

# 룰셋 작성 가이드

2022.11.25

## 목차

1. 개요	2
2. XML 구성	3
2.1 <ccsetting></ccsetting>	3
요소	3
2.2 <cccode></cccode>	4
요소	4
3. 스크립트 작성 방법	6
4. API 함수	7
5. 작성 예시	1

### 1. 개요

본 문서는 품질검토 룰셋 개발 및 적용에 관하여 설명한다.

룰셋은 xml 포맷의 파일로 작성하며, CCL을 기본 확장자로 가진다.

```
<?xml version="1.0" standalone="yes"?>
<CcCodeDataSet xmlns="http://tempuri.org/CcCodeDataSet.xsd">
 <CcSetting>
   <Editable>true</Editable>
   <ShowTypology>false</ShowTypology>
   <InfoUrl></InfoUrl>
   <Id>TEST</Id>
   <Title>품질검토 목록</Title>
   <RulesetName>품질검토 보고서</RulesetName>
   <RulesetUpdateDate></RulesetUpdateDate>
   <ShowSearch>false</ShowSearch>
</CcSetting>
 <CcCode>
   <ld>1</ld>
   <ParentId>-1</ParentId>
   <Number>Pre-check on BIM Model</Number>
   <lsGroup>true</lsGroup>
  </CcCode>
  <CcCode>
     <Id>1003</Id>
     <ParentId>1</ParentId>
     <Number>공간객체의 유무</Number>
     <Desc>모든 영역에 공간객체가 모델링 되어 있어야 한다.</Desc>
     <Script>
       <![CDATA[def Check():
     for storey in SELECT('storey'):
         undef_spaces = storey.SELECT('undefined space')
         if undef_spaces.COUNT() == 0:
             storey.SUCCESS('모든 영역에 공간객체가 모델링 되어 있습니다.')
         else:
             for undefined_space in undef_spaces:
                 area = undefined_space.SELECT('area').UNIT('m2').NUMBER()
                 undefined_space.ERROR('공간객체 없음 (면적: ' + str(area) + 'm2)')]]>
     </Script>
   </CcCode>
</CcCodeDataSet>
```

## 2. XML 구성

2.1 <CcSetting>



<CcSetting>은 룰셋을 정의한다.

요소

- Editable : 룰셋의 코드 편집 가능 여부를 정의한다. true(참)일 경우 사용자가 검토 항목의 입력 및 로직을 수정할 수 있으며, false(거짓)일 경우 수정이 불가능하다.
- ShowTypology : 모델의 건물 유형 메뉴 표시 여부를 정의한다. true(참)일 경우 사용자는
   모델의 건물 유형을 선택할 수 있는 메뉴를 볼 수 있다. 기본 값은 false(거짓)이다.
- InfoUrl : 룰셋의 도움말 URL을 정의한다.
- Id: 룰셋의 ID를 정의한다.
- Title : 룰셋의 이름을 정의한다.
- RulesetName : 품질검토 보고서 표지 이름을 정의한다.
- RulesetUpdateDate : 룰셋을 수정하거나 업데이트한 날짜를 정의한다.

● ShowSearch : 품질검토 목록 창에 검색창 표시 여부를 정의한다. 기본값은 , false(거짓)이 다.

#### <CcSetting>

<Editable>true</Editable> <ShowTypology>false</ShowTypology> <InfoUrl>http://www.inno-lab.co.kr/KBimAssess/RegulationCode.htm</InfoUrl> <Id> TEST </Id> <Id> TEST </Id> <Title>품질검토 목록</Title> <RulesetName>품질검토 보고서</RulesetName> <RulesetUpdateDate>2022-11-23</RulesetUpdateDate> <ShowSearch>false</ShowSearch> </CcSetting>

2.2 <CcCode>



<CcCode>는 검토항목을 정의한다.

요소

- Id: 검토 항목 ID 값을 정의한다.
- Parentld: 검토 항목의 부모 항목 ID 값을 정의한다. 검토 항목의 부모 항목이 없는 경우

ID값은 '-1'이다.

