

# SuperMap UDBX ファイル形式空間データベース仕様書

2025 年 1 月作成

日本スーパーマップ株式会社

事業統括部

**UDBX (Universal Spatial DataBase eXtension)**は SuperMap が提唱し、2019 年上期にリリースした SuperMap GIS 9D(2019)から製品に搭載したオープンデータフォーマットで、空間データの効率的な保存、管理、共有、交換を目的としています。2 次元と 3 次元の統合、ベクタデータとラスタデータの統合をサポートし、単一ファイルにデータを保存することでコピーや配布を容易にします。複数のユーザーによる同時アクセス、デスクトップ、サーバー、モバイル端末での効率的な読み書き、大容量データの保存、多言語に対応するための Unicode エンコーディングもサポートしています。

## 1. UDBX の全体構造

- **ファイル形式:** UDBX は、SuperMap が提唱するオープンなファイル形式であり、空間データの効率的な保存、管理、共有、交換を目的としています。SQLite データベースを基盤とし、ファイル拡張子は「udbx」です。
- **ストレージ:** SpatialLite ストレージと SuperMap カスタムストレージの 2 種類があります。
  - **SpatialLite ストレージ:** ポイント、ライン、ポリゴンなどの基本的なベクタタイプに対応します。
  - **SuperMap カスタムストレージ:** その他のデータタイプに対応します。
- **テーブル:** システムテーブルとデータテーブルの 2 つのカテゴリに分類されます。
  - **システムテーブル:** UDBX のデータセットを管理するために使用されます。
    - SpatialLite のシステムテーブル
    - SuperMap カスタムシステムテーブル
  - **データテーブル:** 実際のデータを格納するために使用されます。

## 2. データセットのタイプ

1 つの UDBX ファイルは、複数のデータセットを格納する 1 つのデータソースです。データセットには、ベクタデータセットとラスタデータセットの 2 つのカテゴリがあります。

- **ベクタデータセット:** 同じ座標系と属性情報を持つ空間オブジェクトを格納します。
- **ラスタデータセット:** 連続したフィールドデータをブロック別に格納および整理します。

**データセットのタイプ:**

- **Tabular:** 幾何オブジェクトを格納しないテーブル
- **Point:** 2D ポイントデータセット
- **PointZ:** 3D ポイントデータセット
- **Line:** 2D ラインデータセット
- **LineZ:** 3D ラインデータセット
- **Region:** 2D ポリゴンデータセット
- **RegionZ:** 3D ポリゴンデータセット
- **Text:** テキストオブジェクト
- **CAD:** 複数のジオメトリオブジェクトを格納する複合データセット
- **Network:** 2D ネットワークデータセット

- Network3D: 3D ネットワークデータセット
- Model: 3D モデルデータセット
- Image: マルチバンド画像データセット
- Grid: デジタルサーフェスモデルに対応し、ピクセル値は地表の特徴量を表します
- VoxelGrid: ボクセルグリッドデータセット
- Mosaic: モザイクデータセット

### 3. Geometry タイプ

2D/3D ポイント、ライン、ポリゴンデータセットに格納されているオブジェクトは、SpatialLite のポイント、ライン、ポリゴンオブジェクトを使用します。オブジェクトの命名規則は以下の通りです。

- "GAIA"プレフィックス: SpatialLite のオブジェクト
- "Geo"プレフィックス: その他のオブジェクトタイプ

#### Geometry タイプ:

- GAIAPoint: 2D ポイントオブジェクト
- GAIAPolygon: 2D ポリゴンオブジェクト
- GAIAMultiLineString: 2D ライン (サブオブジェクト含む)
- GAIAMultiPolygon: 2D ポリゴン (サブオブジェクト含む)
- GAIAPointZ: 3D ポイントオブジェクト
- GAIAMultiLineStringZ: 3D ライン (サブオブジェクト含む)
- GAIAMultiPolygonZ: 3D ポリゴン (サブオブジェクト含む)
- GeoPoint3D: 3D ポイント
- GeoLine3D: 3D ライン (サブオブジェクト含む)
- GeoRegion3D: 3D ポリゴン (サブオブジェクト含む)
- GeoText: テキストオブジェクト
- GeoModel3D: 3D モデルオブジェクト
- GeoRect: 矩形/斜め矩形
- GeoRectRound: 角丸矩形
- GeoCircle: 円
- GeoEllipse: 楕円
- GeoPie: 扇形
- GeoArc: 円アーク
- GeoEllipticArc: 楕円アーク
- GeoCardinal: Cardinal 曲線
- GeoCurve: ペンシルライン
- GeoBSpline: B スプライン曲線

