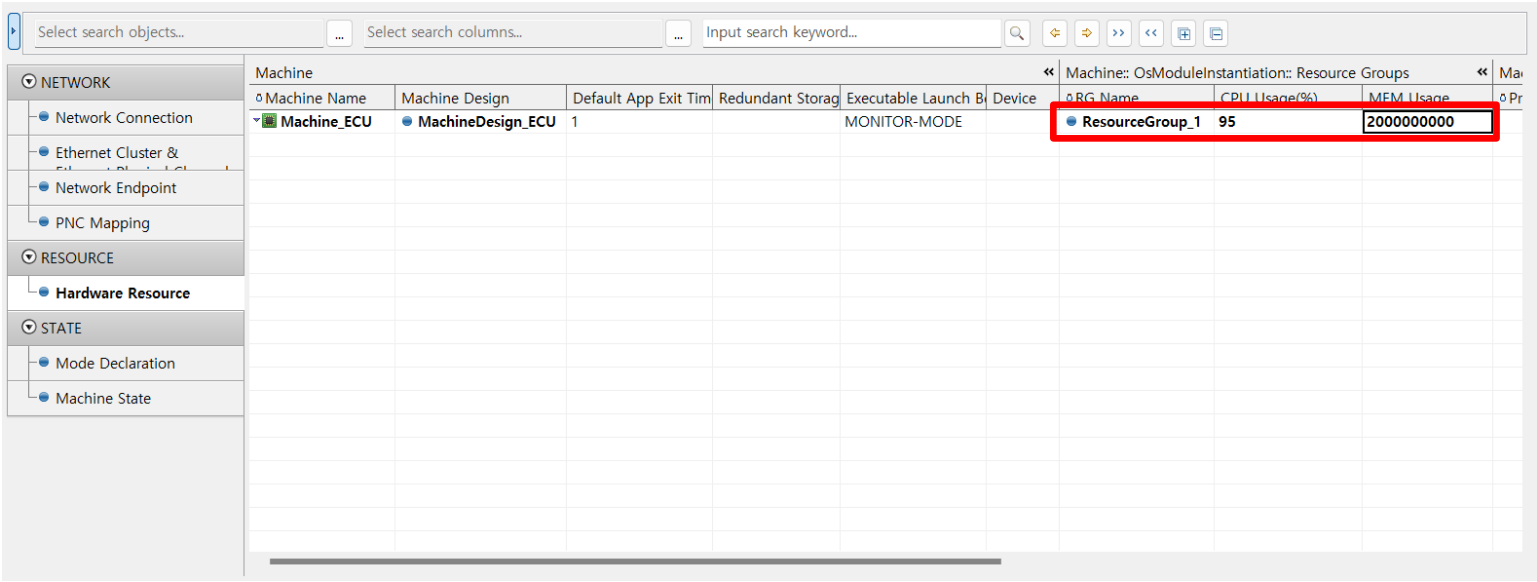
EXEC: Machines 설정

- **Machine 추가 설정 – Resource Groups**
  - ✓ Machine에서 우클릭하여 'Add New ResourceGroup'을 클릭함
  - ✓ 생성된 새로운 Resource Group을 확인하고 다음과 같이 수정함
    - ✓ CPU Usage (%) : '95'
    - ✓ MEM Usage : '2000000000'

Machine Editor



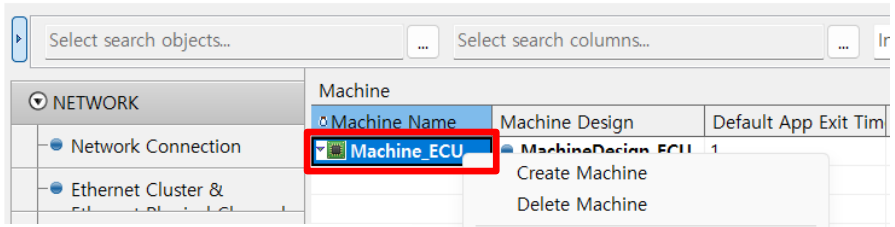
Machine Editor



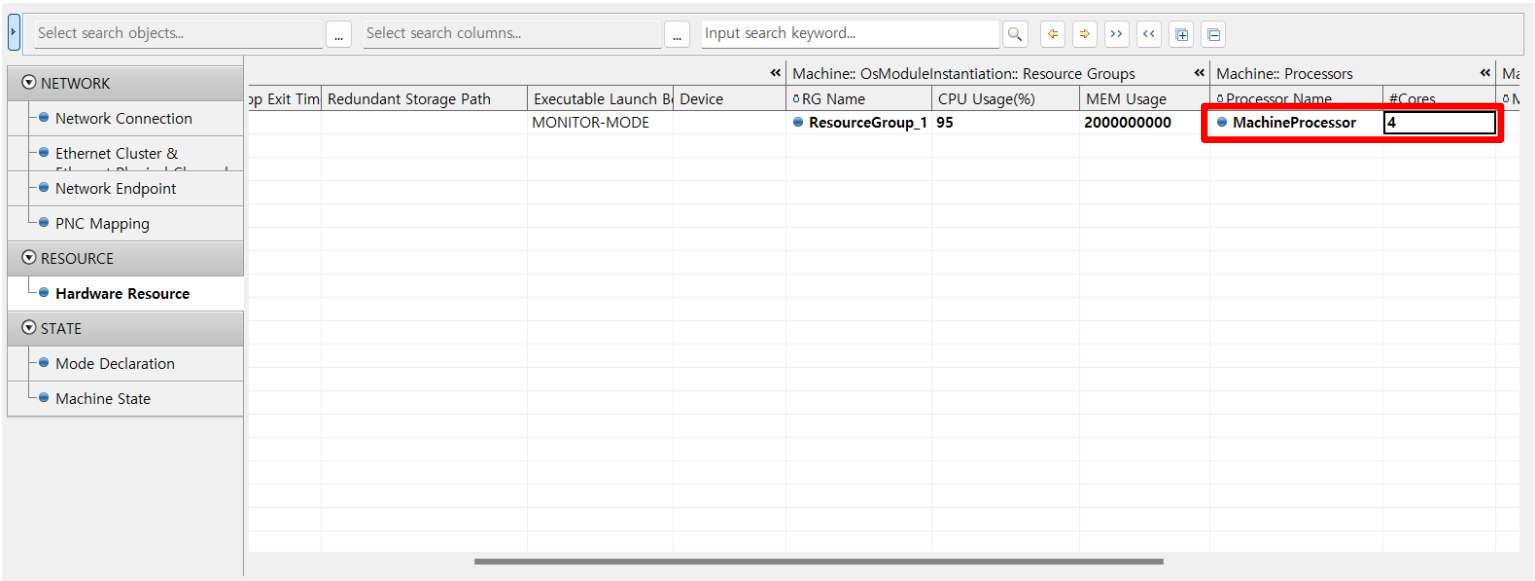
EXEC: Machines 설정

- **Machine 추가 설정 – Processors**
  - ✓ Machine에서 우클릭하여 'Add New Processor'를 클릭함
  - ✓ 생성된 새로운 Processor를 확인하고 다음과 같이 수정함
    - ✓ Processor Name : 'MachineProcessor'
    - ✓ #Cores : '4'

Machine Editor



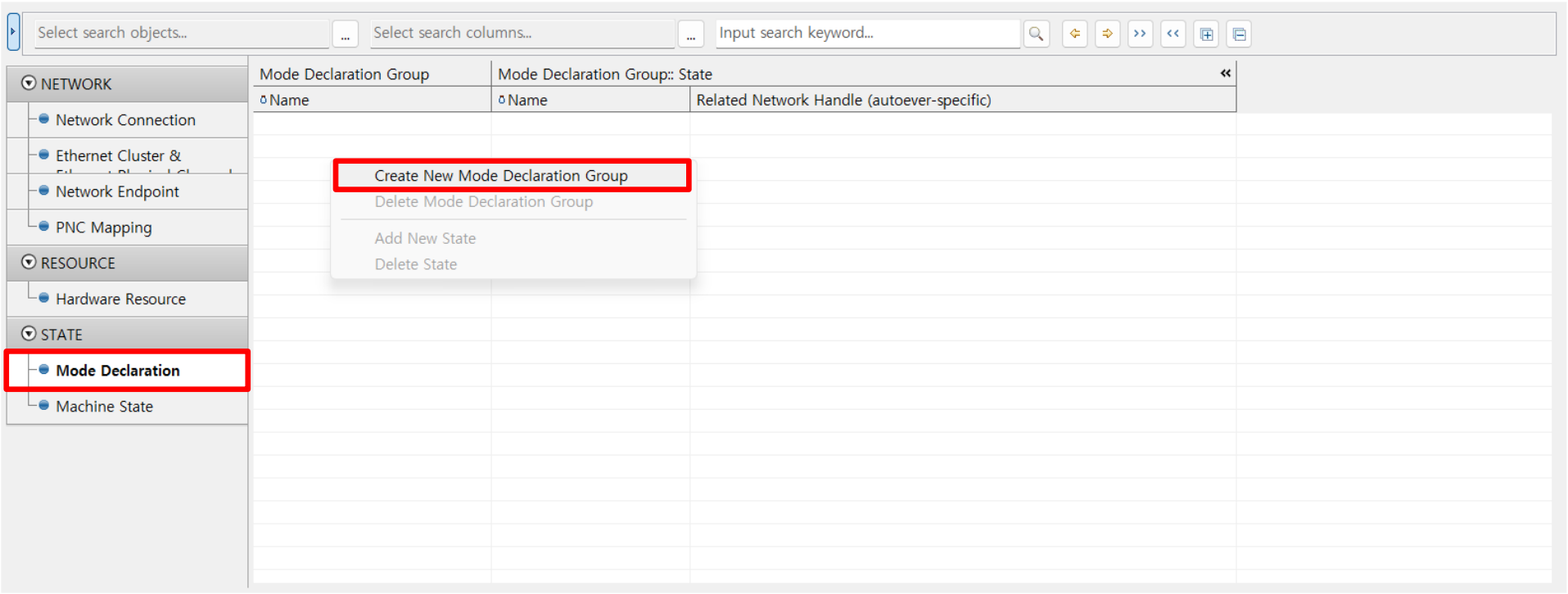
Machine Editor



EXEC: Machines 설정

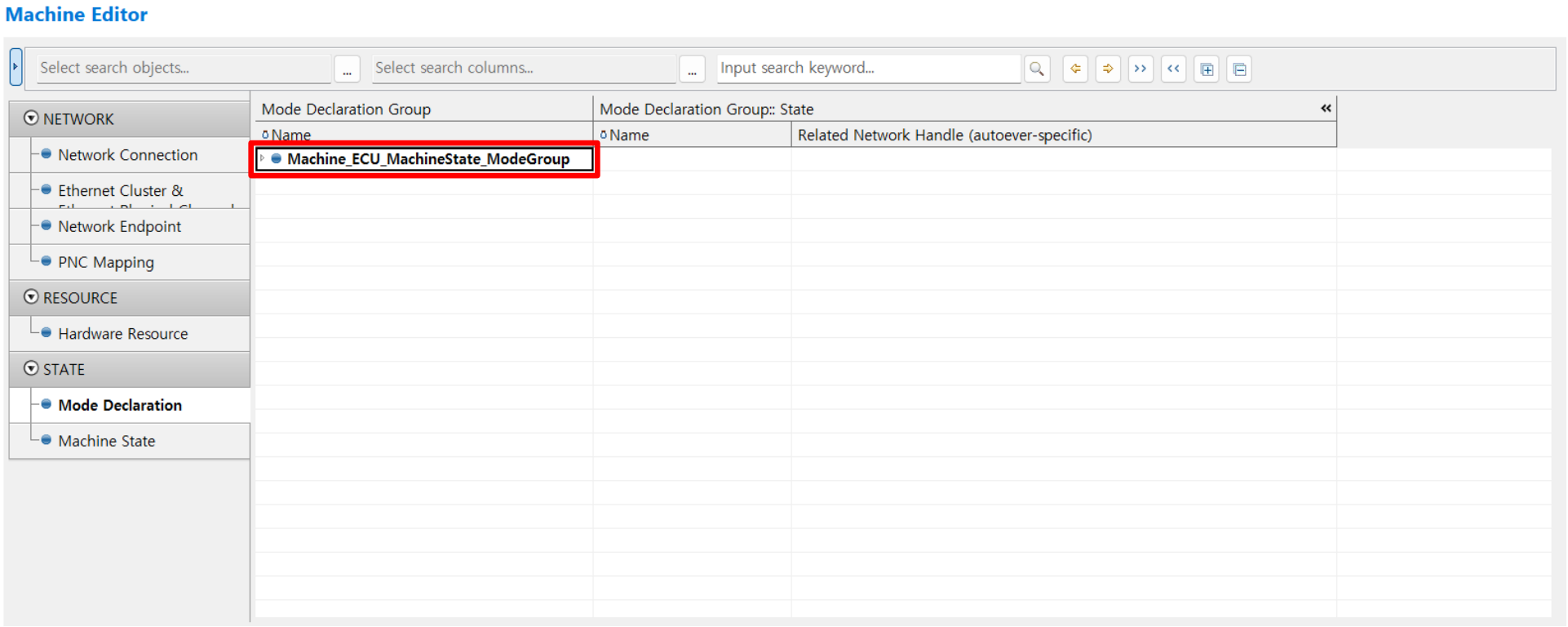
- **Mode Declaration Group 추가**
  - ✓ Mode Declaration Group 추가를 위해 좌측의 'Mode Declaration' 탭으로 이동함
  - ✓ 빈 곳에서 우클릭을 하여 'Create New Mode Declaration Group'을 클릭함

Machine Editor



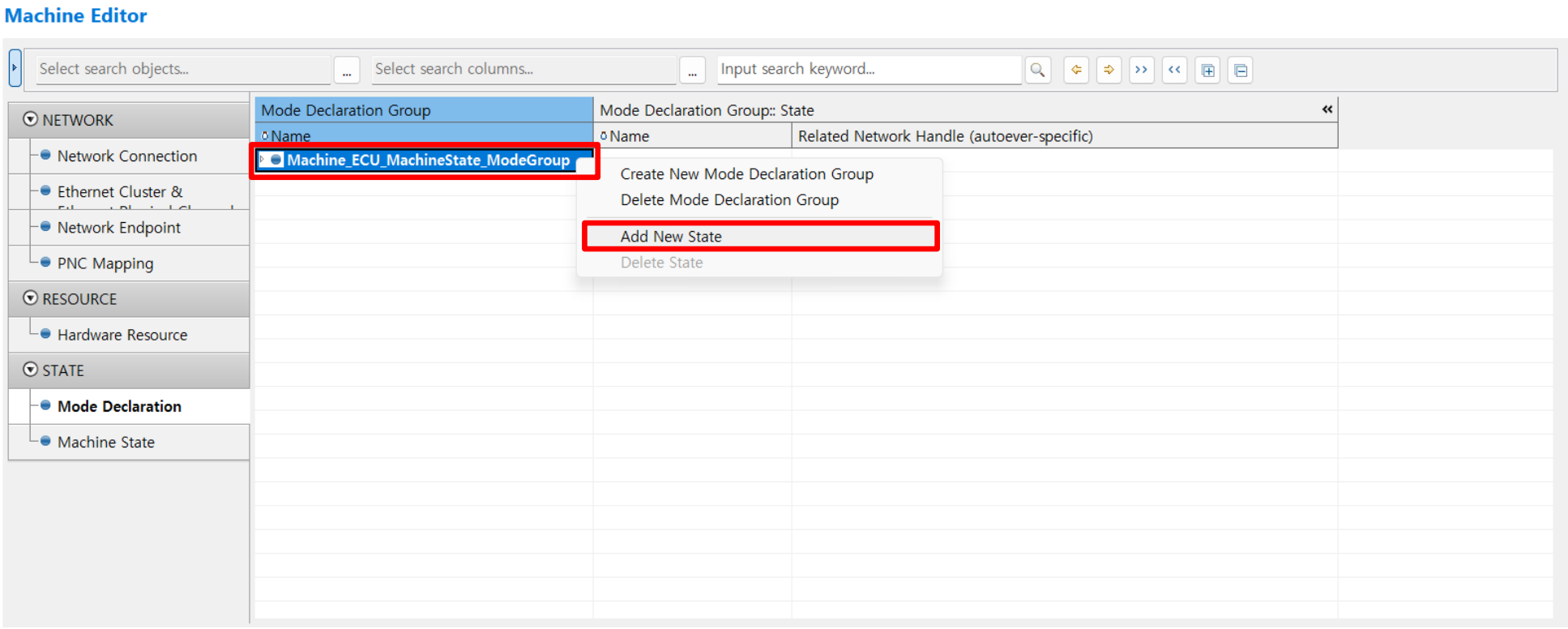
EXEC: Machines 설정

- **Mode Declaration Group 추가 확인 및 설정**
  - ✓ 생성된 새로운 Mode Declaration Group을 확인하고 다음과 같이 수정함
    - ✓ Name : 'Machine_ECU_MachineState_ModeGroup'



