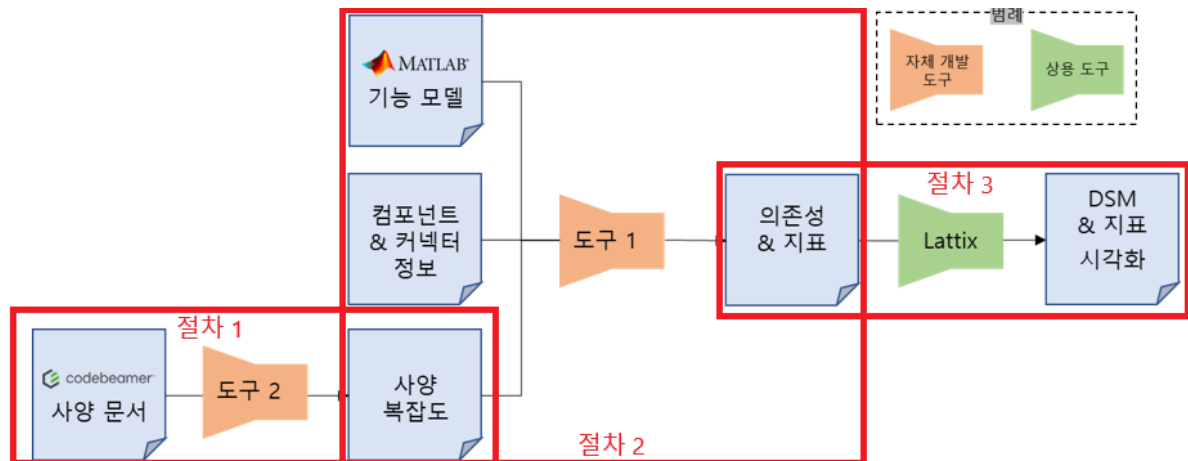


도구 사용 설명서

개요

이 설명서는 도구 2(사양 문서 분석 도구)를 이용해 기초 데이터부터 Lattix 시각화까지 전체 과정을 2가지 절차로 나누어 설명합니다.



<전체 과정의 다이어그램 및 각 절차의 범위>

위 다이어그램에서 각 절차는 한 가지 도구를 조작하여 수행되며 각각 입력과 출력 자료가 존재하는 것을 확인할 수 있습니다. 절차 1부터 절차 3까지 순서대로 수행함으로써 전체 과정을 수행할 수 있습니다.

[절차 1]

목적: 아키텍처 평가, 의존성을 추출하고 대부분의 지표 값을 계산하거나 지표 값 계산을 위한 기초 데이터를 계산함

사용 도구: 도구 1

입력: Excel 파일 (아키텍처 산출물), SLX 파일 (기능 개발 산출물), 사양 복잡도 (도구 2 출력)

출력: LDI 파일 (절차 3에서 입력 자료로 사용)