### 4. システムテーブル

システムテーブルは、データソース内のデータセット情報を格納および管理するために使用されます。システムテーブルは、SpatialLite で定義されたシステムテーブルと SuperMap によって定義されたシステムテーブルに区分されます。

#### SpatialLite システムテーブル:

- 座標系情報テーブル: spatial\_ref\_sys と spatial\_ref\_sys\_aux に格納

- spatial\_ref\_sys:座標系のユニーク ID、作成者、内部 ID、名前、proj.4 形式の座標系、WKT 形式の座標系を格納
- spatial\_ref\_sys\_aux:地理座標系かどうか、軸が反転するかどうか、参照楕円体、中央子午線、測地系、投影方法、座標系の単位、スピンドル名、軸の向きなどを格納
- **ベクタデータセットシステムテーブル:** geometry\_columns に格納
  - geometry\_columns:データテーブルの名前、geometry 列名、geometry 型、座標の次元、座標系、空間インデックスが確立されているかどうかを格納

#### SuperMap システムテーブル:

- **データソースの説明情報テーブル:** SmDataSourceInfo テーブルに格納
  - SmDataSourceInfo:データソース ID、バージョン番号、説明情報、座標系情報、最終更新時刻、データ保存フォーマットを格納
- **ベクタデータセットシステムテーブル**
  - **ベクタデータセットレジストリテーブル:** SmRegister テーブルに記録
    - SmRegister:データセット ID、名前、テーブル名、オプション情報、親データセット ID、データセットの種類、オブジェクト数、地理的範囲、座標系 ID、空間インデックスタイプ、スナップトレランス、角度トレランス、ロングダングルトレランス、ポリゴン積トレランス、最大幾何オブジェクトサイズ、データセット記述情報、拡張情報、作成時刻、最終更新時刻、座標系情報を格納
  - **データセットフィールド情報テーブル:** SmFieldInfo テーブルに格納
    - SmFieldInfo:フィールド ID、データセット ID、フィールド名、識別名、SuperMap に対応するフィールドのタイプ、フォーマット文字列、フィールド ID、フィールドを更新可能かどうか、フィールドを入力するかどうか、デフォルト値、サイズなどを格納
  - **値域情報テーブル:** SmDomains、SmRangeDomains、SmCodeDomains、SmDomainField テーブルに格納
    - SmRangeDomains:範囲値域規則オブジェクトを格納
    - SmCodeDomains:列挙値域規則オブジェクトを格納
    - SmDomains:値域規則 ID、値域名、説明情報、値域タイプを格納
    - SmDomainField:データセット ID、値域規則が適用されるデータセットのフィールド名を格納
- **ラスタデータセットシステムテーブル**
  - **ラスタデータセットレジストリ:** SmImgRegister テーブルに格納
    - SmImgRegister:データセット ID、名前、テーブル名、データセットタイプ、幅、高さ、保存ブロックの大きさ、地理的範囲、作成時刻、作成者、記述情報、クリッピングポリゴン、拡張情報、統計情報、座標系情報を格納
  - **バンド情報レジストリ:** SmBandRegister テーブルに格納
    - SmBandRegister:バンド ID、データセット ID、バンドの順序番号、バンド名、利用可能かどうか、オプション情報、圧縮符号化方法、ピクセル形式、ストレージブロックの最大サイズ、最小/最大ピクセル値、高さ、ピラミッド情報、作成者、作成時刻、Null 値、カラーパレットを格納

## 5. データテーブル

データテーブルは、実際のデータを格納するために使用されます。各データセットは 1 つ以上のテーブルに対応し、ベクタデータセットとラスタデータセットの 2 つのカテゴリに分類されます。

