|  |  |  |  |
| --- | --- | --- | --- |
|  | 2018-10-13 |  | |
|  | |  | |
| [3D 파일 형식 조사]  *컴퓨터 그래픽스* | | | |
|  |  | |  |
|  |  | | 소프트웨어학과 |
|  |  | | 2014041030 |
|  |  | | 손광호 |

내용

[1. STL 2](#_Toc527211018)

[1. 파일 구조 2](#_Toc527211019)

[2. 예제 4](#_Toc527211020)

[2. OBJ 5](#_Toc527211021)

[1. 파일 구조 5](#_Toc527211022)

[2. 예제 8](#_Toc527211023)

[3. FBX 9](#_Toc527211024)

[1. 파일 구조 9](#_Toc527211025)

[2. 예제 11](#_Toc527211026)

[4. AMF 12](#_Toc527211027)

[1. 파일 구조 12](#_Toc527211028)

[2. 예제 13](#_Toc527211029)

[5. X3D 14](#_Toc527211030)

[1. 파일 구조 15](#_Toc527211031)

[2. 예제 16](#_Toc527211032)

[6. 예제 및 Source Code 16](#_Toc527211033)

# 1. STL

3D Systems에서 제작한 광조형 CAD 소프트웨어의 고유한 파일 형식.

STL에는 “표준 삼각 언어” 및 “표준 테셀링 언어”와 같은 여러 개의 after-the-fact backronyms가 있다.

이 파일 형식은 다른 많은 소프트웨어 패키지에서 지원된다. 시제품 제작, 3D 인쇄 및 컴퓨터 보조 제조에 널리 사용된다. STL 파일은 색상, 텍스처 또는 기타 일반적인 CAD 모델 특성을 나타내지 않고 3차원 객체의 표면 형상만 설명한다. ASCII 및 이진 표현을 모두 지정하나 이진 파일이 더 작기 때문에 이진 파일이 더 일반적으로 사용된다.

STL 파일은 3차원 직교 좌표계를 사용하여 삼각형의 단위 법선과 꼭지점(오른손 법칙에 의해 정렬된)에 의해 구조화되지 않은 삼각형 표면을 나타낸다. 즉, 3차원 형상을 무수히 많은 삼각형 면으로 구성하여 표현해 주는 일종의 폴리곤 포멧이기 대문에 삼각형의 크기가 작을수록(또는 개수가 많을수록) 고품질의 출력물 표면을 얻을 수 있다. 이때 규칙이 존재하는데 서로 인접한 삼각형은 정확히 두개의 꼭지점을 공유해야 한다. 원래 모든 STL 좌표는 양수가 요구되었지만 더 이상 이 제한은 강요되지 않으며 오늘날의 STL 파일에서는 흔히 음수 좌표가 나타난다. 또한 STL 파일에는 스케일 정보가 없고 단위가 임의적이다.

### 1. 파일 구조

#### 가. ASCII STL

soild name

facet normal

outer loop

vertex

vertex

vertex

endloop

endfacet

endsolid name

여기서 name은 3D 모델의 이름이다. 이름은 생략할 수 있으나 공백으로 존재해야 한다.

facet ~ endfacet은 삼각형에 대한 정보이며 한번 이상 반복된다. n은 삼각형의 법선이고 v1, v2, v3는 삼각형의 꼭지점이다. 좌표 값은 가수-e-지수 형식의 부동 소수점 수(ex: 3.124000e-002)로 표현된다.

공백(space, tab, newline)은 숫자, 단어를 제외한 모든 곳에서 사용이 가능하다. facet과 normal 사이, outer과 loop 사이는 반드시 띄워야 한다.

#### 나. Binary STL

UINT8[80] – Header

UNIT32 – Number of triangles

foreach triangle

REAL32[3] – Normal vector

REAL32[3] – Vertex 1

REAL32[3] – Vertex 2

REAL32[3] – Vertex 3

UINT16 – Attribute byte count

end

이진 STL 파일은 80자 헤더로 시작한다. 대개 대부분의 STL 파일 판독기에서 무시되며 일반적으로 solid로 시작해서는 안된다. ASCII STL 파일로 인식될 수 있기 때문이다. 헤더 뒤의 총 삼각형 수는 부호 없는 4바이트 정수를 사용하여 표시된다.