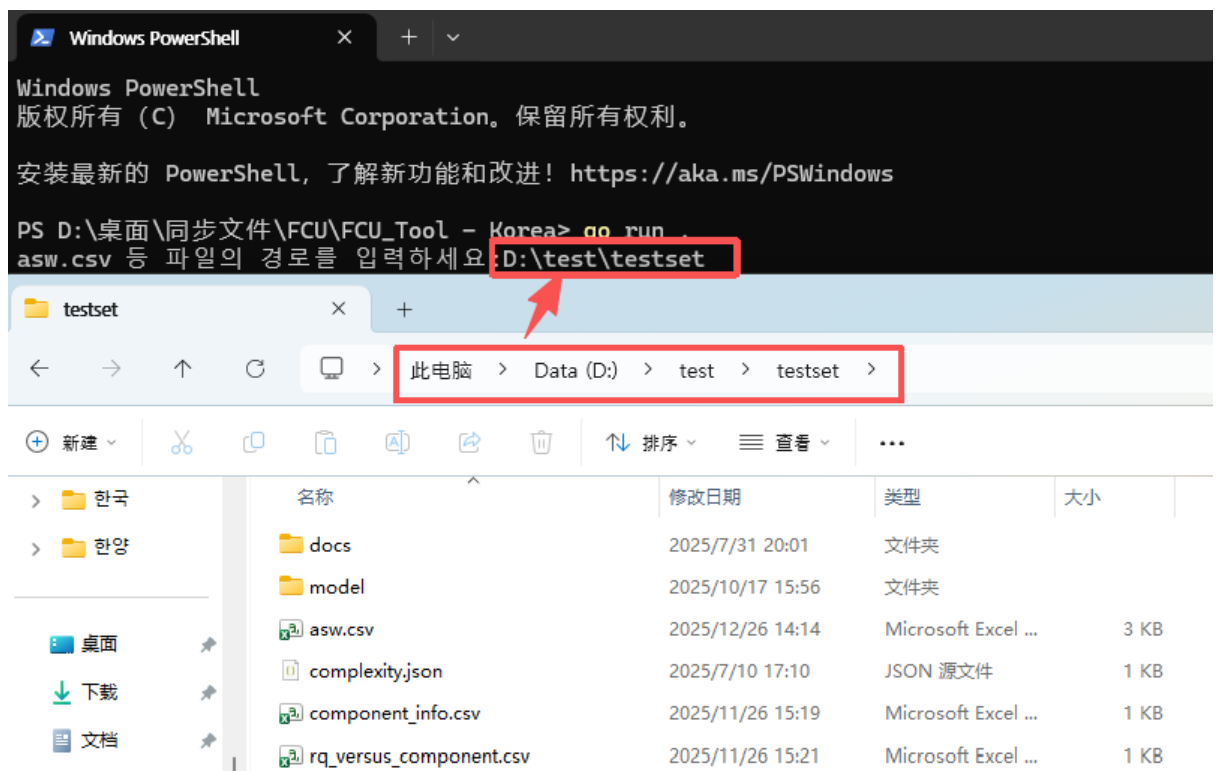
1. 임의의 폴더를 만든 후, 절차 2에서 출력된 complexity.json 파일을 복사하여 붙여넣습니다. (파일 이름은 그대로) 또한 Autosar Connector가 정의된 엑셀 파일을 복사하여 붙여넣고 파일 이름은 asw.csv로 변경합니다. component_info.csv와 rq_versus_component.csv 파일을 생성하고 내용을 채웁니다. (파일 양식과 예시는 testset 폴더를 참고하세요.) 