### ベクタデータセット

- **属性データセット:** 空間データは格納しません。SmID(主キー)、SmUserID(ユーザーカスタム ID)を格納します。
- **2D/3D ポイントデータセット:** SmID(主キー)、SmUserID(ユーザーカスタム ID)、SmGeometry(GAIAPoint または GAIAPointZ オブジェクト)を格納します。
- **2D/3D ラインデータセット:** SmID(主キー)、SmUserID(ユーザーカスタム ID)、SmLength(ラインオブジェクトの長さ)、SmTopoError(トポロジトレランス)、SmGeometry(GAIAMultiLineString または GAIAMultiLineStringZ オブジェクト)を格納します。
- **2D/3D ポリゴンデータセット:** SmID(主キー)、SmUserID(ユーザーカスタム ID)、SmArea(ポリゴンオブジェクトの面積)、SmPerimeter(ポリゴンオブジェクトの周長)、SmGeometry(GAIAMultiPolygon または GAIAMultiPolygonZ オブジェクト)を格納します。
- **テキストデータセット:** SmID(主キー)、SmUserID(ユーザーカスタム ID)、SmGeometry(GeoText オブジェクト)、SmIndexKe(オブジェクトの範囲、GAIAPolygon オブジェクトとして保存)を格納します。
- **複合データセット:** SmID(主キー)、SmUserID(ユーザーカスタム ID)、SmGeoType(Geometry タイプ)、SmGeometry(SuperMap シンプルオブジェクト)、SmIndexKey(SuperMap シンプルオブジェクトの範囲)を格納します。複合データセットに保存されているオブジェクトの種類は以下の通りです。
  - GeoPoint: 2D ポイント
  - GeoLine: 2D ライン(サブオブジェクトを持つ場合があります)
  - GeoRegion: 2D ポリゴン(サブオブジェクトを持つ場合があります)
  - GeoText: テキスト
  - GeoRect: 矩形/斜め矩形
  - GeoRectRound: 丸角矩形
  - GeoCircle: 円
  - GeoEllipse: 楕円
  - GeoPie: 扇形ポリゴン
  - GeoArc: 円アーク
  - GeoEllipticArc: 楕円アーク
  - GeoCardinal: Cardinal 曲線
  - GeoCurve: ペンシルライン
  - GeoBSpline: B タイプ曲線
  - GeoPoint3D: 3D ポイント
  - GeoLine3D: 3D ライン(サブオブジェクトを持つ場合があります)
  - GeoRegion3D: 3D ポリゴン(サブオブジェクトを持つ場合があります)
- **2D/3D ネットワークデータセット:** プライマリテーブルとサブテーブルで構成されます。
  - **プライマリテーブル:** SmID(主キー)、SmUserID(ユーザーカスタム ID)、SmEdgeID(エッジ ID)、SmFNode(起点のポイント ID)、SmTNode(終点結節 ID)、SmResistanceA(正方向バリア)、SmResistanceB(負方向バリア)、SmTopoError(トポロジトレランス)、SmLength(ラインオブジェクトの長さ)、SmGeometry(GAIAMultiLineString または GAIAMultiLineStringZ オブジェクト)を格納します。
  - **サブテーブル:** SmID(主キー)、SmUserID(ユーザーカスタム ID)、SmNodeID(ノード ID)、SmGeometry(GAIAPoint または GAIAPointZ オブジェクト)を格納します。
- **3D モデルデータセット:** メインテーブルとサブテーブルで構成されます。
  - **メインテーブル:** SmID(主キー)、SmUserID(ユーザーカスタム ID)、SmGeometry(GeoModel3D オブジェクト)、SmIndexKey(モデルオブジェクトの範囲)を格納します。

