

9. Surface Object

9.1. Surface의 정의

9.1.1. 좌표에 의한 Surface 정의

행번	스크립트
1	\$ENGINE=VBSCRIPT
2	call geometryData.setAllDefaults()
3	call geometryData.setCreateMethod("planar")
4	call geometryData.addCoords(0,0, 0,0, 0,0)
5	call geometryData.addCoords(6,0, 0,0, 0,0)
6	call geometryData.addCoords(6,0, 6,0, 0,0)
7	call geometryData.addCoords(0,0, 6,0, 0,0)
8	call geometryData.setLowerOrderGeometryType("coordinates")
9	call database.createSurface(geometryData)
1	
2	GeometryData Object의 속성 중에서 별도로 정의하지 않은 값은 기본값을 사용하도록 지정
3	GeometryData Object에 Surface의 종류는 평면으로 지정
4	GeometryData Object에 정의할 Line의 첫 번째 포인트 좌표 입력 (정의할 포인트 수 만큼 반복 지정)
5	GeometryData Object에 정의할 Line의 두 번째 포인트 좌표 입력 (정의할 포인트 수 만큼 반복 지정)
6	GeometryData Object에 정의할 Line의 세 번째 포인트 좌표 입력 (정의할 포인트 수 만큼 반복 지정)
7	GeometryData Object에 정의할 Line의 네 번째 포인트 좌표 입력 (정의할 포인트 수 만큼 반복 지정)
8	GeometryData 에 입력된 값들의 의미를 지정
9	GeometryData Object에 입력된 값들로 Surface 정의

- GeometryData는 Point/Line/Surface/Volume 등의 Geometry를 생성하는데 필요한 정보를 저장하는 Object로, LUSAS 모델러가 내부적으로 가지고 있는 Object이므로 별도로 정의할 필요 없이 그대로 활용하면 됩니다.
- 삼각형이나 오각형 등의 Surface를 작성하고자 하면, 3~6행과 같은 addCoords를 가감하면 됩니다.
- Point를 정의할 때 시계방향 또는 시계반대방향으로 일관되게 정의해야 합니다. Point를 한 방향으로 정의하지 않으면 surface가 구성되지 않으므로 주의하여야 합니다.

Tip

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

9.1.2. 정의되어 있는 포인트를 활용한 Surface 정의

먼저 모델러 상에서 3개 이상의 포인트가 정의되어 있을 때, 아래와 같이 사용할 수 있습니다.

행번	스크립트
1	\$ENGINE=VBSCRIPT
2	call selection.remove("All")
3	call selection.add("Point", "1")
4	call selection.add("Point", "2")
5	call selection.add("Point", "3")
6	call selection.add("Point", "4")
7	call geometryData.setAllDefaults()
8	call geometryData.setLowerOrderGeometryType("points")
9	call selection.createSurface(geometryData)
1	
2	모델러 상에 아무것도 선택된 것이 없도록 초기화
3	모델러 상에서 1번 포인트를 선택된 목록에 추가
4	모델러 상에서 2번 포인트를 선택된 목록에 추가.
5	모델러 상에서 3번 포인트를 선택된 목록에 추가
6	모델러 상에서 4번 포인트를 선택된 목록에 추가.
7	GeometryData Object의 속성 중에서 별도로 정의하지 않은 값은 기본값을 사용하도록 지정
8	GeometryData Object에 선택된 포인트를 활용하여 작업수행할 것을 지정.
9	GeometryData Object에 입력된 값들로 Surface 정의

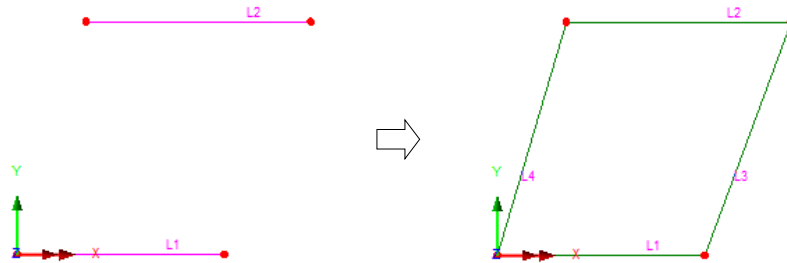
- GeometryData는 Point/Line/Surface/Volume 등의 Geometry를 생성하는데 필요한 정보를 저장하는 Object로, LUSAS 모델러가 내부적으로 가지고 있는 Object이므로 별도로 정의할 필요 없이 그대로 활용하면 됩니다.
- selection 은 모델러 상에 사용자의 마우스 클릭 등으로 선택된 상태로 있는 Object를 의미합니다.
- 포인트를 선택할 때, 포인트 번호 대신 포인트에 부여된 이름을 사용해도 됩니다.