이 작업이 완료되면 임의의 폴더에 총 4개의 파일이 존재합니다.
2. 명령 프롬프트 창에서 FCU_Tool 폴더로 들어간 후, go run . 명령을 실행합니다.

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

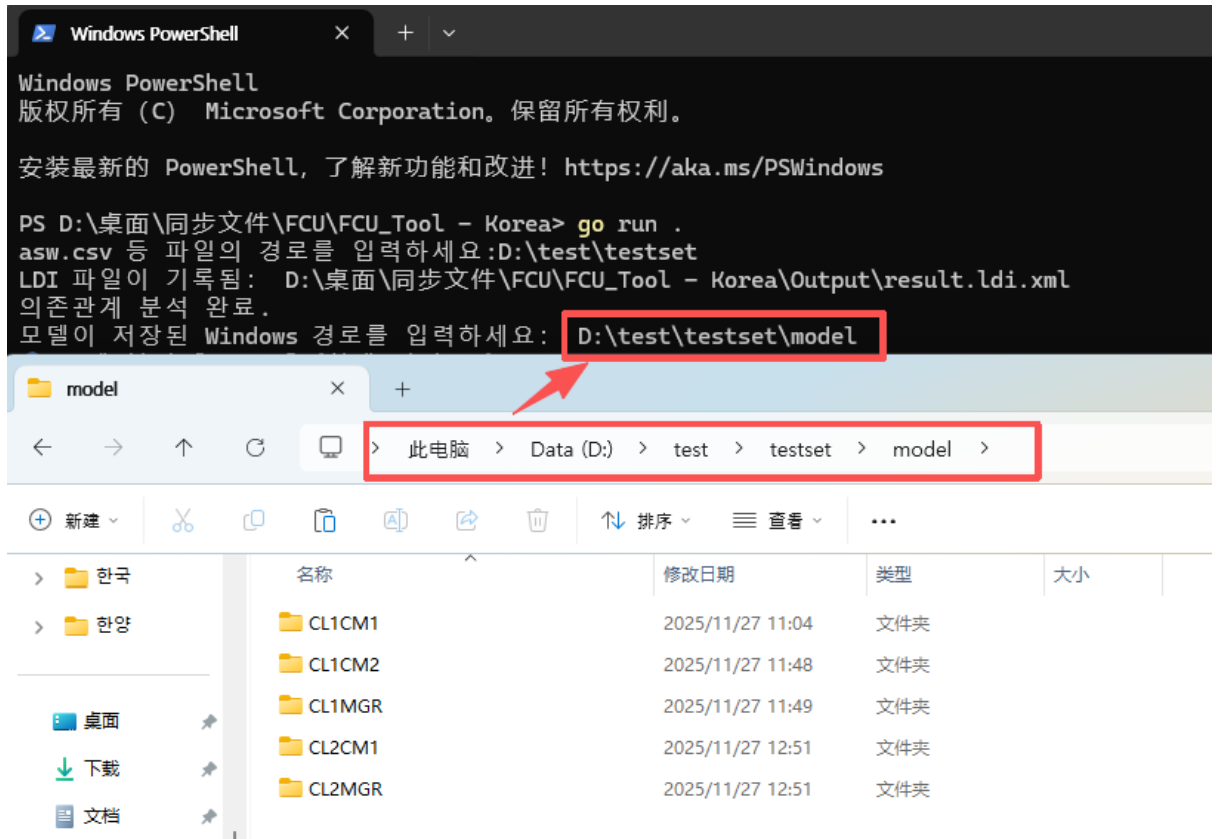
安装最新的 PowerShell, 了解新功能和改进! https://aka.ms/PSWindows

