

## 6. Point Object

### 6.1. Script Recording의 활용

다음을 이용하여 LUSAS 모델러에서 작업한 내용을 스크립트 형태로 기록할 수 있습니다.

1) 모델러에서 다음의 메뉴를 실행하여 스크립트 기록모드를 시작합니다.

(File / Script / Start recording.. )

2) 작업을 수행합니다.

모델러에서 (0,0,0), (10,0,0) 위치에 포인트를 정의합니다.

3) 스크립트 기록모드를 종료합니다.

(File / Script / Stop recording.. )

4) 작업폴더에서 저장된 스크립트 내용을 확인하고,

```
$ENGINE=VBScript
**** Create Line
Call geometryData.setAllDefaults()
call geometryData.setCreateMethod("straight")
call geometryData.addCoords(0.0, 0.0, 0.0)
call geometryData.addCoords(10.0, 0.0, 0.0)
call geometryData.setLowerOrderGeometryType("coordinates")
call database.createLine(geometryData)

**** Modify zoom/rotation
Call view.setScaleToFit(true)
```

일부 내용을 변수로 지정하여 저장합니다.

```
$ENGINE=VBScript
```

```
For i=1 to 10
    Ycoord = i*10
    **** Create Line
    Call geometryData.setAllDefaults()
    call geometryData.setCreateMethod("straight")
    call geometryData.addCoords(0.0, Ycoord, 0.0)
    call geometryData.addCoords(10.0, Ycoord, 0.0)
    call geometryData.setLowerOrderGeometryType("coordinates")
    call database.createLine(geometryData)
```

Next

```
**** Modify zoom/rotation
Call view.setScaleToFit(true)
```

5) 모델러를 초기화하고, 기록된 스크립트를 실행합니다.

(File / Script / Run Script...., 또는  아이콘 클릭하여 스크립트 실행)

## 6.2. Point의 정의

### 6.2.1. 기본형식

6.1을 참조하여 포인트 정의는 아래와 같이 사용할 수 있음을 알 수 있습니다.

행번	스크립트
1	\$ENGINE=VBSCRIPT
2	call geometryData.setAllDefaults()
3	call geometryData.addCoords(0.0, 0.0, 0.0)
4	call geometryData.addCoords(10.0, 0.0, 0.0)
5	call geometryData.setLowerOrderGeometryType("coordinates")
6	call database.createPoint(geometryData)
1	GeometryData 오브젝트의 속성 중에서 별도로 정의하지 않은 값을 기본값을 사용하도록 지정
2	GeometryData 오브젝트에 첫 번째 포인트 좌표 추가
3	GeometryData 오브젝트에 두 번째 포인트 좌표 입력 (정의할 포인트 수 만큼 반복 지정)
4	GeometryData에 입력된 값들의 의미를 지정
5	GeometryData 오브젝트에 입력된 값들로 포인트 정의

GeometryData는 Point/ Line/ Surface/ Volume 등의 Geometry를 생성하는데 필요한 정보를 저장하는 Object로, LUSAS 모델러가 내부적으로 가지고 있는 오브젝트이므로 별도로 정의할 필요 없이 그대로 활용하면 됩니다.

☞ Tip

[LUSAS설치폴더\ Programs\ LPI\Help\ IFGeometryData.html ]에 GeometryData의 속성값들이 정리되어 있습니다.

### 6.2.2. 예제

(0,0,0)으로부터 (10,0,0)까지 1 간격으로 10개의 포인트를 정의하고자 합니다.

#### ■ 스크립트 예

행번	스크립트
1	\$ENGINE=VBSCRIPT
2	call geometryData.setAllDefaults()
3	for xcord = 1 to 10
4	call geometryData.addCoords(xcord, 0.0, 0.0)
5	next
6	call geometryData.setLowerOrderGeometryType("coordinates")
7	call database.createPoint(geometryData)
1	GeometryData 오브젝트의 속성 중에서 별도로 정의하지 않은 값은 기본값을 사용하도록 지정
2	for 문으로 xcord라는 변수값이 1부터 10까지 변화하도록 정의
3	GeometryData 오브젝트에 정의할 포인트 좌표 입력 (xcord 변수를 x 좌표값으로 사용)
4	next
5	GeometryData에 입력된 값들의 의미를 지정
6	GeometryData 오브젝트에 입력된 값들로 포인트 정의합니다. database 역시 별도의 정의가 필요없는 모델러의 오브젝트로 현재의 모델파일을 의미합니다.

### 6.3. 정의된 Point Object 선택

#### 6.3.1. 지정된 1개 포인트를 오브젝트 변수로 지정

행번	스크립트
1	Set point = database.getObject("Point", 1)
1	모델러에 정의되어 있는 것 중 1번 포인트를 point라는 Object로 지정.