Tip

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

9.1.3. 대응되는 2개의 Line을 이용한 Surface 정의

먼저 모델러 상에서 2개의 마주보는 Line이 정의되어 있을 때, 이 Line들을 묶는 Surface를 아래와 같이 정의할 수 있습니다.



행번	스크립트
1	\$ENGINE=VBSCRIPT
2	call selection.remove("All")
3	call selection.add("Line", "1")
4	call selection.add("Line", "2")
5	call geometryData.setAllDefaults()
6	call geometryData.setMaximumDimension(2)
7	set returnedSetA = selection.join(geometryData, selectionMem)
8	call database.updateMesh()
1	
2	모델러 상에 아무것도 선택된 것이 없도록 초기화
3	모델러 상에서 1번 Line을 선택된 목록에 추가
4	모델러 상에서 2번 Line을 선택된 목록에 추가.
5~6	GeometryData Object의 속성 중에서 별도로 정의하지 않은 값은 기본값을 사용하도록 지정
7	Join기법에 의한 상위 Geometry 정의
8	이 작업으로 요소 구성이 변화될 수 있다면 요소망 구성 업데이트 실행

- Line/Surface/Volume 등을 정의할 때에는 Script Recording 기능을 사용하여 쉽게 명령문 조합을 확인할 수 있으므로, 위 예문을 직접 타이핑하는 것 보다 유용합니다.

9.1.4. 예제

아래 변수를 이용하여 변의 길이(length)가 6인 Surface를 가로(cols) 5개, 세로(rows) 5개가 생성하는 스크립트를 작성하되, 9.1.1을 활용합니다.

cols=5, rows=5, length=6

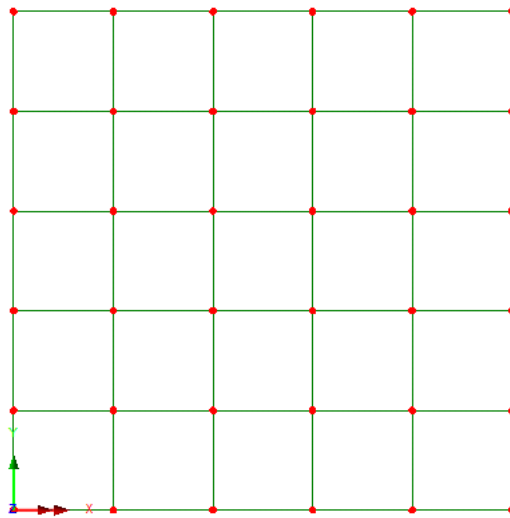
Tip

1) 배열을 정의하는 방법

dim a

a = array(0,1,2,3)

이와 같이 하면 0,1,2,3 을 각각 a(0),a(1),a(2),a(3) 의 값으로 갖는 배열변수 a가 정의됩니다.



\$ENGINE=VBSCRIPT

```
cols=5
rows=5
length=6
```

```
dim a
a=array(0,length,0,length)
```

```
for j=0 To (cols-1)*length step length
  for i=0 To (rows-1)*length step length
    call geometryData.setAllDefaults()
    call geometryData.setCreateMethod("planar")
    call geometryData.addCoords(a(0)+j, a(2)+i, 0,0)
    call geometryData.addCoords(a(1)+j, a(2)+i, 0,0)
    call geometryData.addCoords(a(1)+j, a(3)+i, 0,0)
    call geometryData.addCoords(a(0)+j, a(3)+i, 0,0)
    call geometryData.setLowerOrderGeometryType("coordinates")
    call database.createSurface(geometryData)
  Next
Next
```