PS D:\桌面\同步文件\FCU\FCU_Tool - Korea> go run .
```

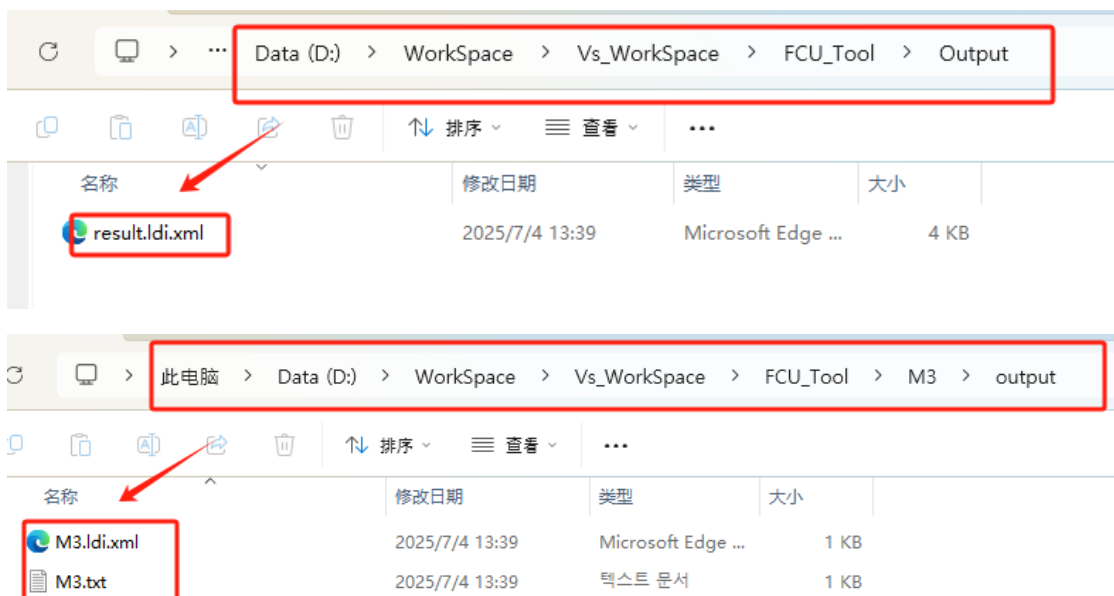
3. 과정 1에서 진행한 임의의 폴더 경로를 복사한 후 프로그램이 "asw.csv 등 파일의 경로를..." 질의에 붙여 넣은 후 Enter를 입력합니다.



4. 두번째 "모델이 저장된 Windows 경로를..." 질의에 SLX 파일이 모여있는 루트 경로를 복사하여 붙여 넣은 후 Enter를 입력합니다.



5. 프로그램이 성공하면 Output(O 대문자) 폴더에 result.ldi.xml 파일이 생성됩니다. 마찬가지로 M1 ~ M6 폴더가 생성되며 각 폴더 내부에 output(o 소문자) 폴더가 존재하고, 지표값을 저장하는 xml 파일과 규칙을 위반하는 내용이 있는 경우 txt 파일이 저장됩니다.



[절차 2]

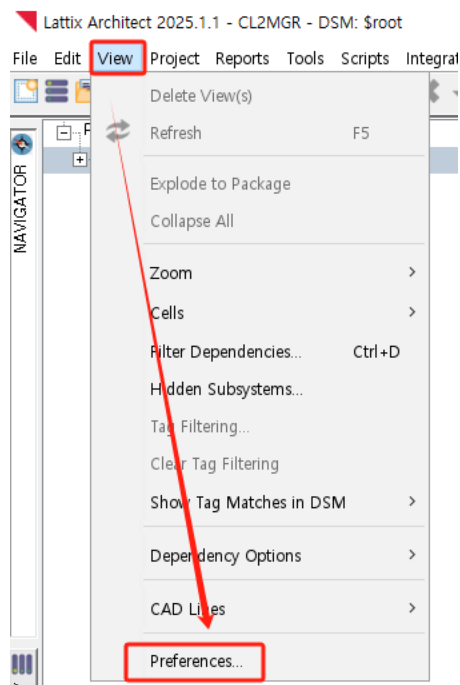
목적: 계산된 데이터를 Lattix로 시각화

사용 도구: Lattix

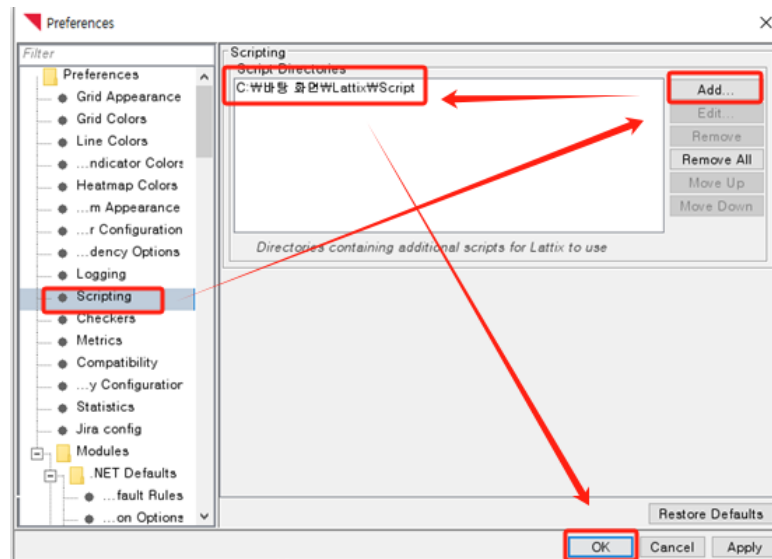
입력: LDI 파일

출력: Lattix의 GUI

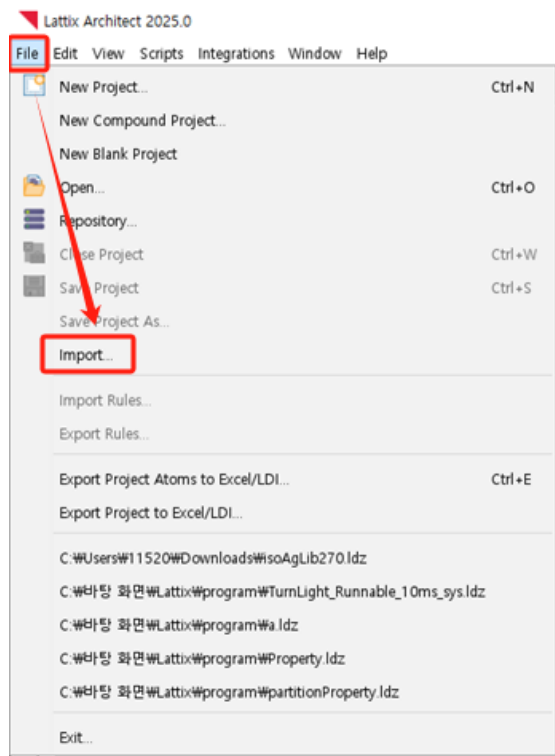
1. Lattix 소프트웨어를 열고 상단 메뉴 → View → Preferences를 선택합니다.

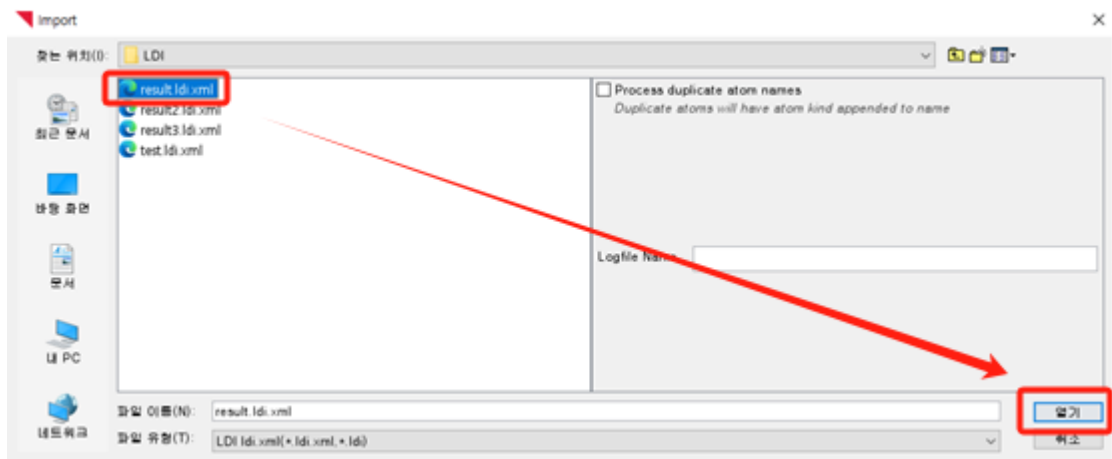


2. Preferences 창의 Scripting 탭에서 Script Code 폴더의 경로를 추가합니다.



3. 절차 2에서 출력된 result.Idi.xml 파일을 상단 메뉴 → File → Import에서 선택합니다.





4. GUI에서 DSM과 Metric 탭에서 지표를 확인할 수 있습니다.

Lattix Architect 2025.1.1 - CL2MGR - DSM: \$root

File Edit View Project Reports Tools Scripts Integrations Window Help

Heatmap Selection

NAVIGATOR

REPOSITORY

Heatmap Selection

Component	CL1CM1	CL1CM2	CL1MGR	CL2CM1	CL2MGR
CL1CM1	20%	1			
CL1CM2		20%	1		
CL1MGR	1		20%		
