REST Interface - Entity Model

- Introduction
- Shared Properties
- Entity-specific Properties
- Spatial Properties
- Extended Block Model Metadata
- Extended Drill Hole Metadata
- Extended Voxelized Mesh Metadata
- Extended Geo-chemical Metadata

Introduction

A Voxel Farm project is a collection of entities. Each entity is a collection of key-value pairs. Both key and value are strings. These pairs will be called “properties” in this document.

Each entity will have a special property “ID” which uniquely identifies the entity. This ID is often used in REST calls to identify the entity.

Entities also have a “name” property. This is a readable name that will be used in the UI for users to identify the entity.

Another special property, “type” describes the type of entity. The entity type can be one of these:

“PROJECT”	Specifies the entity is a Voxel Farm project. A project entity can contain other entities.
“FILE”	An entity that contains data, either data that was imported into the system or that was produced by the system
“REGION”	A special type of entity that describes a 3D scope

File entities can be placed inside Folder entities. The “file_folder” in the File entity property specifies the ID of the Folder that contains it.

Most entities in a project will be of “FILE” type, as these are the ones holding the source data and processed results. To differentiate them, the “file_type” property is used. These are the different file types that can appear in Voxel Farm projects:

“FOLDER”	A special type of file entity inside a project that can contain other file entities.
“RAWPC”	A raw point could, that is, a point cloud that has been uploaded by users and remains in an unprocessed format.
“IDXPC”	A point cloud that has been processed by the system and inserted into a spatial index.
“VOXSURF”	A Voxel Terrain entity, usually derived from a heightmap or raw point cloud.
“RAWHM”	A raw raster set defining heights, also known as a heightmap.
“RAWBM”	A raw Block Model, typically stores a CVS file and column metadata for the block model

“RAWMESH”	A raw mesh object.
“IDXMESH”	A mesh that has been processed and indexed into the project.
"VOXMESH"	A voxelized mesh set with attributes
“VOXBM”	A voxelized Block Model. This is the output of processing a “RAWBM” entity.
“RAWDH”	A raw drill hole dataset.
"IDXDH"	An indexed drill hole dataset.
“ORTHO”	A raw imagery raster set, typically used to apply ortho imagery to terrains. Includes images and world registration files.
"IDXORTHO"	An indexed imagery set. A single imagery set can supply different formats for the image, depending on what the clients can read.
“PROGRAM”	An entity that stores a Python program for a custom report, view or spatial object.
“VOXGEN”	An entity that generates volumetric content on-the-fly using a Python program that runs on the servers.
“VOXOP”	A real-time terrain operation layer.
“VIEW”	Describes a view in the project. A view is a mashup of different entities and viewing options used for rendering
“REPORT”	An entity that describes a report
“EXPORT”	An entity that holds data that can be downloaded out of the system.
“UNITY”	An entity that contains a Unity project already preconfigured for this Voxel Farm project
“VOXMT”	A Material Tracking Layer. See the Material Tracking API for additional REST calls available for this entity type.
"VOXPC"	A voxelized point cloud. This entity produces a volumetric model out of a set of spherical point cloud capture sets.
"RAWGEOCHEM"	A raw Geo-chemical samples dataset
"GEOCHEM"	A processed Geo-chemical samples dataset.

Shared Properties

The following properties are found in most entities:

ID	Contains the unique identifier for the entity. This property is used to index all entities.
name	Contains the readable name of the entity.
description	Contains a textual description of the entity.
type	Contains a type identifier for the entity. These are the types currently supported: 1. “PROJECT” 2. “REGION” 3. “FILE”
project	Contains the unique identifier for the project that contains the entity. For project entities, this property must have the same value as the “ID” property.
state	Defines the current state of the entity. It can be one of these values: “COMPLETE”, “PARTIAL” and “ERROR”.
partial_progress	If the “state” property is set to “PARTIAL”, this property may contain a percentage indicator for the current stage of processing.
partial_status	If the “state” property is set to “PARTIAL”, this property may contain a readable sentence that describes the current stage of processing.
error_info	If the “state” property is set to “ERROR”, this property may contain additional information about the error. This property contains a Base64 string.
file_type	If the “type” property is set to “FILE”, this property contains a more specific type for the entity. The following subsections are dedicated to these types.
file_size	If the “type” property is set to “FILE”, this property contains the size of the entity’s data in bytes.
file_folder	If the “type” property is set to “FILE”, this property contains the ID of the folder that contains the entity. The special value “0” means the entity is in the root folder of the project.
virtual	Can be set to "1" or "0". If set to "1", this signals the application that this entity should be hidden in the User Interface. This property is used in entities that are building blocks for a larger entity.

Entity-specific Properties

File VIEW

Defines a View entity. Views store a pre-configured way of viewing multiple datasets. This type of entity will have these additional properties:

view_type	Specifies the type of view. A view that contains children, should have its "view_type" property set to "container", and should include a property named "entity_container" which lists the view children. If the view is not a container, its "view_type" must be set to the ID of a View Program entity. Please review the View Programs section for more information on these programs.
entity_container	Used only if the "view_type" property is set to "container". This property contains the list of children views for the view entity. This property is made by concatenating the IDs of the children views separated by spaces.
input_value_#id#	Used only for views that use a View Program. Contains the value for the input. Replace the #id# placeholder by the actual id used by the input. Depending on the value specified for "input_type_#id#" (see types below in this table) the value will have a different interpretation: 0 – A numeric value, can have decimals 1 – A date-time, a 64bit integer value representing a timestamp. Typically a Linux epoch time. 2 – An alphanumeric value 3 – An identifier for an entity 4 – A list of attributes, separated by spaces 5 – Ignored, attribute query properties are listed below in this table. 6 – A Boolean value, "0" for False and "1" for True. 7 – Ignored, color legend properties are listed below in this table. 8 – Ignored, drill hole properties are listed below in this table.
input_label_#id#	Used only for views that use a View Program. Contains the label for the input. Replace the #id# placeholder by the actual id used by the input.

input_type_#id#	Used only for views that use a View Program. Specifies type of the input. Replace the #id# placeholder by the actual id used by the input. The possible input types are: 0 – Numeric 1 – Date 2 – String 3 – Entity 4 – Attribute Set 5 – Query Expression 6 – Boolean (Checkbox) 7 – Color Legend 8 – Drill-hole Settings
input_filter_#id#	Used only for views that use a View Program. Specifies the type of the input. Replace the #id# placeholder by the actual id used by the input. Must be non-zero for type 3 (Entity), contains a numeric value containing a binary mask identifying which types of entities can be potentially selected. Use binary OR to merge different types into a single mask. The values identifying each identity type are (in hexadecimal): 0x01 – Voxel Terrain 0x02 – Voxel Block Model 0x04 – Voxel Generator 0x08 – Realtime Voxel Terrain 0x10 – Indexed Point Cloud
color_legend_mode	Used only when an input of type "Color Legend" is included in the object. Defines which color model should be applied to the object view. Can have one of these two values: 1. The view will use a single color to the entire object 2. The will will apply a gradient based on data values
color_legend_static_r	Used only when an input of type "Color Legend" is included in the object and "color_legend_mode" is set to "1". This property contains the Red component in the static color. This property ranges from 0 to 255.
color_legend_static_g	Used only when an input of type "Color Legend" is included in the object and "color_legend_mode" is set to "1". This property contains the Green component in the static color. This property ranges from 0 to 255.
color_legend_static_b	Used only when an input of type "Color Legend" is included in the object and "color_legend_mode" is set to "1". This property contains the Blue component in the static color. This property ranges from 0 to 255.
color_legend_attribute	Used only when an input of type "Color Legend" is included in the object and "color_legend_mode" is set to "2". This property contains the identifier of the attribute that will be used to derive the color for the object.

