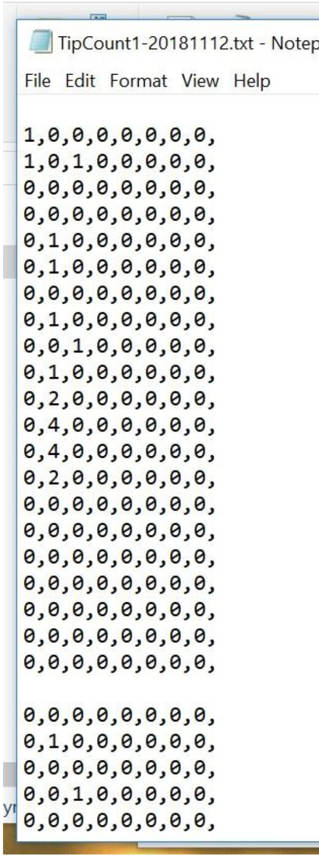
Collecting / Capturing Output Data

How do I capture output data? I want some real numbers to crunch.

Data output is controlled by the file OutputRules.xml in the base of your config directory. You can define which characteristics you wish captured, how often and the name of the file to write them to. For a list of output (and input) characteristics, see List of Characteristics in the Documentation directory. You can also look in the View menu on ROOTMAP for a list of characteristics.

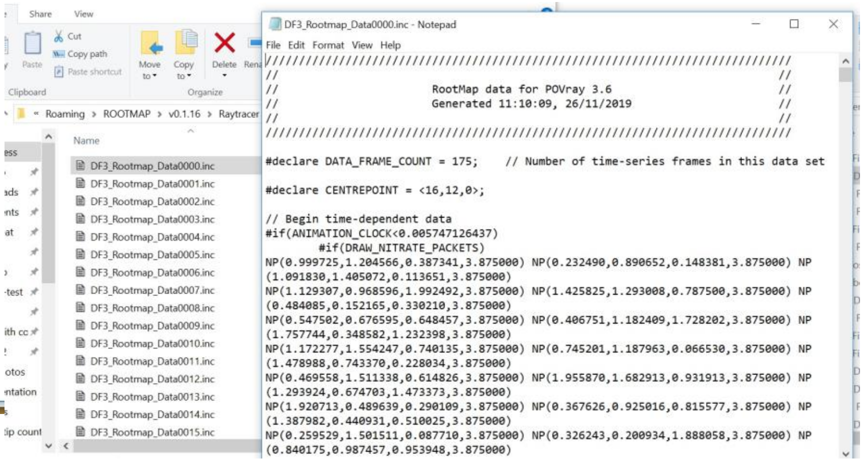


```
1,0,0,0,0,0,0,0,
1,0,1,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,1,0,0,0,0,0,0,
0,1,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,1,0,0,0,0,0,0,
0,0,1,0,0,0,0,0,
0,1,0,0,0,0,0,0,
0,2,0,0,0,0,0,0,
0,4,0,0,0,0,0,0,
0,4,0,0,0,0,0,0,
0,2,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,
0,1,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,1,0,0,0,0,0,
0,0,0,0,0,0,0,0,
```

Common types of data captured are numeric data, for mathematical calculations, and raytracing data, for animations based on the simulation's growth over time.

Numeric data that is spatial in nature, such as the number of root tips in each cell of the Scoreboard, will be written in a text file, broken into planes along the Y axis. The image to the left shows a section of an output file. The scoreboard being modelled is 8 cells wide (X), 8 cells deep (Y) and 20 cells down (Z). Each X-Z slice is printed with commas separating cells, line breaks separating Z layers and additional line breaks separating Y slices. This can be altered in the Output Rules.



```
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//                          RootMap data for POVray 3.6                       //
//                          Generated 11:10:09, 26/11/2019                    //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////

#declare DATA_FRAME_COUNT = 175;    // Number of time-series frames in this data set

#declare CENTREPOINT = <16,12,0>;

// Begin time-dependent data
#if(ANIMATION_CLOCK<0.005747126437)
        #if(DRAW_NITRATE_PACKETS)
NP(0.999725,1.204566,0.387341,3.875000) NP(0.232490,0.890652,0.148381,3.875000) NP
(1.091830,1.405072,0.113651,3.875000)
NP(1.129307,0.968596,1.992492,3.875000) NP(1.425825,1.293008,0.787500,3.875000) NP
(0.484085,0.152165,0.330210,3.875000)
NP(0.547502,0.676595,0.648457,3.875000) NP(0.406751,1.182409,1.728202,3.875000) NP
(1.757744,0.348582,1.232398,3.875000)
NP(1.172277,1.554247,0.740135,3.875000) NP(0.745201,1.187963,0.066530,3.875000) NP
(1.478988,0.743370,0.228034,3.875000)
NP(0.469558,1.511338,0.614826,3.875000) NP(1.955870,1.682913,0.931913,3.875000) NP
(1.293924,0.674703,1.473373,3.875000)
NP(1.920713,0.489639,0.290109,3.875000) NP(0.367626,0.925016,0.815577,3.875000) NP
(1.387982,0.440931,0.510025,3.875000)
NP(0.259529,1.501511,0.087710,3.875000) NP(0.326243,0.200934,1.888058,3.875000) NP
(0.840175,0.987457,0.953948,3.875000)
```