Function(부함수)을 사용하여 다음과 같이 정의할 수 있습니다. 부함수는 스크립트 내에서 동일 작업을 반복적으로 수행할 때 유용하게 사용됩니다.

```
$ENGINE=VBSCRIPT

cols=5
rows=5
length=6

dim Xcoord
For i=1 to cols
  X1=(i-1)*length
  X2=i*length
  Xcoord = array(X1,X2,X2,X1)

  dim Ycoord
  For j=1 to rows
    Y1=(j-1)*length
    Y2=j*length
    Ycoord = array(Y1,Y1,Y2,Y2)

    dim Zcoord
    Zcoord = array(0,0,0,0)

    Call CreateSurface(Xcoord,Ycoord,Zcoord)
  Next
Next

Function CreateSurface(Xarray,Yarray,Zarray)

  call geometryData.setAllDefaults()
  call geometryData.setCreateMethod("planar")
  call geometryData.addCoords(Xarray(0), Yarray(0),Zarray(0))
  call geometryData.addCoords(Xarray(1), Yarray(1),Zarray(1))
  call geometryData.addCoords(Xarray(2), Yarray(2),Zarray(2))
  call geometryData.addCoords(Xarray(3), Yarray(3),Zarray(3))
  call geometryData.setLowerOrderGeometryType("coordinates")
  call database.createSurface(geometryData)

End Function
```

9.2. 정의된 Surface Object 선택

9.2.1. 지정된 1개 Surface을 Object 변수로 지정

행번	스크립트
1	Set Surface = database.getObject("Surface",1)
1	모델러에 정의되어 있는 것 중 1번 Surface를 Surface 이라는 Object로 지정.

9.2.2. 여러 개의 Surface들을 배열변수로 지정

행번	스크립트
1	SurfaceArray = database.getObjects("Surface")
1	모델러에 정의되어 있는 모든 Surface Object를 SurfaceArray 라는 배열로 저장 즉, SurfaceArray(0)은 모델러 상에 처음 정의된 Surface Object SurfaceArray(1)은 모델러 상에 두 번째 정의된 Surface Object

■ 참고

- selection.getObjects("Surface") 으로 하면, 모델러 상에서 선택되어져 있는 Object 중에서 Surface Object만을 가져오게 됩니다. selection 은 별도의 정의가 필요없이 모델러 상에서 선택한 상태에 있는 모든 것을 의미합니다.

9.3. Surface 관련 Method

9.3.1. getArea()

- 파라미터 : 없음
- 내용 : surface의 면적 출력
- 출력값 : surface의 면적 출력
- 스크립트 예 1 (모델러에서 surface 1개를 정의한 후 테스트)

행번	스크립트
1	\$ENGINE=VBSCRIPT
2	Set surface = database.getObject("surface", 1)
3	surfacearea = surface.getArea()
4	msgbox "the surface area is"&"("&surfacearea&")"
1	
2	모델러에 정의되어 있는 1번 surface를 surface 라는 Object로 정의
3	1번 Surface의 면적을 surfacearea라는 배열변수에 저장.
4	결과를 메시지 창에 출력

9.3.2. getSurfaceTypeCode()

- 파라미터 : 없음
- 내용 : surface의 종류 출력
- 출력값 : surface의 종류를 정수로 출력

0: Affine (Plannar)	3: Conical	6: Web	9: IsoParametric
1: Cylindrical	4: NURB	7: TransSweep	10: Offset
2: Spherical	5: Coon	8: RotSweep	11: Ruled

- 스크립트 예 1 (모델러에서 surface 1개를 정의한 후 테스트)

행번	스크립트
1	\$ENGINE=VBSCRIPT
2	Set surface = database.getObject("surface", 1)
3	surfacetype = surface.getSurfaceTypeCode()
4	
5	If (surfacetype=0) Then
6	msgbox "the surface type is Affine (Plannar)"
7	Elseif (surfacetype =1) Then
8	msgbox "the surface type is Cylindrical"
9	Elseif (surfacetype =2) Then
10	msgbox "the surface type is Spherical"
11	Elseif (surfacetype =3) Then
12	msgbox "the surface type is Conical"
13	Elseif (surfacetype =4) Then
14	msgbox "the surface type is NURB"
15	Elseif (surfacetype =5) Then
16	msgbox "the surface type is Coon"
17	Elseif (surfacetype =6) Then
18	msgbox "the surface type is Web"
19	Elseif (surfacetype =7) Then
20	msgbox "the surface type is TransSweep"
21	Elseif (surfacetype =8) Then
22	msgbox "the surface type is RotSweep"
23	Elseif (surfacetype =9) Then
24	msgbox "the surface type is IsoParametric"
25	Elseif (surfacetype =10) Then
26	msgbox "the surface type is Offset"
27	Elseif (surfacetype =11) Then
28	msgbox "the surface type is Ruled"
29	End If
1	
2	모델러에 정의되어 있는 1번 surface를 surface 라는 Object로 정의
3	1번 Surface의 종류를 surfacetype라는 배열변수에 저장.
4	결과를 메시지 창에 출력
5~29	정수형태로 출력되는 결과에 따라 해당 surface 종류의 이름으로 출력