#### 6.3.2. 여러 개의 포인트를 배열변수로 지정

행번	스크립트
1	pointArray = database.getObjects("Point")
1	모델러에 정의되어 있는 모든 포인트 오브젝트를 pointArray라는 배열로 저장 즉, pointArray(0)은 모델러 상에 처음 정의된 포인트 오브젝트 pointArray(1)은 모델러 상에 두 번째 정의된 포인트 오브젝트

## 6.4. Point 관련 Method

### 6.4.1. getX(), getY(), getZ()

- 파라미터 : 없음
- 내용 : 포인트의 x 좌표값 (현재 모델러에서 Active되어 있는 지역좌표계가 있으면 지역좌표계에 의한 값)
- 출력값 : 포인트의 좌표값
- 스크립트 예 1 (모델러에 여러 개의 포인트를 정의한 후 테스트)

행번	스크립트
1	\$ENGINE=VBSCRIPT
2	set point = database.getObject("Point",1)
3	msgbox point.getX()
1	
2	모델러에 정의되어 있는 1번 포인트를 point 라는 오브젝트로 정의
3	1번 포인트의 X좌표값을 메시지창으로 표시

☞ Tip

point.getX(false) 또는 point.getX(0)라고 하면, 항상 글로벌 좌표값으로만 출력합니다.

- 스크립트 예 2 (모델러에 여러 개의 포인트를 정의한 후 테스트)

행번	스크립트
1	\$ENGINE=VBSCRIPT
2	pointArray = database.getObjects("Point")
3	For i=0 To UBound(PointArray)
4	MsgBox pointArray(i).getX()
5	Next
1	
2	모델러에 정의되어 모든 포인트를 pointArray 배열로 저장 for문으로 정의된 포인트 개수만큼 반복
3	(Ubound는 배열의 크기를 표시. 10개의 포인트가 있다면, 9라는 값을 가지므로, i는 0~9까지 변화)
4	포인트의 X좌표값을 순차적으로 메시지창으로 표시
5	next

### 6.4.2. getXYZ(x,y,z)

- **파라미터** : 포인트의 좌표값이 저장될 변수명 3개 임의 지정
- **내용** : 포인트의 x,y,z 좌표값을 x,y,z 변수에 저장
- **출력값** : 없음. (call을 사용하여 실행)
- **스크립트 예 1** (모델러에 여러 개의 포인트를 정의한 후 테스트)

행번	스크립트
1	\$ENGINE=VBSCRIPT
2	pointArray = database.getObjects("Point")
3	For i=0 To UBound(PointArray)
4	call pointArray(i).getXYZ(x1,y1,z1)
5	msgbox (i+1)&"번째 포인트 좌표는 ("&x1&","&y1&","&z1&") 입니다."
6	Next
1	모델러에 정의되어 모든 포인트를 pointArray 배열로 저장
2	for문으로 정의된 포인트 개수만큼 반복
3	(Ubound는 배열의 크기를 표시. 10개의 포인트가 있다면, 9라는 값을 가지므로, i는 0~9까지 변화)
4	포인트의 좌표값을 x1,y1,z1에 각각 저장
5	좌표값을 메시지창으로 표시
6	next

☞ **Tip**

getXYZ(x,y,z,false) 또는 getXYZ(x,y,z,0)을 사용하면 항상 글로벌 좌표값으로 출력됩니다.

### 6.4.3. getID()

- **파라미터** : 없음
- **내용** : 포인트의 일련번호를 확인
- **출력값** : 포인트의 일련번호
- **참고** : Geometry 관련 Object에 공통 사용

## 6.4.4. getName()

- **파라미터** : 없음
- **내용** : 포인트에 부여된 이름을 확인. 별도로 지정한 이름이 없으면 일련번호가 출력됨.
- **출력값** : 포인트에 부여된 이름
- **참고** : Geometry 관련 Object에 공통 사용  
getXYZ(x,y,z,false) 또는 getXYZ(x,y,z,0)을 사용하면 항상 글로벌 좌표값으로 출력
- **스크립트 예 1** (모델러에 여러 개의 포인트를 정의한 후 테스트)

행번	스크립트
1	\$ENGINE=VBSCRIPT
2	pointArray = database.getObjects("Point")
3	For i=0 To UBound(PointArray)
4	pointID = pointArray(i).getID()
5	pointName = pointArray(i).getName()
6	msgbox (i+1)&"번째 포인트의 번호는 "&pointID&"이고, 이름은 "&pointName&"입니다"
7	Next
1	모델러에 정의되어 모든 포인트를 pointArray 배열로 저장
2	for문으로 정의된 포인트 개수만큼 반복 (Ubound는 배열의 크기를 표시. 10개의 포인트가 있다면, 9라는 값을 가지므로, i는 0~9까지 변화)
3	포인트의 일련번호를 pointID 변수에 저장
4	포인트의 이름을 pointName 변수에 저장
5	포인트의 번호와 이름을 메시지창으로 표시
6	next