- **サブテーブル:** SmID(主キー)、SmHashCode(ユニーク制約)、SmGeometry(実体オブジェクト)を格納します。

#### ラスタデータセット

- **Image データセット、Grid データセット、VoxelGrid データセット:** 同じデータ編成と保存方式を使用します。
  - **データテーブル:** SmRow(ブロックの行番号)、SmColumn(ブロックの列番号)、SmBandID(グリッドデータ層 ID)、SmSize(ブロックデータバイトストリーム大きさ)、SmBand(ブロックデータ)を格納します。
- **モザイクデータセット:** マスターテーブル(SmImgRegter)と、アウトライン、境界、クリッピングの 3 つのサブデータセットで構成されます。
  - **アウトラインサブデータセット:** SmID(主キー)、SmUserID(ユーザーカスタム ID)、SmArea(ポリゴンの面積)、SmPerimeter(ポリゴンの周長)、SmGeometry(幾何オブジェクト)、SmFileName(フックされたファイルの名前)、SmMinPS(最小表示解像度)、SmMaxPS(最大表示解像度)、SmLowPS(元の解像度)、SmHighPS(ピラミッドの解像度)、SmCategory(元のファイルかどうか)、SmPath(画像ファイルパス)、SmInfo(統計情報)、SmZOrder(予約フィールド)、SmFileHash(ファイルに対応する Hash 値)、SmOverviewLevel(ピラミッドレベル)を格納します。
  - **境界サブデータセット:** 2D ポリゴンデータセットと同じフィールドを格納します。
  - **クリッピングサブデータセット:** 2D ポリゴンデータセットのフィールドに加えて、FootprintID(アウトラインサブデータセットの SmID フィールドに関連付けられる)を格納します。

## 6. オブジェクト保存構造

UDBX 内の様々なオブジェクトのバイナリ保存構造について説明します。バイトオーダーはリトルエンディアンです。

#### 基本タイプ

- byte: 1 バイト
- bool: 1 バイト、0 または 1
- int16: 2 バイト、-32768 から 32767
- uint16: 2 バイト、0 から 65535
- int32: 4 バイト、-2147483648 から 2147483647
- uint32: 4 バイト、0 から 4294967295
- int64: 8 バイト、 $-2^{63}$  から  $(2^{63}-1)$
- uint64: 8 バイト、0 から  $(2^{64}-1)$
- float: 4 バイト、単精度浮動小数点型
- double: 8 バイト、倍精度浮動小数点型
- wchar: 2 バイト、ワイド文字型
- String: Unicode でコード化された String オブジェクトで記述され、文字セットは UTF8 として定義されています。
  - int32 length: バイト数
  - byte str[length]: データ内容
- Point: x,y で表される 2D 座標点。
  - double x: x 座標
  - double y: y 座標
- PointZ: x,y,z で表される 3D 座標点。
  - double x: x 座標
  - double y: y 座標

- double z: z 座標
- Rect: 左上の点と右下の点で表される矩形。
  - Point pntLB: 左下角点
  - Point pntRT: 右上角点
- BoundingBox: バウンディングボックス、最大ベクトルと最小ベクトルで表されます。
  - PointZ boxMax: 最大ベクトル
  - PointZ boxMin: 最小ベクトル
- Ring: ポイントによって首尾相互接続されるリング。
  - int32 numPoints: 点数
  - Point[] pnts[numPoints]: 点座標
- RingZ: 3D ポイントによって首尾相互接続されるリング。
  - int32 numPoints: 点数
  - PointZ[] pnts[numPoints]: 点座標
- Vector3D: 3D ベクトル、保存構造は PointZ と同じ。
- Color: rgba の 4 つのコンポーネントで構成される uint32 値。
  - byte a: alpha 値
  - byte b: blue 値
  - byte g: green 値
  - byte r: red 値

#### SpatialLite の単純なオブジェクト

- GAIAInfo: SpatialLite の各種単一オブジェクトのヘッダー情報。
  - byte byteOrdering: バイト順、小端保存
  - int32 srid: 座標系 ID
  - Rect mbr: オブジェクトの座標範囲
- GAIAPoint: SpatialLite の 2D ポイントオブジェクト。
  - byte gaiaStart: バイナリストリームの開始タグ
  - GAIAInfo info: 幾何オブジェクトの基本情報
  - int32 geoType: Geometry タイプ ID
  - Point geopnt: ポイントオブジェクトの座標値
  - byte gaiaEnd: バイナリストリームの終了タグ
- GAIAPointZ: SpatialLite の 3D ポイントオブジェクト。
  - byte gaiaStart: バイナリストリームの開始タグ
  - GAIAInfo info: 幾何オブジェクトの基本情報
  - int32 geoType: Geometry タイプ ID
  - PointZ geoPntZ: ポイントオブジェクトの座標値
  - byte gaiaEnd: バイナリストリームの終了タグ
- GAIAMultiLineString: SpatialLite の 2D マルチラインオブジェクト。
  - byte gaiaStart: バイナリストリームの開始タグ
  - GAIAGeoInfo info: 幾何の基本情報
  - int32 geoType: Geometry タイプ ID
  - int32 numLineStrings: サブオブジェクトの数