EXEC: Machines 설정

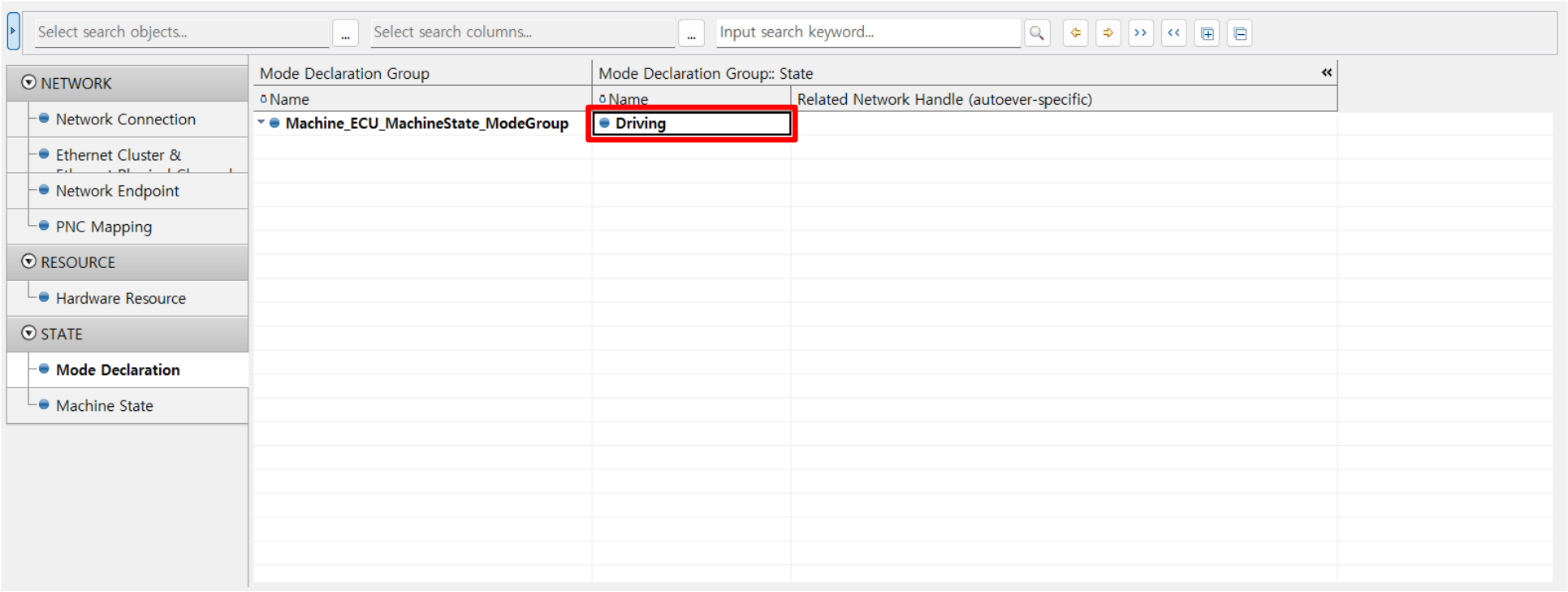
- **State 추가**
  - ✓ Mode Declaration Group에서 우클릭하여 'Add New State'를 클릭함



EXEC: Machines 설정

- **State 추가 확인 및 설정**
  - ✓ 생성된 새로운 State를 확인하고 다음과 같이 수정함
    - ✓ Name : 'Driving'

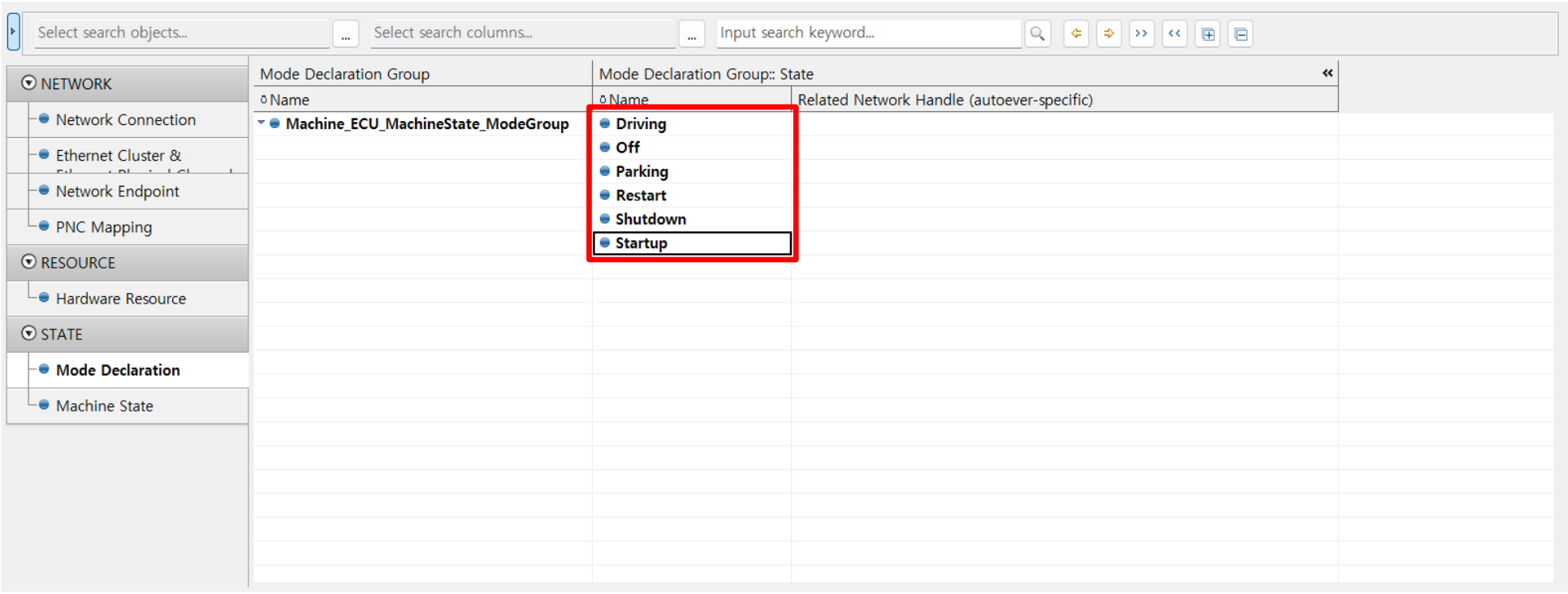
Machine Editor



EXEC: Machines 설정

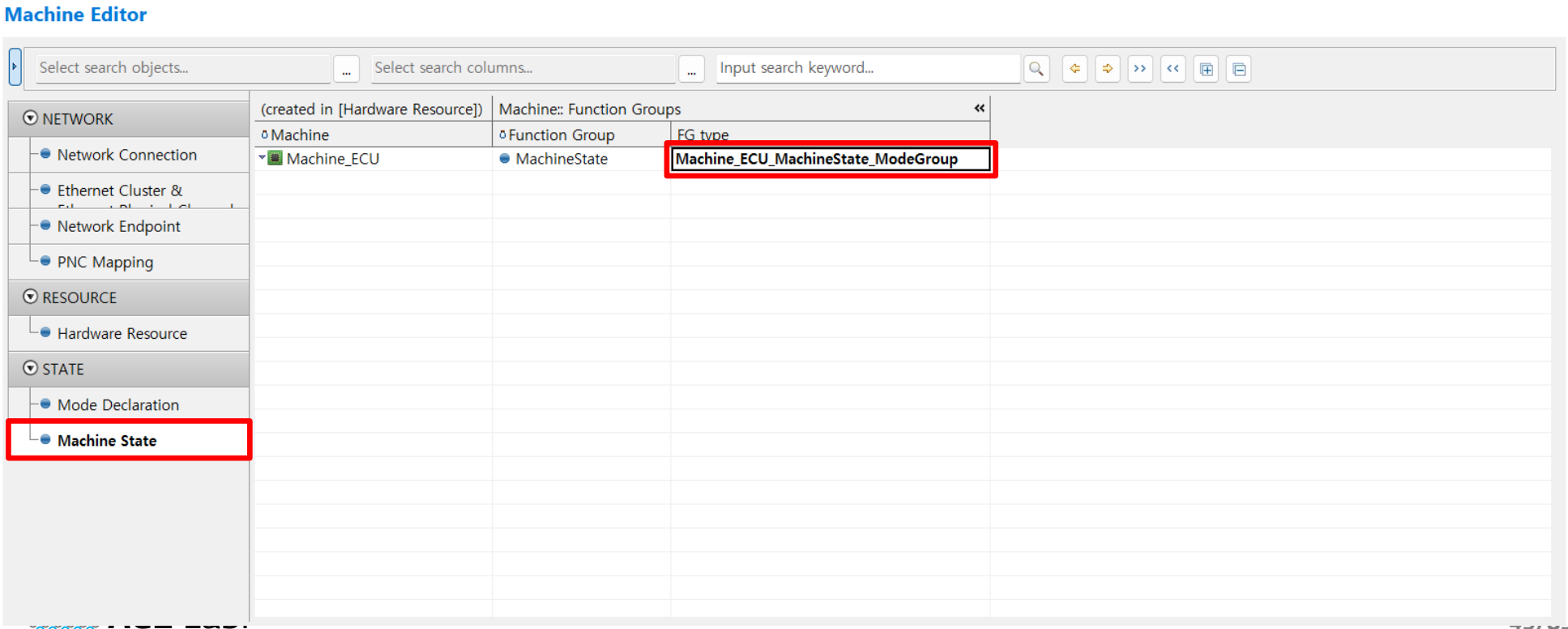
- **State 추가 설정**
  - ✓ 동일한 과정을 통해 State를 추가하고 설정함

Machine Editor



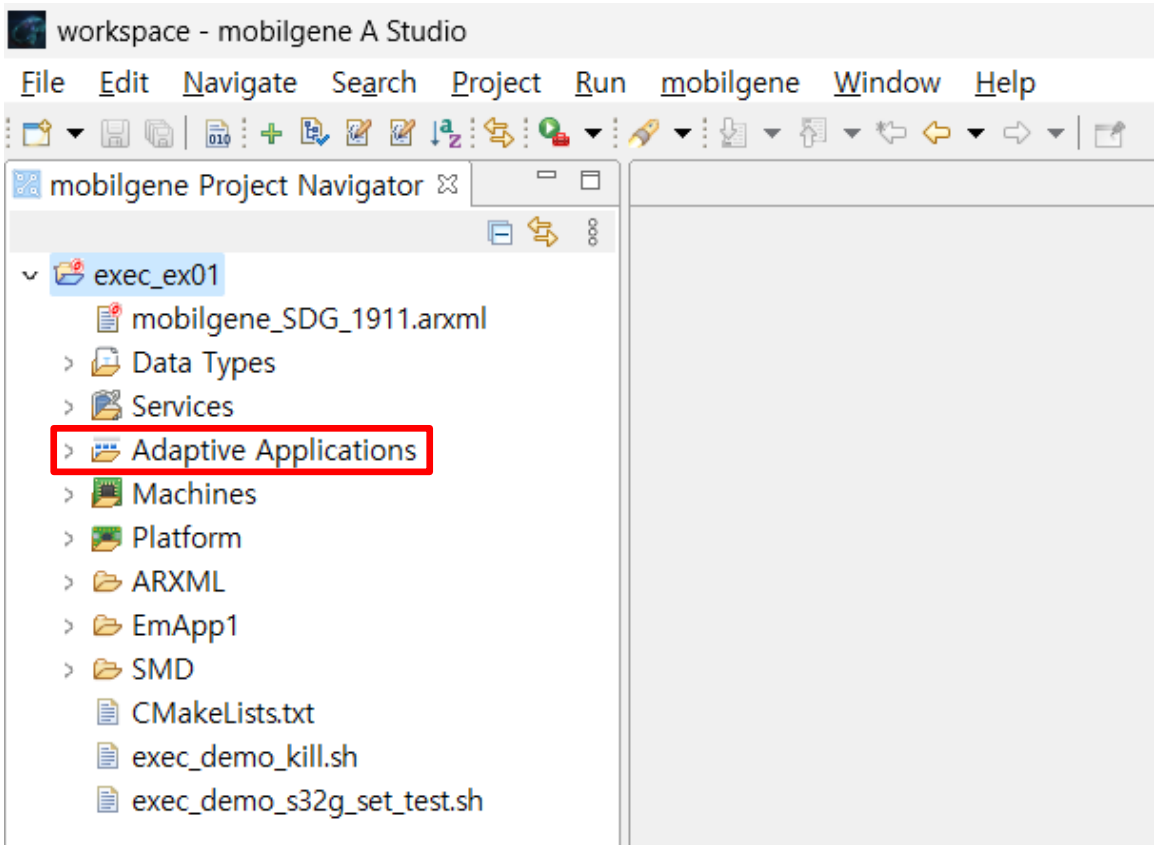
EXEC: Machines 설정

- **Machine State 설정**
  - ✓ Machine의 Machine State를 설정하기 위해 좌측의 'Machine State' 탭으로 이동함
  - ✓ 생성한 Machine의 Machine State에 대응하는 FG type을 더블 클릭하여 다음과 같이 수정함
    - ✓ FG type : 'Machine_ECU_MachineState_ModeGroup'



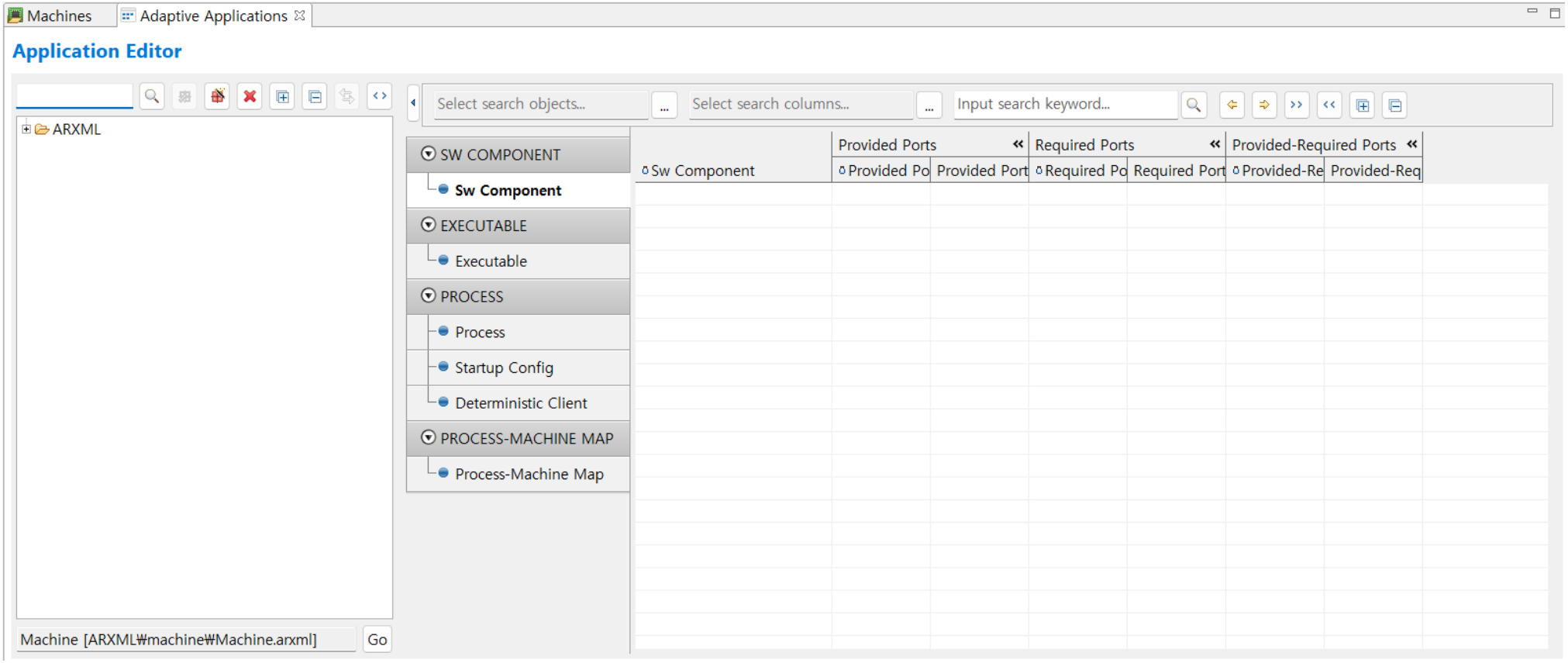
EXEC: Adaptive Applications 설정

- **Adaptive Applications Editor 활성화**
  - ✓ 왼쪽의 'mobilgene Project Navigator' 창에서 해당 Project의 'Adaptive Applications'를 더블 클릭함



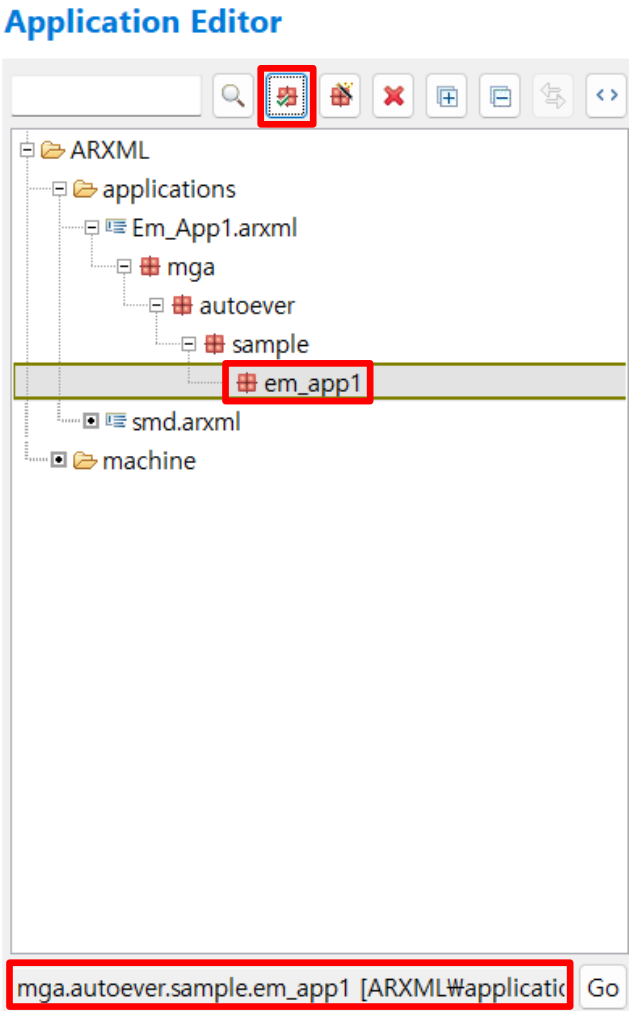
EXEC: Adaptive Applications 설정

- **Adaptive Applications Editor 활성화 확인**
  - ✓ 활성화 된 Adaptive Applications Editor를 확인함



