



3DML 开源说明书

泰瑞数创科技（北京）有限公司

2019 年 7 月

概述

3DML 格式是针对传输和浏览大量 3D 空间数据设计的，包括地形表面、3D 城市模型和 BIM/CAD。3DML 基于开源的 SQLite 数据库和扩展组件 SpatialLite，它定义了包括分层分级的模型数据、矢量信息和工程信息的数据库结构。

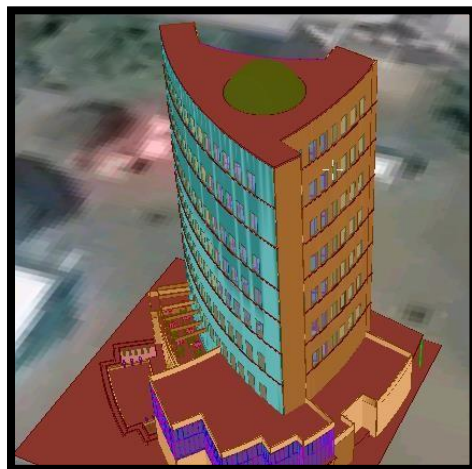
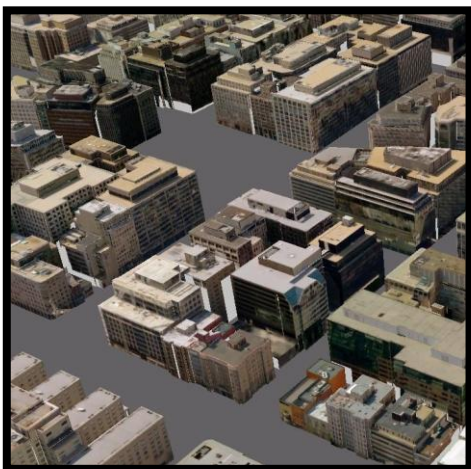
3DML 数据库所包括的相关的块数据格式以及相关的矢量信息都是开源的，都不依赖其他特殊厂商的解决方案、技术或者产品。

3DML 数据库可以用于本地 3D 模型资源或者在线的网络服务。

3DML 的主要作用

3DML 主要用于：

- 地形网格模型
- 城市网格模型
- 单个模型
- BIM 和 CAD 模型



3DML 文件格式

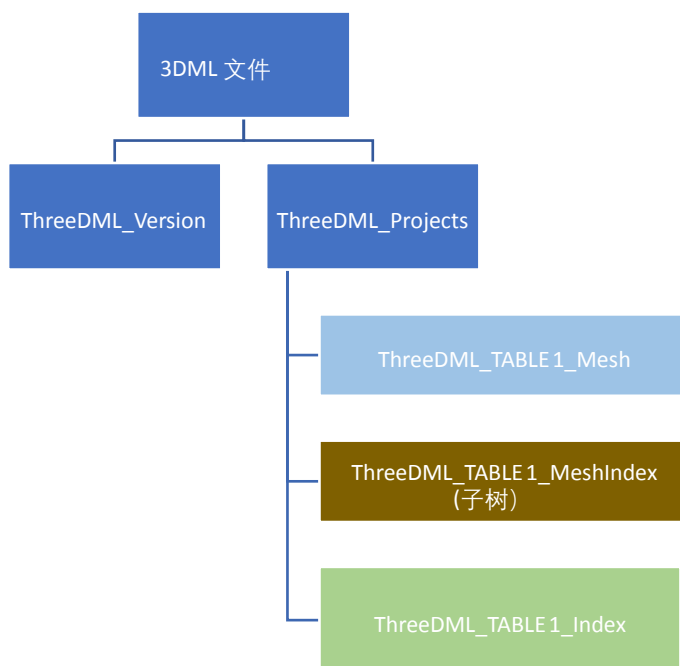
3DML 文件包含网格数据、矢量图层和元数据。不同分辨率的瓦片数据是按照树状结构进行组织，树上的每个节点都包含更多的子节点。每个节点和其父节点表示相同的信息，但是子节点的分辨率更高。

3DML 文件包含以下表格：

- Version 表-保存 3DML 版本信息。
- Projects 表-保存 3DML 数据集相关信息，包括：
 - BBox
 - 唯一 GUID
 - WKT 格式的数据集坐标系统
- Index 表-以子树形式存储网格模型的索引信息。
- mesh 表-包括网格模型的节点块数据，包括几何、贴图和地表。
- Feature 表-存储用于分类的面状图层的空间表（不在本文档中详细描述）。