이후는 삼각형에 대한 정보이다. 각 삼각형은 12개의 32비트 부동 소수점 숫자로 표현된다. ASCII STL 파일과 마찬가지로 처음 3개의 숫자는 삼각형의 법선에 대한 좌표이다. 나머지 9개의 숫자는 각 꼭지점의 좌표이다.

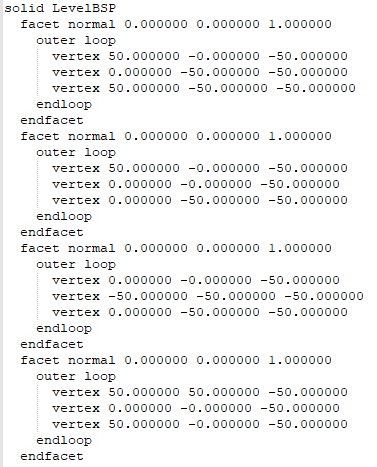
꼭지점 다음 2바이트의 ‘속성 바이트 수’가 있다. 대부분의 경우 이 값은 0으로 설정되어 2개의 삼각형 사이에 공백으로 작동한다. 일부 소프트웨어는 이것을 사용하여 삼각형의 추가 정보(ex: 색상)를 인코딩한다.

### 2. 예제

UE4로 cube 생성 후 export

#### ASCII STL

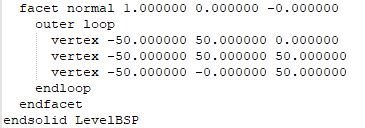
파일 크기: 10.27KB

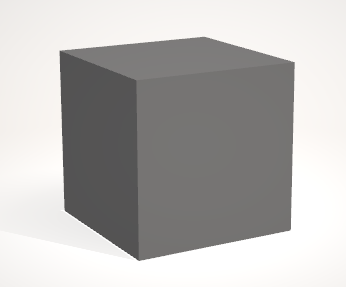


.

.

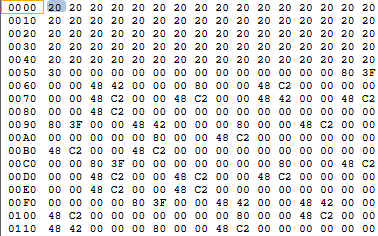
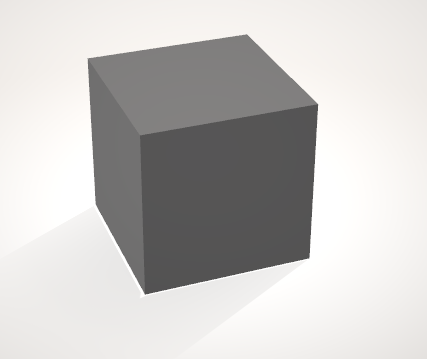
.





#### 나. Binary STL

파일 크기: 2.43KB



# 2. OBJ

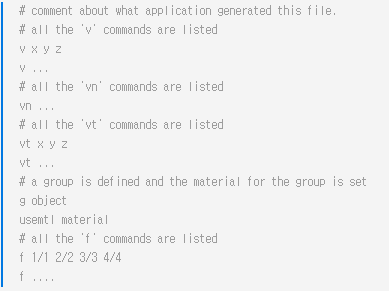
Advanced Visualizer 애니메이션 패키지를 위해 Wavefront Technologies에서 처음 개발한 지오메트리 정의 파일 형식. OBJ 파일 형식은 3D 지오메트리만을 나타내는 간단한 데이터 형식.

OBJ 파일 형식은 오픈 소스이며 중립적이다.

ASCII 형태로 데이터를 저장할 수도 있고 binary 형식으로 저장할 수도 있다. ASCII 형태인 OBJ로 저장을 하더라도 그렇게 크기가 커지지 않으며 MTL이라는 별도의 material 파일을 사용하기 때문에 모델 정보 이외에 texture map이나 material에 대한 데이터를 함께 옮길 때에는 관련 MTL 데이터도 찾아 같이 옮겨야 한다.