CL2CM1				15%	1
CL2MGR					25%

System Architecture Object Oriented ☒ Custom

Metric	Value
AutosarViolations	Undefined
CertViolations	Undefined
DecisionCoverage	Undefined
EntryCoverage	Undefined
StatementCoverage	Undefined
MCDCPairsCoverage	Undefined
MisraViolations	Undefined
M1(Component Complexity)	2.506
CL1CM1	173
CL1CM2	173
CL1MGR	887
CL2CM1	90
CL2MGR	1.183
M2(Specification Complexity)	41
M2_1(Percentage of Horizontal Dependency Inversions)	3.73
M3(Number of Vertical Dependency Inversions)	1
M3_A (M3 / log10(M3Demo + 1))	0.93
M3Demo(Number of Vertical Dependencies)	11
M3Percent(Percentage of Vertical Dependency Inversions)	9.09%
M4(Number of Horizontal Separation Violations)	0
M4_A (M4 / log10(M4Demo + 1))	0.00
M4Demo(Number of Horizontal Separations)	11
M4Percent(Percentage of Horizontal Separation Violations)	0.00%
M5(Safety Partition Impact)	2
M5Demo(Number of Safety Partitions)	5
M5Percent(Percentage of Safety Partition Impact)	40.00%
M6(Communication Interference Impact)	1
M6Demo(Number of Communication Interferences)	11
M6Percent(Percentage of Communication Interference Impact)	9.09%

M6 Demo Metric: 11.0

saveSplitPos: lattix.gui.rightSplit: 761

[보충1]

