

일반, 중요, 불필요, 무관		
이름	Define	용도
*MATERIAL_LIST	ID_MATERIAL_LIST	머테리얼 목록 시작
*MATERIAL_COUNT	ID_MATERIAL_COUNT	머테리얼의 수 (쓰면 줄고 안써도 됨)
*MATERIAL %d	ID_MATERIAL	새 머테리얼의 시작
*MATERIAL_NAME	ID_MATNAME	머테리얼의 이름
*MATERIAL_AMBIENT	ID_AMBIENT	머테리얼의 엠비언트 (R, G, B) 밑에 3개다 같다
*MATERIAL_DIFFUSE	ID_DIFFUSE	머테리얼의 디퓨즈
*MATERIAL_SPECULAR	ID_SPECULAR	머테리얼의 스펙큘라
*MAP_DIFFUSE	ID_MAP_DIFFUSE	텍스처의 시작
*BITMAP	ID_BITMAP	텍스처의 경로 (경로 주소의 수정이 필요)
*MAP_BUMP	ID_MAP_BUMP	노말맵 (아직 이거 쓰지 않음)
지오메트리 파트		
*GEOMOBJECT	ID_GEOMETRY	새 지오메트리 시작
*NODE_NAME	ID_NODE_NAME	노드가 가질 이름
*NODE_PARENT	ID_NODE_PARENT	노드의 부모 (이 이름을 가진 부모의 자식으로 달면 됨)
*NODE_TM	ID_NODE_TM	노드의 트랜스폼 매트릭스 설정 시작
*TM_ROW0~3	ID_TM_ROW0~3	트랜스폼 매트릭스의 각각의 열 (2, 3번 가로줄을 바꾸고 2, 3번 세로줄을 바꿔줘야 함) ※ 순서가 2번 바뀐다 주의하자 (4번째 세로줄에 각각 0, 0, 0, 1이 들어감, Identity로 초기화 하고 채우면 됨)
*MESH	ID_MESH	메쉬 파싱의 시작
*MESH_NUMVERTEX	ID_MESH_NUMVERTEX	메쉬의 버텍스 수 (vecV의 총 갯수)
*MESH_NUMFACES	ID_MESH_NUMFACES	메쉬의 페이스의 수 우리페이스 1개 대신 3개의 버텍스가 생김 (vecVertex의 수 = 메쉬의 페이스의 수 * 3)
*MESH_VERTEX_LIST	ID_MESH_VERTEX_LIST	메쉬의 버텍스 리스트 *MESH_NUMVERTEX의 수만큼 존재