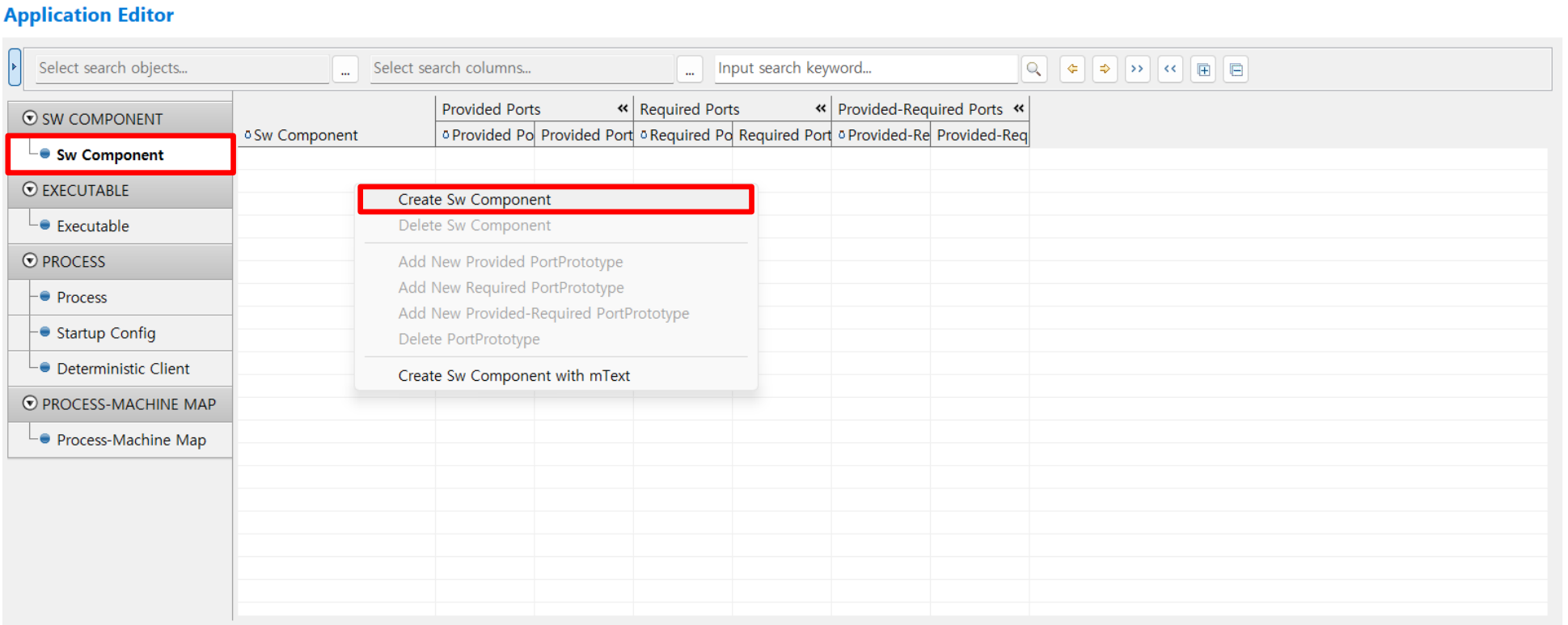
EXEC: Adaptive Applications 설정

- **Default Package 설정**
  - ✓ 설정이 저장될 Default Package를 설정함
    - ✓ Default Package로 설정할 'em_app1' Package를 선택함
    - ✓ 우측 상단의 'Set Default Package' 버튼을 클릭함
    - ✓ 하단에 설정된 Default Package를 확인함



EXEC: Adaptive Applications 설정

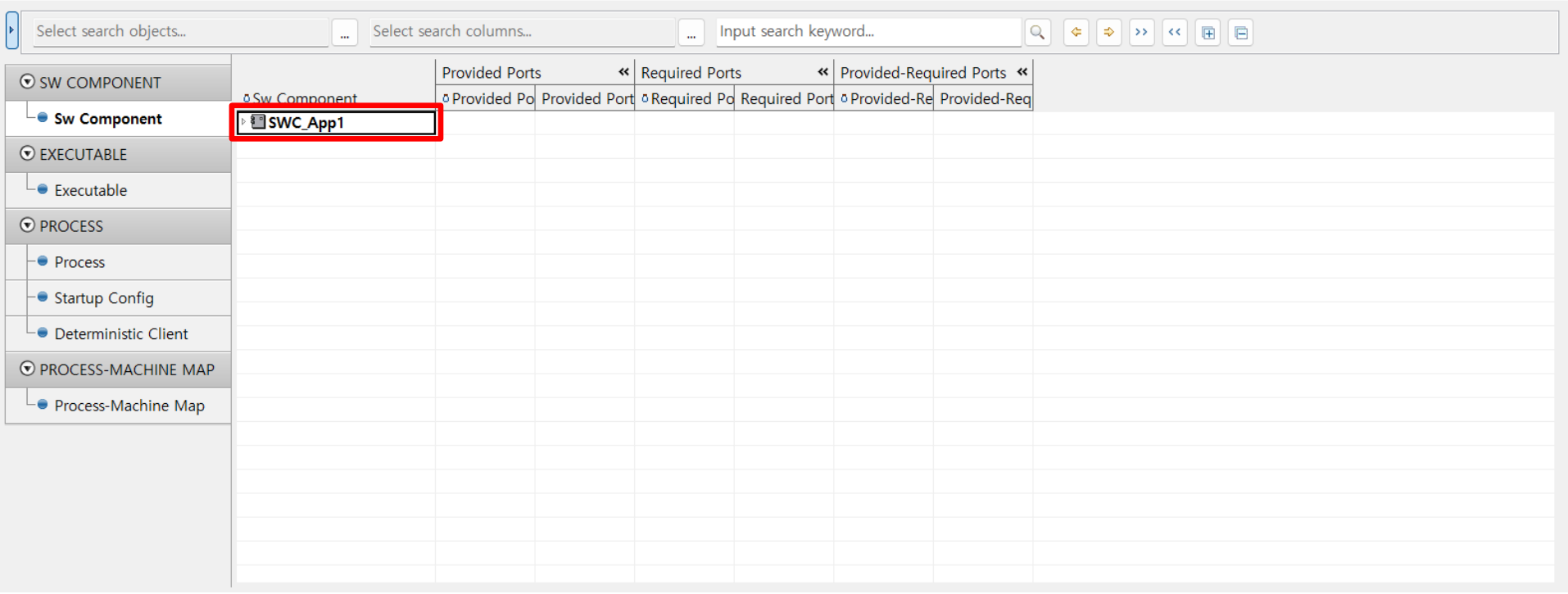
- **Sw Component 추가**
  - ✓ Sw Component 추가를 위해 좌측의 'Sw Component' 탭으로 이동함
  - ✓ 빈 곳에서 우클릭을 하여 'Create Sw Component'를 클릭함



EXEC: Adaptive Applications 설정

- **Sw Component 추가 확인 및 설정**
  - ✓ 생성된 새로운 Sw Component를 확인하고 다음과 같이 수정함
    - ✓ Sw Component : 'SWC_App1'

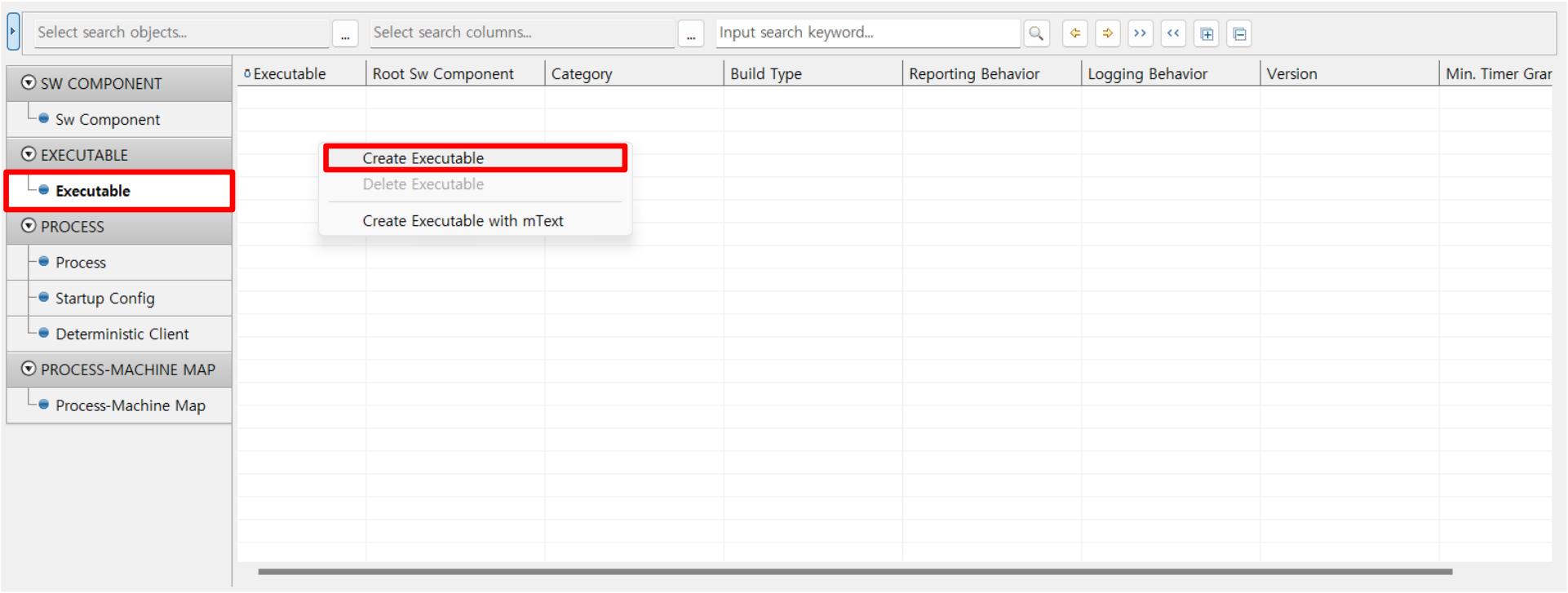
Application Editor



EXEC: Adaptive Applications 설정

- **Executable 추가**
  - ✓ Executable 추가를 위해 좌측의 'Executable' 탭으로 이동함
  - ✓ 빈 곳에서 우클릭을 하여 'Create Executable'을 클릭함

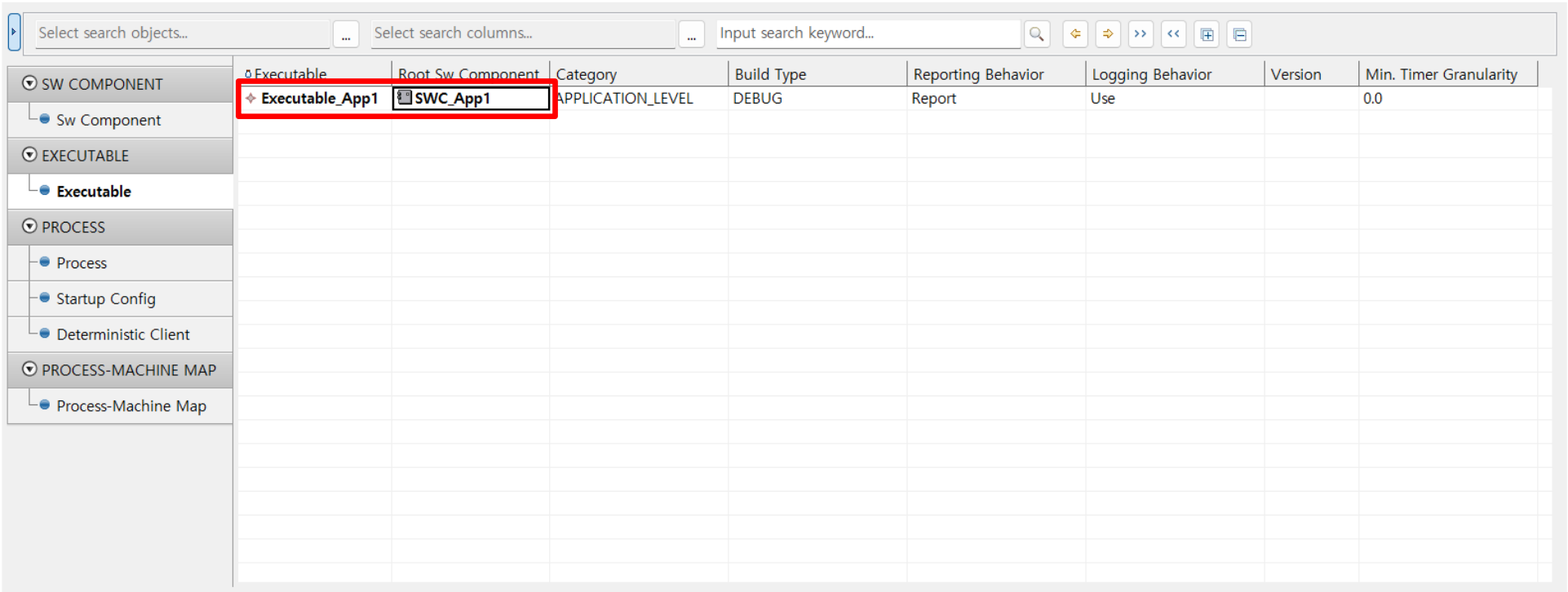
Application Editor



EXEC: Adaptive Applications 설정

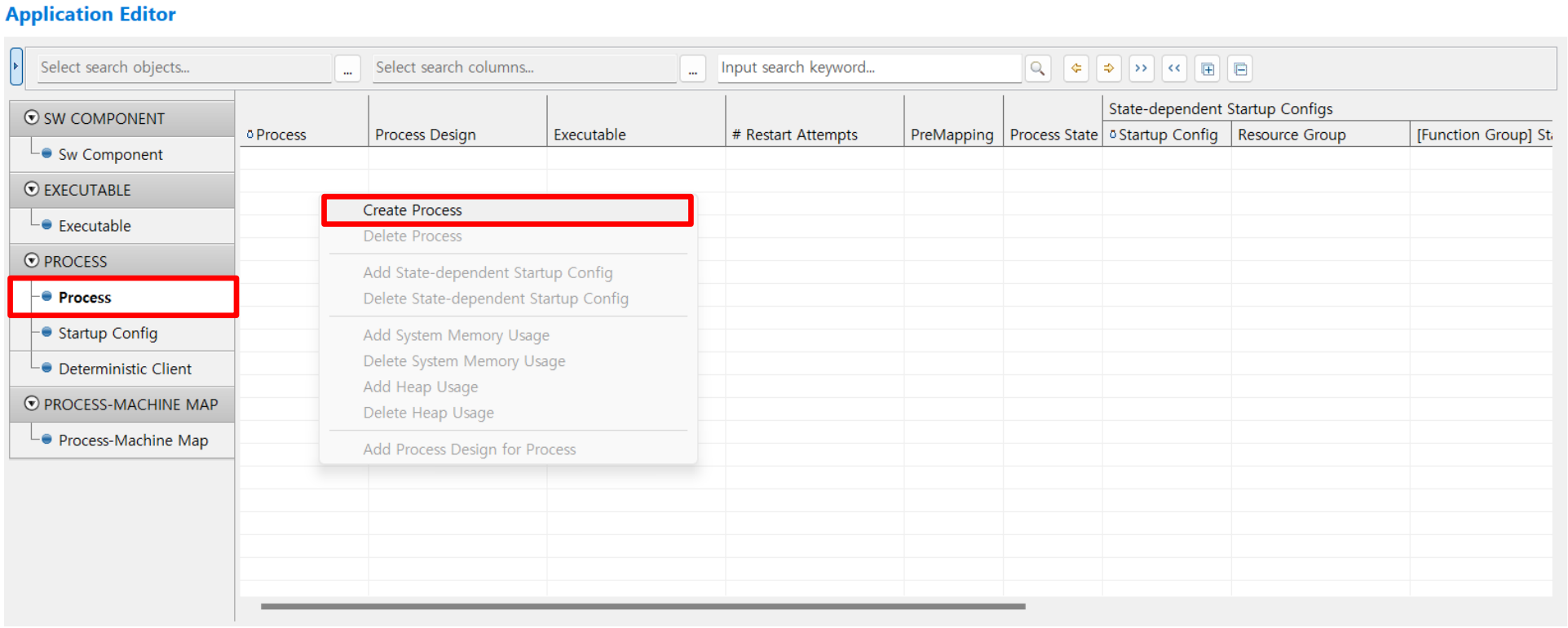
- **Executable 추가 확인 및 설정**
  - ✓ 생성된 새로운 Executable을 확인하고 다음과 같이 수정함
    - ✓ Executable : 'Executable_App1'
    - ✓ Root Sw Component : 'SWC_App1'

Application Editor



EXEC: Adaptive Applications 설정

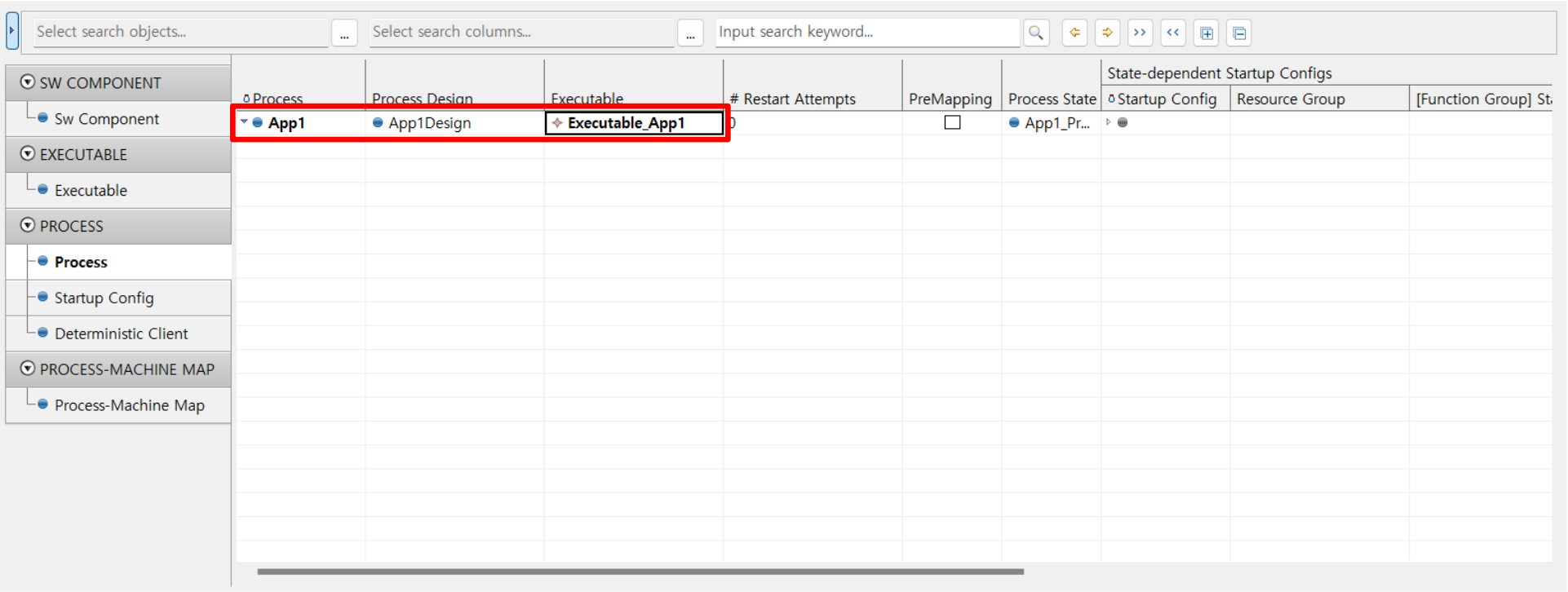
- **Process 추가**
  - ✓ Process 추가를 위해 좌측의 'Process' 탭으로 이동함
  - ✓ 빈 곳에서 우클릭을 하여 'Create Process'를 클릭함