```

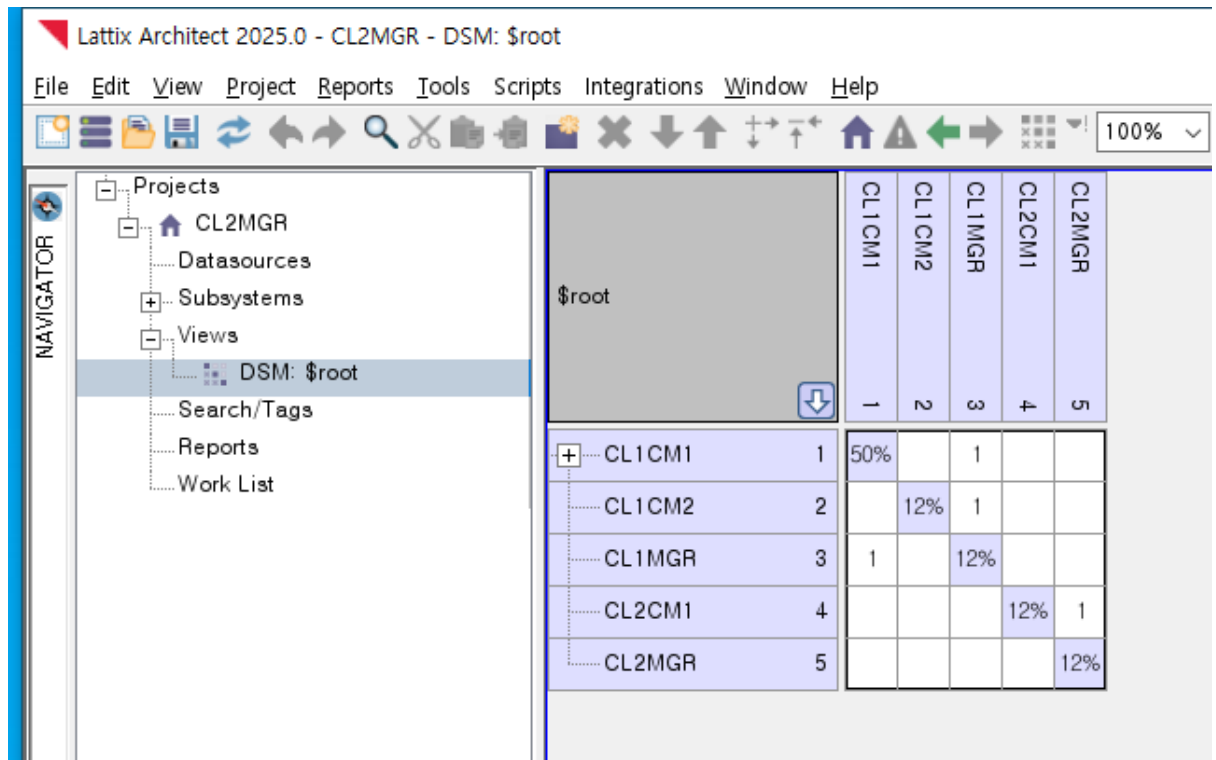
▼<element name="CL2MGR">
  <uses provider="CL2CM1" strength="5"/>
  <property name="coverage.m1">204.0000</property>
  <property name="coverage.m2">8</property>
  <property name="coverage.m3">0</property>
  <property name="coverage.m3demo">5</property>
  <property name="coverage.m4">0</property>
  <property name="coverage.m4demo">5</property>
  <property name="coverage.m5">0</property>
  <property name="coverage.m5demo">1</property>
  <property name="coverage.m6">0</property>
  <property name="coverage.m6demo">5</property>
</element>

```

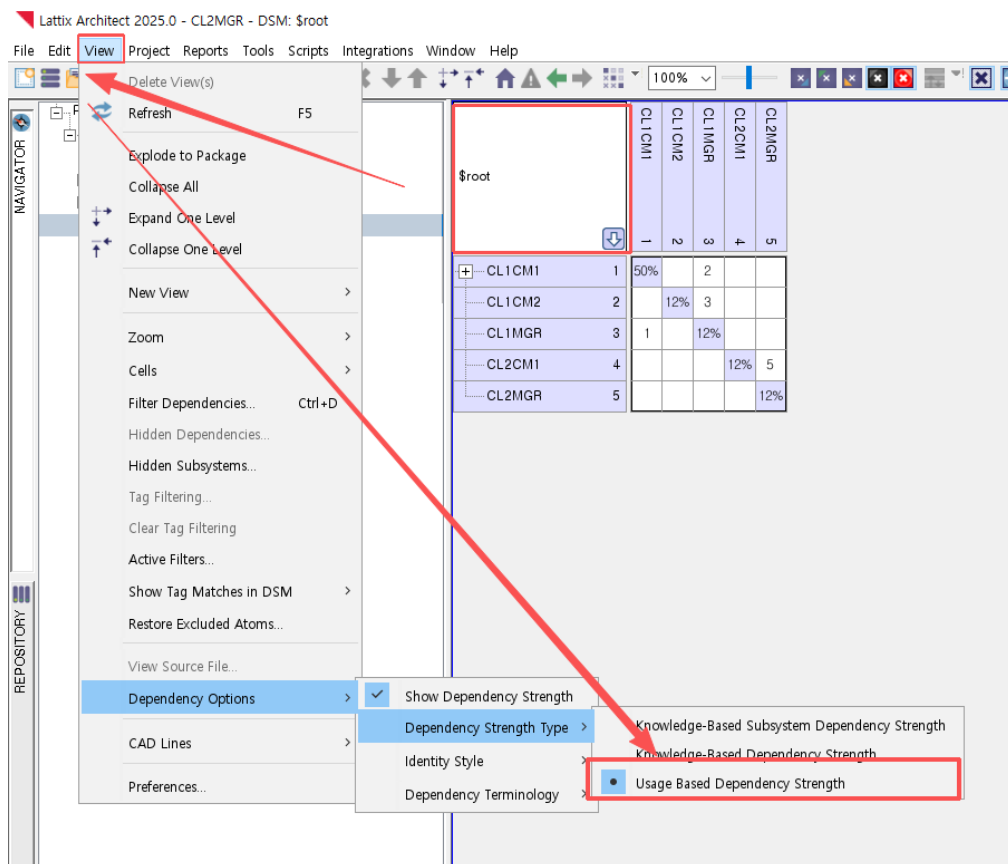
result.lidi.xml에 나오는 지표 값은 최종 값이 아닙니다.(아래의 값은 result.lidi.xml에 나온 값은 기준으로 설명합니다.)