### 1. 파일 구조

#### 가. OBJ



# comment line

-> #은 해당 줄이 주석임을 나타낸다. 첫 번째 행은 보통 파일을 생성한 프로그램을 지정하는 주석이다.

v x y z

-> 3개의 좌표로 정점을 지정한다. 정점은 파일에서 발견된 순서에 따라 암시적으로 이름이 지정된다. 예를 들어 첫번째 정점은 ‘1’, 두번째 정점은 ‘2’가 되는 식이다. 일부 응용 프로그램은 y 다음에 rgb 값을 넣어 정점의 색상을 지원하기도 한다. 이때, 색상 값의 범위는 0에서 1 사이이다.

vt u v [w]

-> vertex texture 커맨드는 텍스처 맵의 포인트를 지정한다. u와 v는 텍스처 맵의 x와 y좌표이다. 값의 범위는 0에서 1사이이다. 실질적인 의미를 가지고 있지 않으며 f 커맨드에서 꼭짓점으로 그룹화 되어야한다.

vn x y z

-> vertex normal 커맨드는 법선 벡터를 지정한다. x, y, z는 법선 벡터의 성분이다. 아직 어떤 정점에도 연결되어 있지 않으며, vt 명령과 마찬가지로 f 커맨드를 사용하여 정점을 그룹화해야 한다. f 커맨드로 정점을 다각형면으로 그룹화 할 때, 정점 좌표로부터의 법선 벡터와 정점이 나타나는 순서를 자동으로 결정하기 때문에 vertex normal 커맨드는 많은 파일에서 생략된다.

f v1[/vt1][/vn1] v2[/vt2][/vn2] v3[/vt3][/vn3] …

-> face 커맨드는 나열된 정정 리스트로 만들어진 다각형을 지정한다. 원하는 만큼의 정점을 가질 수 있다. 예로 f 1 2 3 4는 정점 1~4에서 생성된 면을 의미한다. 각 꼭짓점에 대해 vn을 연관시킬 수 있다. 마찬가지로 vt를 정점에 연결할 수도 있다. 이때, vt는 텍스처 매핑을 결정한다. vt, vn을 지정하면 모든 정점에 대해 지정해야 한다.

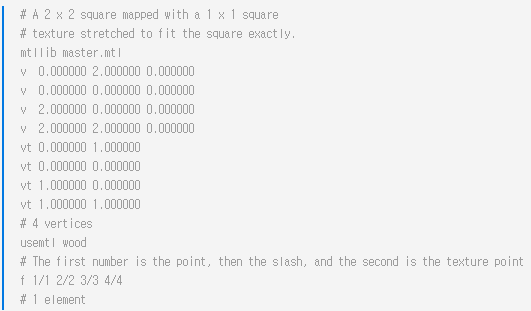
g name

-> group name 커맨드는 하위 오브젝트 그룹을 지정한다. 뒤따르는 f 커맨드들은 모두 같은 그룹에 있는 것으로 간주된다. 재질 유형과 같은 객체의 특정 정보를 다시 사용하려는 경우 유용하다.

usemtl name

-> use material 커맨드를 사용하여 사용할 material의 이름을 지정할 수 있다. 뒤따르는 모든 f 커맨드들은 다른 usemtl 명령이 나올 때까지 동일한 material을 사용한다.

#### 나. MTL



mtlib name

-> MTL 파일의 이름이다. mtlib 명령은 usemtl 명령 앞에 와야한다.

newmtl name

-> 재료의 특성을 정의한다.

Ka R G B

-> 재료의 주변 색을 정의한다. 각 값의 범위는 0~1이다.

Kd R G B

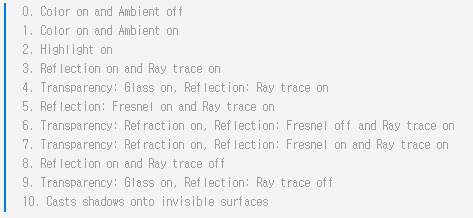
-> 재료의 확산 색을 정의한다.

d value

-> 재료의 투명도를 지정한다. 값의 범위는 0(투명)~1(불투명)이다.

illum value

-> 조명 모드를 지정한다. 선택 사항은 다음과 같다.