- LineStringEntity[] lineStrings[numLineStrings]: LineString のジオメトリデータ
- byte gaiaEnd: バイナリストリームの終了タグ
- LineStringEntity:
  - byte gaiaEntityMark: サブオブジェクト ID
  - int32 geoType: Geometry タイプ ID
  - int32 numPoints: 点数
  - Point[] pnts[numPoints]: 各点の座標値
- GAIAMultiLineStringZ: SpatialLite の 3D マルチラインオブジェクト。
  - byte gaiaStart: バイナリストリームの開始タグ
  - GAIAGeoInfo info: 幾何オブジェクトの基本情報
  - int32 geoType: Geometry タイプ ID
  - int32 numLineStrings: サブオブジェクトの数
  - LineStringZEntity[] lineStrings[numLineStrings]: LineString のジオメトリデータ
  - byte gaiaEnd: バイナリストリームの終了タグ
  - LineStringZEntity:
    - byte gaiaEntityMark: サブオブジェクト ID
    - int32 geoType: Geometry タイプ ID
    - int32 numPoints: 点数
    - PointZ[] pnts[numPoints]: 各点の座標値
- GAIAPolygon: SpatialLite の 2D ポリゴンオブジェクト。
  - byte gaiaStart: バイナリストリームの開始タグ
  - GAIAGeoInfo info: 幾何オブジェクトの基本情報
  - PolygonData data: ポリゴンジオメトリデータ
  - byte gaiaEnd: バイナリストリームの終了タグ
  - PolygonData:
    - int32 geoType: Geometry タイプ ID
    - int32 numInteriors: 内円の数
    - Ring interiorRing: 外円オブジェクト
    - Ring[] interiorRings[numInteriors]: 内円オブジェクト
- GAIAMultiPolygon: SpatialLite の 2D マルチフェイスオブジェクト。
  - byte gaiaStart: バイナリストリームの開始タグ
  - GAIAGeoInfo info: 幾何オブジェクトの基本情報
  - int32 geoType: Geometry タイプ ID
  - int32 numPolygon: サブオブジェクトの数
  - PolygonEntity[] polygons[numPolygon]: ポリゴンエンティティ
  - PolygonEntity:
    - byte gaiaEntityMark: サブオブジェクト ID
    - PolygonData data: サブオブジェクトデータ
- GAIAMultiPolygonZ: SpatialLite の 3D ポリゴンオブジェクト。
  - byte gaiaStart: バイナリストリームの開始タグ
  - GAIAGeoInfo info: 幾何オブジェクトの基本情報

- int32 geoType: Geometry タイプ ID
- int32 numPolygon: サブオブジェクトの数
- PolygonEntity[] polygons[numPolygon]: ポリゴンエンティティ
- PolygonEntity:
  - byte gaiaEntityMark: サブオブジェクト ID
  - PolygonZData data: サブオブジェクトデータ
- PolygonZData:
  - int32 geoType: Geometry タイプ ID
  - int32 numInteriors: 内円の数
  - RingZ exteriorRing: 外円オブジェクト
  - RingZ[] interiorRings[numInteriors]: 内円オブジェクト

#### 複合データセットに保存されているオブジェクト

- GeoHeader: オブジェクトヘッド。
  - int32 geoType: オブジェクトタイプ
  - int32 styleSize: オブジェクトスタイルが占めるバイト数
  - Style Style: スタイルコンテンツ
- Style: オブジェクトの寸法によってポイントシンボル、ラインシンボル、ポリゴンフィルスタイルに分割されます。
  - StyleMarker: ポイントシンボル。
    - int32 length: バイトストリームの長さ
    - int32 markerStyle: シンボルライブラリ内のポイントシンボルの ID
    - int32 markerSize: シンボルサイズ、精度 0.1 ミリメートル
    - int32 markerAngle: シンボルの回転角度(0.1 度単位)
    - color markerColor: シンボルの色
    - int32 markerwidth: シンボルの幅
    - int32 markerHeight: シンボルの高さ
    - byte reservedLength: 予約バイトの長さ
    - byte[] reservedData[reservedLength+4]: 予約データ
    - byte fillOpaqueRate: フィル透明度
    - byte fillGradientType: グラデーションフィルのタイプ
    - int16 fillAngle: フィル角度
    - int16 fillCenterOffsetX: フィル中心位置水平オフセットの割合
    - int16 fillCenterOffsetY: フィル中心位置垂直オフセットの割合
    - color fillBackcolor: フィル背景色
  - StyleLine: ラインシンボル。
    - int32 lineStyle: マークライブラリにあるラインシンボルの ID 番号
    - int32 lineWidth: 線の太さ、0.1mm
    - color lineColor: 線の色
    - byte reservedLength: 予約バイトの長さ
    - byte[] reservedData[reservedLength+4]: 予約データ
  - StyleFill: ポリゴンフィルスタイル。
    - int32 lineStyle: マークライブラリ内のラインシンボルの ID