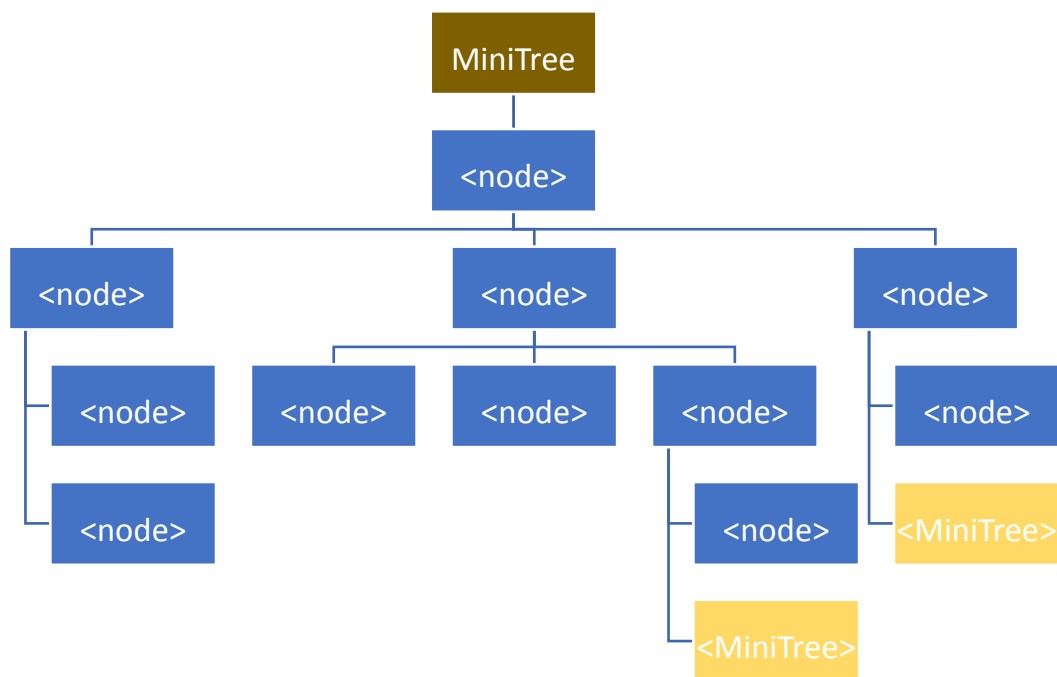
关于表格及表格内容更详细的信息请查看[表格](#)章节。

3DML 文件结构图:

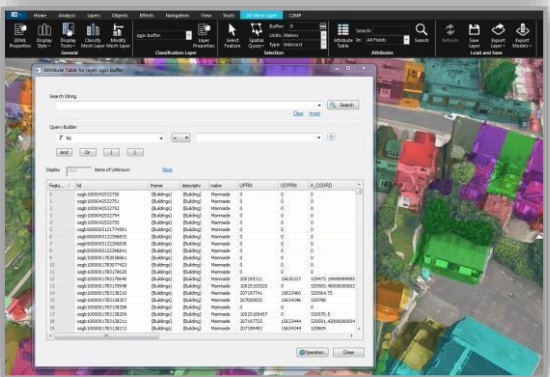
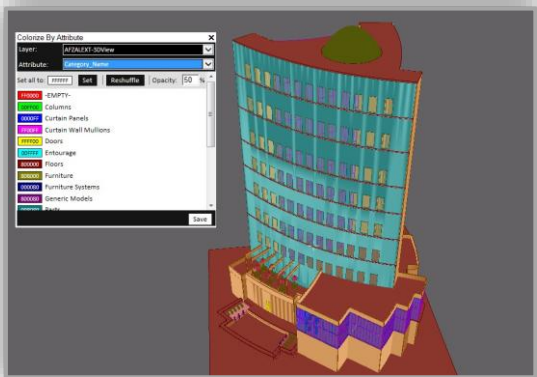
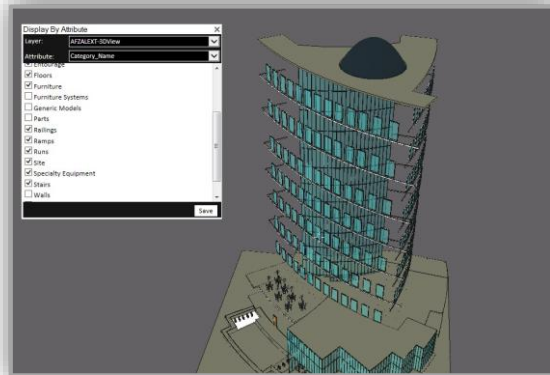
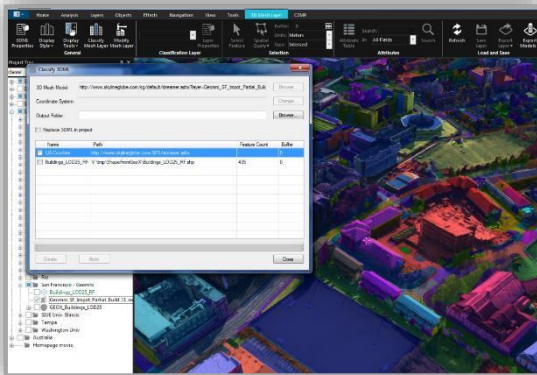