color_legend_attribute_index	Used only when an input of type "Color Legend" is included in the object and "color_legend_mode" is set to "2". This property contains the index of the attribute that will be used to derive the color for the object. This index is respective to the full list of available attributes in the object.
color_legend_range_min	Used only when an input of type "Color Legend" is included in the object and "color_legend_mode" is set to "2". This property contains the minimum attribute value to be mapped to the lower end of the color gradient.
color_legend_range_max	Used only when an input of type "Color Legend" is included in the object and "color_legend_mode" is set to "2". This property contains the maximum attribute value to be mapped to the higher end of the color gradient.
color_legend_range_step	Used only when an input of type "Color Legend" is included in the object and "color_legend_mode" is set to "2". This property defines the magnitude of each step in the attribute's space that will mapped as a single color in the the color gradient.
attribute_query_mode	Used only when an input of type "Query Expression" is included in the object. This property defines which attribute filter mode will be used. There are two possible modes: 1. Use attribute query builder 2. Use attribute query string
attribute_query_string	Used only when an input of type "Query Expression" is included in the object and the "attribute_query_mode" is set to "1". This property contains a query string.
attribute_query_builder	Used only when an input of type "Query Expression" is included in the object and the "attribute_query_mode" is set to "1". This property contains a list of space-separated terms. Each term contains four items separated by commas. Each of these items corresponds to: 1. A string denoting the attribute identifier 2. Minimum value for the attribute (inclusive) 3. Maximum value for the attribute (exclusive) 4. Boolean operator to concatenate the expression following in the next term. Use "&" to perform a boolean AND and use "|" to perform a boolean OR.
drillholes_show_collars	Used only when an input of type "Drill-hole Settings" is included in the object. Set to "1" to make the drill hole collar geometry appear in the view. Set to "0" to hide drill hole collars for the object.
drillholes_show_collar_labels	Used only when an input of type "Drill-hole Settings" is included in the object. Set to "1" to make the drill hole collar labels appear in the view. Set to "0" to hide drill hole labels for the object.

drillholes_show_cylinders	Used only when an input of type "Drill-hole Settings" is included in the object. Set to "1" to make the drill hole cylinder geometry appear in the view. Set to "0" to hide drill hole cylinders for the object.
drillholes_show_attributes	Used only when an input of type "Drill-hole Settings" is included in the object. Set to "1" to make the drill hole attribute labels appear in the view. Set to "0" to hide drill hole attribute labels for the object.
drillholes_attributes	Used only when an input of type "Drill-hole Settings" is included in the object. Set to "1" to make the drill hole collar geometry appear in the view. Set to "0" to hide drill hole collars for the object.
tube_legend_attribute	Specifies the attribute ID for the attribute that will be used to compute cylinder diameter.
tube_legend_radius	Used when "tube_legend_scale_mode" equals "1", this defines the radius of all cylinders.
tube_legend_radius_max	Used when "tube_legend_scale_mode" equals "2", this defines the maximum radius of cylinders.
tube_legend_radius_min	Used when "tube_legend_scale_mode" equals "2", this defines the minimum radius of cylinders.
tube_legend_range_max	Used when "tube_legend_scale_mode" equals "2", this defines the maximum value for the attribute that will be mapped to the maximum cylinder radius.
tube_legend_range_min	Used when "tube_legend_scale_mode" equals "2", this defines the minimum value for the attribute that will be mapped to the minimum cylinder radius.
tube_legend_range_step	Used when "tube_legend_scale_mode" equals "2", this defines the discrete step increases in attribute value.
tube_legend_scale_mode	Set to "1" for constant cylinder radius. Set to "2" for attribute-dependent cylinder radius.

File VOXSURF

Defines a Voxel Terrain Surface. This type of entity will have these additional properties:

source	Contains the ID of the entity that was used to produce the Voxel Terrain Surface
source_type	Contains the type of the entity that was used to produce the Voxel Terrain Surface. Can be either “RAWPC” (raw point cloud) or “HM” (heightmaps).

source_ortho	Contains the ID of the raw orthoimagery set used for the Voxel Terrain Surface.
normals	Contains “1” when the Voxel Terrain Surface contains normal maps, “0” otherwise.

File IDXORTHO

Defines an indexed ortho-imagery dataset. This type of entity will have these additional properties:

source	Contains the ID of the ORTHO entity that was used to produce the imagery dataset
image_format	Specifies which image formats should be included in the processed imagery dataset. It is possible to specify multiple targets. Target can be identify by the following values, using space as a separator between targets: • DXT – Hardware compressed textures for Desktop and Web Browsers. • ETC - Hardware compressed textures for Mobile • BGRA - Raw, uncompressed imagery • JPG - Software compressed textures using JPEG compression

File RAWBM

Defines a raw Block Model CSV file. This type of entity will have these additional properties:

column_meta	This property is optional. Please look into the following section to see how to provide information about the columns in the raw Block Model: Extended Block Model Metadata Defines how to interpret and process each column (attribute) found in the Block Model's CSV file. This value must be the Base64 encoding of a single string composed of multiple comma-separated headers. There should be one of these headers for each column in the CVS file. Each header contains four terms separated by spaces. The first term is a unique identifier for the column in the model. It must not contain any spaces nor commas. The second term identifies the type of column, it can be one of the following: • SKIP – The column will be ignored • VALUE – The column contains a numeric value of unknown type • UNIT – The column contains a numeric value that represents a quantity, like tonnage • GRADE – The column contains a numeric value that represents a fraction of bigger whole, e.g. a percentage, grade or parts-per-million. • INT – The column contains a signed integer value • UINT – The column contains an unsigned integer value • ENUM – The column contains values belonging to a discrete set of values, which can be alphanumeric if needed • XC – X coordinate for block centroid • YC – Y coordinate for block centroid • ZC – Z coordinate for block centroid • XL – X coordinate for block origin • YL – Y coordinate for block origin • ZL – Z coordinate for block origin • XS – Block dimension along the X axis • YS – Block dimension along the Y axis • ZS – Block dimension along the Z axis The third term specifies the Level of Detail operation that will be performed for the column. These are the possible values: • AVERAGE – Average the values • ADD – Will add values • MIN – Preserve the smallest absolute values • MAX – Preserve the largest absolute values • MINORITY – Will preserve the least occurring value • MAJORITY – Will preserve the most occurring value • MULTIPLY – Will preserve the smallest volume-weighted values • WEIGHTED(COL_ID) – Performs weighted average using a secondary column, specified by COL_ID.

block_model_type	Optional property to specify the type of block model. The possible values are: "octree" - Denotes an octree-based block model. This is the default value. "varz" - Denotes a variable-z octree (DataMine)

File VOXBM

Defines a Voxel Block Model. This type of entity will have these additional properties:

source	Contains the ID of the raw Block Model data that was used to process the Voxel Block Model
translate_x	Number of units to translate the block model along the X direction.
translate_y	Number of units to translate the block model along the Y direction.
translate_z	Number of units to translate the block model along the Z direction.
scale_x	A proportion (0..1) to scale the block model along the X direction.
scale_y	A proportion (0..1) to scale the block model along the Y direction.
scale_z	A proportion (0..1) to scale the block model along the Z direction.
rotate_x	Euler angle rotation for X axis.
rotate_y	Euler angle rotation for Y axis.
rotate_z	Euler angle rotation for Z axis.
rotation_order	Rotation order. Must be one of: ZXZ, XYX, YZY, ZYZ, XZX or YXY

File VOXMT

Defines a Material Tracking Layer. This type of entity will have these additional properties:

attributes	Contains a list of comma separated attributes that will be tracked by the layer.
tags	Contains a list of comma separated tags that will be tracked by the layer. Unlike attributes, tags are not originally included in the resource model attribute schema.
resource_model	Contains a Base64 string that defines a list of resource models, allowing to define a timestamp and a series of tags for each model . The models are listed as comma-separated terms, where each term corresponds to one resource model. Each term is a list of space-separated entries, following this sequence: <ENTITY_ID> <timestamp> [tag0=value] [tag1=value] ... The first two terms are required: • ENTITY_ID holds the unique identifier for the resource model entity • timestamp is the time value at which the resource model was included in the tracking system The following terms contain custom tag values, in the form: tagId=value The tags specified here should belong to the same set of tags that is specified for the "tags" attribute. Not all tags are required.
terrain_model	Contains the identifier for a Terrain Model entity. This terrain model will be used as a starting point for depletion of the resource models
resource_models_initialized	Set to zero to signal the resource models require re-processing. The system will set this to one after processing is complete.