EXEC: Adaptive Applications 설정

- **Process 추가 확인 및 설정**
  - ✓ 생성된 새로운 Process를 확인하고 다음과 같이 수정함
    - ✓ Process : 'App1'
    - ✓ Executable : 'Executable_App1'

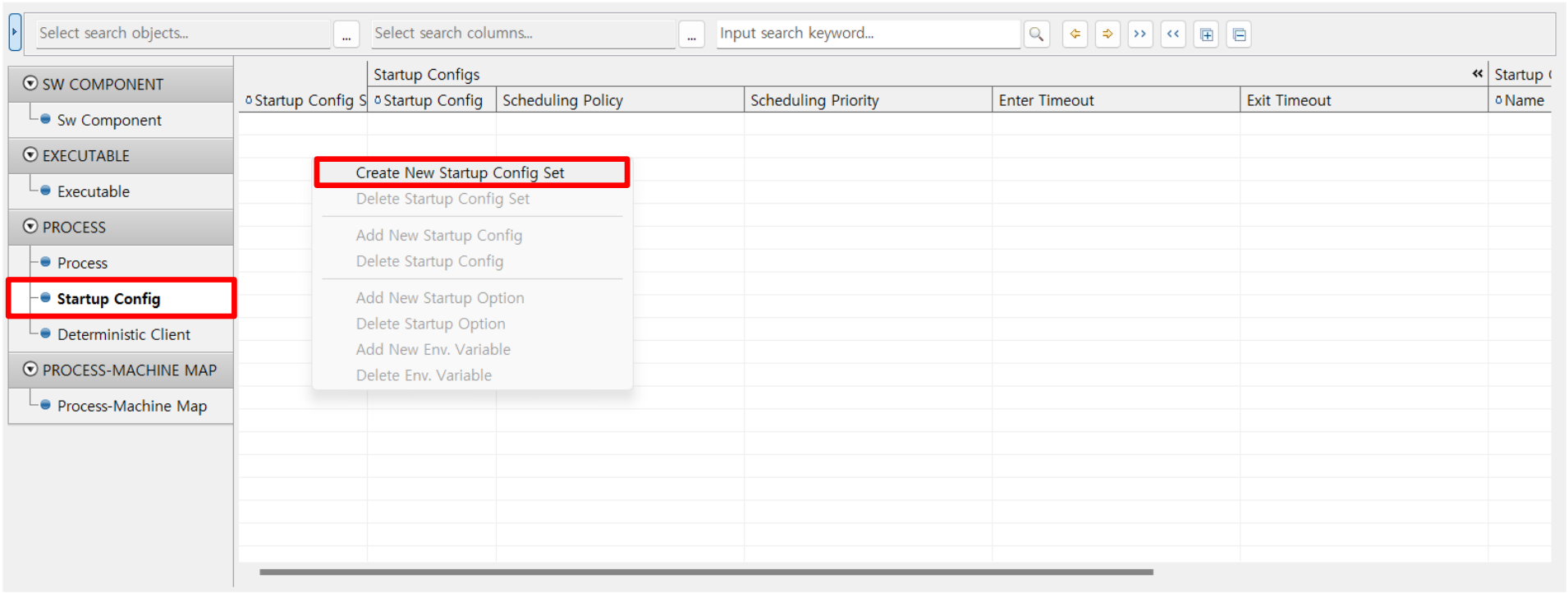
Application Editor



EXEC: Adaptive Applications 설정

- **Startup Config Set 추가**
  - ✓ Startup Config Set 추가를 위해 좌측의 'Startup Config' 탭으로 이동함
  - ✓ 빈 곳에서 우클릭을 하여 'Create New Startup Config Set'을 클릭함

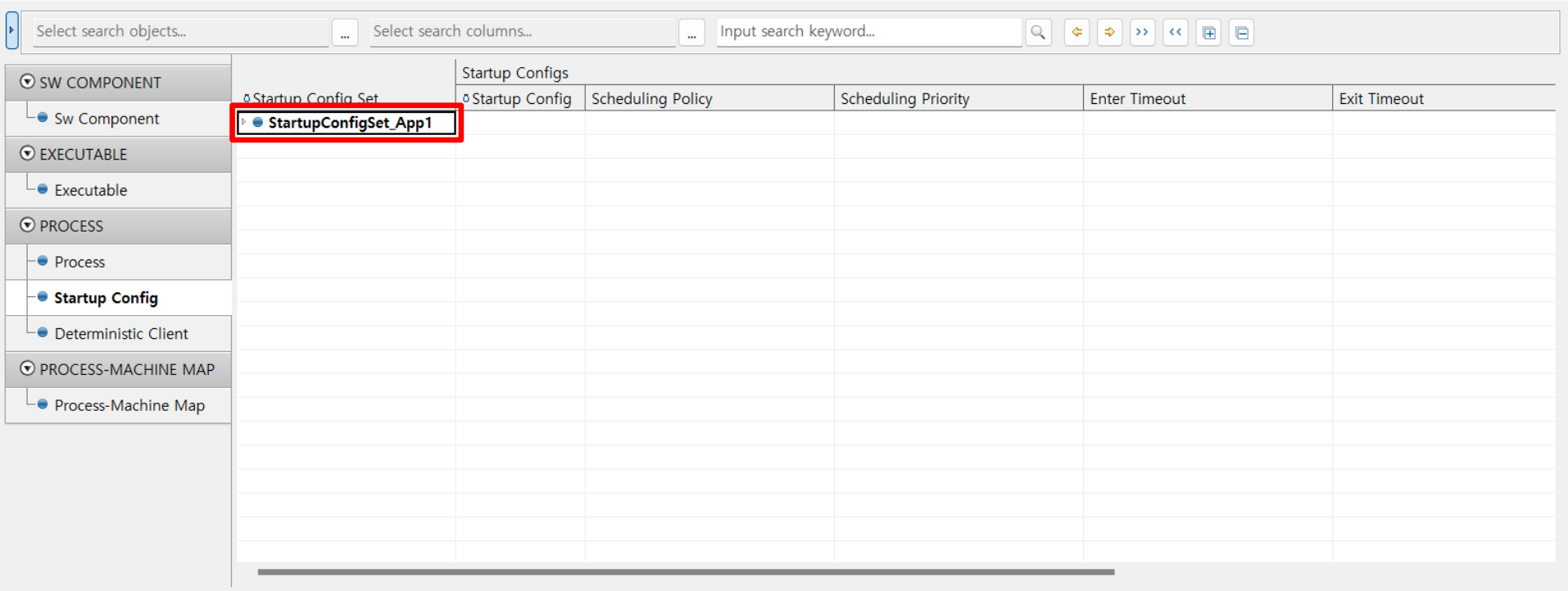
Application Editor



EXEC: Adaptive Applications 설정

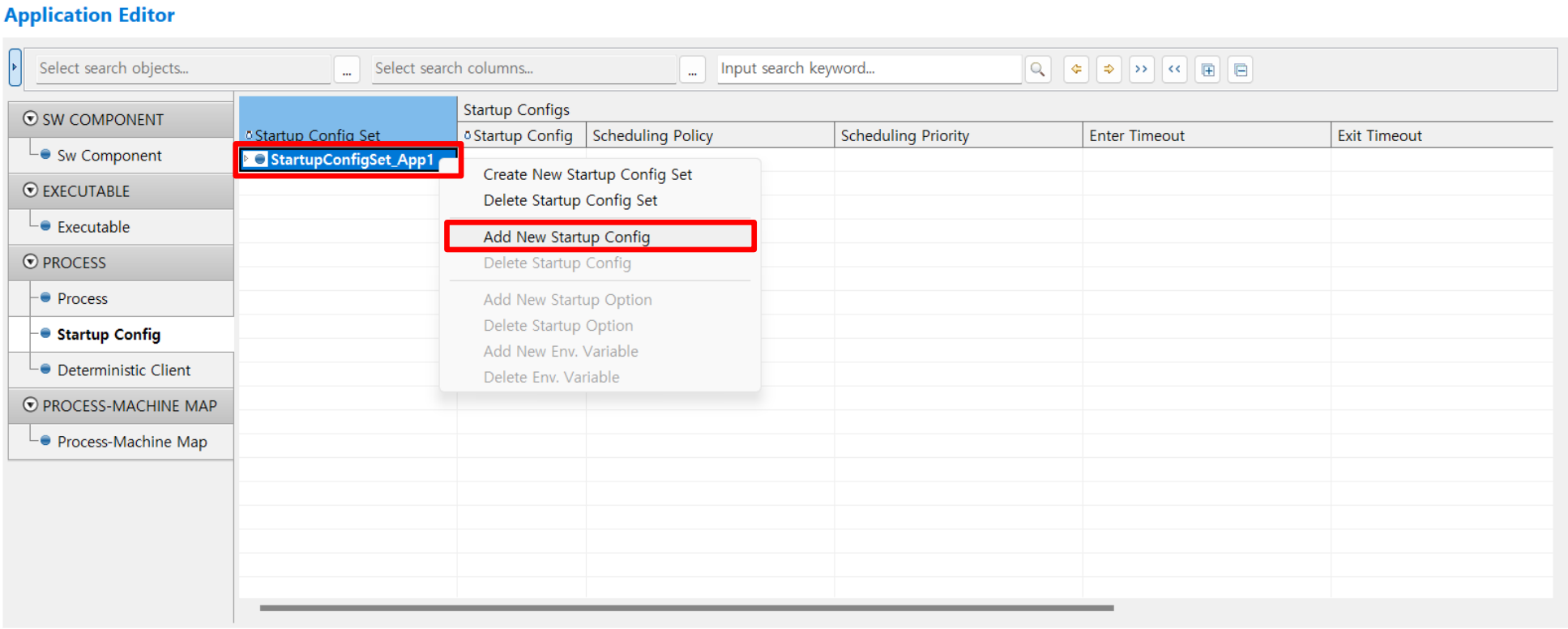
- **Startup Config Set 추가 확인 및 설정**
  - ✓ 생성된 새로운 Startup Config Set을 확인하고 다음과 같이 수정함
    - ✓ Startup Config Set : 'StartupConfigSet_App1'

Application Editor



EXEC: Adaptive Applications 설정

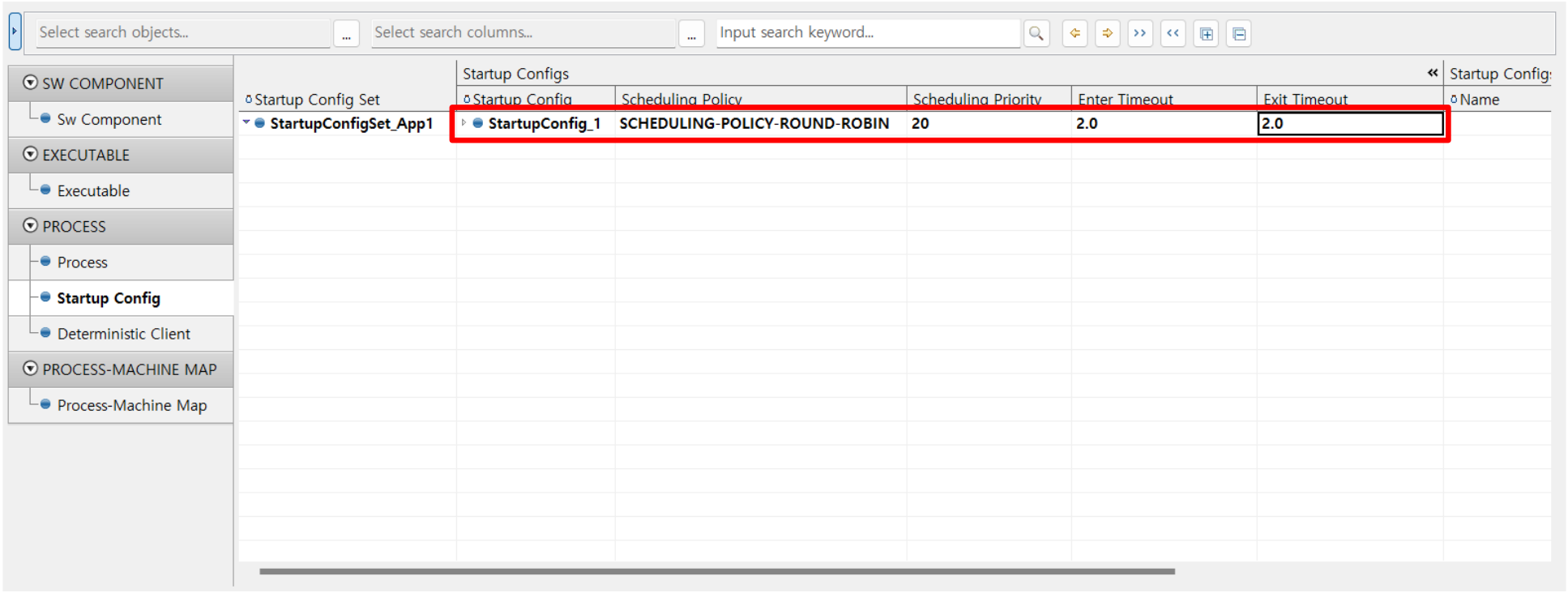
- **Startup Config 추가**
  - ✓ 생성한 Startup Config Set에서 우클릭하여 'Add New Startup Config'를 클릭함



EXEC: Adaptive Applications 설정

- **Startup Config 추가 확인 및 설정**
  - ✓ 생성된 새로운 Startup Config를 확인하고 다음과 같이 수정함
    - ✓ Scheduling Policy : 'SCHEDULING-POLICY-ROUND-ROBIN'
    - ✓ Scheduling Priority : '20'
    - ✓ Enter/Exit Timeout : '2.0'