子树以二进制数据形式保存了网格图层分层结构。每个子树记录保存了包括 BBox、数据块的像素大小、参考和相对子节点的参考等信息。

子树的结构图：



嵌套式子树的结构图：



地面层

3DML 文件格式可以单独读取网格图层的地面信息。这使得网格图层可以替换 3DML 范围内的影像和高程，这样让类似道路线等贴地表的对象可以直接贴在 3DML 地面层上，可能有一点随机的网格地面误差。



压缩存储优化

3DML 格式是基于 SQLite 数据库和它的 SpatialLite 扩展，用于存储海量的三维空间数据集，可以用在移动端、网页端和桌面端。3DML 是一个单独的二进制文件，存储了所有关于 3DML 数据集的信息，包括 BBox、坐标系、网格模型块以及用于分类的面图层，这样的存储形式非常方便存储和管理。

数据流优化

网格数据存储在多级文件数据库中，这样在加载的时候可以先加载低精度级别的数据，当用户放大浏览到高精度的区域的时候再动态加载高精度的数据。从文件中读取数据时不需要一定是连续的数据，可以是不连续的数据。

表格说明

所有的 3DML 都包括了以下通用表格：

- [ThreeDML_Version](#)
- [ThreeDML_Projects](#)

每个网格图层都包括如下表格

- [ThreeDML_Table1_Mesh](#)
- [ThreeDML_Table1_Index](#)

通用表格

ThreeDML_Version

3DML 1.0 当前版本

SQL:

```
CREATE TABLE IF NOT EXISTS `ThreeDML_Version` (  
    `Major` INTEGER NOT NULL,  
    `Minor` INTEGER NOT NULL  
);  
INSERT INTO `ThreeDML_Version` VALUES (1,0);
```

	类型	描述
Major	INTEGER	主版本
Minor	INTEGER	小版本

ThreeDML_Projects

这个表格保存了 3DML 数据集的常规信息，包括：

- BBox
- 唯一 GUID
- WKT 形式的坐标系统
- 版本
- 子树根节点

通常一个 3DML 文件包括多个 3D 网格图层，但实际上我们只包括一个网格图层，所以 Projects 表里只有一个工程（一行）。

这个工程的编号是 1，它的所有数据都被保存在前缀是 ThreeDML_Table1 的表格里 (ThreeDML_TABLE1_Mesh, ThreeDML_TABLE1_Index, ThreeDML_TABLE1_Meshindex)。

SQL:

```
CREATE TABLE IF NOT EXISTS `ThreeDML_Projects` (  
    `Id` INTEGER NOT NULL,  
    `Name` TEXT NOT NULL,  
    `Prefix` TEXT NOT NULL,  
    `MinX` REAL NOT NULL,  
    `MaxX` REAL NOT NULL,  
    `MinY` REAL NOT NULL,  
    `MaxY` REAL NOT NULL,  
    `MinZ` REAL NOT NULL,  
    `MaxZ` REAL NOT NULL,  
    `BestResolution` REAL NOT NULL,  
    `WorstResolution` REAL NOT NULL,  
    `Projection` TEXT NOT NULL,  
    `Version` INTEGER NOT NULL,  
    `GUID` guid NOT NULL,  
    `Tags` TEXT NOT NULL,  
    `Description` TEXT NOT NULL,  
    `StartMeshIndexId` INTEGER NOT NULL,
```

```

`State` INTEGER NOT NULL,
`Checksum` TEXT NOT NULL,
`EngineVersion` TEXT NOT NULL,
`WKTFull` TEXT,
PRIMARY KEY(`Id`)
);