File IDXPC

Defines an Indexed Point Cloud. This type of entity will have these additional properties:

source	Contains the ID of the raw Point Cloud data that was used to process the Indexed Point Cloud.

File VOXPC

Defines an voxelized point cloud. This type of entity will have these additional properties:

translate_x	Number of units to translate the point cloud along the X direction.
translate_y	Number of units to translate the point cloud along the Y direction.
translate_z	Number of units to translate the point cloud along the Z direction.
scale_x	A proportion (0..1) to scale the point cloud along the X direction.
scale_y	A proportion (0..1) to scale the point cloud along the Y direction.
scale_z	A proportion (0..1) to scale the point cloud along the Z direction.
rotate_x	Euler angle rotation for X axis.
rotate_y	Euler angle rotation for Y axis.
rotate_z	Euler angle rotation for Z axis.
rotation_order	Rotation order. Must be one of: ZXZ, XYX, YZY, ZYZ, XZX or YXY

File IDXMESH

Defines an indexed mesh. This type of entity will have these additional properties:

source	Contains the ID of the raw Block Model data that was used to process the Voxel Block Model.
translate_x	Number of units to translate the mesh along the X direction.

translate_y	Number of units to translate the mesh along the Y direction.
translate_z	Number of units to translate the mesh along the Z direction.
scale_x	A proportion (0..1) to scale the mesh along the X direction.
scale_y	A proportion (0..1) to scale the mesh along the Y direction.
scale_z	A proportion (0..1) to scale the mesh along the Z direction.
rotate_x	Euler angle rotation for X axis.
rotate_y	Euler angle rotation for Y axis.
rotate_z	Euler angle rotation for Z axis.
rotation_order	Rotation order. Must be one of: ZXZ, XYX, YZY, ZYZ, XZX or YXY

File VOXMESH

Defines a voxelized set of meshes. This type of entity will have these additional properties:

source	Contains the ID of the raw mesh entity used to process the Voxelized Mesh.
metadata	Contains a sequence of comma-separated terms. Each term describes an attribute for the voxelized set, and it is made from three space-separated items. These items are, in listing order: 1. Attribute ID - This identifies the attribute in the dataset 2. Attribute Type - Defines which type of attribute to use. Can be set to UID or VALUE. 3. Attribute Default Value - Default value for the attribute. Leave empty for UID type. The set must have one attribute defined as UID, which will allow to link individual meshes in the set to their attributes in the property file. See the Extended Voxelized Mesh Metadata topic for more information on how metadata for Voxelized Mesh Sets is defined.

translate_x	Number of units to translate the mesh along the X direction.
translate_y	Number of units to translate the mesh along the Y direction.
translate_z	Number of units to translate the mesh along the Z direction.
scale_x	A proportion (0..1) to scale the mesh along the X direction.
scale_y	A proportion (0..1) to scale the mesh along the Y direction.
scale_z	A proportion (0..1) to scale the mesh along the Z direction.
rotate_x	Euler angle rotation for X axis.
rotate_y	Euler angle rotation for Y axis.
rotate_z	Euler angle rotation for Z axis.
rotation_order	Rotation order. Must be one of: ZXZ, XYX, YZY, ZYZ, XZX or YXY

File RAWHM

Defines a set of orthoimagery that describes elevation. This type of entity will have these additional properties:

elevation_min	Contains the minimum elevation value for the set. This value maps to a black pixel.
elevation_max	Contains the maximum elevation value for the set. This value maps to a white pixel.

File RAWDH

Defines source data for a Drill Hole entity. While this entity does not have any special properties, the entity must include a process.meta file. This is covered in the Extended Drill Hole Metadata section.

File PROGRAM

Defines a voxel generator program. This type of entity will have these additional properties:

code_origin	Contains the type of code used by the program. Currently, two types are supported: • "PYTHON" - The code is a Python program • "NETASM" - The code is provided as a compiled .Net assembly
code	If code_origin is "PYTHON", this property contains the Python code for the program, in Base64 format. If code_origin is "NETASM", this property identifies the entry-point class for the Lambda program in the form: <FullyNamespacedClassName>:<DLLName>
program_type	Contains the type of the program. The program type can be one of the following: • “VOXEL” – The program is for a voxel entity generator. This type only accepts "PYTHON" as code_origin. • “VIEW” – The program creates a view This type only accepts "PYTHON" as code_origin. • “REPORT” – The program creates a report. Reports can be have "PYTHON" or "NETASM" as code_origin

File VOXGEN

Defines a spatial entity that is the output of a voxel generator program. This type of entity will have these additional properties:

program	Contains the ID of the voxel generator program that will be evaluated.
input_value_#id#	Contains the value for the input. Replace the #id# placeholder by the actual id used by the input.
input_label_#id#	Contains the label for the input. Replace the #id# placeholder by the actual id used by the input.
input_type_#id#	Specifies type of the input. Replace the #id# placeholder by the actual id used by the input. The possible input types are: 0 – Numeric 1 – Date 2 – String 3 – Entity 4 – Attribute Set 5 – Query Expression

input_filter_#id#	For Specifies type of the input. Replace the #id# placeholder by the actual id used by the input. Must be non-zero for type 3 (Entity), contains a numeric value containing a binary mask identifying which types of entities can be potentially selected. Use binary OR to merge different types into a single mask. The values identifying each identity type are (in hexadecimal): 0x01 – Voxel Terrain 0x02 – Voxel Block Model 0x04 – Voxel Generator 0x08 – Realtime Voxel Terrain 0x10 – Indexed Point Cloud

File REPORT

Defines a report entity. A report entity takes a Program and runs it over a specified Region. The program gets to inspect the spatial data int the region and it can compute useful information by combining the different spatial attributes using custom logic.

Creating a report entity is not enough to produce the desired report files. After the entity has been created, it is also necessary to trigger its processing by a second REST call. Please visit the “Triggering reports” section to see how report jobs can be triggered.

This type of entity will have these additional properties:

program	Contains the ID of the report program that will be evaluated.
lod	Specifies at which Level of Detail the report will run.
input_value_#id#	Contains the value for the input. Replace the #id# placeholder by the actual id used by the input.
input_label_#id#	Contains the label for the input. Replace the #id# placeholder by the actual id used by the input.
input_type_#id#	Specifies type of the input. Replace the #id# placeholder by the actual id used by the input. The possible input types are: 0 – Numeric 1 – Date 2 – String 3 – Entity 4 – Attribute Set 5 – Query Expression

input_filter_#id#	For Specifies type of the input. Replace the #id# placeholder by the actual id used by the input. Must be non-zero for type 3 (Entity), contains a numeric value containing a binary mask identifying which types of entities can be potentially selected. Use binary OR to merge different types into a single mask. The values identifying each identity type are (in hexadecimal): 0x01 – Voxel Terrain 0x02 – Voxel Block Model 0x04 – Voxel Generator 0x08 – Realtime Voxel Terrain 0x10 – Indexed Point Cloud

File EXPORT

Defines an export entity. Export entities contain files in standard formats that were produced out of the spatial data in the project. For instance, a user may desire to export a mashup of terrain surfaces as a single terrain mesh. In this case, there would be one export entity that stores the mesh as an internal file.

Creating an export entity is not enough to produce the desired files. After the entity has been created, it is also necessary to trigger its processing by a second REST call. Please visit the “Triggering jobs” section to see how export jobs can be triggered.