- int32 lineWidth: 線の太さ、精度 0.1mm
- color lineColor: 線の色
- int32 fillStyle: フィルライブラリ内のシンボルの ID
- Color fillForecolor: フィル前景色
- Color fillBackColor: フィル背景色
- byte fillOpaquerate: フィル透明度
- byte fillGadientType: グラデーションフィルタイプ
- int16 fillAngle: フィル角度、精度 0.1°
- int16 fillCenterOffsetX: フィル中心位置水平オフセットの割合
- int16 fillCenterOffsetY: フィル中心位置垂直オフセットの割合
- byte reserved1Length: 予約バイトの長さ
- byte[] reserved1Data[reserved1Length+4]: 予約データ
- byte reserved2Length: 予約バイトの長さ
- byte[] reserved2Data[reserved2Length+4]: 予約データ
- GeoPoint: 2D ポイントオブジェクト。
  - GeoHeader header: ヘッダー
  - Point pnt: 点座標値
- GeoLine: 2D ラインオブジェクト。
  - GeoHeader header: ヘッダー
  - uint32 numSub: サブオブジェクトの数
  - int32 subPointCount[numSub]: 各サブオブジェクト点数
  - Point[] pnts[allPntCount]: すべてのサブオブジェクトの点座標数の和
- GeoRegion: 2D ポリゴンオブジェクト。
  - GeoHeader header: ヘッダー
  - uint32 numSub: サブオブジェクトの数
  - int32 subPointCount[numSub]: 各サブオブジェクト点数
  - Point[] pnts[allPntCount]: すべてのサブオブジェクトの点座標数の和
- GeoPoint3D: 3D ポイントオブジェクト。
  - GeoHeader header: ヘッダー
  - PointZ pnt: 点座標
- GeoLine3D: 3D ラインオブジェクト。
  - GeoHeader header: ヘッダー
  - uint32 numSub: サブオブジェクトの数
  - int32 subPointCount[numSub]: 各サブオブジェクト点数
  - PointZ[] pnts[allPntCount]: すべてのサブオブジェクトの点座標数の和
- GeoRegion3D: 3D ポリゴンオブジェクト。
  - GeoHeader header: ヘッダー
  - uint32 numSub: サブオブジェクトの数
  - int32 subPointCount[numSub]: 各サブオブジェクト点数
  - PointZ[] pnts[allPntCount]: すべてのサブオブジェクトの点座標数の和
- GeoRect: 矩形オブジェクト。

- GeoHeader header: ヘッダー
- Point pntCenter: 中心位置座標値
- double width: 幅
- double height: 高さ
- int32 angle: 回転角度に 10 を掛け、丸め値を四捨五入
- GeoRectRound: 丸角矩形オブジェクト。
  - GeoHeader header: ヘッダー
  - Point pntCenter: 中心位置座標値
  - double width: 幅
  - double height: 高さ
  - int32 angle: 回転角度に 10 を掛け、丸め値を四捨五入
  - double radiusX: 丸角半長軸
  - double radiusY: 丸角半短軸
- GeoCircle: 円オブジェクト。
  - GeoHeader header: ヘッダー
  - Point pntCenter: 中心位置座標値
  - double radius: 半径
- GeoEllipse: 楕円オブジェクト。
  - GeoHeader header: ヘッダー
  - Point pntCenter: 中心位置座標値
  - double semimajoraxis: 半長軸
  - double semiminoraxis: 半短軸
  - int32 angle: 回転角度に 10 を掛け、丸め値を四捨五入
- GeoPie: 扇形オブジェクト。
  - GeoHeader header: ヘッダー
  - Point pntCenter: 中心位置座標値
  - double semimajoraxis: 半長軸
  - double semiminoraxis: 半短軸
  - int32 rotationangle: 回転角度に 10 を掛け、丸め値を四捨五入
  - int32 startangle: 開始角度に 10 を掛け、丸め値を四捨五入
  - int32 endangle: 終了角度に 10 を掛け、丸め値を四捨五入
- GeoArc: 円アークオブジェクト。
  - GeoHeader header: ヘッダー
  - Point pntStart: 開始点の座標値
  - Point pntMiddle: 中間点の座標値
  - Point pntEnd: 終端点座標値
- GeoEllipticArc: 楕円アークオブジェクト。
  - GeoHeader header: ヘッダー
  - Point pntCenter: 中心位置座標値
  - double semimajoraxis: 半長軸
  - double semiminoraxis: 半短軸