1. M1 지표(컴포넌트 구조 복잡도) 값은 측정식: $SCC = n_{class} \cdot port_{asr} \cdot port_{sim}$ 에 따라 나옵니다(최종 결과만 바로 출력합니다.).
2. M2 지표(상대적 시스템 구조 복잡도) 값은 측정식: $RSSC = \frac{n_E}{cmx}$ 의 분모까지 계산합니다. 분자는(컴포넌트 간 연결의 총 개수)는 M3의 m3demo 값들의 총합으로 구할 수 있습니다. 이어서 lattix에서(M2에 관련 Script Code를 통해 조리함) M2_1 coverage indicator from LDI import라고 하는 값은 M2최종 값입니다.
3. M3 지표 측정식: $VDI = \frac{\sum_{e \in VE} diff(e)}{\log(n_E + 1)}$ 의 분자는 result.lidi.xml에 나타나는 m3 지표의 합입니다. 분자 중 n_E 는 result.lidi.xml에 나타나는 m3demo 지표의 합입니다. 계산 과정은 Script Code에서 수행됩니다.
4. M4 지표(횡단 분리성 위반) 측정식: $HSV = \frac{n_{HE}}{\log(n_E + 1)}$ 의 분자는 result.lidi.xml에 나타나는 m4 지표의 합입니다. 분자 중 n_E 는 result.lidi.xml에 나타나는 m4demo 지표의 합입니다. 계산 과정은 Script Code에서 수행됩니다.
