

Triggering Export Jobs

Export entities allow to extract information from Voxel Farm Servers into a standard format. Currently, data can be exported in the following formats:

- An .OBJ file containing a mesh
- One or more image tiles in TIF/TFW format
- A CVS file containing a point cloud

Before exporting any data, the export entity must be properly configured. This section will cover how to set up an export job using the REST interface.

To successfully set up an export entity, it is necessary to:

1. Define which region of space will be exported
2. Define which type of data will be exported
3. Define which other entity will provide the data to be exported

Providing a 3D region

The Voxel Farm system currently supports one type of 3D region, which is defined as a horizontal 2D polygon that has been extruded along the vertical dimension.

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

Putting the export entity together

The following table lists the properties that must be set in order to create an export entity:

name	Set to the readable name of the export job.
description	Set to a textual description of the export job.
type	Must be set to “FILE”.
state	Set to “PARTIAL”
file_type	Set to “EXPORT”
file_folder	Set to the ID of the folder that will contain the export entity. Set to “0” to use the project’s root folder.
entity	Set to the ID of the program entity that will provide the data to be exported
lod	Set to the desired resolution for the exported data. Zero is the highest possible resolution, each subsequent number will decrease the resolution by a factor of two.
region	Set to region string.
export_type	Set to the desired type of data. The possible values are: • mesh – Exports the surface mesh for the entity (.OBJ format) • points – Exports a point cloud (CVS format) • ortho – Exports ortho-imagery (TIF+TFW)

This set of values can be used to create the entity, as described in the “Creating an entity” section. Once the entity is successfully created, the actual export job needs to be triggered. To see how to do so, see the “Triggering jobs” section.

Downloading the exported data

Once the export job completes, it is possible to download the data using the REST interface.

The exported data is always created inside a ZIP archive, and this archive is attached to the export interface using the filename “export.zip”.

The “Getting entity files” section describes how to retrieve a file once its name inside the entity is known.

Export example

This example will use the REST interface to create an export job that produces a mesh for a large sphere. The sphere will be defined as a custom volumetric object using a Python program. This example does not rely on any external data or entities that may already exist in the project.

This is the Python code for the sphere object:

```
import voxelfarm as vf
import math
vf.init()
x = vf.input("x", "Sphere Center X")
y = vf.input("y", "Sphere Center Y")
z = vf.input("z", "Sphere Center Z")
r = vf.input("r", "Sphere Radius")
vf.set_entity_bounds_x(x - r, x + r)
vf.set_entity_bounds_y(y - r, y + r)
vf.set_entity_bounds_z(z - r, z + r)
for v in vf.field:
    p = vf.get_field_origin(v)
    dx = x - p[0]
    dy = y - p[1]
    dz = z - p[2]
    d = math.sqrt(dx * dx + dy * dy + dz * dz)
    vf.set_field(v, d - r)
```

This program must be converted to Base64 before it can be stored as a Program entity:

```
aW1wb3J0IHZveGVsZmFybSBhcyB2ZgppbXBvcnQgbWF0aAoKdmYuaW5pdCgpCgp4ID0gdmYuaW5wdXQoIngiLCAiU3BoZ
XJlIENlbnRlciBYIikKeSA9IHZmLmlucHV0KCJ5JiwiZGllNwAGVyZSBDZW50ZXIgwSlpCnogPSB2Zi5pbnB1dCgieiIsICJTCgHlcmUgQ
2VudGVyIFoiKQpyID0gdmYuaW5wdXQoInIiLCAiU3BoZXJlIFJhZGl1cyIpCgp2Zi5zZXRfZW50aXR5X2JvdW5kc194KHggLSByL
CB4ICsgcikKdmYuc2V0X2VudGloev9ib3VuZHNfeSh5IC0gcwgeSArIHIpCnZmLnNldF9lbnRpdHlfYm91bmRzX3ooeiAtIHIsIH
gKyByKQoKZm9yIHlYgaW4gdmYuZmllbGQ6CiAgICBwID0gdmYuZ2V0X2ZpZWxkX29yaWdpbi2KQogICAgZHggaPSB4IC0gc
FswXQogICAgZHKgaPSB5IC0gcFswXQogICAgZHKgaPSB6IC0gcFsyXQogICAgZCA9IG1hdGguc3FydChkeCAqIGR4ICsgZHKgKiBk
eSArIGR6ICogZHopCiAgICB2Zi5zZXRfZmllbGQodiwgZCAtIHIpCg==
```