### 6.4.5. setName(newName)

- 파라미터 : 포인트/선/면/입체에 부여할 이름 (문자열 혹은 문자를 보관한 변수)
- 내용 : Object에 이름을 부여
- 출력값 : 없음. (call을 사용하여 실행)
- 참고 : Geometry 관련 Object에 공통 사용  
**이름의 첫문자는 숫자가 되지 않도록 지정.**
- 스크립트 예 1 (모델러에 여러 개의 포인트를 정의한 후 테스트)

행번	스크립트
1	\$ENGINE=VBSCRIPT
2	pointArray = database.getObjects("Point")
3	For i=0 To UBound(pointArray)
4	newName1 = "내 포인트"&i
5	Call pointArray(i).setName(newName1)
6	pointID = pointArray(i).getID()
7	pointName = pointArray(i).getName()
8	msgbox (i+1)&"번째 포인트의 번호는 "&pointID&"이고, 이름은 "&Chr(34)&pointName&Chr(34)&"입니다"
9	Next
1	모델러에 정의되어 모든 포인트를 pointArray 배열로 저장
2	for문으로 정의된 포인트 개수만큼 반복 (Ubound는 배열의 크기를 표시. 10개의 포인트가 있다면, 9라는 값을 가지므로, i는 0~9까지 변화)
3	포인트의 이름을 변수로 지정
4	포인트에 이름을 부여
5	포인트의 일련번호를 pointID 변수에 저장
6	포인트의 이름을 pointName 변수에 저장
7	포인트의 번호와 이름을 메시지창으로 표시
8	next

#### ☞ Tip

파라미터에 입력한 변수(문자)나 문자열을 제대로 인식하지 못하는 경우 다음과 같이 정의할 수 있습니다.

##### 1) 빈 문자열 삽입

Call pointArray(i).setName("'"&newName1)

##### 2) 문자열로 정의

Call pointArray(i).setName(CStr(newName1))

반대로, 숫자로 인식하지 못하는 경우에는 다음과 같이 정의할 수 있습니다.

##### 1) 값에 영향을 미치지 않는 숫자 연산 추가

Call pointArray(i).setName(newName1+0)

또는,

Call pointArray(i).setName(newName1\*1)

## 6.4.6. distanceFrom(obj)

- **파라미터** : 포인트 Object
- **내용** : Method를 실행하는 포인트 Object와 파라미터로 주어진 포인트 Object 사이의 거리
- **출력값** : 포인트 사이의 거리
- **스크립트 예 1** (모델러에 여러 개의 포인트를 정의한 후 테스트)

행번	스크립트
1	\$ENGINE=VBSCRIPT
2	pointArray = database.getObjects("Point")
3	For i=1 To UBound(PointArray)
4	Set point1 = pointArray(i-1)
5	Set point2 = pointArray(i)
6	distance = point2.distanceFrom(point1)
7	distance = round(distance,2)
8	msgbox i&"번째 포인트와 "&(i+1)&" 번째 포인트 거리는 "&distance&"입니다"
9	Next
1	모델러에 정의되어 모든 포인트를 pointArray 배열로 저장
2	for문으로 정의된 포인트 개수만큼 반복
3	(Ubound는 배열의 크기를 표시. 10개의 포인트가 있다면, 9라는 값을 가지므로, i는 0~9까지 변화)
4	포인트 Object 정의
5	포인트 Object 정의
6	두 포인트 사이의 거리를 distance 변수에 저장
7	거리값을 소수점 2번째 자리까지만 표시하도록 조정하여 distance 변수에 저장
8	포인트 거리를 표시하는 메시지 박스
9	next

### 6.5. 연습문제

모델러에 임의의 포인트를 10개 정의한 후에 실행시키면 아래와 같이 textWindow에 포인트 번호와 좌표값을 출력시키는 스크립트 작성

출력결과 : (아래 숫자는 정의한 포인트에 따라 달라질 수 있음)

1번 포인트 좌표 = 0, 0, 0

2번 포인트 좌표 = 2, 4, 5

...

10번 포인트 좌표 = 10, 30, 0

```
$ENGINE=VBSCRIPT

set textwindow=gettextwindow()
call geometryData.setAllDefaults()
for i=1 to 10 step 1
  call geometryData,addcoords(i,i+1,0)
Next

call geometryData.setLowerOrderGeometryType("Coordinates")
call database,createPoint(geometryData)
pointArray=database.getObjects("point")

for j=0 to UBound(pointArray)
  textwindow.writeline(j+1& "번 point 좌표 = "&pointArray(j).getX()&" "& pointArray(j).getY()& ,
  "&pointArray(j).getZ())
Next
```