- Number: 검토 항목의 이름을 정의한다.
- IsGroup: 검토 항목이 자식 항목을 가지고 있는지 유무를 표시한다.
- Desc: 검토 항목에 대해 설명한다.
- Script: 검토 항목 로직을 정의한다. 항목들의 검토순서, 사전 정의 요소(공간분류코드, 조 달청 세부 공종 코드 등)들을 파이썬 코드로 작성한다. 자세한 스크립트 작성 방법은 <u>3.</u>
   스크립트 작성 방법을 참고한다.

✓ Pre-check on BIM Model
□ 공간객체의 유무
첨부된 이미지의 항목들을 룰셋으로 정의한 예시이다.
<cccode></cccode>
<id>1</id>
<parentid>-1</parentid>
<number>Pre-check on BIM Model</number>
<isgroup>true</isgroup>
#해당 항목은 부모 항목이 없고 자식 항목만 있으므로 IsGroup 요소가 true이며 검토항목에 따로 설명이나 로직이 없
어 Desc와 Script는 작성하지 않았다. 보통 목차의 성격을 가진 항목일 경우 위와 같이 작성을 한다.
<cccode></cccode>
<ld>1001</ld>
<parentid>1</parentid>
<number>공간객체의 유무</number>
<desc>모든 영역에 공간객체가 모델링 되어 있어야 한다.</desc>
<script></td></tr><tr><td><![CDATA[def Check():</td></tr><tr><td>for storey in SELECT('storey'):</td></tr><tr><td>undef_spaces = storey.SELECT('undefined space')</td></tr><tr><td>if undef_spaces.COUNT() == 0:</td></tr><tr><td>storey.SUCCESS('모든 영역에 공간객체가 모델링 되어 있습니다.')</td></tr><tr><td>else:</td></tr><tr><td>for undefined_space in undef_spaces:</td></tr><tr><td>area = undefined_space.SELECT('area').UNIT('m2').NUMBER()</td></tr><tr><td>undefined_space.ERROR('공간객체 없음 (면적: ' + str(area) + 'm2)')]]></td></tr><tr><td></script>
#해당 항목은 실질적으로 검토를 하는 항목이므로 Desc와 Script를 작성하였다.
#lsGroup의 경우 기본 값이 false이므로 부모 항목이 아닌 경우 작성하지 않는다.

## 3. 스크립트 작성 방법

스크립트는 python으로 작성하고, python 버전은 2.7을 기준으로 한다.

메인 함수명은 'Check()'로 한다.

<scrip< th=""><th>ot&gt;</th><th></th><th></th><th></th><th></th></scrip<>	ot>				
	def</td><td><mark>Check():</mark></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td>Print('룰</td><td>셋</td><td>작성</td><td>가이드')</td></tr><tr><td></td><td>				
<td>ript&gt;</td> <td></td> <td></td> <td></td> <td></td>	ript>				

BIM 모델 정보를 가져오기 위해서는 API 함수를 이용한다.

API 함수에 대한 자세한 설명은 <u>4. API 함수</u>를 참고한다.

<script></th></tr><tr><td><![CDATA[def Check():</td></tr><tr><td>for storey in <mark>SELECT</mark>('storey'):</td></tr><tr><td>undef_spaces = storey.<mark>SELECT</mark>('undefined space')</td></tr><tr><td>if undef_spaces.<mark>COUNT()</mark> == 0:</td></tr><tr><td>storey.<mark>SUCCESS</mark>('모든 영역에 공간객체가 모델링 되어 있습니다.')</td></tr><tr><td>else:</td></tr><tr><td>for undefined_space in undef_spaces:</td></tr><tr><td>area = undefined_space.SELECT('area').<mark>UNIT('m2').</mark>NUMBER()</td></tr><tr><td>undefined_space<mark>.ERROR</mark> ('공간객체 없음 (면적: ' + str(area) + 'm2)')]]></td></tr><tr><td></script>
--

## 4. API 함수

BIM 모델 정보를 가져오기 위해서는 다음과 같은 API 함수를 이용한다.