- int32 rotationangle: 回転角度に 10 を掛け、丸め値を四捨五入
- int32 startangle: 開始角度に 10 を掛け、丸め値を四捨五入
- int32 endangle: 終了角度に 10 を掛け、丸め値を四捨五入
- **曲線:** Cardinal カーブ、フリーカーブ、および B スプラインの保存構造は同じです。
  - GeoHeader header: ヘッダー
  - uint32 numPnts: 曲線参照点の数
  - Point[] pnts[numPnts]: 曲線参照点の座標

### テキストオブジェクト

- **GeoText:** テキストオブジェクトは、テキストデータセットまたは複合データセットに保存できます。テキストデータセットに保存されている場合は、GeoHeader の styleSize は 0 です。
  - GeoHeader header: オブジェクト先頭
  - int32 subCount: text サブオブジェクトの数
  - TextStyle textStyle: テキストスタイル
  - GeoSubText サブテキスト[subCount]: text サブオブジェクト
  - GeoSubText:
    - Point pntAnchor: アンカーポイント
    - int32 subAngle: 回転角度、実際の回転角度は 10 の丸め値で乗算されます
    - String subText: テキストサブオブジェクトのテキスト内容
  - TextStyle:
    - Color color: テキストカラー
    - TextStyleBit textStyleBit: テキストビットごとスタイル
    - Color bgColor: テキスト背景色
    - Double fontWidth: テキストフォント幅
    - Double fontHeight: テキストフォント高さ
    - Point pntAnchor: テキストアンカーポイント
    - String faceName: テキストフォント名
  - TextStyleBit:
    - Byte fixedSize: 固定サイズ
    - Byte weight: ストローク幅
    - byte styleFlag: 太字、斜体、下線など
    - byte alignFlag: テキストの配置

### 3D モデルオブジェクト

- **GeoModel3D:** ローカル座標系を持つモデルオブジェクト(ModelNode)とその配置場所、姿勢などの情報で構成されます。
  - static int32 type: オブジェクトタイプ
  - static int32 hasStyle: スタイルがあるかどうか
  - uint32 version: オブジェクトバージョン番号
  - bool isSpherePlaced: 球ポリゴンに配置するかどうか
  - Vector3D vecPosition: オブジェクト位置
  - Vector3D vecScale: オブジェクトスケール値
  - Vector3D vecRotate: オブジェクト回転値

- BoundingBox bbox: バウンディングボックス
- ModelRenderer modelNode: モデルノード
- ModelRenderer:
  - int32 numLODs: LOD 層数
  - PagedLOD pagedLODs[numLODs]: LOD 層データ
  - int32 numPatches: 精緻モデル Patch の数
  - Patch patches[numPatches]: 精緻モデル Patch データ
- PagedLOD:
  - RangeMode rangeMode: 範囲モードを切り替えます
  - int32 lodNum: LOD 層番号
  - int32 numPatches: 当該層の Patch の数
  - Patch patches[numPatches]: Patche データ
- Patch:
  - float lodDistance: 距離を切り替え
  - int32 index: 現在データ層 Patch のインデックス番号
  - int32 parentIndex: 親ノードのインデックス番号、-1 は親ノードがないことを示します
  - int32 numChildren: サブノードの数
  - int32 childrenIndexes[numChildren]: サブノードのインデックス番号