This type of entity will have these additional properties:

entity	Contains the ID of the entity that will be used as a source for the export operation
lod	Specifies at which Level of Detail the report will run.
region	A region can be expressed in the REST interface as a single string, containing the following format: 1,<min_height>,<max_height>,<polygon_point0_x>, <polygon_point0_y>, <polygon_point1_x>, <polygon_point1_y>, ... The format is a list of numbers separated by commas. The first number must be 1, this identifies this is an extruded polygon region. The second two numbers are world coordinates for the vertical bounds of the region. The following numbers are the X, Y coordinates of each region point. The polygon points are defined in counter-clockwise order. For example, the following region string defines a 10-unit cube centered in the project’s origin: 1,-5,5,-5, 5,5,5,5,-5,-5,-5

export_type	Specifies the type of data that will be exported. Can be one of these: • mesh – Exports the surface mesh for the entity (.OBJ format) • points – Exports a point cloud (CVS format) • ortho – Exports ortho-imagery (TIF+TFW)

Export entities have one special file attachment. This file is named “export.zip” and contains a ZIP archive of the exported data.

Region entity

Entities of type “REGION” describe a 3D scope where export jobs, reports and other tasks will be performed.

This type of entity will have these additional properties:

region	A region can be expressed in the REST interface as a single string, containing the following format: 1,<min_height>,<max_height>,<polygon_point0_x>, <polygon_point0_y>, <polygon_point1_x>, <polygon_point1_y>, ... The format is a list of numbers separated by commas. The first number must be 1, this identifies this is an extruded polygon region. The second two numbers are world coordinates for the vertical bounds of the region. The following numbers are the X, Y coordinates of each region point. The polygon points are defined in counter-clockwise order. For example, the following region string defines a 10-unit cube centered in the project’s origin: 1,-5,5,-5, 5,5,5,5,-5,-5,-5

Spatial Properties

Some entities are of spatial nature, in these cases, the entity will have a set of properties related to the coordinate system used by the entity. The following table lists these properties:

<table>
<tr><td>coord_hdatum</td><td>Contains the horizontal datum for the Coordinate Reference Systems (CRS). These are the types currently supported:

<table>
<tr><td>wgs72</td><td>World Geodetic System 1972 (WGS 72)</td></tr>
<tr><td>wgs84</td><td>World Geodetic System 1984 (WGS 84)</td></tr>
<tr><td>grs80</td><td>Geodetic Reference System 1980 (GRS 80)</td></tr>
<tr><td>nad27</td><td>North American Datum 1927 (NAD27)</td></tr>
<tr><td>nad83</td><td>North American Datum (NAD83)</td></tr>
<tr><td>nad83_2011</td><td>North American Datum (NAD83 2011)</td></tr>
<tr><td>nad83_harn</td><td>North American Datum (NAD83 HARN)</td></tr>
<tr><td>nad83_csrs</td><td>North American Datum (NAD83 CSRS)</td></tr>
<tr><td>gda94</td><td>Geocentric Datum of Australia 1994 (GDA94)</td></tr>
<tr><td>etrs89</td><td>European Terrestrial Reference System 1989 (ETRS89)</td></tr>
<tr><td>agd66</td><td>Australian Geodetic Datum 1966 (AGD66)</td></tr>
<tr><td>clarke80</td><td>Modified Clarke 1880</td></tr>
</table></td></tr>
<tr><td>coord_origin_x</td><td>Contains the coordinate of axis X for the origin of the project.</td></tr>
</table>

<table>
<tr><td>coord_origin_y</td><td>Contains the coordinate of axis Y for the origin of the project.</td></tr>
<tr><td>coord_origin_z</td><td>Contains the coordinate of axis Z for the origin of the project.</td></tr>
<tr><td>coord_projection</td><td>Contains the data projection for the Coordinate Reference Systems (CRS). These are the types currently supported:

<table><tr><td>latlong</td><td>Latitude Longitude</td></tr><tr><td>longlat</td><td>Longitude Latitude</td></tr><tr><td>ecef</td><td>Earth Centered Earth Fixed</td></tr><tr><td>merc</td><td>World Mercator</td></tr><tr><td>webmerc</td><td>Web Mercator</td></tr><tr><td>utm</td><td>Universal Transverse Mercator (UTM)</td></tr><tr><td>tm</td><td>Transverse Mercator</td></tr><tr><td>lcc</td><td>Lambert conformal conic</td></tr><tr><td>aeac</td><td>AEAC</td></tr><tr><td>amg</td><td>AMG</td></tr><tr><td>mga</td><td>MGA</td></tr><tr><td>lo</td><td>Lo</td></tr></table></td></tr>
<tr><td>coord_projection_aeac_falseEasting</td><td>Contains the false Easting. It is only required if the projection is aeac.</td></tr>
<tr><td>coord_projection_aeac_falseNorthing</td><td>Contains the false Northing. It is only required if the projection is aeac.</td></tr>
<tr><td>coord_projection_aeac_firstStdParallelDeg</td><td>Contains the first Std Parallel Degree. It is only required if the projection is aeac.</td></tr>
<tr><td>coord_projection_aeac_latOfOriginDeg</td><td>Contains the latitud of Origin Degree. It is only required if the projection is aeac.</td></tr>
<tr><td>coord_projection_aeac_longOfOriginDeg</td><td>Contains the longitud of Origin Degree. It is only required if the projection is aeac.</td></tr>
<tr><td>coord_projection_aeac_secondStdParallelDeg</td><td>Contains the second Std Parallel Degree. It is only required if the projection is aeac.</td></tr>
<tr><td>coord_projection_amg_zone</td><td>Contains the zone value. It is a number between 48 and 58. It is only required if the projection is amg.</td></tr>
</table>

coord_projection_mga_zone	Contains the zone value. It is a number between 48 and 58. It is only required if the projection is mga.
coord_projection_lo_zone	Contains the zone value. It is one of these numbers: [15, 17, 19, 21, 23, 25, 27, 31, 33]. It is only required if the projection is lo.
coord_projection_lcc_falseEasting	Contains the false Easting. It is only required if the projection is lcc.
coord_projection_lcc_falseNorthing	Contains the false Northing. It is only required if the projection is lcc.
coord_projection_lcc_firstStdParallelDeg	Contains the first Std Parallel Degree. It is only required if the projection is lcc.
coord_projection_lcc_latOfOriginDeg	Contains the latitud of Origin Degree. It is only required if the projection is lcc.
coord_projection_lcc_longOfOriginDeg	Contains the longitud of Origin Degree. It is only required if the projection is lcc.
coord_projection_lcc_secondStdParallelDeg	Contains the second Std Parallel Degree. It is only required if the projection is lcc.
coord_projection_tm_falseEasting	Contains the false Easting. It is only required if the projection is tm.
coord_projection_tm_falseNorthing	Contains the false Northing. It is only required if the projection is tm.
coord_projection_tm_latOriginDeg	Contains the latitude of Origin Degree. It is only required if the projection is tm.
coord_projection_tm_longMeridianDeg	Contains the longitude of Origin Degree. It is only required if the projection is tm.
coord_projection_tm_scaleFactor	Contains the scale factor. It is only required if the projection is tm.
coord_unit	Contains the unit of measure of the horizontal data. These are the types currently supported: • meter • feet • survey_feet
coord_vunit	Contains the unit of measure of the horizontal data. These are the types currently supported: • meter • feet • survey_feet

coord_vdatum	Contains the vertical datum for the Coordinate Reference Systems (CRS). These are the types currently supported:

none	None
wgs84	World Geodetic System 1984 (WGS 84)
ngvd29	National Geodetic Vertical Datum 1929 (NGVD 29)
navd29	North American Vertical Datum 29 (NAVD 29)
cgvd2013	Canadian Geodetic Vertical Datum 2013 (CGVD 2013)
cgvd28	Canadian Geodetic Vertical Datum 1928 (CGVD 28)
dvr90	Denmark Vertical Reference (DVR 90)
nn2000	Norway Normal Null 2000 (NN2000)
nn54	Norway Normal Null 1954 (NN54)
dhhn92	Deutsches Haupthoehennetz 1992 (DHHN92)
navd88	North American Vertical Datum 1988 (NAVD 88)
navd88_geoid12b	North American Vertical Datum 1988 (NAVD 88 GEOID12B)
navd88_geoid12a	North American Vertical Datum 1988 (NAVD 88 GEOID12A)
navd88_geoid12	North American Vertical Datum 1988 (NAVD 88 GEOID12)
navd88_geoid09	North American Vertical Datum 1988 (NAVD 88 GEOID09)
navd88_geoid06	North American Vertical Datum 1988 (NAVD 88 GEOID06)
navd88_geoid03	North American Vertical Datum 1988 (NAVD 88 GEOID03)
navd88_geoid99	North American Vertical Datum 1988 (NAVD 88 GEOID99)
navd88_geoid96	North American Vertical Datum 1988 (NAVD 88 GEOID96)

Extended Block Model Metadata

When submitting files for a new Block Model raw dataset (file_type RAWBM), the system expects an additional file named:

- **process.meta**

This is a text file in INI format that contains additional information on how the block model dataset must be constructed out of the accompanying files.