• SELECT(string query, object arg = null)

SELECT은 공간정보(storey, space), 객체(element, floor, beam, door, window), 객체 속성정 보(name, width, height, length, area, prop) 등을 가져올 수 있는 함수이다.

parameter	가져오는 BIM 모델 정보
storey	층 정보
space	공간 정보
undefined space	공간 객체가 정의되지 않은 공간 정보
element	객체 정보
floor	바닥
door	문
window	창문
name	객체 이름
width	객체의 width
height	객체의 height
length	객체의 length
area	객체의 area
class code	공간분류코드
element id	객체 아이디
	객체의 속성정보
prop	prop는 SELECT('prop', '불연성재료')와 같이 속성 이름을 함께 작
	성하면 해당 속성정보 값을 가져온다.

```
<Script>
<![CDATA[def Check():
for storey in SELECT('storey'):
#층 정보를 가져온다.
spaces = storey.SELECT('space')
#층에서 공간 정보를 가져온다.
for space in spaces:
wd = space.SELECT('window')
#공간에서 창문을 가져온다.
wdWidth = wd.SELECT('width').NUMBER()
#창문에서 너비 정보를 가져온다.
if wdWidth > 5:
```

#### wd.SUCCESS('창문의 너비가 조건을 만족합니다.')]]>

</Script>

```
#해당 보가 내화구조인지 검토하는 함수이다.
<Script>
<![CDATA[def Check():
for storey in SELECT('storey'):
#층 정보를 가져온다.
for beam in storey.SELECT('beam'):
#층에서 보를 가져온다.
if beam.SELECT('prop', '내화구조여부'). BOOL():
#층에서 보의 속성정보 중 '내화구조여부'를 가져온다.
beam.SUCCESS('해당 보는 내화구조입니다.')]]>
</Script>
```

• SUCCESS(string msg ="")

SUCCESS는 검토결과가 적합할 때 사용하는 함수이다.

Msg에 검토 결과 메시지를 넣어준다.

```
#모든 영역에 공간객체가 모델링이 되어있는지 검토하는 함수이다.
<Script>
 <![CDATA[def Check():
for storey in SELECT('storey'):
#층 정보를 가져온다.
 undef_spaces = storey.SELECT('undefined space')
 #층에서 공간 객체가 정의되어 있지 않은 공간정보를 가져온다.
 if undef_spaces.COUNT() == 0:
 #공간 객체가 정의되어 있지 않은 공간이 없으면
 #검토결과가 적합하다는 검토 결과를 보여준다.
 storey.SUCCESS('모든 영역에 공간객체가 모델링 되어 있습니다.')]]>
```

ERROR(string msg = "")

ERROR는 검토결과가 부적합할 때 사용하는 함수이다.

Msg에 검토 결과 메시지를 넣어준다.

```
#모든 영역에 공간객체가 모델링이 되어있는지 검토하는 함수이다.
<Script>
```

<![CDATA[def Check(): for storey in SELECT('storey'): # 층 정보를 가져온다. undef\_spaces = storey.SELECT('undefined space') #층에서 공간 객체가 정의되어 있지 않은 공간정보를 가져온다. if undef\_spaces.COUNT() > 0: #공간 객체가 정의되어 있지 않은 공간이 있으면 #검토결과가 부적합하다는 검토 결과를 보여준다. storey.ERROR('모든 영역에 공간객체가 모델링 되어있지 않습니다.')]]>

• COUNT()

COUNT는 주어진 범위의 데이터의 개수를 구하는 함수이다.

<Script> <![CDATA[def Check(): for storey in SELECT('storey'): undef\_spaces = storey.SELECT('undefined space') if undef\_spaces.<mark>COUNT()</mark> == 0: storey.SUCCESS('모든 영역에 공간객체가 모델링 되어 있습니다.')]]> </Script>

• UNIT(string unit)

UNIT은 기존의 단위(m)에서 입력된 단위(unit)로 변환을 하는 함수이다.