The following REST call creates the Program entity for the sphere:

```
Invoke-WebRequest -Uri "http://localhost:58697/entity.ashx?project=myproject" -Method "POST" -Headers
@{"Origin"="http://localhost:58697"; "Accept-Encoding"="gzip, deflate, br"; "Accept-Language"="en-
US,en;q=0.9,es;q=0.8"; "User-Agent"="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/73.0.3683.103 Safari/537.36"; "Accept"="*/*";
"Referer"="http://localhost:58697/cloud/project.html?id=myproject"} -ContentType "application/json; charset=UTF-8" -
Body
"{`"name`":`"Sphere`",`"type`":`"FILE`",`"state`":`"COMPLETE`",`"code`":`"aW1wb3J0IHZveGVsZmFybSBhcyB2ZgppbXBvcnQgbWF0aAoKdmYuaW5pdCgpCgp4ID0gdmYuaW5wdXQoIngiLCAiU3BoZXJlIENlbnRlciBYlikKeSA9IHZmLmlucHV0KCJ5IiwgIiNwaGVyZSBDZW50ZXIgaW5pdCgpCgp2Zi5pbnB1dCgieiIsIjTcGhlcmUgQ2VudGVyIiFoiKQpyID0gdmYuaW5wdXQoInliLCAiU3BoZXJlIiFhZGI1cyIpCgp2Zi5zZXRFZW50aXR5X2JvdW5kc194KHggLSByLCB4ICsgcikKdmYuc2V0X2VudGI0eV9ib3VuZHNfeSh5IC0gcicwgeSARlHlpCnZmLnNldF9lbnRpdHlfYm91bmRzX3ooeiAtIHIsIHogKyByKQoKZm9yIHYgaW4gdmYuZmllbGQ6CiAgICBwID0gdmYuZ2V0X2ZpZWxkX29yaWdpbiH2KQogICAgZHGgPSB4IC0gcFswXQogICAgZHKgPSB5IC0gcFswXQogICAgZHGgPSB6IC0gcFsyXQogICAgZCA9IG1hdGguc3FydChkeCAqIGR4ICsgZHKgKiBkeSARlGR6ICogZHopCiAgICB2Zi5zZXRFZmllbGQodiwgZCAtIHlpCg==`,`"program_type`":`"VOXEL`",`"file_type`":`"PROGRAM`",`"file_size`":`"0`",`"file_date`":`"1557157137959`",`"file_folder`":`"0`"}`"}
```

The server returns the ID of the newly created program:

```
{"result" : "success", "id" : "03318353117A40D6AA8B6C09A85A1060"}
```

The next step is to create the export entity. The entity will use “1,-50,50,-50,50,50,50,50,-50,-50,-50” for the region, and will have its export_type property set to “mesh”. The following REST request creates the entity:

```
Invoke-WebRequest -Uri "http://localhost:58697/entity.ashx?project=myproject" -Method "POST" -Headers
@{"Origin"="http://localhost:58697"; "Accept-Encoding"="gzip, deflate, br"; "Accept-Language"="en-
US,en;q=0.9,es;q=0.8"; "User-Agent"="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/73.0.3683.103 Safari/537.36"; "Accept"="*/*";
"Referer"="http://localhost:58697/cloud/project.html?id=myproject"} -ContentType "application/json; charset=UTF-8" -
Body "{`"name`":`"Sphere
Mesh`",`"type`":`"FILE`",`"file_type`":`"EXPORT`",`"export_type`":`"mesh`",`"lod`":`"0`",`"region`":`"1,-50,50,-
50,50,50,50,50,-50,-50,-
50`",`"entity`":`"398446B631524AABAE0D3183C138A62`",`"state`":`"PARTIAL`",`"file_size`":`"0`",`"file_date`":`"1557231988008`",`"file_folder`":`"0`"}`"}
```

The server returns the ID of the newly created export entity:

```
{"result" : "success", "id" : "93E4A859BAD74A30AABFFE02D2403695"}
```

The following REST call starts the processing of the export entity:

```
Invoke-WebRequest -Uri
"http://localhost:58697/events.ashx?project=myproject&id=93E4A859BAD74A30AABFFE02D2403695&org=2343243456
678890" -Method "POST" -Headers @{"Origin"="http://localhost:58697"; "Accept-Encoding"="gzip, deflate, br"; "Accept-
Language"="en-US,en;q=0.9,es;q=0.8"; "User-Agent"="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36"; "Accept"="*/*";
"Referer"="http://localhost:58697/cloud/project.html?id=myproject"} -ContentType "multipart/form-data; boundary=----
WebKitFormBoundaryEIB7ohnHD6E1hAjI" -Body ([System.Text.Encoding]::UTF8.GetBytes("-----
WebKitFormBoundaryEIB7ohnHD6E1hAjI$([char]13)$([char]10)Content-Disposition: form-data;
name=`"process`"$([char]13)$([char]10)$([char]13)$([char]10)EXPORT$([char]13)$([char]10)-----
WebKitFormBoundaryEIB7ohnHD6E1hAjI--$([char]13)$([char]10)"))
```

This example does not use the callback URL, it will poll until the state of the export entity is set to “COMPLETE”. The following REST request will retrieve the export entity:

```
Invoke-WebRequest -Uri "http://localhost:58697/entity.ashx?id=93E4A859BAD74A30AABFFE02D2403695" -Headers
@{"Upgrade-Insecure-Requests"="1"; "User-Agent"="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36";
"Accept"="text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3"; "Accept-Encoding"="gzip, deflate, br"; "Accept-Language"="en-US,en;q=0.9,es;q=0.8"}
```

The server will respond with a JSON object that contains all properties for the export entity. By inspecting the value of the “state” property, it is possible to determine if the entity has completed processing and whether any errors may have occurred. When the processing completes, the “state” property will switch to either “COMPLETE” or “ERROR”.

Once the export job has completed, the data generated can be downloaded using the REST API:

```
Invoke-WebRequest -Uri
"http://localhost:58697/file.ashx?org=2343243456678890&project=myproject&id=93E4A859BAD74A30AABFFE02D2403
695&filename=export.zip&mime=application/zip&namehint=U3BoZXJITWVzaC56aXA=" -Headers @{"Upgrade-Insecure-
Requests"="1"; "User-Agent"="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36";
"Accept"="text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3"; "Referer"="http://localhost:58697/cloud/project.html?id=myproject"; "Accept-Encoding"="gzip, deflate,
br"; "Accept-Language"="en-US,en;q=0.9,es;q=0.8"}
```

This retrieves a ZIP archive that contains the .OBJ mesh for the sphere:

Revision #2

Created 17 March 2025 15:59:30 by admin

Updated 17 March 2025 16:08:04 by admin