The file must contain a single section, named [Configuration]

Column_Count	Specifies how may attribute columns will be read from the block model.
Column<N>_ID	Specifies the name for the attribute as listed in the block model headers. The <N> placeholder is replaced by the attribute index, which is zero-based.
Column<N>_Class	Specifies the class for the attribute. The <N> placeholder is replaced by the attribute index, which is zero-based. The possible values are: XL – X coordinate of the block's origin XC – X coordinate of the block's centroid XR – X coordinate of the block's end XS – Block's dimension along the X axis YL – Y coordinate of the block's origin YC – Y coordinate of the block's centroid YR – Y coordinate of the block's end YS – Block's dimension along the Y axis ZL – Z coordinate of the block's origin ZC – Z coordinate of the block's centroid ZR – Z coordinate of the block's end ZS – Block's dimension along the Z axis Value – The values in this column will be interpreted as numbers. This setting should be used for grades and any other numeric quantity. Enum – The values in this column belong to a set with a limited number of elements that appear many times over in the block model.

Column<N>_Aggregator	Specifies the aggregator for the attribute. The <N> placeholder is replaced by the attribute index, which is zero-based. The possible values are: AVG – Average the values. WAVG - Performs weighted average. SUM – Will add values. MIN – Preserve the smallest absolute values MAX – Preserve the largest absolute values MINORITY – Will preserve the least occurring value MAJORITY – Will preserve the most occurring value MULTIPLY – Will multiply values
Column<N>_DefaultValue	Specifies the default value for the attribute. The <N> placeholder is replaced by the attribute index, which is zero-based.
Column<N>_Unit	Specifies the unit of measure for the attribute. The <N> placeholder is replaced by the attribute index, which is zero-based.
Column<N>_Label	Specifies a readable label for the attribute name. The <N> placeholder is replaced by the attribute index, which is zero-based.

Example process.meta file:

[Configuration]

Column_Count=35

Column0_ID=xlength

Column0_Class=XS

Column0_Aggregator=Avg

Column0_DefaultValue=0

Column1_ID=xworld

Column1_Class=XC

Column1_Aggregator=Avg

Column1_DefaultValue=0

Column2_ID=ylength

Column2_Class=YS

Column2_Aggregator=Avg

Column2_DefaultValue=0

Column3_ID=yworld

Column3_Class=YC

Column3_Aggregator=Avg

Column3_DefaultValue=0

Column4_ID=zlength

Column4_Class=ZS

Column4_Aggregator=Avg

Column4_DefaultValue=0

Column5_ID=zworld

Column5_Class=ZC

Column5_Aggregator=Avg

Column5_DefaultValue=0

Column6_ID=volume

Column6_Class=Value

Column6_Aggregator=Avg

Column6_DefaultValue=0

Column7_ID=flag_block_count

Column7_Class=Value

Column7_Aggregator=Avg

Column7_DefaultValue=0

Column8_ID=metreg_cyclone_feed_al2o3

Column8_Class=Value

Column8_Aggregator=Avg

Column8_DefaultValue=0

Column9_ID=metreg_cyclone_feed_fe

Column9_Class=Value

Column9_Aggregator=Avg

Column9_DefaultValue=0

Column10_ID=metreg_cyclone_feed_sio2

Column10_Class=Value

Column10_Aggregator=Avg

Column10_DefaultValue=0

Column11_ID=metreg_cyclone_feed_tonnes

Column11_Class=Value

Column11_Aggregator=Avg

Column11_DefaultValue=0

Column12_ID=mine_fines_al2o3

Column12_Class=Value

Column12_Aggregator=Avg

Column12_DefaultValue=0

Column13_ID=mine_fines_fe

Column13_Class=Value

Column13_Aggregator=Avg

Column13_DefaultValue=0

Column14_ID=mine_fines_sio2

Column14_Class=Value

Column14_Aggregator=Avg

Column14_DefaultValue=0

Column15_ID=mine_fines_tonnes

Column15_Class=Value

Column15_Aggregator=Avg

Column15_DefaultValue=0

Column16_ID=mine_lump_al2o3

Column16_Class=Value

Column16_Aggregator=Avg

Column16_DefaultValue=0

Column17_ID=mine_lump_fe

Column17_Class=Value

Column17_Aggregator=Avg

Column17_DefaultValue=0

Column18_ID=mine_lump_sio2

Column18_Class=Value

Column18_Aggregator=Avg

Column18_DefaultValue=0

Column19_ID=mine_lump_tonnes

Column19_Class=Value

Column19_Aggregator=Avg

Column19_DefaultValue=0

Column20_ID=mine_tails_al2o3

Column20_Class=Value

Column20_Aggregator=Avg

Column20_DefaultValue=0

Column21_ID=mine_tails_fe

Column21_Class=Value

Column21_Aggregator=Avg

Column21_DefaultValue=0

Column22_ID=mine_tails_sio2

Column22_Class=Value

Column22_Aggregator=Avg

Column22_DefaultValue=0

Column23_ID=mine_tails_tonnes

Column23_Class=Value

Column23_Aggregator=Avg

Column23_DefaultValue=0

Column24_ID=pit_al2o3

Column24_Class=Value

Column24_Aggregator=Avg

Column24_DefaultValue=0

Column25_ID=pit_density_dry_gp

Column25_Class=Value

Column25_Aggregator=Avg

Column25_DefaultValue=0

Column26_ID=pit_fe

Column26_Class=Value

Column26_Aggregator=Avg

Column26_DefaultValue=0

Column27_ID=pit_sio2

Column27_Class=Value

Column27_Aggregator=Avg

Column27_DefaultValue=0

Column28_ID=pit_tonnes_gp

Column28_Class=Value

Column28_Aggregator=Avg

Column28_DefaultValue=0

Column29_ID=pit_volume

Column29_Class=Value

Column29_Aggregator=Avg

Column29_DefaultValue=0

Column30_ID=flag_water_table_status

Column30_Class=Enum

Column30_Aggregator=Set

Column30_DefaultValue="-"

Column31_ID=geo_domain

Column31_Class=Enum

Column31_Aggregator=Set

Column31_DefaultValue="-"

Column32_ID=id_product

Column32_Class=Enum

Column32_Aggregator=Set

Column32_DefaultValue="-"

Column33_ID=id_product_3

Column33_Class=Enum

Column33_Aggregator=Set

Column33_DefaultValue="-"

Column34_ID=id_product_4

Column34_Class=Enum

Column34_Aggregator=Set

Column34_DefaultValue="-"

Extended Drill Hole Metadata

When submitting files for a new Drill Hole raw dataset (file_type RAWDH), the system expects an additional file named:

- **process.meta**

This is a text file in INI format that contains additional information on how the drill hole dataset must be constructed out of the accompanying files.

The file must contain a single section, named [Configuration]

This section must contain a property named "Desurveyed", which indicates if the Drill Hole dataset must be submitted to a desurveying process before ingestion.

Set Desureveyed = 0 to request the system to desurvey the holes and create segment interpolation.

Set Desureveyed = 1 to ingest the drill hole segments as they appear in the source files.

The rest of the properties found in the [Configuration] section depend on whether desurveying was chosen. The two following sections cover these alternatives.