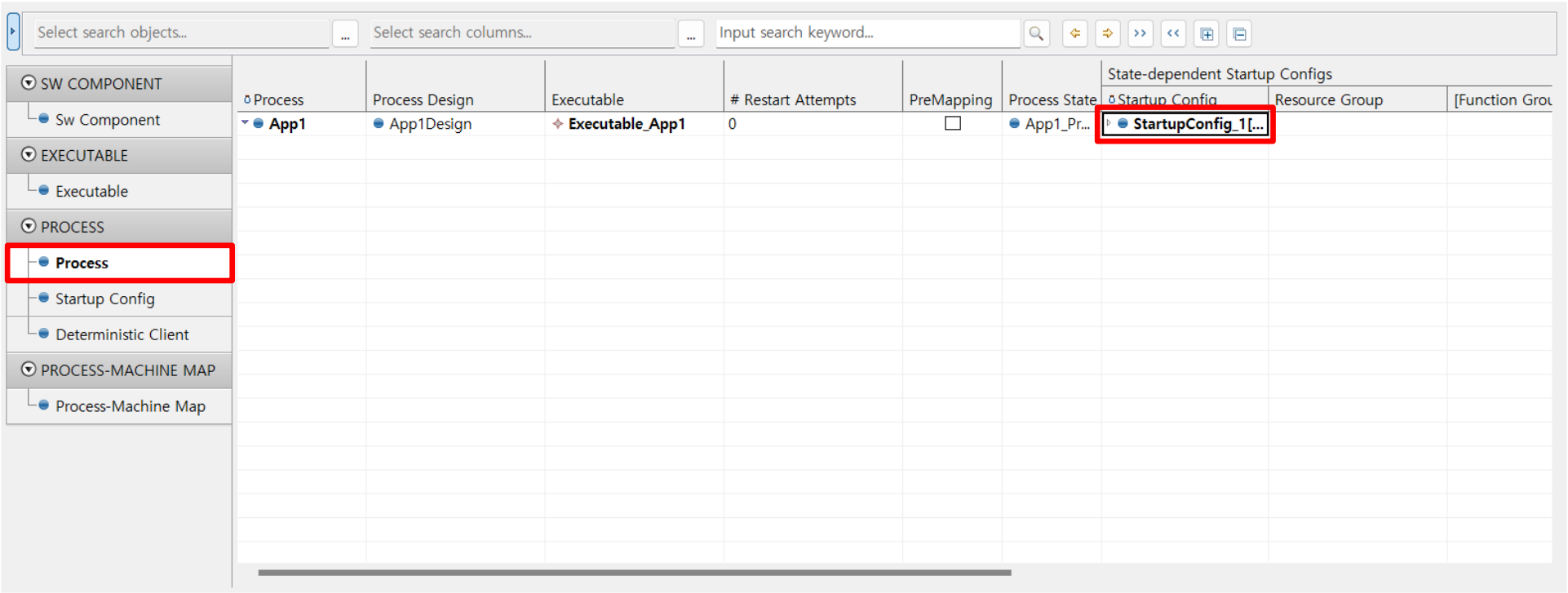
Application Editor



EXEC: Adaptive Applications 설정

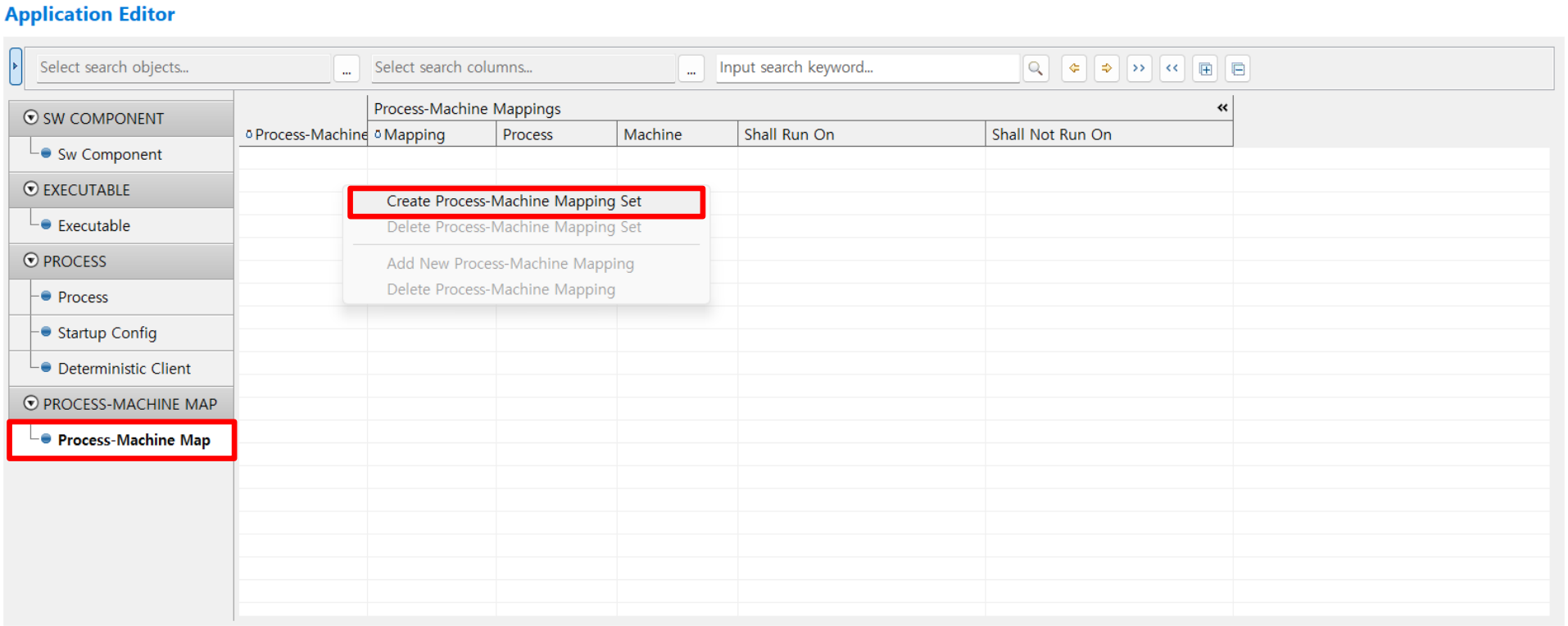
- **Startup Config 연결**
  - ✓ 생성한 Startup Config를 Process에 연결하기 위해 좌측의 'Process' 탭으로 이동함
  - ✓ Process의 Startup Config를 생성한 'StartupConfig_1'로 수정함

Application Editor



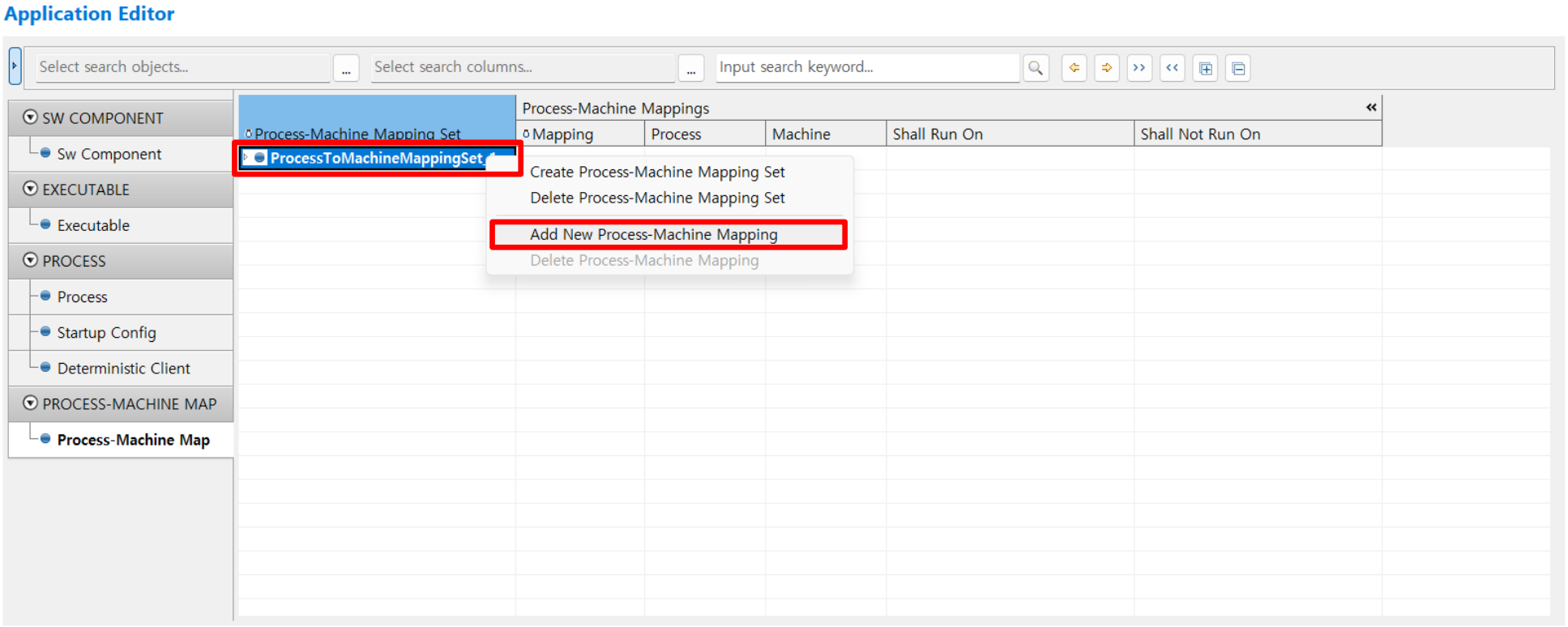
EXEC: Adaptive Applications 설정

- **Process-Machine Mapping Set 추가**
  - ✓ Process-Machine Mapping Set 추가를 위해 좌측의 'Process-Machine Map' 탭으로 이동함
  - ✓ 빈 곳에서 우클릭을 하여 'Create Process-Machine Mapping Set'을 클릭함



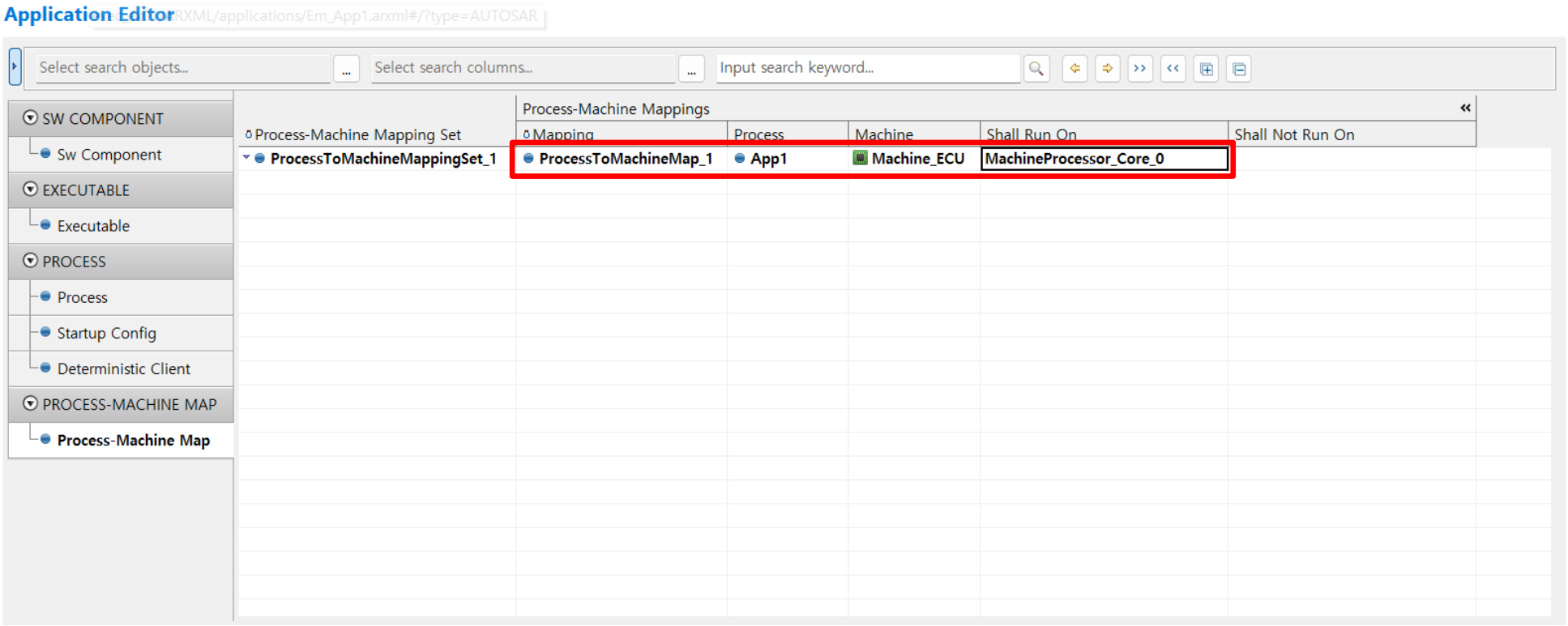
EXEC: Adaptive Applications 설정

- **Process-Machine Mapping Set 추가 확인 및 Process-Machine Mapping 추가**
  - ✓ 생성한 Process-Machine Mapping Set에서 우클릭하여 'Add New Process-Machine Mapping'을 클릭함



EXEC: Adaptive Applications 설정

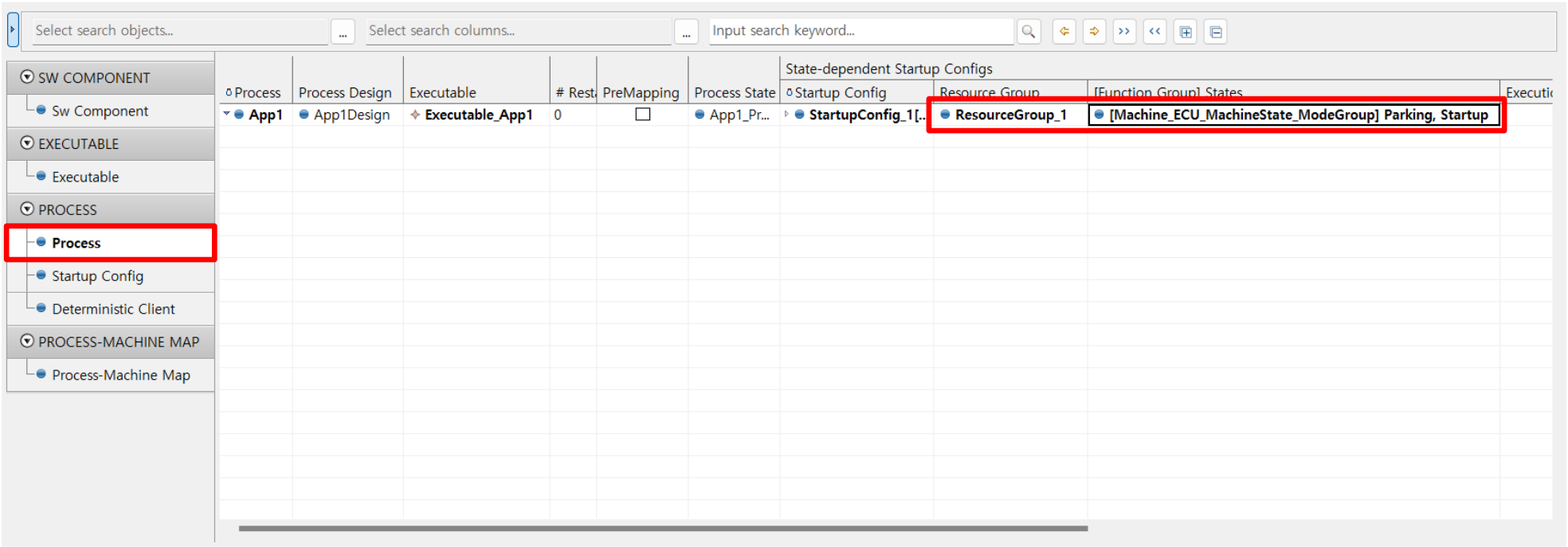
- **Process-Machine Mapping 추가 확인 및 설정**
  - ✓ 생성된 새로운 Process-Machine Mapping을 확인하고 다음과 같이 수정함
    - ✓ Process : ‘App1’
    - ✓ Machine : 'Machine_ECU’
    - ✓ Shall Run On : ‘MachineProcessor_Core_0’



EXEC: Adaptive Applications 설정

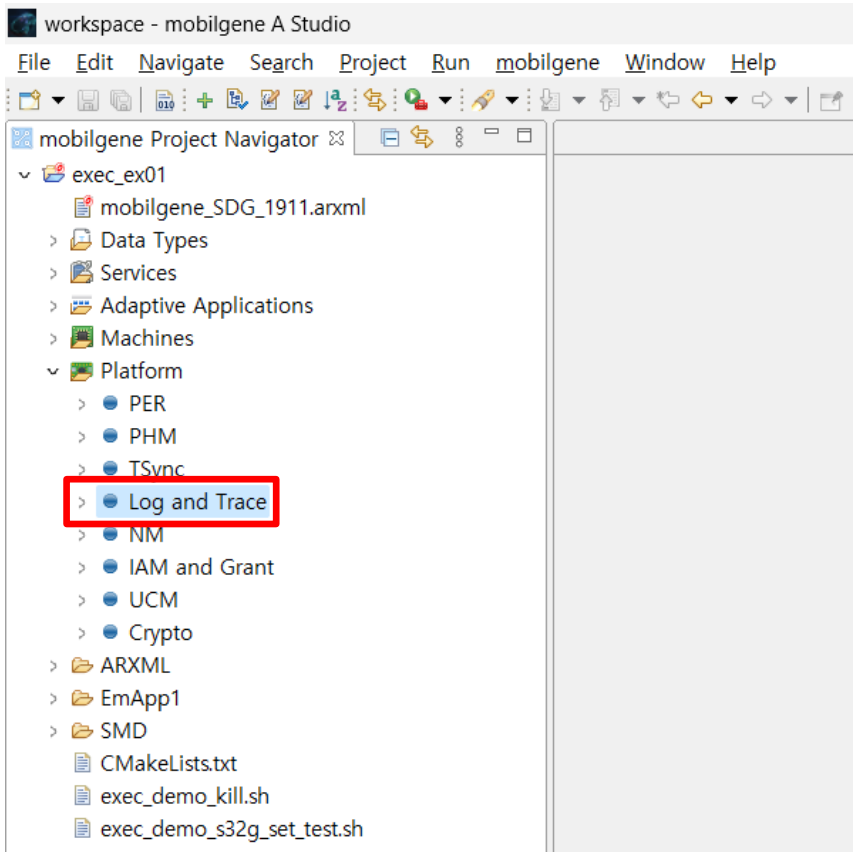
- **Process 추가 설정**
  - ✓ Machine과 관련된 Process 설정을 추가하기 위해 좌측의 'Process' 탭으로 이동함
  - ✓ Process에 대해 다음과 같이 추가적으로 수정함
    - ✓ Resource Group : 'ResourceGroup_1'
    - ✓ [Function Group] States : 'Parking, Startup'

Application Editor



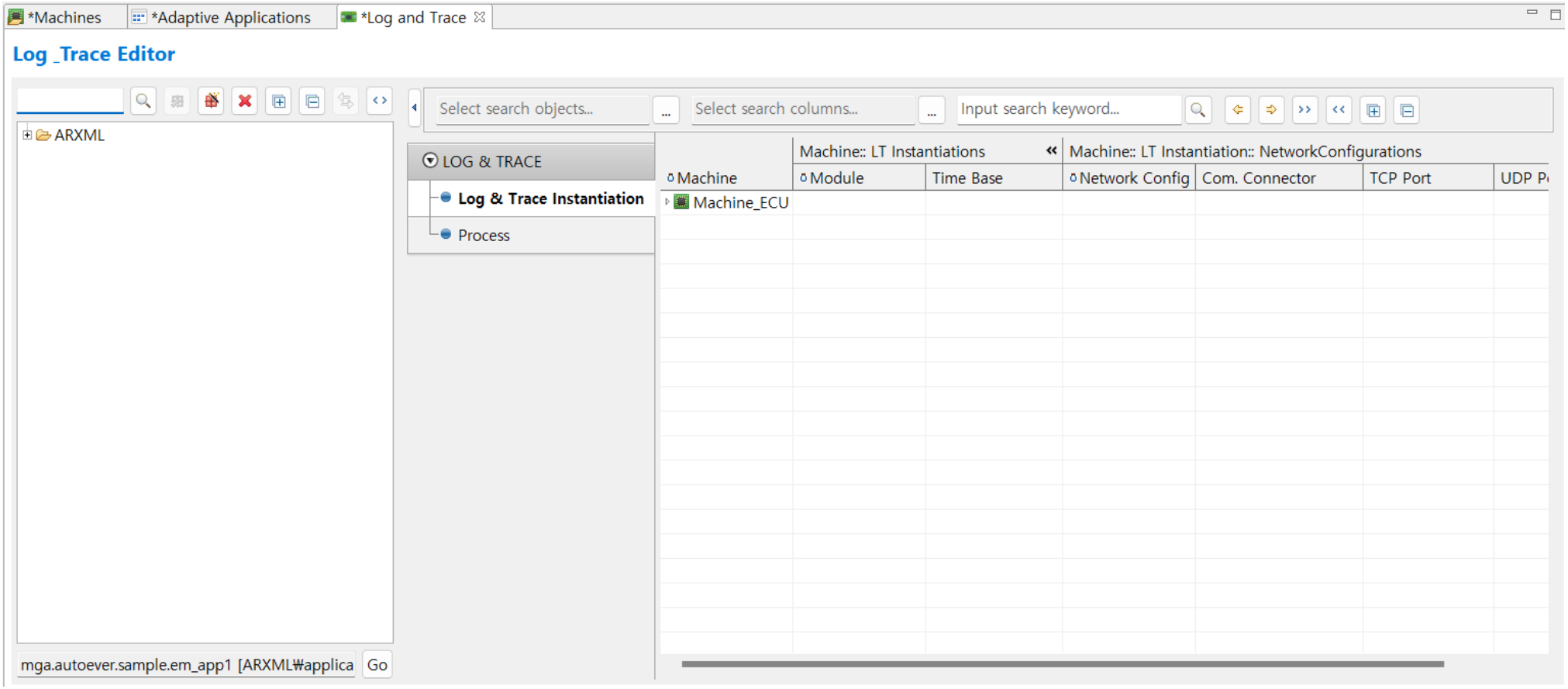
EXEC: Log 설정

- **Log and Trace Editor 활성화**
  - ✓ 왼쪽의 'mobilgene Project Navigator' 창에서 해당 Project의 'Platform' - 'Log and Trace'를 더블 클릭함