```

	类型	描述
Id	INTEGER	网格图层的 ID
Name	TEXT	网格图层的名称
Prefix	TEXT	用于存储网格数据的表格前缀。例如 “ThreeDML_Table1”。
MinX	REAL	存储的数据最小坐标 X 值，在工程坐标系统中
MaxX	REAL	存储的数据最大坐标 X 值，在工程坐标系统中
MinY	REAL	存储的数据最小坐标 Y 值，在工程坐标系统中
MaxY	REAL	存储的数据最大坐标 Y 值，在工程坐标系统中
MinZ	REAL	存储的数据最小坐标 Z 值（高程），在工程坐标系统中
MaxZ	REAL	存储的数据最大坐标 Z 值（高程），在工程坐标系统中
BestResolution	REAL	最高网格分辨率，单位：单元/像素，用于元数据，例如 0.02
WorstResolution	REAL	最差网格分辨率，单位：单元/像素，用于计算 3D 窗口开始加载 3DML 时的距离

Projection	TEXT	数据集 WKT 格式的坐标系统 Without COMPD_CS[].
Version	TEXT	对于 Version 1.0 值是 2
GUID	guid	网格的唯一编码。3DML 每次更新时会变化。 例如：12345678-12341234-1234- 123456789abc.
Tags	TEXT	可选，分隔元数据标记。保留使用，值为空
Description	TEXT	可选，用于描述网格图层。保留使用，值为空.
StartMeshIndexId	INTEGER	在[Prefix]_MeshIndex 表 (例如 ThreeDML_Table1_MeshIndex)中的根节点 索引
State	INTEGER	版本 1.0 中使用 65535
CheckSum	TEXT	可选，值为空
EngineVersion	TEXT	版本 1.0 中使用 7.0.0.1836
WKTFullII	TEXT	可选，完整的坐标系统 WKT 描述。如果高程 基准不需要提供，值和 Projection 相同
MinZ	REAL	存储的数据最小坐标 Z 值（高程），在工程坐 标系统中
MaxZ	REAL	存储的数据最大坐标 Z 值（高程），在工程坐 标系统中
BestResolution	REAL	最高网格分辨率，单位：单元/像素，用于元数 据，例如 0.02
WorstResolution	REAL	最差网格分辨率，单位：单元/像素，用于计 算 3D 窗口开始加载 3DML 时的距离
Projection	TEXT	数据集 WKT 格式的坐标系统 Without COMPD_CS[].

Version	TEXT	对于 Version 1.0 值是 2
GUID	guid	网格的唯一编码。3DML 每次更新时会变化。 例如：12345678-12341234-1234-123456789abc.
Tags	TEXT	可选，分隔元数据标记。保留使用，值为空
Description	TEXT	可选，用于描述网格图层。保留使用，值为空.
StartMeshIndexId	INTEGER	在[Prefix]_MeshIndex 表 (例如 ThreeDML_Table1_MeshIndex)中的根节点索引
State	INTEGER	版本 1.0 中使用 65535
CheckSum	TEXT	可选，值为空
EngineVersion	TEXT	版本 1.0 中使用 7.0.0.1836
WKTFullI	TEXT	可选，完整的坐标系统 WKT 描述。如果高程基准不需要提供，值和 Projection 相同

网格图层表格

ThreeDML_Table1_Mesh

这个表格存储网格节点数据。

表格中的每个节点可以包括：

3D-Mesh (几何、贴图、FLID) --详情请查看网格 SLBLOB

3D-Floor (几何) -不在规格说明书范围。

节点之间的关系（父节点和子节点）被存储于子树结构中，在 ThreeDML_Table1_MeshIndex 和 Mini-tree SLBLOB 中进行描述。子树是一个小的树结构，用于通过节点编号指向节点或通过树编号指向另一个子树。

SQL:

```
CREATE TABLE IF NOT EXISTS `ThreeDML_Table1_Mesh` (
  `Id`    INTEGER NOT NULL,
  `Data`  Blob NOT NULL,
  PRIMARY KEY(`Id`)
);
```