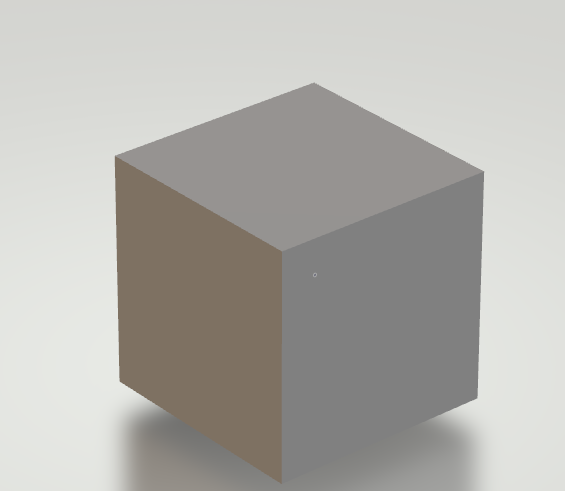


map\_Ka square.tga

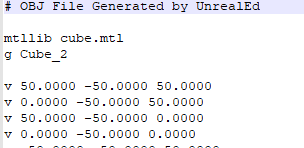
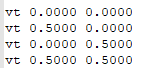
-> 모든 속성을 텍스처 맵을 통해 선언할 수 있다. 이 구문에서는 확장자가 .tga인 별도의 텍스처 맵이다.

### 2. 예제

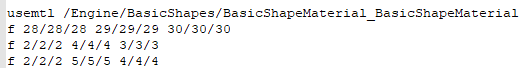
UE4로 cube 생성 후 export



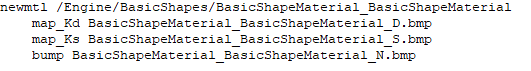
#### 가. OBJ







#### 나. MTL



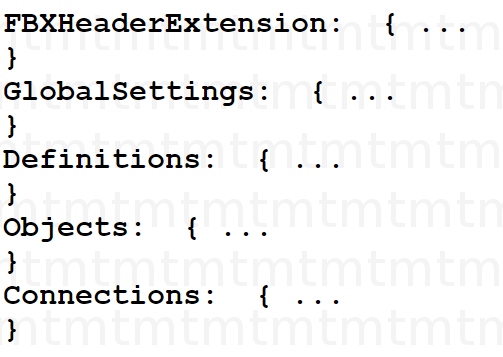
# 3. FBX

Kaydara가 개발하고 2006년부터 Autodesk가 소유한 독점적인 파일형식이다. 이 파일 형식은 디지털 컨텐츠 제작 응용 프로그램 간의 상호 운용성을 제공하는데 사용된다. 또한 비디오 게임 미들웨어 시리리즈인 Autodesk Gameware의 일부이다.

정점, Indices, Normals, UV cords, Materials, 애니메이션을 지원한다. 다른 유형의 정보도 지원되지만 게임 엔진에는 전혀 사용되지 않는다.

형식은 ASCII 또는 2진 형식으로 되어있다.

### 1. 파일 구조



기본적으로 위와 같은 구조로 되어있다.

; comment line

-> 세미콜론은 주석을 나타낸다.

FBXHeaderExtension

-> 초기 주석 뒤에 있다. 파일 구조의 첫번째 주 노드이다. format 버전을 나타내는 두 개의 속성(FBXHeaderVersion, FBXVersion)을 갖고 있다. CreationTimeStamp라는 생성 날짜를 나타내는 속성도 가지고 있다.

Objects

-> 가장 중요한 노드로 모델의 정점, 인덱스, normal, UV 좌표 및 Material이 포함되어 있다.

Vertices: v1\_x, v1\_y, v1\_z, v2\_x, v2\_y, v2\_z, …

-> Model 서브 노드의 Vertices라는 속성에서 찾을 수 있다. 좌표는 십진법으로 표현된다.

PolygonVertexIndex: i1, i2, -i3, i4, i5, -i6

-> 3면체 또는 4면체를 익스포트할 수 있다. 다각형을 구성하는 인덱스는 순서대로 나타내어 지고 음의 인덱스는 마지막 인덱스를 의미한다. 후에 음수는 양수로 만들어야 하고 그 다음 1을 빼야한다(XOR). 일반적으로 3면체(삼각형)으로 익스포트한다.

Normals: n1\_x, n1\_y, n1\_z, n2\_x, n2\_y, n2\_z, …

-> LayerElementNormal 아래에서 찾을 수 있다. Vertices와 구조가 비슷하다. MappingInformationType 속성에 주의해야 한다.

• ByPolygon: 모델의 모든 polygon에 법선이 있음을 의미한다.

• ByPolygonVertex: 모델의 모든 polygon의 모든 정점에 법선이 있음을 의미한다.

• ByVertex: 모델의 모든 정점에 법선이 있음을 의미한다.

• ByEdge: 모델의 모든 가장자리에 법선이 있음을 의미한다.

• AllSame: 모델의 모든 정점에 대해 동일한 법선이 있음을 의미한다.