EXEC: Log 설정

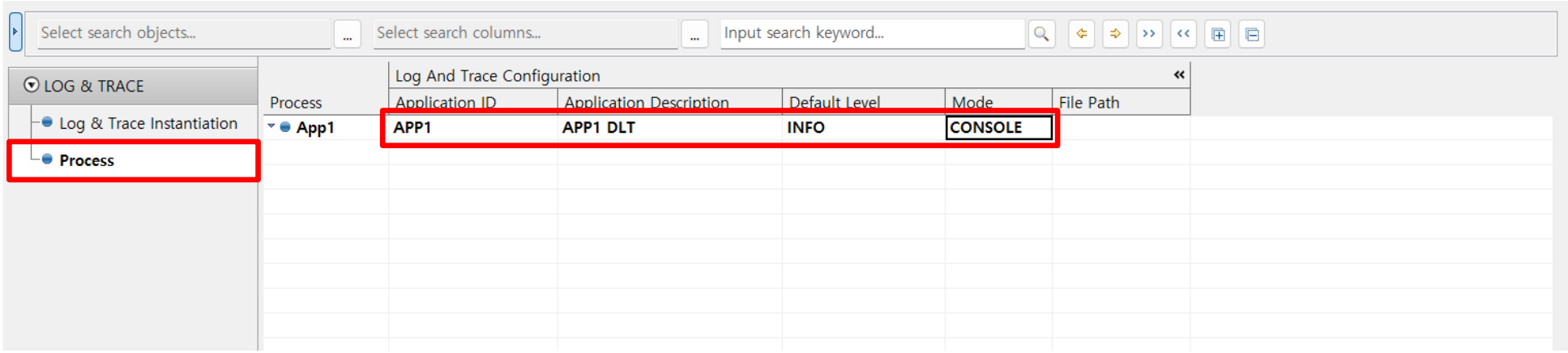
- **Log and Trace Editor 활성화 확인**
  - ✓ 활성화 된 Log and Trace Editor를 확인함



EXEC: Log 설정

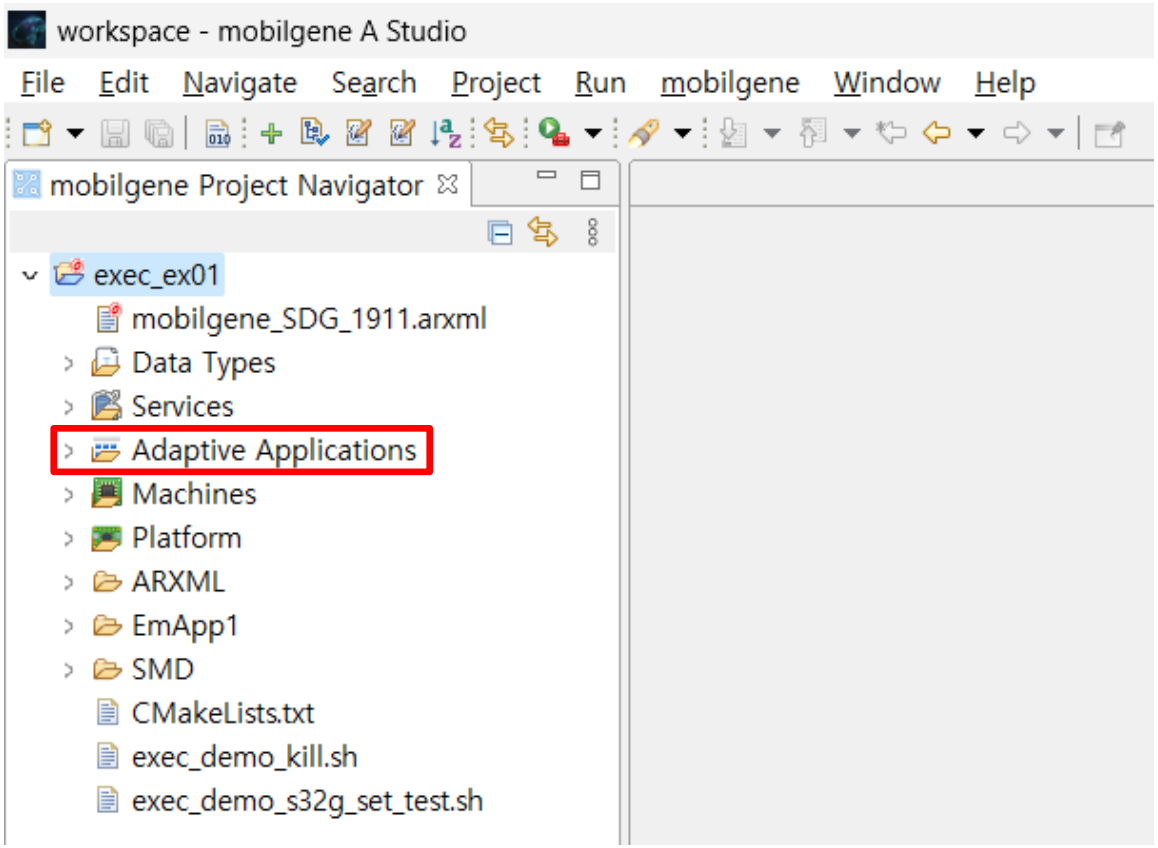
- **Process Log 설정**
  - ✓ Process에 대한 Log 설정을 하기 위해 좌측의 'Process' 탭으로 이동함
  - ✓ 생성한 Process에 대한 Log 설정을 다음과 같이 수정함
    - ✓ Application ID : 'APP1'
    - ✓ Application Description : 'APP1 DLT'
    - ✓ Default Level : 'INFO'
    - ✓ Mode : 'CONSOLE'

Log _Trace Editor



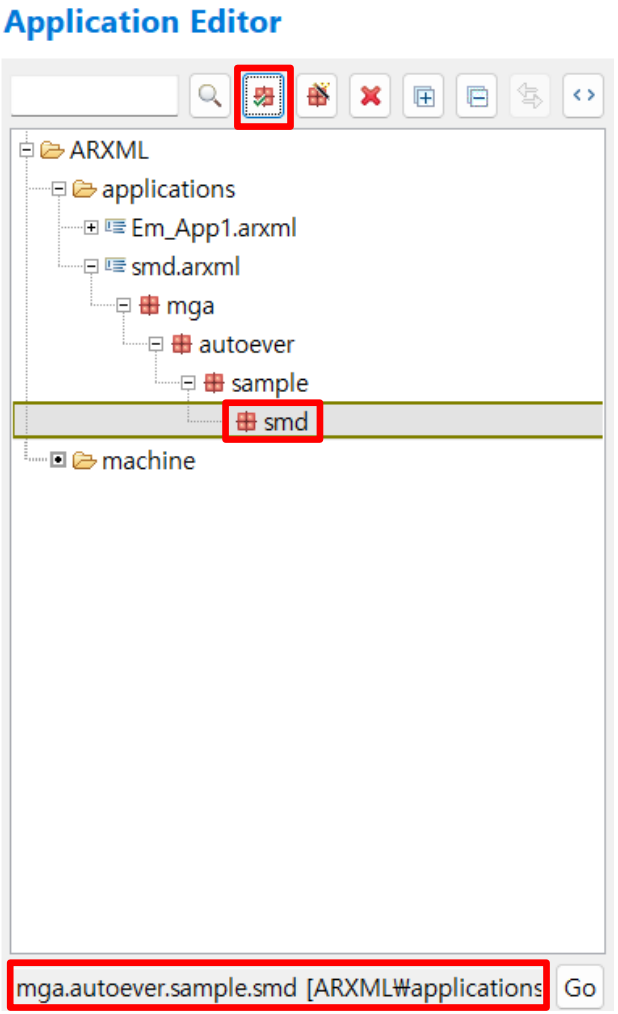
EXEC: Adaptive Applications 추가 설정

- **Adaptive Applications Editor 활성화**
  - ✓ 왼쪽의 'mobilgene Project Navigator' 창에서 해당 Project의 'Adaptive Applications'를 더블 클릭함



EXEC: Adaptive Applications 추가 설정

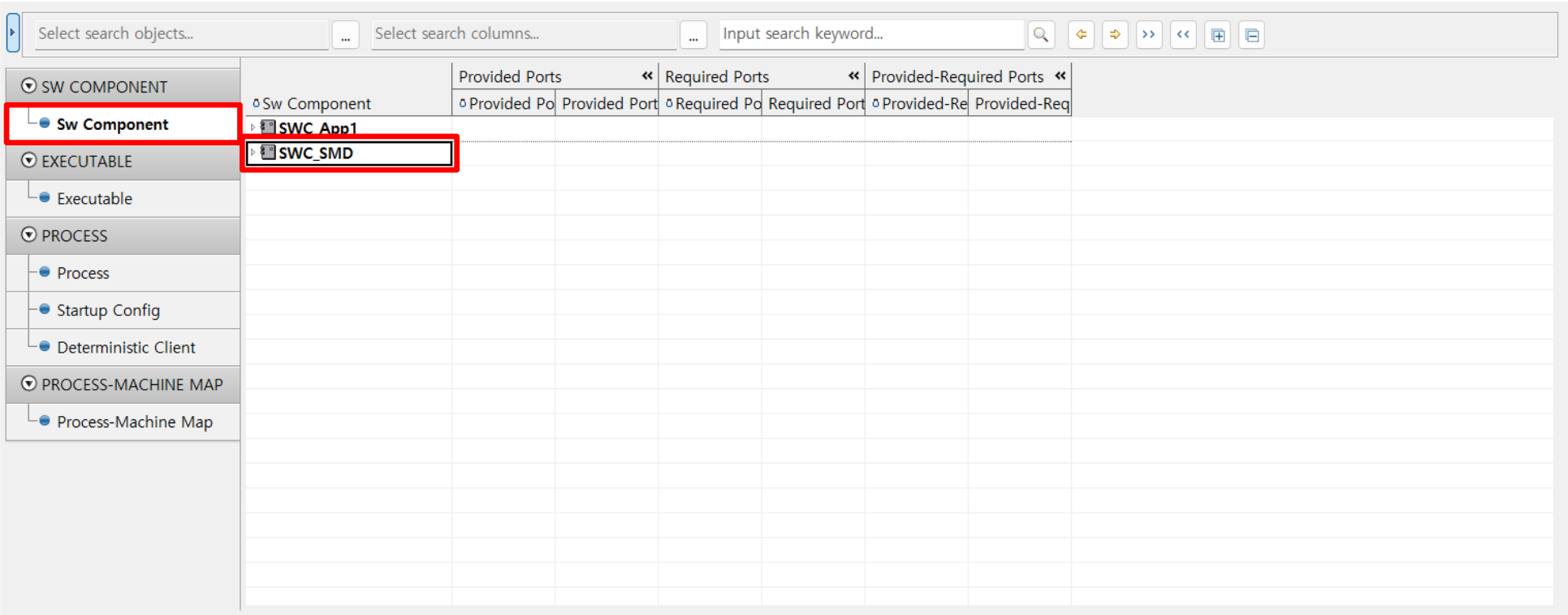
- **Default Package 설정**
  - ✓ 설정이 저장될 Default Package를 설정함
    - ✓ Default Package로 설정할 'smd' Package를 선택함
    - ✓ 우측 상단의 'Set Default Package' 버튼을 클릭함
    - ✓ 하단에 설정된 Default Package를 확인함