	Type	Description
Id	INTEGER	节点编号
Data	Blob	节点 SLBLOB 二进制数据

ThreeDML_Table1_MeshIndex

这个表格存储了子树结构，描述了节点的分层关系。子树存储了子节点 ID 和其他子树的 ID，从而可以描述整个网格图层的所有节点。

SQL:

```
CREATE TABLE IF NOT EXISTS `ThreeDML_Table1_MeshIndex` (
  `Id`    INTEGER NOT NULL,
  `Data`  Blob NOT NULL,
  PRIMARY KEY(`Id`)
);
```

	Type	Description
Id	INTEGER	子树编号

Data	Blob	子树 SLBLOB 二进制数据
------	------	-----------------

ThreeDML_Table1_Index

这个表格存储了关于网格图层的元数据，用于描述每个节点。

SQL:

```
CREATE TABLE IF NOT EXISTS `ThreeDML_Table1_Index` (
  `Id`      INTEGER NOT NULL,
  `TreeId`   INTEGER NOT NULL,
  `Radius`   REAL NOT NULL,
  `Resolution` REAL NOT NULL,
  `CenterX`  REAL NOT NULL,
  `CenterY`  REAL NOT NULL,
  `CenterZ`  REAL NOT NULL,
  `RadiusMinX` REAL NOT NULL,
  `RadiusMaxX` REAL NOT NULL,
  `RadiusMinY` REAL NOT NULL,
  `RadiusMaxY` REAL NOT NULL,
  `RadiusMinZ` REAL NOT NULL,
  `RadiusMaxZ` REAL NOT NULL,
  `MinX`     REAL NOT NULL,
  `MaxX`     REAL NOT NULL,
  `MinY`     REAL NOT NULL,
  `MaxY`     REAL NOT NULL,
  `MinZ`     REAL NOT NULL,
  `MaxZ`     REAL NOT NULL,
  `MinRange` REAL NOT NULL,
  `MeshSize` INTEGER NOT NULL,
  `ParentId` INTEGER NOT NULL,
  `MeshId`   INTEGER NOT NULL,
```

```

`Source`      INTEGER NOT NULL,
`PredictableLayerUID`      TEXT NOT NULL,
`PredictableItemUID` TEXT NOT NULL,
PRIMARY KEY(`Id`)
);

```

	Type	Description
Id	INTEGER	网格节点编号
TreeId	TreeId	指向网格节点的子树的编号
Radius	REAL	节点半径，单位：米/CS
Resolution	REAL	网格节点的分辨率，单位：米
CenterX	REAL	网格节点中心点坐标 X 值（网格图层坐标系统）
CenterY	REAL	网格节点中心点坐标 Y 值（网格图层坐标系统）
CenterZ	REAL	网格节点中心点坐标 Z 值（高程）（网格图层坐标系统）
RadiusMinX	REAL	网格节点半径 X 最小值，在工程坐标系统中
RadiusMaxX	REAL	网格节点半径 X 最大值，在工程坐标系统中
RadiusMinY	REAL	网格节点半径 Y 最小值，在工程坐标系统中
RadiusMaxY	REAL	网格节点半径 Y 最大值，在工程坐标系统中

RadiusMinZ	REAL	网格节点半径 Z（高程）最小值，在工程坐标系统中
RadiusMaxZ	REAL	网格节点半径 Z（高程）最大值，在工程坐标系统中
MinX	REAL	网格节点坐标 X 最小值，在工程坐标系统中
MaxX	REAL	网格节点坐标 X 最大值，在工程坐标系统中
MinY	REAL	网格节点坐标 Y 最小值，在工程坐标系统中
MaxY	REAL	网格节点坐标 Y 最大值，在工程坐标系统中
MinZ	REAL	网格节点坐标 Z（高程）最小值，在工程坐标系统中
MaxZ	REAL	网格节点坐标 Z（高程）最大值，在工程坐标系统中
MinRange	REAL	显示网格节点的最小距离（单位：米）
MeshSize	INTEGER	网格节点大小（单位：KB）
ParentId	INTEGER	父节点的编号
MeshId	INTEGER	网格节点编号
Source	INTEGER	版本 1.0 中使用 1
PredictableLayerUID	TEXT	保留使用，值为空
PredictableItemUID	TEXT	保留使用，值为空

SLBLOB 数据

在 ThreeDML_Table1_Meshindex 和 ThreeDML_Table1_Mesh 表格中存储了 SLBLOB 类型数据，它是加密流格式。

SLBLOB = [marker][xor_7zdata]