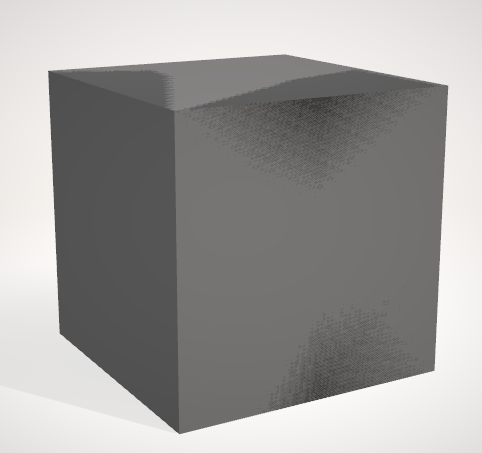
ReferenceInformationType 속성도 중요하다.

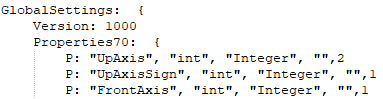
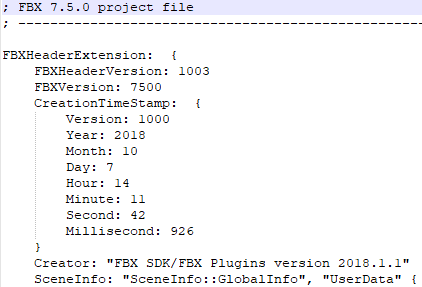
• Direct: 법선이 순서대로 부여된다.

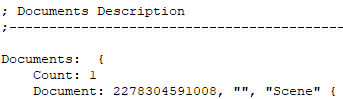
• IndexToDirect: 법선의 순서가 NormalsIndex 속성에 의해 부여된다.

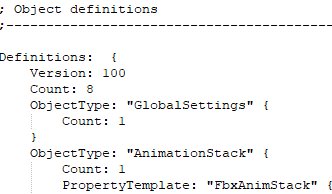
### 2. 예제

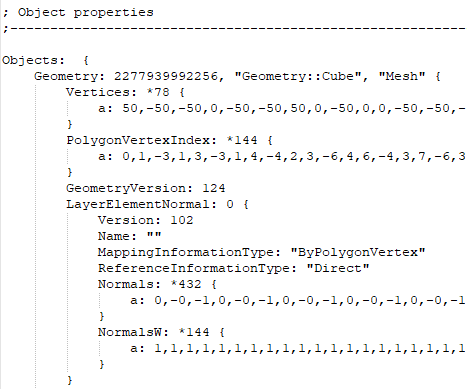
UE4로 cube 생성 후 export

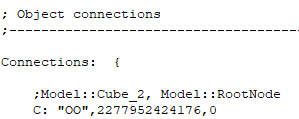


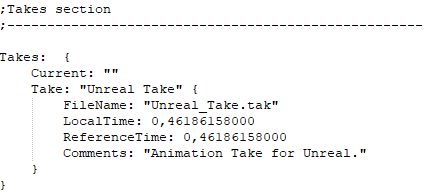












# 4. AMF

AMF는 XML 기반이고 텍스트 기반이다. CAD 프로그램에서 3D 인쇄용 개체를 묘사하는데 사용된다. 측정 단위, 곡선 패치, 통일성, 작은 파일 용량(STL 대비 50% 우수)라는 특징을 갖고 있다. STL 형식과 달리 색상(RGBA), material, 격자 및 constellation을 지원한다. 삼각형 메쉬를 사용하여 곡면을 만든다. 하지만 객체 생성 방법에 대한 정보를 제공하지는 않는다. 파일은 객체의 작성자, 이름, 저작권 등의 메타데이터를 저장하며 구조가 단순하고 확장이 가능하다.

### 1. 파일 구조

XML 버전 및 인코딩을 지정하는 선언 행으로 시작한다. 파일은 <amf>로 시작해 </amf>로 끝난다. 단위계도 지정할 수 있다. 없다면 밀리미터가 사용된다.

최상위 요소 5개가 존재한다.

<object>

-> 재료의 볼륨을 정의한다. 각 볼륨은 인쇄할 재료 ID와 연관된다. 최소 하나의 object가 있어야 하며 추가적인 object는 선택사항이다.

<material>

-> 선택사항인 material 요소는 한 개 이상을 정의한다. 각각은 material ID를 가지고 있다.