EXEC: Adaptive Applications 추가 설정

- **Sw Component 추가 설정**
  - ✓ 동일한 방법으로 다음과 같은 설정을 추가함

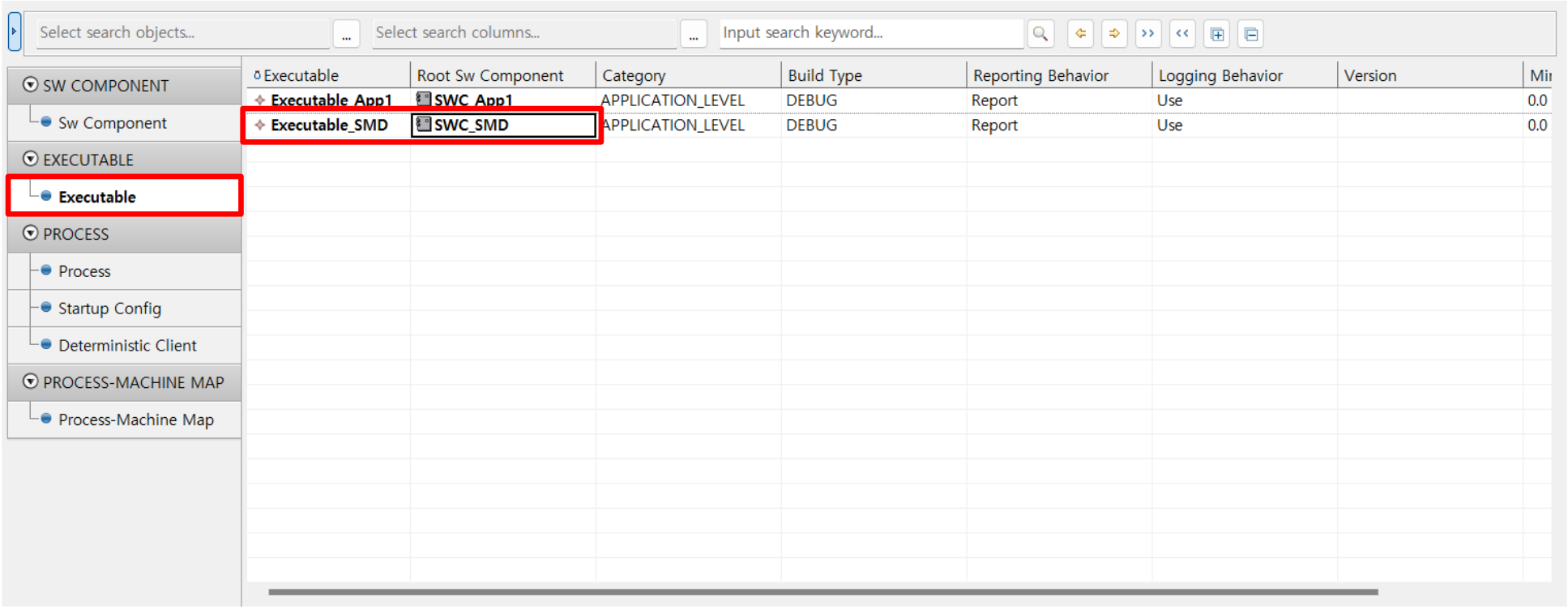
Application Editor



EXEC: Adaptive Applications 추가 설정

- **Executable 추가 설정**
  - ✓ 동일한 방법으로 다음과 같은 설정을 추가함

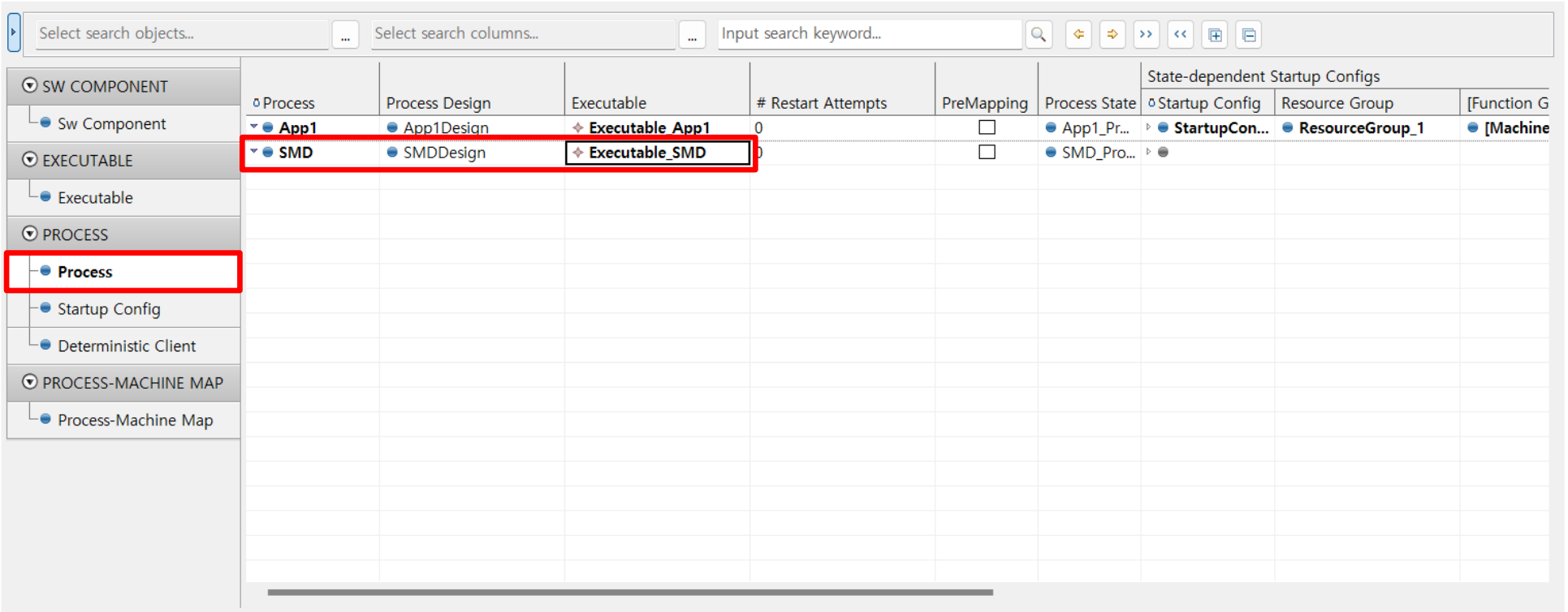
Application Editor



EXEC: Adaptive Applications 추가 설정

- **Process 추가 설정**
  - ✓ 동일한 방법으로 다음과 같은 설정을 추가함

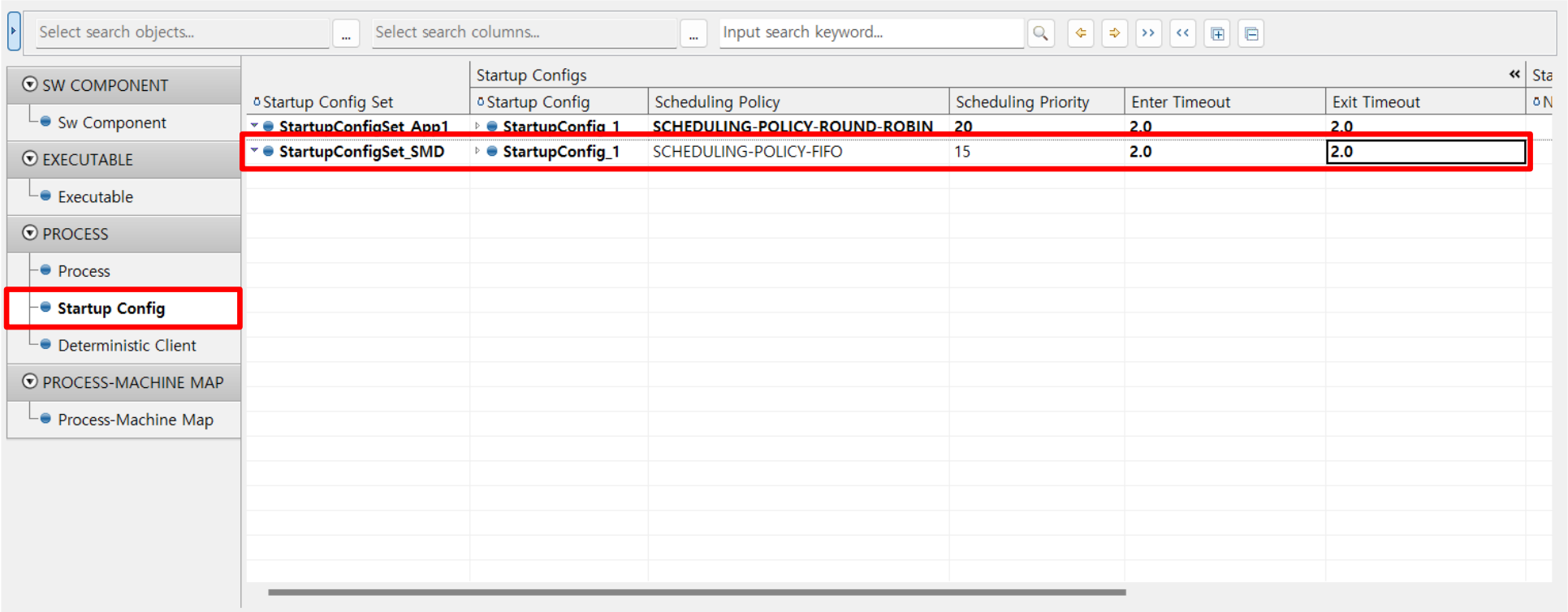
Application Editor



EXEC: Adaptive Applications 추가 설정

- **Startup Config 추가 설정**
  - ✓ 동일한 방법으로 다음과 같은 설정을 추가함

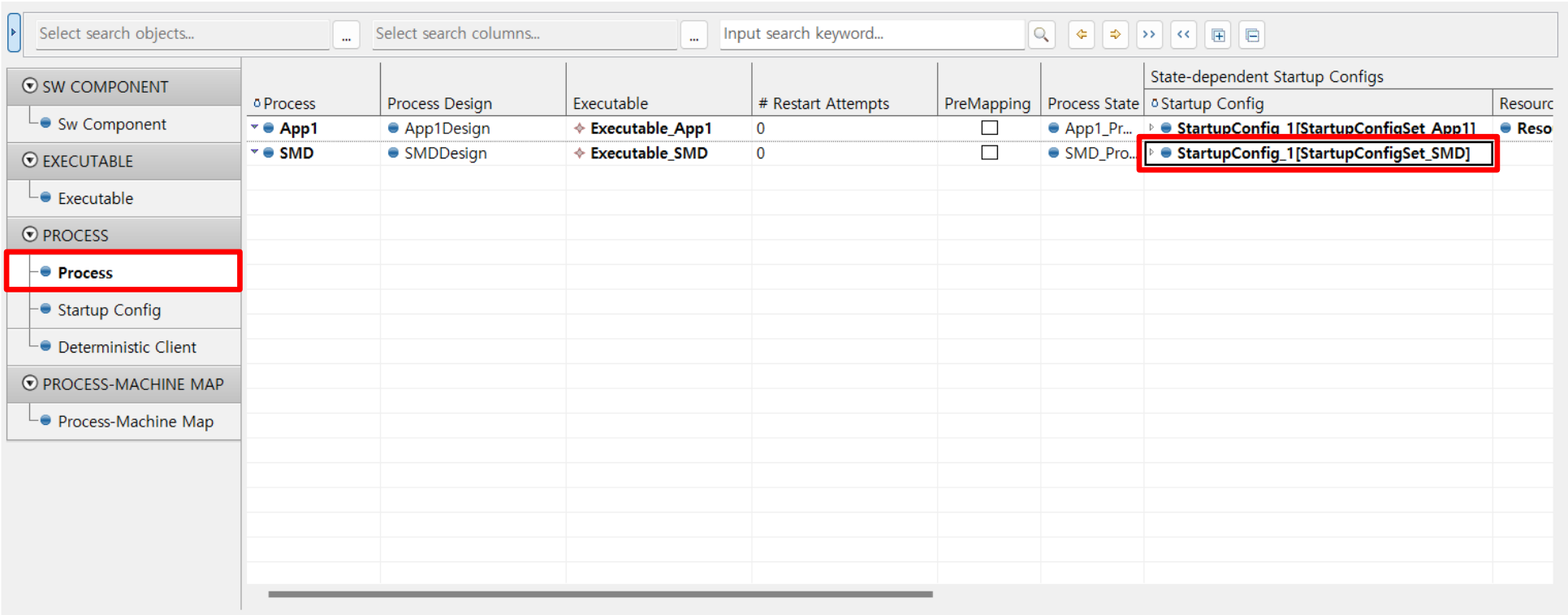
Application Editor



EXEC: Adaptive Applications 추가 설정

- **Process 추가 설정**
  - ✓ 동일한 방법으로 다음과 같은 설정을 추가함

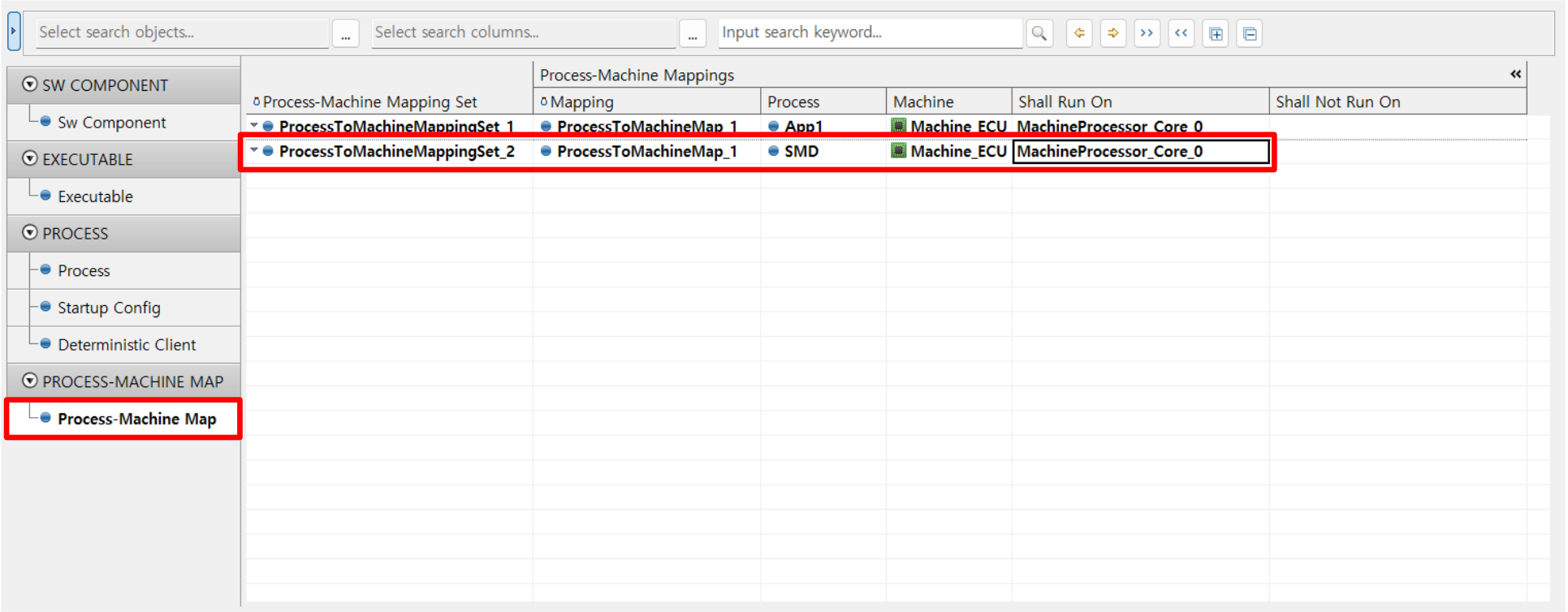
Application Editor



EXEC: Adaptive Applications 추가 설정

- **Process-Machine Map 추가 설정**
  - ✓ 동일한 방법으로 다음과 같은 설정을 추가함

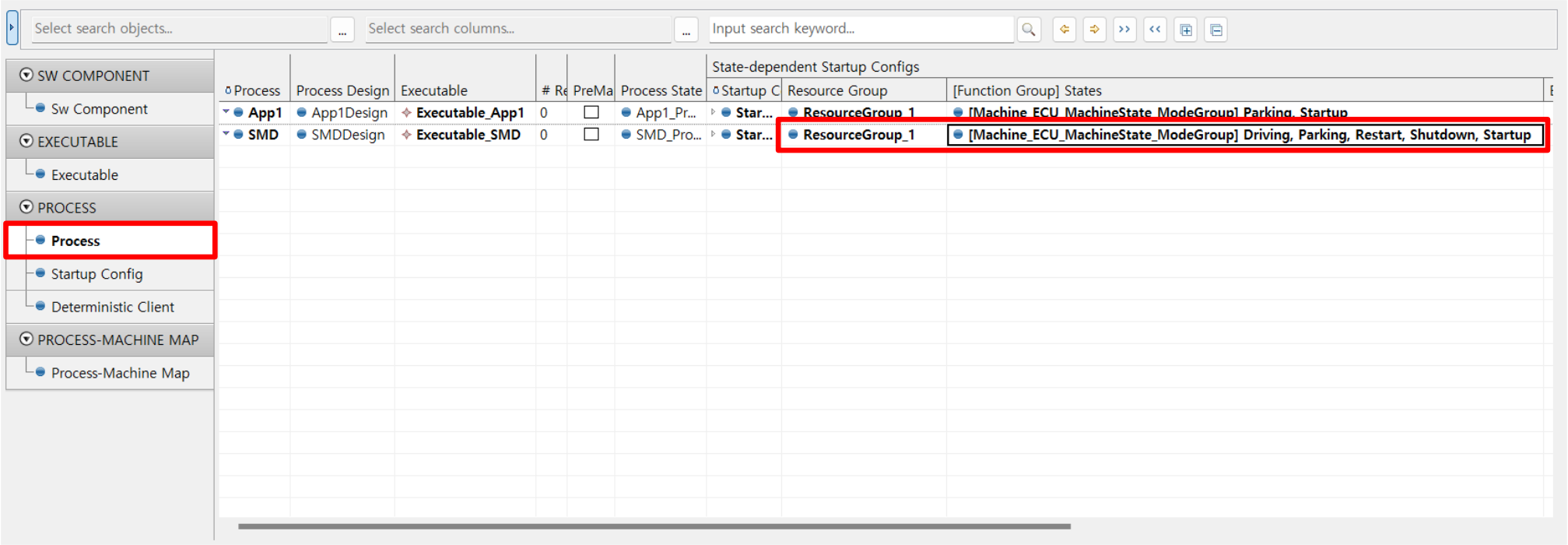
Application Editor



EXEC: Adaptive Applications 추가 설정

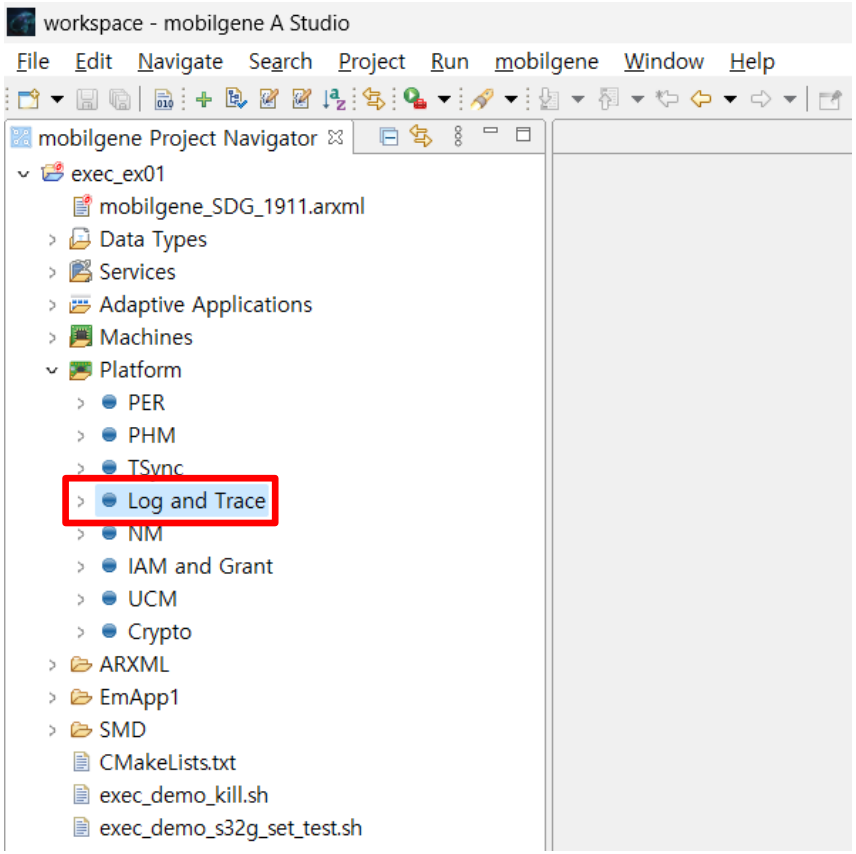
- **Process 추가 설정**
  - ✓ 동일한 방법으로 다음과 같은 설정을 추가함

Application Editor



EXEC: Log 추가 설정

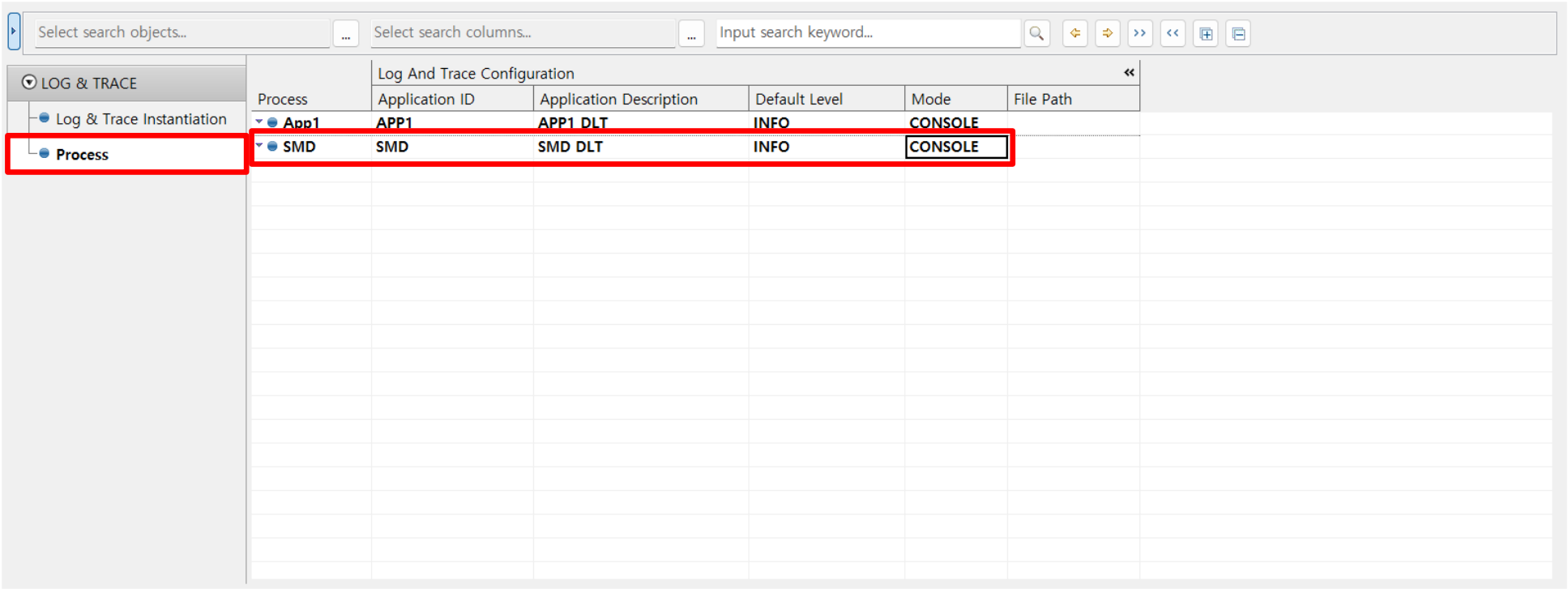
- **Log and Trace Editor 활성화**
  - ✓ 왼쪽의 'mobilgene Project Navigator' 창에서 해당 Project의 'Platform' - 'Log and Trace'를 더블 클릭함