<script></th></tr><tr><td><![CDATA[def Check():</td></tr><tr><td>for storey in SELECT('storey'):</td></tr><tr><td>undef_spaces = storey.SELECT('undefined space')</td></tr><tr><td>for undefined_space in undef_spaces:</td></tr><tr><td>area = undefined_space.SELECT('area').<mark>UNIT('m2')</mark>.NUMBER()</td></tr><tr><td>#area는 단위가 m가 아니므로 UNIT함수를 이용하여 'm2'으로 단위를 변환시켜준다.</td></tr><tr><td>undefined_space.ERROR('공간객체 없음 (면적: ' + str(area) + 'm2)')]]></td></tr><tr><td></script>
---

• NUMBER(double def = 0), STRING(string def = null), BOOL(bool def = false)

NUMBER는 해당 값을 double 값으로, STRING은 string, BOOL은 bool 값으로 각각 변환 시켜준다.

```
#해당 공간이 최소 면적 기준을 통과하는지 검토하는 함수이다.
<Script>
<![CDATA[def Check():
for storey in SELECT('storey'):
spaces = storey.SELECT('space')
for space in spaces:
area = space.SELECT('area').UNIT('m2').NUMBER()
#area는 단위가 m가 아니므로 UNIT함수를 이용하여 'm2'으로 단위를 변환시켜준다.
#또한 NUMBER 함수를 이용하여 double 값으로 변환시켜 준다.
min_area = 300
If area < min_area:
space.SUCCESS('해당 공간이 최소 면적 기준을 통과하지 못했습
니다. ')]]>
```

\*이외의 함수는 '[품질검토] 내부함수.xlsx' 파일을 참고한다.

#### 5. 작성 예시

1. XML 편집기를 연다. 여기서는 Visual Studio Code를 사용했다.



2. 'testRuleset.ccl' 파일을 생성한다.



3. 모델에서 공간 객체를 가져와 공간분류코드 속성이 있는지 확인하는 룰을 작성해보자.

XML 구성에 맞춰 'CcSetting' 태그를 작성한다.

<?xml version="1.0" standalone="yes"?> <CcCodeDataSet xmlns="http://tempuri.org/CcCodeDataSet.xsd"> <CcSetting> <Editable>true</Editable> <ShowTypology>false</ShowTypology> <InfoUrl> </InfoUrl> <Id>TestRuleSet </Id> <Title>품질검토 테스트 룰셋 목록</Title> <RulesetName> </RulesetName> <RulesetUpdateDate> </RulesetUpdateDate> <ShowSearch>false</ShowSearch> </CcSetting> </CcCodeDataSet> 4. 'CcCode' 태그도 작성한다.

```
<?xml version="1.0" standalone="yes"?>
<CcCodeDataSet xmlns="http://tempuri.org/CcCodeDataSet.xsd">
   <CcSetting>
     <Editable>true</Editable>
     <ShowTypology>false</ShowTypology>
     <InfoUrl> </InfoUrl>
     <Id>TestRuleSet </Id>
     <Title>품질검토 테스트 룰셋 목록</Title>
     <RulesetName> </RulesetName>
     <RulesetUpdateDate> </RulesetUpdateDate>
     <ShowSearch>false</ShowSearch>
   </CcSetting>
   <CcCode>
       <Id>1001</Id>
       <ParentId>-1</ParentId>
       <Number>공간객체 속성 입력</Number>
       <Desc>공간 객체에는 공간분류코드 속성이 있어야 한다.</Desc>
       <Script> </Script>
    </CcCode>
</CcCodeDataSet>
```

#### <u>스크립트 작성 방법</u>을 참고하여 공간객체에 공간분류코드 속성이 있는지 확인하는 스크 립트를 작성한다.

```
<?xml version="1.0" standalone="yes"?>
<CcCodeDataSet xmlns="http://tempuri.org/CcCodeDataSet.xsd">
<CcSetting>
<Editable>true</Editable>
<ShowTypology>false</ShowTypology>
<InfoUrl> </InfoUrl>
<Id>TestRuleSet </Id>
<Title>품질검토 테스트 룰셋 목록</Title>
<RulesetName> </RulesetName>
<RulesetUpdateDate> </RulesetUpdateDate>
<ShowSearch>false</ShowSearch>
</CcSetting>
<CcCode>
<Id>1001</Id>
<ParentId>-1</ParentId>
```



6. 룰셋을 완성하면 'KBim Assess-Lite'에서 작성한 룰셋을 검토한다.