Metadata for Desurveyed (Composite) Drill Hole datasets

In this variant, the process.meta file must contain the following properties:

Desurveyed	Must be equal to 1.
AssayFileCount	Specifies how many assay files will be included.
AssayFile<N>	Specifies an assay file name. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for AssayFile0, AssayFile1 and AssayFile2. The <N> placeholder is replaced by the file index.

Column_Assay<N>_ID	Specifies the index of the column in the assay file that contains the drill hole ID. This is an unique identifier per hole, which is shared across several files in the dataset. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_ID, Column_Assay1_ID and Column_Assay2_ID. The <N> placeholder is replaced by the file index.
Column_Assay<N>_From	Specifies the index of the column in the assay file that contains drill hole segment start depths. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_From, Column_Assay1_From and Column_Assay2_From. The <N> placeholder is replaced by the file index.
Column_Assay<N>_To	Specifies the index of the column in the assay file that contains drill hole segment end depths. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_To, Column_Assay1_To and Column_Assay2_To. The <N> placeholder is replaced by the file index.
Column_Assay<N>_X	Specifies the index of the column in the assay file that contains drill hole segment start X coordinate. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_X, Column_Assay1_X and Column_Assay2_X. The <N> placeholder is replaced by the file index.
Column_Assay<N>_Y	Specifies the index of the column in the assay file that contains drill hole segment start Y coordinate. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_Y, Column_Assay1_Y and Column_Assay2_Y. The <N> placeholder is replaced by the file index.
Column_Assay<N>_Z	Specifies the index of the column in the assay file that contains drill hole segment start Z coordinate. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_Z, Column_Assay1_Z and Column_Assay2_Z. The <N> placeholder is replaced by the file index.
Column_Assay<N>_Attribute_Columns	Specifies how may attribute columns will be read from an assay file. There will be one of these properties for each assay file. The <N> placeholder is replaced by the file index.

Column_Assay<N>_Attribute_Column<M>_Index	Specifies the index of the column that contains the attribute values for a given attribute in a given assay file. The <N> placeholder is replaced by the assay file index, and the <M> placeholder is replaced by the attribute index. For instance, if the third assay file has three attributes to define, the properties.meta file will include values for Column_Assay2_Attribute_Column0_Index, Column_Assay2_Attribute_Column1_Index and Column_Assay2_Attribute_Column2_Index.
Column_Assay<N>_Attribute_Column<M>_Name	Specifies the name of an attribute in a given assay file. The <N> placeholder is replaced by the assay file index, and the <M> placeholder is replaced by the attribute index. For instance, if the third assay file has three attributes to define, the properties.meta file will include values for Column_Assay2_Attribute_Column0_Name, Column_Assay2_Attribute_Column1_Name and Column_Assay2_Attribute_Column2_Name.
Column_Assay<N>_Attribute_Column<M>_Type	Specifies the type of an attribute in a given assay file. 0 - Attribute is of Float type 1 - Attribute of of Alphanumerical type Other types than float are not currently supported. The <N> placeholder is replaced by the assay file index, and the <M> placeholder is replaced by the attribute index. For instance, if the third assay file has three attributes to define, the properties.meta file will include values for Column_Assay2_Attribute_Column0_Type, Column_Assay2_Attribute_Column1_Type and Column_Assay2_Attribute_Column2_Type.
Column_Assay<N>_Attribute_Column<M>_Unit	Specifies the Units of Measure for the attribute
Column_Assay<N>_Attribute_Column<M>_Label	Specifies a readable label for the attribute name. The <N> placeholder is replaced by the attribute index, which is zero-based

Example of properties.meta for a desurveyed (composite) drill hole dataset:

[Configuration]

Desurveyed=1

AssayFileCount=2

AssayFile0=Granofels_composites.csv

AssayFile1=Pyroxenite_composites.csv

Column_Assay0_ID=0

Column_Assay0_From=4

Column_Assay0_To=5

Column_Assay0_X=1

Column_Assay0_Y=2

Column_Assay0_Z=3

Column_Assay0_Attribute_Columns = 11

Column_Assay0_Attribute_Column0_Index=7

Column_Assay0_Attribute_Column0_Name=A0

Column_Assay0_Attribute_Column0_Type=0

Column_Assay0_Attribute_Column1_Index=8

Column_Assay0_Attribute_Column1_Name=B0

Column_Assay0_Attribute_Column1_Type=0

Column_Assay0_Attribute_Column2_Index=9

Column_Assay0_Attribute_Column2_Name=C0

Column_Assay0_Attribute_Column2_Type=0

Column_Assay0_Attribute_Column3_Index=10

Column_Assay0_Attribute_Column3_Name=PGE

Column_Assay0_Attribute_Column3_Type=0

Column_Assay0_Attribute_Column4_Index=11

Column_Assay0_Attribute_Column4_Name=PT

Column_Assay0_Attribute_Column4_Type=0

Column_Assay0_Attribute_Column5_Index=12

Column_Assay0_Attribute_Column5_Name=PD

Column_Assay0_Attribute_Column5_Type=0

Column_Assay0_Attribute_Column6_Index=13

Column_Assay0_Attribute_Column6_Name=RH

Column_Assay0_Attribute_Column6_Type=0

Column_Assay0_Attribute_Column7_Index=14

Column_Assay0_Attribute_Column7_Name=AU

Column_Assay0_Attribute_Column7_Type=0

Column_Assay0_Attribute_Column8_Index=15

Column_Assay0_Attribute_Column8_Name=CU

Column_Assay0_Attribute_Column8_Type=0

Column_Assay0_Attribute_Column9_Index=16

Column_Assay0_Attribute_Column9_Name=NI

Column_Assay0_Attribute_Column9_Type=0

Column_Assay0_Attribute_Column10_Index=17

Column_Assay0_Attribute_Column10_Name=SG

Column_Assay0_Attribute_Column10_Type=0

Column_Assay1_ID=0

Column_Assay1_From=4

Column_Assay1_To=5

Column_Assay1_X=1

Column_Assay1_Y=2

Column_Assay1_Z=3

Column_Assay1_Attribute_Columns = 10

Column_Assay1_Attribute_Column0_Index=7

Column_Assay1_Attribute_Column0_Name=A0

Column_Assay1_Attribute_Column0_Type=0

Column_Assay1_Attribute_Column1_Index=8

Column_Assay1_Attribute_Column1_Name=B0

Column_Assay1_Attribute_Column1_Type=0

Column_Assay1_Attribute_Column2_Index=9

Column_Assay1_Attribute_Column2_Name=PGE

Column_Assay1_Attribute_Column2_Type=0

Column_Assay1_Attribute_Column3_Index=10

Column_Assay1_Attribute_Column3_Name=PT

Column_Assay1_Attribute_Column3_Type=0

Column_Assay1_Attribute_Column4_Index=11

Column_Assay1_Attribute_Column4_Name=PD

Column_Assay1_Attribute_Column4_Type=0

Column_Assay1_Attribute_Column5_Index=12

Column_Assay1_Attribute_Column5_Name=RH

Column_Assay1_Attribute_Column5_Type=0

Column_Assay1_Attribute_Column6_Index=13

Column_Assay1_Attribute_Column6_Name=AU

Column_Assay1_Attribute_Column6_Type=0

Column_Assay1_Attribute_Column7_Index=14

Column_Assay1_Attribute_Column7_Name=CU

Column_Assay1_Attribute_Column7_Type=0

Column_Assay1_Attribute_Column8_Index=15

Column_Assay1_Attribute_Column8_Name=NI

Column_Assay1_Attribute_Column8_Type=0

Column_Assay1_Attribute_Column9_Index=16

Column_Assay1_Attribute_Column9_Name=SG

Column_Assay1_Attribute_Column9_Type=0

Metadata for Drill Hole datasets that require desurveying

In this variant, the process.meta file must contain the following properties:

Desurveyed	Must be equal to 0.
CollarFile	Specifies the filename for the file that contains the drill hole collars.
Column_Collars_ID	Specifies the index of the column in the collars file that contains drill hole IDs.
Column_Collars_X	Specifies the index of the column in the collars file that contains the X coordinate for the drill hole collar.
Column_Collars_Y	Specifies the index of the column in the collars file that contains the Y coordinate for the drill hole collar.
Column_Collars_Z	Specifies the index of the column in the collars file that contains the Z coordinate for the drill hole collar.
Column_Collars_Depth	Specifies the index of the column in the collars file that contains the depth (length) for the drill hole.
SurveyFile	Specifies the filename for the file that contains the drill hole surveys.
Column_Survey_ID	Specifies the index of the column in the survey file that contains drill hole IDs.
Column_Survey_Depth	Specifies the index of the column in the surveys file that contains the depth for each survey.
Column_Survey_Azimuth	Specifies the index of the column in the surveys file that contains the azimuth for each survey.
Column_Survey_Dip	Specifies the index of the column in the surveys file that contains the dip for each survey.
Column_Survey_Type	Specifies the type of interpolation that will be used to reconstruct drill hole paths in space. Specify one of the following values: 0 - Tangential First (use for acQuire sets) 1 - Tangential Second 2 - Minimum Curvature 3 - Balanced Tangential 4 - Average Angle 5 - Radius of Curvature
AssayFileCount	Specifies how many assay files will be included.