<texture>

-> 선택사항인 텍스처 요소는 색상 또는 매핑을 위한 하나 이상의 이미지나 텍스처를 정의한다. 각각은 texture ID를 가지고 있다.

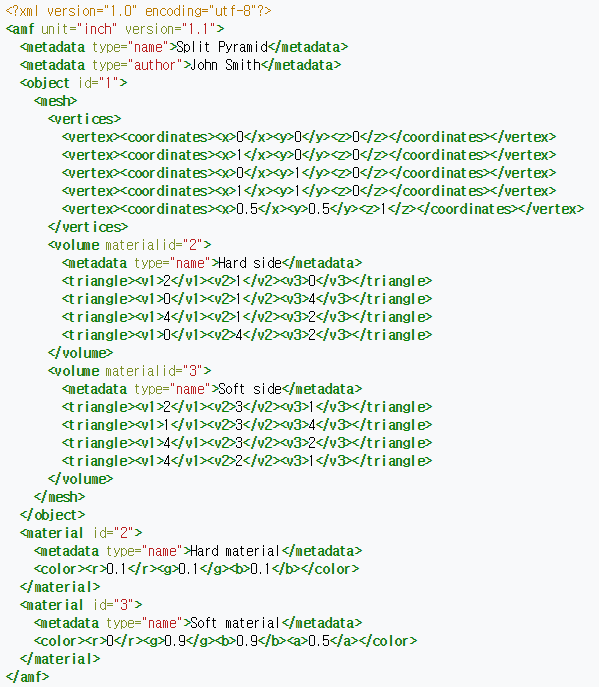
<constellation>

-> 선택사항인 constellation 요소는 인쇄에 있어 상대적인 패턴으로 object와 다른 constellation을 결합한다.

<metadata>

-> 선택사항인 메타데이터 요소는 파일에 포함된 object와 요소들에 대해 추가적인 정보를 설명한다.

### 2. 예제



# 5. X3D

VRML을 모델로 한 현재 웹 3D 파일 형식이며 XML 기반 형식이다. 파일을 보기 위해 웹 브라우저 플러그인의 독립형 뷰어가 필요하다. 응용 프로그램에 포함된 실시간 그래픽 콘텐츠를 저장, 검색 그리고 재생할 수 있는 시스템을 제공하여 광범위한 도메인 및 사용자 시나리오를 지원하는 ISO/IEC 국제 인증 표준이다. 공학 및 과학의 시각화, CAD 및 아키텍처, 3D 인쇄 및 스캐닝 등의 사용에 적합한 기능을 가지고 있다.

STL과 달리 색상 정보를 인코딩하기 때문에 컬러 3D 프린터로 모델을 인쇄할 수 있다는 장점이 있다. 또한 X3D 뷰어에 다운로드한 X3D 파일을 삽입함으로써 자신의 사이트 같은 다른 장소에 3D 모델을 보여줄 수 있다.

### 파일 구조



X3D 파일은 다음과 같이 이루어져 있다.

• 파일 헤더

• 버전과 프로필 속성을 포함하는 루트 노드의 시작

• 구성 요소와 메타 상태가 있는 헤드 섹션

• scene 그래프와 자식 노드

• 루트 노드의 끝

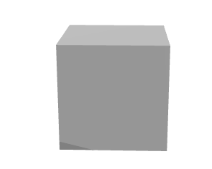
헤더에는 XML 선언과 선택적 DTD가 필요하다. XML 스키마는 버전 및 프로필 속성을 포함하는 루트 노드에서 선언된다. 또한 헤더에서 필요한 구성요소를 보다 정확하게 지정할 수 있다.

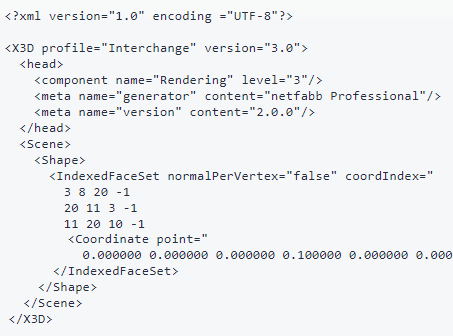


메타 문장은 X3D 장면 전체에 대한 정보를 제공한다.

각 X3D 파일은 XML 원칙에 따라 하나의 X3D 장면만 포함할 수 있다.

### 2. 예제





# 6. 예제 및 Source Code

본문에 사용한 예제와 각 Viewer 프로그램의 Source Code는 압축파일로 별도 첨부함