*MESH_VERTEX	ID_MESH_VERTEX	메쉬의 버텍스 (INDEX, X, Z, Y)
*MESH_FACE_LIST	ID_MESH_FACE_LIST	메쉬의 페이스 리스트
*MESH_FACE	ID_MESH_FACE	메쉬의 페이스 (INDEX, 0번, 2번, 1번 vecV의 INDEX)
*MESH_MTLID		메쉬의 머테리얼 아이디(?)로 추정 (안씀)
*MESH_NUMTVERTEX	ID_MESH_NUMTVERTEX	메쉬의 텍스처 버텍스의 수 (vecVT의 총 갯수)
*MESH_TVERTLIST	ID_MESH_TVERTLIST	메쉬의 텍스처 버텍스 리스트 *MESH_NUMTVERTEX의 수만큼 존재
*MESH_TVERT	ID_MESH_TVERT	메쉬의 텍스처 버텍스 (INDEX, X, Y, 미사용) *Y는 1.0f - Y로 쓸 것
*MESH_NUMTVFACES	ID_MESH_NUMTVFACES	메쉬의 텍스처 페이스의 수 *MESH_NUMFACES와 같다 (안써도 무방)
*MESH_TFACELIST	ID_MESH_TFACELIST	메쉬의 텍스처 페이스 리스트
*MESH_TFACE	ID_MESH_TFACE	메쉬의 텍스처 페이스 (INDEX, 0번, 2번, 1번 vecVT의 INDEX)
*MESH_NORMALS	ID_MESH_NORMALS	메쉬의 노말 벡터 리스트
*MESH_FACENORMAL	ID_MESH_FACENORMAL	메쉬의 페이스의 노말 벡터 (여기서 버텍스의 인덱스를 뺌으면 좋다)
*MESH_VERTEXNORMAL	ID_MESH_VERTEXNORMAL	메쉬의 버텍스의 노말 벡터 (미사용, X, Z, Y) (0번, 2번, 1번 인덱스로 각각의 버텍스마다 값을 넣는다) ※ 순서가 2번 바뀐다 주의하자
*MATERIAL_REF	ID_MATERIAL_REF	지오메트리가 쓸 머테리얼의 인덱스 번호
※ 기본 좌표에 월드 매트릭스가 적용 되어있음		
※ 출력시 월드 매트릭스가 아닌 아이덴티티로 초기화한 값을 써서 출력하면 정상적으로 나옴		
※ 파일의 좌표 순서는 x, z, y 순서로 기록되어 있어 파싱하면서 순서를 바꿔서 적어줄 필요가 있음		
※ 삼각형의 인덱스도 0, 2, 1번 순서로 기록되어 있음		
※ 텍스처의 y축이 뒤집혀있어 1.0f - y로 사용해야 다이렉트에서 정상적인 텍스처 좌표로 출력할 수 있다		

MESH의 LocalTM = NODE_TM_행렬 * (부모가 있으면 부모의 WorldTM의 역행렬)					
MESH의 Local좌표 =	Vertex.p에 NODE_TM_행렬의 역행렬로 Coord				초월적인 어휘력
	Vertex.n에는 Normal				
일단 MESH의 좌표계를 파싱을 마치면 적용되어있는 월드 좌표계를 로컬 좌표계로 바꿔주어야 한다					
이름	Define	용도			
*TM_ANIMATION	ID_TM_ANIMATION	TM 애니메이션 파싱 시작			
*NODE_NAME	ID_NODE_NAME	애니메이션이 들어갈 노드의 이름 ( <a href="#">넣고싶으면 넣자...</a> )			
*CONTROL_POS_TRACK	ID_POS_TRACK	이동 애니메이션 목록			
*CONTROL_POS_SAMPLE	ID_POS_SAMPLE	이동 애니메이션 (키프레임, X, Z, Y)			
*CONTROL_ROT_TRACK	ID_ROT_TRACK	회전 애니메이션 목록			
*CONTROL_ROT_SAMPLE	ID_ROT_SAMPLE	회전 애니메이션 (키프레임, X, Z, Y, W) x *= sin(w / 2) y *= sin(w / 2) z *= sin(w / 2) w = cos(w / 2) (회전 애니메이션은 이전 애니메이션에서 누적연산해서 저장해야 한다)			
벡터의 프론트와 백을 써서 제일 작은것보다 작은 경우 작은걸로 적용, 큰 경우 큰 걸로 적용					
로컬 TM의 계산					
회전 애니메이션 계산 =	로테이션 트랙이 비어있는 경우	회전 매트릭스 = 로컬TM			
		회전 매트릭스의 이동 좌표는 0으로 초기화			
	로테이션 트랙이 있는 경우	키프레임에 알맞는 선형 보간된 QUATERNION으로 회전 매트릭스 생성			
		D3DXQuaternionSlerp			
		D3DXMatrixRotationQuaternion			
이동 애니메이션 계산 =	포스 트랙이 비어있는 경우	로컬TM에서 이동 좌표만 가져와서 이동 매트릭스 생성			
	포스 트랙이 있는 경우	키프레임에 알맞는 선형 보간된 D3DVECTOR3로 이동 매트릭스 생성			
		D3DXVec3Lerp			
월드TM = 회전 매트릭스 * 이동 매트릭스 * (부모가 있다면 부모의 월드TM)					
출력에는 월드TM만 사용					