AssayFile<N>	Specifies an assay file name. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for AssayFile0, AssayFile1 and AssayFile2. The <N> placeholder is replaced by the file index.
Column_Assay<N>_ID	Specifies the index of the column in the assay file that contains the drill hole ID. This is an unique identifier per hole, which is shared across several files in the dataset. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_ID, Column_Assay1_ID and Column_Assay2_ID. The <N> placeholder is replaced by the file index.
Column_Assay<N>_From	Specifies the index of the column in the assay file that contains drill hole segment start depths. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_From, Column_Assay1_From and Column_Assay2_From. The <N> placeholder is replaced by the file index.
Column_Assay<N>_To	Specifies the index of the column in the assay file that contains drill hole segment end depths. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_To, Column_Assay1_To and Column_Assay2_To. The <N> placeholder is replaced by the file index.
Column_Assay<N>_X	Specifies the index of the column in the assay file that contains drill hole segment start X coordinate. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_X, Column_Assay1_X and Column_Assay2_X. The <N> placeholder is replaced by the file index.
Column_Assay<N>_Y	Specifies the index of the column in the assay file that contains drill hole segment start Y coordinate. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_Y, Column_Assay1_Y and Column_Assay2_Y. The <N> placeholder is replaced by the file index.
Column_Assay<N>_Z	Specifies the index of the column in the assay file that contains drill hole segment start Z coordinate. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_Z, Column_Assay1_Z and Column_Assay2_Z. The <N> placeholder is replaced by the file index.
Column_Assay<N>_Attribute_Columns	Specifies how may attribute columns will be read from an assay file. There will be one of these properties for each assay file. The <N> placeholder is replaced by the file index.

Column_Assay<N>_Attribute_Column<M>_Index	Specifies the index of the column that contains the attribute values for a given attribute in a given assay file. The <N> placeholder is replaced by the assay file index, and the <M> placeholder is replaced by the attribute index. For instance, if the third assay file has three attributes to define, the properties.meta file will include values for Column_Assay2_Attribute_Column0_Index, Column_Assay2_Attribute_Column1_Index and Column_Assay2_Attribute_Column2_Index.
Column_Assay<N>_Attribute_Column<M>_Name	Specifies the name of an attribute in a given assay file. The <N> placeholder is replaced by the assay file index, and the <M> placeholder is replaced by the attribute index. For instance, if the third assay file has three attributes to define, the properties.meta file will include values for Column_Assay2_Attribute_Column0_Name, Column_Assay2_Attribute_Column1_Name and Column_Assay2_Attribute_Column2_Name.
Column_Assay<N>_Attribute_Column<M>_Type	Specifies the type of an attribute in a given assay file. 0 - Attribute is of Float type 1 - Attribute of of Alphanumerical type Other types than float are not currently supported. The <N> placeholder is replaced by the assay file index, and the <M> placeholder is replaced by the attribute index. For instance, if the third assay file has three attributes to define, the properties.meta file will include values for Column_Assay2_Attribute_Column0_Type, Column_Assay2_Attribute_Column1_Type and Column_Assay2_Attribute_Column2_Type.
Column_Assay<N>_Attribute_Column<M>_Units	Specifies the Units of Measure for the attribute
Column_Assay<N>_Attribute_Column<M>_Label	Specifies a readable label for the attribute name. The <N> placeholder is replaced by the attribute index, which is zero-based

Example of properties.meta for a drill hole dataset that requires desurveying:

[Configuration]

CollarFile="CollarAll.csv"

SurveyFile="SurveyAll.csv"

AssayFileCount=2

AssayFile0="GeologyAll.csv"

AssayFile1="AssayAll.csv"

Column_Collars_ID=0

Column_Collars_X=2

Column_Collars_Y=3

Column_Collars_Z=4

Column_Collars_Depth=5

Column_Survey_ID = 0

Column_Survey_Depth = 2

Column_Survey_Azimuth = 3

Column_Survey_Dip = 4

Column_Survey_Type = 5

Column_Assay0_ID = 0

Column_Assay0_From = 2

Column_Assay0_To = 3

Column_Assay0_Attribute_Columns = 1

Column_Assay0_Attribute_Column0_Index=4

Column_Assay0_Attribute_Column0_Name=Abbrv

Column_Assay0_Attribute_Column0_Type=1

Column_Assay1_ID = 0

Column_Assay1_From = 3

Column_Assay1_To = 4

Column_Assay1_Attribute_Columns = 10

Column_Assay1_Attribute_Column0_Index=5

Column_Assay1_Attribute_Column0_Name=Ag_ppm_BEST

Column_Assay1_Attribute_Column0_Type=0

Column_Assay1_Attribute_Column1_Index=6

Column_Assay1_Attribute_Column1_Name=As_pct_BEST

Column_Assay1_Attribute_Column1_Type=0

Column_Assay1_Attribute_Column2_Index=7

Column_Assay1_Attribute_Column2_Name=Cd_ppm_BEST

Column_Assay1_Attribute_Column2_Type=0

Column_Assay1_Attribute_Column3_Index=8

Column_Assay1_Attribute_Column3_Name=Co_ppm_BEST

Column_Assay1_Attribute_Column3_Type=0

Column_Assay1_Attribute_Column4_Index=9

Column_Assay1_Attribute_Column4_Name=Cu_ppm_BEST

Column_Assay1_Attribute_Column4_Type=0

Column_Assay1_Attribute_Column5_Index=10

Column_Assay1_Attribute_Column5_Name=Fe_pct_BEST

Column_Assay1_Attribute_Column5_Type=0

Column_Assay1_Attribute_Column6_Index=11

Column_Assay1_Attribute_Column6_Name=Mg_pct_BEST

Column_Assay1_Attribute_Column6_Type=0

Column_Assay1_Attribute_Column7_Index=12

Column_Assay1_Attribute_Column7_Name=Ni_ppm_BEST

Column_Assay1_Attribute_Column7_Type=0

Column_Assay1_Attribute_Column8_Index=13

Column_Assay1_Attribute_Column8_Name=Pb_pct_BEST

Column_Assay1_Attribute_Column8_Type=0

Column_Assay1_Attribute_Column9_Index=14

Column_Assay1_Attribute_Column9_Name=Zn_pct_BEST

Column_Assay1_Attribute_Column9_Type=0

Extended Voxelized Mesh Metadata

In order to produce voxelized mesh sets, the mesh source RAWMESH entity requires additional files included along with the source OBJ meshes:

- **meta_attributes.csv**

The file is used to enumerate the meshes that will be included in the mesh dataset, and it also allows assigning different attribute values to the volume occupied by the mesh.

The first row in the file contains the attribute identifiers used for each column. There must be one column that lists the mesh file names names.