[7zdata]是用常规的 7zip 压缩算法(lzma16)进行压缩，在 7zdata 字节数组中包括 1 个或多个文件。

如果数据超过 64 字节，则可以使用以下方法对字节数据进行 xor 操作：

```
std::vector<byte> _7zdata

if (_7zdata.size()>64)
{
    DWORD* Data;
    DWORD* Mask;    for
(int i=0;i<8;i++)
    {
        Data=(DWORD*)(_7zdata.data()+i*sizeof(DWORD));
        Mask=(DWORD*)(_7zdata.data()+(8+i)*sizeof(DWORD));
        *Data=*Data ^ *Mask;
    }
}

[xor_7zdata]=xor([7zdata])
```

网格 SLBLOB

网格 SLBLOB 定义使用了标记：

```
[marker]= dword(0x31637078) [0x78,0x70,0x63,0x31] 'xpc1'
```

Xpc1 网格数据格式: [Mesh Node SLBLOB](#).

子树 SLBLOB

子树 SLBLOB 定义使用了标记：

```
[marker]= dword(0x30304449)
```

```
[0x49,0x44,0x30,0x30] 'ID00'
```

子树数据是个二进制数据，格式如下：

“id” 文件是使用[marker]、[root_link]和[node]列表创建的。

[marker]4 字节

[root_link]指向文件的第一个节点（这应该是列表中一个节点的字节偏移量）

每个节点都包含[node_data]-这是节点信息和可选的[son_link]

[son_link]是文件中用于链接另一个节点(node_link_struct)或数据库中另一个记录的另一个子树(tree_link_struct)

下面是每个结构的不同流数据:

```
[marker] =      dword(0x30304449) or unsigned char [0x49,0x44,0x30,0x30] ->
                'ID00'
```

```
[root_link] word(0x0004),word(0x0004),dword(first_node_offset)
```

```
[node]      word(0x0001),word(0x0001/0x0002)  [node_data] + optional ( [son_link] )
```

```
[node_data] word(0x0002),word(0x0034), data_struct
```

[son_link] word(0x0003),word(num), node_link_struct X num | tree_link_struct X num

data_struct

{

double Center[3]; // position of mesh pivot

float Radius; // radius =
 $\sqrt{\text{SizeDiv2}[0]*\text{SizeDiv2}[0]+\text{SizeDiv2}[1]*\text{SizeDiv2}[1]+\text{SizeDiv2}[2]*$

$\text{SizeDiv2}[2])}$ float SizeDiv2[3]; // half size bounding box

float MinRange; // size in pixels when this node should be

use

LONGLONG Fid; // use -1 to have no mesh

}

node_link_struct

{

LONGLONG Offset; // Offset in bytes for the node

DWORD Type; // 0x00000003 - Offset in file

}

tree_link_struct

{

LONGLONG TID; // Tree ID


```
DWORD   Type;           // 0x00000001 - Tree ID  
  
}
```

每个子树 SLBLOB 都被存储在 ThreeDML_Table1_Meshindex，都有一个私有的 ID。这个 ID 在 tree_link_struct 中被当做 TID 和从一个子树到另一个子树的链接。

网格节点 SLBLOB

3D 网格节点格式:

3D 网格节点是一个 SLBLOB 数据，包括以下文件:

“1.sdkmesh”

“1.dds” /jpeg/png (请查看贴图格式)

“1.sdkmesh” 是一个存储网格几何信息的二进制文件– sdkmesh 文件可以存储拥有多个材质或贴图的网格。大部分情况下，我们只使用一个几何和一个贴图。

“1.dds” 是一个存储网格贴图的二进制文件。

如果不止一个贴图，贴图文件名将为 2.....n.dds/jpg/png

网格拆分成矢量对象不在本次介绍中（ FLID ）

贴图格式：

dds - DXT1 压缩贴图格式。(不透明)

DXT3 压缩贴图格式。(可透明)

大部分情况下 3DML 包含 dds 文件，当然也可以使用 png 或 jpg，但是当用户使用时客户端桌面应用会实时转换成 dxt1/dxt3。

SmartEarth 产品技术中心

技术支持邮箱：info@terra-it.cn

官网：<http://www.smartearth.cn>

微信公众账号：泰瑞数创

官方 QQ 群：538859839