Raytracing data is written as a text file with an .inc suffix, in the format read by POV-Ray as above. See the POV-Ray documentation for further explanation of the .include files and setting up a 3D animation run.

During the model run, when output files are being written to, the on-screen Progress window will show the caption "Output File Alarm". Numeric output files will be written to the directory C:\[you the user]\AppData\Roaming\ROOTMAP\[current version number]. Raytracing output files will be written to a Raytracing directory in the same location.

Give me an example or three

Here’s an output rule that collects numeric data on the root length of the plant, in this case, plant 1.

```
<outputrule>
<type>ScoreboardData</type>
<source>PlantCoordinator</source>
<!-- characteristic is meaningful for the type/source combination It applies in the general non-
scoreboard sense of the term also, you see -->
<characteristic>Root Length Wrap None Plant 1</characteristic>
<stratum>Soil</stratum>
<!-- accepts % formatting codes for strftime -->
<!-- Pre-processes URL % formatted values before passing to strftime eg. use %20 for trailing spaces
-->
<!-- Pre-processes extra format specifiers: %C counter for number of outputs for this object so far
(UNSUPPORTED) %R raw timestamp in seconds from T=0 (UNSUPPORTED) both of these accept
the usual %d format specifiers, eg. "%.3C" produces a 3-digit leading-zero-padded. The following
example produces a filename like RootLength_YYYYmmdd-HHMMSS.txt PLEASE ensure you
don't confuse the lowercase and uppercase 'm'. m=month, M=Minute -->
<filename>RootLengthPlant1.txt</filename>
<!-- specification1 and specification2 are meaningful to the type. For "ScoreboardData",
specification1 is the dimension order. -->
<specification1>X,Z,Y</specification1>
<!-- how to handle current contents when opening a non-empty file: append|overwrite
(default=overwrite) -->
<reopen>append</reopen>
<when>
        <!-- a count of zero is to repeat ad infinitum -->
        <count>0</count>
        <!-- For exporting "every" so often, use the initialtime and frequency tags
        Year,Month,Day,Hour,Minute,Second -->
        <initialtime>0,0,0,0,0,0</initialtime>
        <interval>0,0,0,1,0,0</interval>

</when>
</outputrule>
```

Here’s an output rule that captures the tap root length of Plant 2 in Week 4.

```
<outputrule>
<type>ScoreboardData</type>
<source>PlantCoordinator</source>
<characteristic>Root Length Wrap None Plant 2 RootOrder0</characteristic>
<stratum>Soil</stratum>
<filename>TapRootLengthwk4.txt</filename>
<specification1>X,Z,Y</specification1>
<reopen>append</reopen>
<when>
        <!-- a count of zero is to repeat ad infinitum -->
        <count>0</count>
        <!-- For exporting "every" so often, use the initialtime and frequency tags
        Year,Month,Day,Hour,Minute,Second -->
        <interval>0,0,28,0,0,0</interval>
        <initialtime>0,0,0,0,0,0</initialtime></when>

</outputrule>
```