For instance, assume the user wants to import a set of five meshes. For each mesh, the user will supply one OBJ file, named:

1. MESH_01.OBJ
2. MESH_02.OBJ
3. MESH_03.OBJ
4. MESH_04.OBJ
5. MESH_05.OBJ

Let's also assume each mesh assigns a different value for the following attributes: ROCKID, DOMAIN, ZONE, INDEX

In this case, the meta_attributes.csv would look like:

Filename	INDEX	ROCKID	DOMAIN	DENSITY
MESH_01	1	23	10	15
MESH_02	2	34	1	17
MESH_03	3	85	3	21
MESH_04	4	23	12	37
MESH_05	5	1	7	1

Note the file names do not include the .OBJ extension here.

The schema for this file must be defined in the "metadata" property for the RAWMESH entity.

This property contains a sequence of comma-separated terms.

Each term describes an attribute for the voxelized set, in the same order as they appear in the meta_attributes.csv file,

and it is made from four space-separated items. These items are, in listing order:

1. Attribute ID - This identifies the attribute in the dataset
2. Attribute Type - Defines which type of attribute to use. Can be set to UID, VALUE or SET.
3. Attribute Default Value - Default value for the attribute. Leave empty for UID type.
4. Attribute Unit of Measure - It is an optional text describing the Unit of Measure for the attribute. Leave empty for UID type.

The set must have one attribute defined as UID, which will allow to link individual meshes in the set to their attributes in the property file.

For the earlier example, the metadata property would look like this:

Filename UID,INDEX VALUE 0,ROCKID VALUE 0,DOMAIN VALUE 0,DENSITY VALUE 0 kg/m3

- ## meshes.meta

This file is optional, and it describes additional properties per mesh. Like the previous file in this section, this is also a CSV file. The file must contain two columns, with two headers named UID and volumetric (lowercase).

The "UID" column lists the mesh names. These identifiers must match the ones included in meta_attributes.csv.

The "volumetric" column specifies whether the mesh should be processed as volumetric.

If this file is omitted, or a particular mesh is not found in the list, the ingestion system assumes the mesh is not volumetric.

If a mesh is set as volumetric, but it fails a mesh consistency check, it will not be included in the processed set. The processing log for the entity will contain a warning for the mesh.

UID	volumetric
GM_Macro-Stratigraphy_BVstrat - CMZ - GM_Macro-Stratigraphy_BVstrat_-	1

GM_Macro-Stratigraphy_BVstrat - GCZ - GM_Macro-Stratigraphy_BVstrat_-	1
GM_Macro-Stratigraphy_BVstrat - LowerZone - GM_Macro-Stratigraphy_BVstrat_-	1
GM_Macro-Stratigraphy_BVstrat - Upper - GM_Macro-Stratigraphy_BVstrat_-	1
GM_Macro-Stratigraphy_BVstrat - Xenolith - GM_Macro-Stratigraphy_BVstrat_-	1
GM_Macro-Stratigraphy_BVstrat - Archean Crystalline - GM_Macro-Stratigraphy_BVstrat_-	0
GM_Macro-Stratigraphy_BVstrat - Critical Zone - GM_Macro-Stratigraphy_BVstrat_-	0
GM_Macro-Stratigraphy_BVstrat - Footwall - GM_Macro-Stratigraphy_BVstrat_-	0
GM_Macro-Stratigraphy_BVstrat - FT Granite - GM_Macro-Stratigraphy_BVstrat_-	0
GM_Macro-Stratigraphy_BVstrat - Granite - GM_Macro-Stratigraphy_BVstrat_-	0
GM_Macro-Stratigraphy_BVstrat - Main Zone - GM_Macro-Stratigraphy_BVstrat_-	0

Extended Geo-chemical Metadata

When submitting files for a new Geo-chemical raw dataset (file_type RAWGEOCHEM), the system expects an additional file named:

- **process.meta**

This is a text file in INI format that contains additional information on how the drill hole dataset must be constructed out of the accompanying files.

The file must contain a single section, named [Configuration] with the following properties:

SampleFile	Specifies the filename for the file that contains the samples.
SampleFile_ID	Specifies the index of the column in the collars file that contains sample IDs.
SampleFile_X	Specifies the index of the column in the samples file that contains the X coordinate for the drill hole collar.
SampleFile_Y	Specifies the index of the column in the samples file that contains the Y coordinate for the drill hole collar.
SampleFile_Z	Specifies the index of the column in the samples file that contains the Z coordinate for the drill hole collar.
SampleFile_Attribute_Columns	Specifies how may attribute columns will be read from the sample file.
SampleFile_Attribute_Column<N>_Index	Specifies the index of the column that contains the attribute values for a given attribute in the sample file. The <N> placeholder is replaced by the attribute index. For instance, if the sample file has three attributes to define, the process.meta file will include values for SampleFile_Attribute_Column0_Index, SampleFile_Attribute_Column1_Index and SampleFile_Attribute_Column2_Index.

SampleFile_Attribute_Column<N>_Name	Specifies the name of an attribute in the sample file. The <N> placeholder is replaced by the attribute index. For instance, if the sample file has three attributes to define, the process.meta file will include values for SampleFile_Attribute_Column0_Name, SampleFile_Attribute_Column1_Name and SampleFile_Attribute_Column2_Name.
SampleFile_Attribute_Column<N>_Type	Specifies the type of an attribute in the sample file. Must be set to one the following values: 0 - Attribute is of Float type 1 - Attribute of Alphanumerical type The <N> placeholder is replaced by the attribute index. For instance, if the sample file has three attributes to define, the process.meta file will include values for SampleFile_Attribute_Column0_Type, SampleFile_Attribute_Column1_Type and SampleFile_Attribute_Column2_Type.
SampleFile_Attribute_Column<N>_Unit	A string identifying the Units of Measurement for the attribute
AssayFileCount	Specifies how many assay files will be included.
AssayFile<N>	Specifies an assay file name. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for AssayFile0, AssayFile1 and AssayFile2. The <N> placeholder is replaced by the file index.
Column_Assay<N>_ID	Specifies the index of the column in the assay file that contains the drill hole ID. This is a unique identifier per hole, which is shared across several files in the dataset. There will be one of these properties for each assay file. For instance, if AssayFileCount = 3, the file must include values for Column_Assay0_ID, Column_Assay1_ID and Column_Assay2_ID. The <N> placeholder is replaced by the file index.
Column_Assay<N>_Attribute_Columns	Specifies how may attribute columns will be read from an assay file. There will be one of these properties for each assay file. The <N> placeholder is replaced by the file index.

Column_Assay<N>_Attribute_Column<M>_Index	Specifies the index of the column that contains the attribute values for a given attribute in a given assay file. The <N> placeholder is replaced by the assay file index, and the <M> placeholder is replaced by the attribute index. For instance, if the third assay file has three attributes to define, the properties.meta file will include values for Column_Assay2_Attribute_Column0_Index, Column_Assay2_Attribute_Column1_Index and Column_Assay2_Attribute_Column2_Index.
Column_Assay<N>_Attribute_Column<M>_Name	Specifies the name of an attribute in a given assay file. The <N> placeholder is replaced by the assay file index, and the <M> placeholder is replaced by the attribute index. For instance, if the third assay file has three attributes to define, the process.meta file will include values for Column_Assay2_Attribute_Column0_Name, Column_Assay2_Attribute_Column1_Name and Column_Assay2_Attribute_Column2_Name.
Column_Assay<N>_Attribute_Column<M>_Type	Specifies the type of an attribute in a given assay file. Must be set to one the following values: 0 - Attribute is of Float type 1 - Attribute of Alphanumerical type The <N> placeholder is replaced by the assay file index, and the <M> placeholder is replaced by the attribute index. For instance, if the third assay file has three attributes to define, the process.meta file will include values for Column_Assay2_Attribute_Column0_Type, Column_Assay2_Attribute_Column1_Type and Column_Assay2_Attribute_Column2_Type.
Column_Assay<N>_Attribute_Column<M>_Unit	A string identifying the Units of Measurement for the attribute
Column_Assay<N>_Attribute_Column<M>_Label	Specifies a readable label for the attribute name. The <N> placeholder is replaced by the attribute index, which is zero-based