5. M5 지표(안전 분할 영향도) 측정식: $SDI = \frac{n_{DI}}{n_I}$ 의 분자는 result.lidi.xml에 나타나는 m5 지표의 합입니다. 분자는 result.lidi.xml에 나타나는 m5demo 지표의 합입니다. 계산 과정은 Script Code에서 수행됩니다
6. M6 지표(통신 간섭 영향도) 측정식: $SCII = \frac{n_{LHC}}{n_C}$ 의 분자는 result.lidi.xml에 나타나는 m6 지표의 합입니다. 분자는 result.lidi.xml에 나타나는 m6demo 지표의 합입니다. 계산 과정은 Script Code에서 수행됩니다

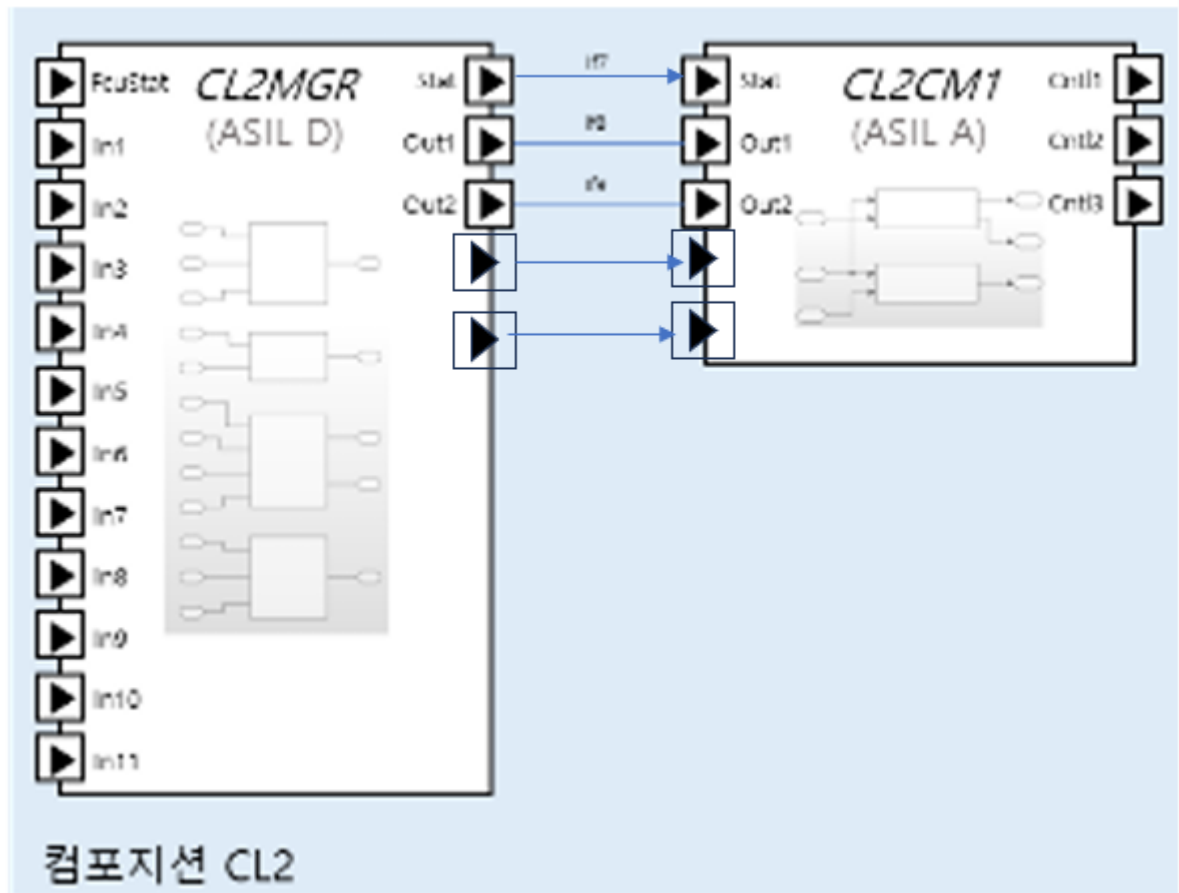
[보충2]



모델간 의존 강도가 1로 고정되면 아래 그림에 따라 설치하면 진짜 값을 나옵니다.



[보충3]



위의 그림을 통해 CL2MGR에서 CL2CM1로 향하는 선이 5개 있다는 것을 알 수 있습니다. 아래 DSM에서의 1개 선과 비교했을 때, 의존 강도가 5인 것과 대응된다고 볼 수 있습니다.

	CL1CM1	CL1CM2	CL1MGR	CL2CM1	CL2MGR
\$root	1	2	3	4	5
CL1CM1	1	50%	2		
CL1CM2	2	12%	3		
CL1MGR	3	1	12%		
CL2CM1	4			12%	5
CL2MGR	5				12%