Here’s one that makes files for raytracing a 3D model using POV-Ray.

```
<outputrule>
<!-- For "RaytracerData", the <source> tag controls both (A) What will be the primary source for raytraced visualisation, and (B) Where the the raytraced scene will be centred. If the <source> is a Plant name, its roots will be drawn and halfway down the Scoreboard from its <origin> point will be the centre. (see Plants.xml) If the <source> is the PlantCoordinator, all Plants' roots will be drawn and halfway down the Scoreboard from the geometric centre of all Plants' origins will be the centre. If the <source> is the ScoreboardRenderer, all Scoreboard boxes will be drawn and the geometric centre of the Scoreboards will be the centre. -->
<source>PlantCoordinator</source>
<type>RaytracerData</type>
<stratum>Soil</stratum>
<!-- Filename should be left as the default: Raytracer/DF3_Rootmap_Data%04u.inc (or Raytracer/SB_Rootmap_Data%04u.inc). The %04u suffix indicates that each frame's filename will be serialised with a 4-digit [U]nsigned integer (i.e. 0000, 0001, 0002, 0003, etc). The first file generated contains a list of all (zero or more) subsequent filenames. It is referenced by the main Rootmap_Scene.pov file to load every frame into the POV-ray scene. (One serialised include file is generated for each frame) Count starts at zero due to convention. If changed, the new filename must be substituted for DF3_Rootmap_Data0000.inc in Rootmap_Scene.pov. The unserialised filename prefix must not end with a digit (i.e. Rootmap1%04u.inc is illegal). File extension must be .inc. -->
<filename>Raytracer/DF3_Rootmap_Data%04u.inc</filename>
<!-- specification1 is meaningful to the type. For "RaytracerData", specification1 is a comma-delimited list of extra Process names to be raytraced. (1) Type "all" to raytrace output from everything: the <source>, plus all raytracer-capable Processes, plus visualisation of all Scoreboard boxes (see Windows.xml : ViewCoordinator and View3DCoordinator). (2) Type a comma-delimited list of one or more Processes and/or "ScoreboardRenderer" to raytrace output from these in addition to the <source>. E.g. <specification1>Nitrate, ScoreboardRenderer</specification1> (3) Leave <specification1> blank (or absent entirely) to raytrace output from the <source> only. -->
<specification1>Nitrate,VolumeObjectCoordinator</specification1>
<!-- specification2 is used for setting the method of ScoreboardBox rendering. <specification2>DF3</specification2> uses POV-ray's voxel-based density field format to represent the whole Scoreboard, <specification2>SB</specification2> uses a rectangular prism of translucent media to represent each ScoreboardBox. As a rule of thumb, DF3 is better for mathematically regular Scoreboard boundaries, and SB is better for irregular Scoreboard box sizes. -->
<specification2>DF3</specification2>
<!-- <reopen> should always be set to OVERWRITE. POV-ray relies on the files generated by Rootmap being formatted correctly; appending data to existing files would corrupt them -->
<reopen>overwrite</reopen>
<when>
<!-- a count of zero is to repeat ad infinitum -->
<count>175</count>
<!-- If only <count> element is present (or if <interval> is set to all zeros), [value of <count>] frames will be produced, evenly distributed throughout the simulation. Of course if <count> is zero in this case, no output will be produced. -->
<!--interval>0,0,2,12,0,0</interval> <initialtime>0,0,0,18,0,0</initialtime-->
</when>

</outputrule>
```

Here’s an output rule that collects non-spatial data, in this case, the nitrogen uptake of the plant (which varies with time but has no relation to any location on the Scoreboard). There’s currently an error in the program which means this type of data is not being collected.

```
<outputrule>
<source>Nitrate</source>
<type>NonSpatialData</type>
<characteristic>Cumul Plant Nitrate Uptake</characteristic>
<stratum>Soil</stratum>
<!-- Use a filename without any strftime % formatting codes -->
<!-- and <reopen>append</reopen> to write all data to one file, -->
<!-- suitable for graphing -->
<filename>CumulNUptake.csv</filename>
<!-- <specification1> is the variation name(s), e.g. "Plant" - see files in ./shared_attributes/ -->
<specification1>Plant,RootOrder</specification1>
<!-- <specification2>n</specification2> specifies a data dimensionality of n, where 0<=n<=2. Use
<type>ScoreboardData</type> for data that is spatially relevant. -->
<specification2>csv</specification2>
<reopen>append</reopen>
<when>
        <!-- a count of zero is to repeat ad infinitum -->
        <count>0</count>
        <!-- For exporting "every" so often, use the initialtime and frequency tags
        Year,Month,Day,Hour,Minute,Second -->
        <initialtime>0,0,0,1,0,0</initialtime>
        <interval>0,0,0,0,0,0</interval></when>

</outputrule>
```

Troubleshooting

If you make a mistake in OutputRules.xml, for example misspelling characteristic names or plant names or not putting in the right time, the model will run fine and just not write you any output files for whatever's screwed up.

If output data is not being captured, check these three things first.

1. Check that you’ve exactly matched the real characteristic name you meant to capture. This includes variations such as Wrap None or Volume Object 1 where relevant. The full name acts as a single string and is processed as such.
2. Check the timing. Ensure the <when> element is set up to actually produce output during the simulation's runtime. For example, if it has an <interval> of 30 days and the simulation runs for 20 days, it'll never produce output. Note that the interval starts after the initial time and that one-time-outputs may need <interval> filled out. Your one output will happen <interval> after <initial>, there is no <initial> output if <interval> is set to 0.
3. Make sure you have replaced all references to “Plant 120” or “Rootmap Plant” or any other name with whatever your simulation’s plant is called.

Note that there’s currently (at time of writing) a bug somewhere that means non-spatial data isn’t being recorded.