EXEC: Log 추가 설정

- **Process Log 추가 설정**
  - ✓ 동일한 방법으로 다음과 같은 설정을 추가함

Log _Trace Editor



EXEC: Code 구현

- **'em_app1.h' 파일 작성**
  - ✓ 다음과 같은 'em_app1.h' 파일 작성 ('EmApp1' - 'include' - 'em_app1.h')

Machines | Adaptive Applications | Log and Trace | em_app1.h

```
#ifndef MGA_AUTOEVER_SAMPLE_EM_APP1_H_
#define MGA_AUTOEVER_SAMPLE_EM_APP1_H_

#include <ara/log/logging.h>

namespace mga {
namespace autoever {
namespace sample {

class EmApp1 {
public:
    EmApp1();
    ~EmApp1();

    void Init();
    void Act();

private:
    ara::log::Logger& em_app1_logger;
};

}  // namespace sample
}  // namespace autoever
}  // namespace mga

#endif  // MGA_AUTOEVER_SAMPLE_EM_APP1_H_
```

EXEC: Code 구현

- **'em_app1.cpp' 파일 작성**
  - ✓ 다음과 같은 'em_app1.cpp' 파일 작성 ('EmApp1' - 'src' - 'em_app1.cpp')

Machines | Adaptive Applications | Log and Trace | em_app1.h | em_app1.cpp | main.cpp

```
/**********************************************************************************************************************
 *  INCLUDES
 *********************************************************************************************************************/
#include "em_app1.h"


/**********************************************************************************************************************
 *  VARIABLES
 *********************************************************************************************************************/


/**********************************************************************************************************************
 *  FUNCTIONS
 *********************************************************************************************************************/

namespace mga {
namespace autoever {
namespace sample {

EmApp1::EmApp1()
    : em_app1_logger{ara::log::CreateLogger("CTX1", "EM APP1", ara::log::LogLevel::kVerbose)} {}

EmApp1::~EmApp1() {}

void EmApp1::Init()
{
    em_app1_logger.LogInfo() << "APP1 Init Function start.";
}

void EmApp1::Act()
{
    // em_app1_logger.LogInfo() << "APP1 Act Function start.";
}

}  // namespace sample
}  // namespace autoever
}  // namespace mga
```

EXEC: Code 구현

- **'main.cpp' 파일 작성**
  - ✓ 다음과 같은 'main.cpp' 파일 작성 ('EmApp1' - 'src' - main.cpp')

Machines | Adaptive Applications | Log and Trace | em_app1.h | em_app1.cpp | main.cpp

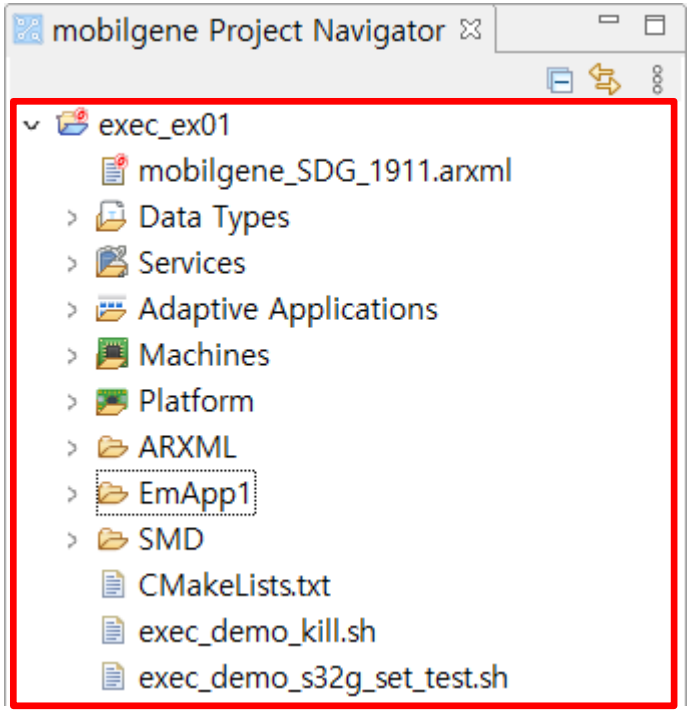
```
/**********************************************************************************
 *  INCLUDES
 **********************************************************************************/
#include <thread>
#include <chrono>
#include <cstdio>
#include <cstdint>
#include <csignal>

#include <ara/exec/execution_client.h>
#include <ara/log/logging.h>

#include "em_app1.h"

/**********************************************************************************
 *  VARIABLES
 **********************************************************************************/
ara::exec::ExecutionClient exec_client;
std::atomic<bool>exit_requested(false);
ara::log::Logger& logger{CreateLogger("CTX1", "EM APP1", ara::log::LogLevel::kVerbose)};

/**********************************************************************************
 *  FUNCTIONS
 **********************************************************************************/
void ThreadAct1()
{
    logger.LogInfo() << "==================================== APP1 is running.";
    mga::autoever::sample::EmApp1 app1;
    app1.Init();

    while (1) {
        app1.Act();

        // sleep
        std::this_thread::sleep_for(std::chrono::milliseconds(400));

        if(exit_requested.load()) {
            break;
        }
    }

    logger.LogInfo() << "Application is Run exit.";
}
```

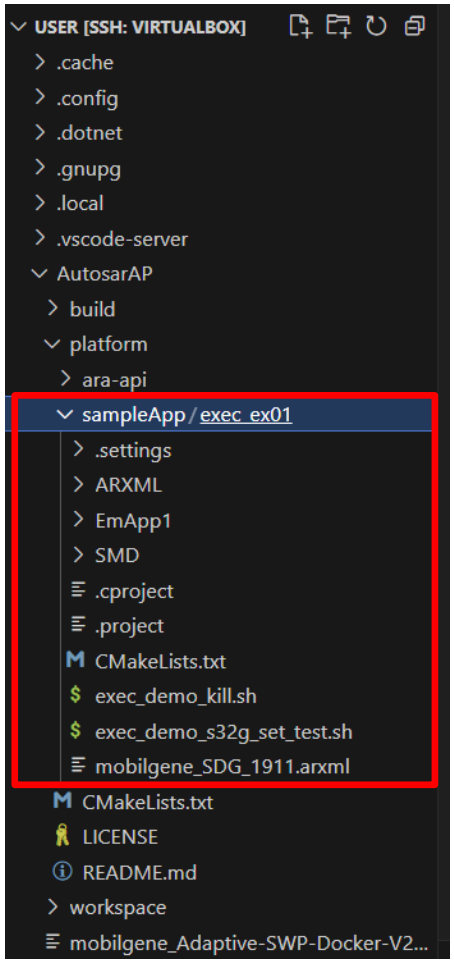
Machines | Adaptive Applications | Log and Trace | em_app1.h | em_app1.cpp | main.cpp

```
    logger.LogInfo() << "Application is Run exit.";
}

void handle_signal(int _signal) {
    if(!exit_requested.load() && (_signal == SIGINT || _signal == SIGTERM)) {
        logger.LogInfo() << "received signal:" << _signal;
        exit_requested.store(true);
    }
}

int main(int argc, char* argv[])
{
    exec_client.ReportExecutionState(ara::exec::ExecutionState::kRunning);

    logger.LogInfo() << "WAIT signal:";
    signal(SIGINT, handle_signal);
    signal(SIGTERM, handle_signal);

    logger.LogInfo() << "Application will be initialized.";
    std::thread act1(ThreadAct1);

    act1.join();

    logger.LogInfo() << "APP1:Application shutdown initiated.";

    exec_client.ReportExecutionState(ara::exec::ExecutionState::kTerminating);

    logger.LogInfo() << "Application shutdown is done.";

    return 0;
}
```

EXEC: Build

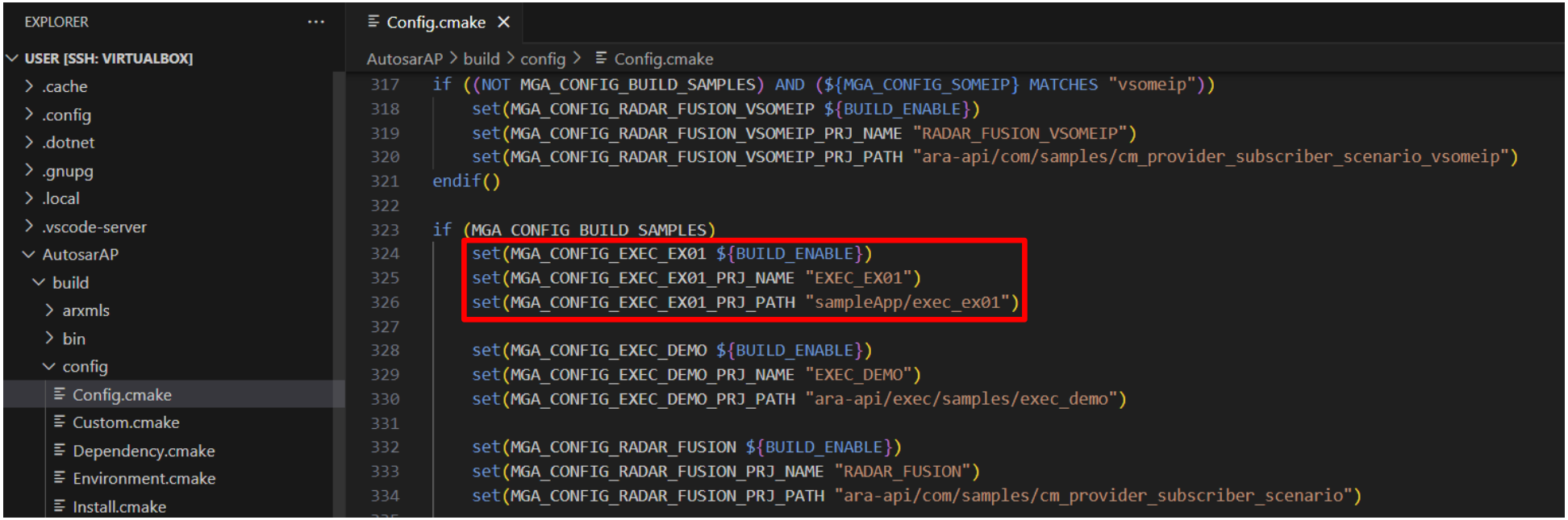
- 개발 내용 빌드 환경으로 복사
  - ✓ mobilgene A Studio에서 개발한 Adaptive Application을 빌드 환경으로 복사





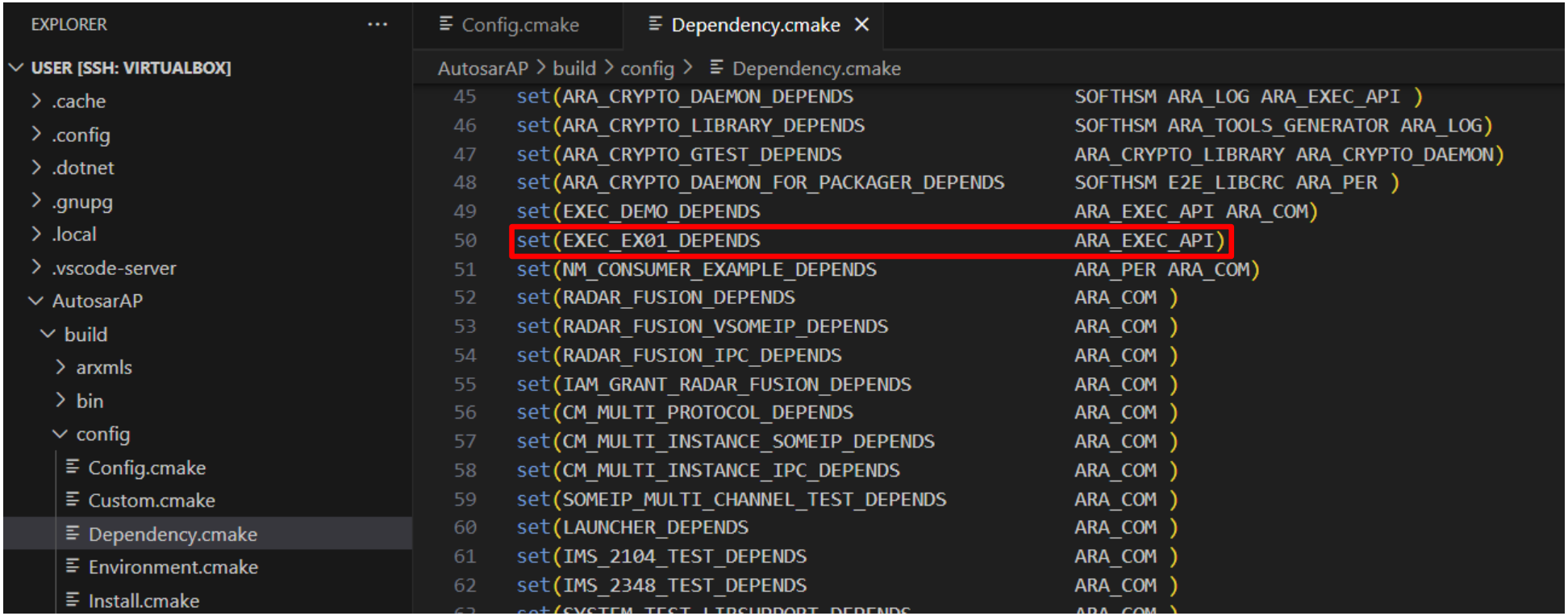
EXEC: Build

- **Adaptive Application 관련 매크로 설정 추가**
  - ✓ 'build' – 'config' - 'Config.cmake'에 추가하고자 하는 Adaptive Application 관련 매크로 설정 추가



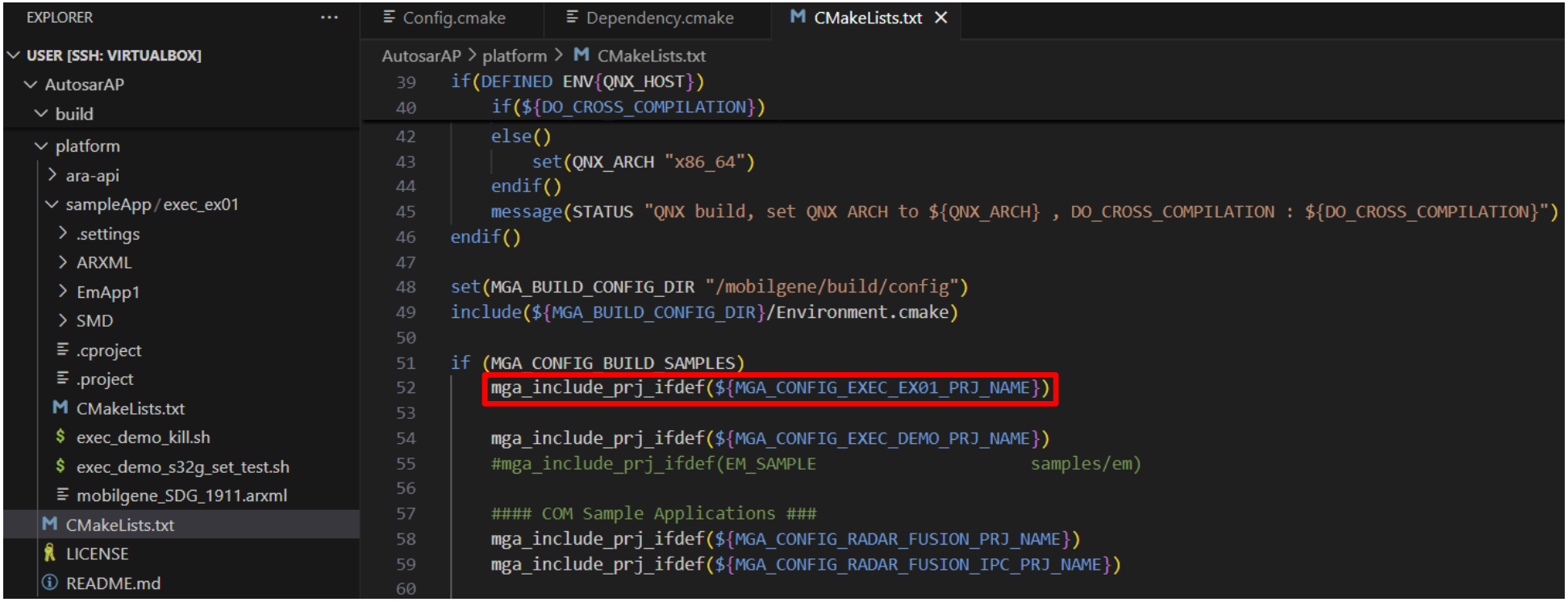
EXEC: Build

- **Adaptive Application 관련 의존성 설정 추가**
  - ✓ 'build' – 'config' - 'Dependency.cmake'에 추가하고자 하는 Adaptive Application 관련 의존성 설정 추가



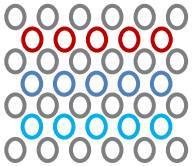
EXEC: Build

- **Adaptive Application을 Build 목록에 추가**
  - ✓ Adaptive Application을 'platform' - 'CMakeLists.txt'에 추가하여 Build 목록에 추가



Q & A

Thank you for your attention



Architecture and
Compiler
for Embedded Systems Lab.

School of Electronics Engineering, KNU

ACE Lab (junho7513@knu.ac.kr)