

Web Visualization of a Trajectory Generated from the General Mission Analysis Tool (Part 1)

By Daniel A. O'Neil

Introduction

A Web-based Mission Visualization System (WMVS) can educate the public about past and future space missions and engage interested individuals in the design and analysis of potential space missions. NASA provides a free open source General Mission Analysis Tool (GMAT) for designing and analyzing space missions. With GMAT, a mission designer can specify a spacecraft, trajectories, parameters for external forces, and select from a variety of propagators to move a spacecraft along a trajectory. The GMAT Graphical User Interface (GUI) presents an animated 3D view of the mission. Data can be exported from GMAT for use in other programs. This document describes a work-flow for exporting a simple trajectory from GMAT, processing the data to produce a JavaScript object, and creating a web page with an embedded interactive visualization of the trajectory.

Objectives

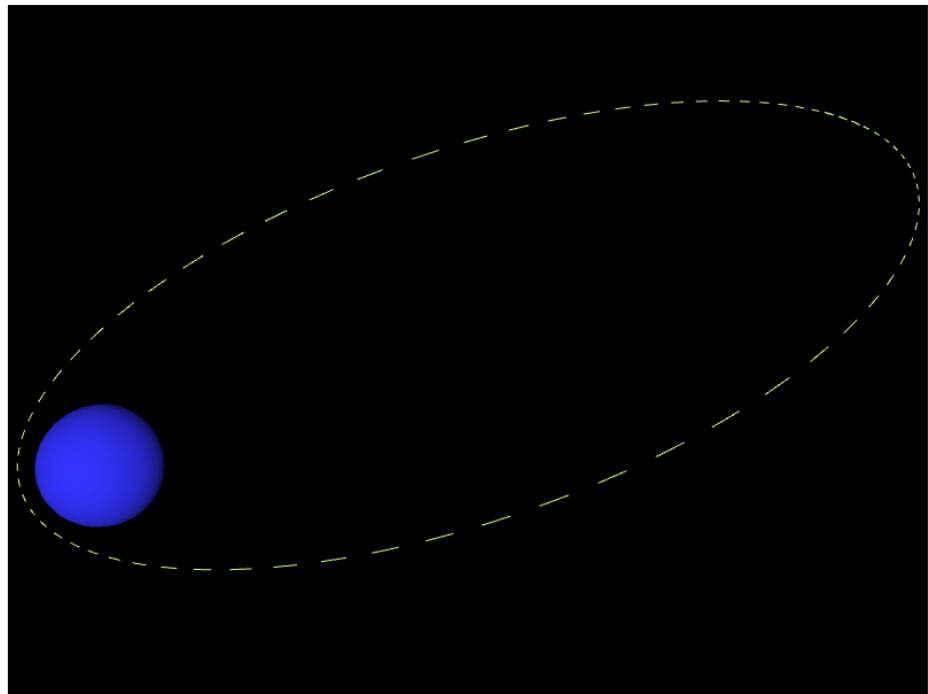
Resources identified in this document include websites, research guidance, and code snippets. Figure 1 depicts a web visualization of a trajectory generated from GMAT. The following Uniform Resource Locator (URL) leads to this trajectory visualization.

http://daoneil.github.io/spacemission/X3Dom/Trajectory_JSON.html

Learning objectives for this tutorial include:

1. Installing GMAT
2. Introductory YouTube GMAT tutorial
3. Adding a report generator to a GMAT script
4. Using Excel to convert a data file to JavaScript Object Notation (JSON)
5. Web Graphics Language (WebGL) code libraries
6. Writing a JavaScript + WebGL trajectory visualization using X3Dom

Prototype Web Visualization of a GMAT Generated Trajectory



Mission Start Date: 22 Jul 2014 11:29:10.811

Mission End Date: 25 Jul 2014 11:29:10.811

Figure 1 Image of a Web Visualization of a GMAT Generated Trajectory

Installing and Learning GMAT

This tutorial refers to GMAT-R2014a, which you can download from Sourceforge.

<http://sourceforge.net/projects/gmat/files/GMAT/>

A Windows installer is available. To run the program on Linux or Mac OS, download and compile the source code. Compiling GMAT is beyond the scope of this tutorial.

After installing GMAT, watch the YouTube video orbit simulation tutorial.

<https://www.youtube.com/watch?v=2nbMj05ELjE>

Read the GMAT R2014a User Guide for a tour of the GMAT GUI and tutorials for designing and simulating missions. <http://gmat.sourceforge.net/docs/R2014a/help-letter.pdf>

Adding a Report Generator to a GMAT Script

After performing the introductory tutorial, the GMAT GUI Resources Tree will include a script in the Scripts folder. You can double-click the script to open and edit it. Read the ReportFile documentation under the Help file Reference Guide Resources section and the Report documentation under the Reference Guide Commands section. These sections explain the commands and provide examples. The following code snippet creates a report file and identifies data to be captured in the report file.

```
%-----  
%----- Mission Sequence  
%-----  
Create ReportFile aReport;  
GMAT aReport.Filename = 'ReportFile1.txt';  
GMAT aReport.Precision = 6;  
GMAT aReport.Add = {Sat.UTCGregorian, Sat.EarthMJ2000Eq.X, Sat.EarthMJ2000Eq.Y,  
Sat.EarthMJ2000Eq.Z};  
GMAT aReport.ColumnWidth = 15;  
%_____
```

The % indicates a comment in the script. The Create ReportFile command generates a data object or handle for referencing the file. Lines that follow the create command set attributes of the report file. These attributes include a file name, precision, fields, and column width. The precision attribute determines the number of decimal places. The Add attribute identifies data fields to include in the report file; in this case, the fields include a date and time stamp, and the Cartesian coordinates, X, Y, and Z of the spacecraft.

After executing the script, GMAT will produce a report file. The following data snippet presents the fixed width data file generated from the GMAT script.

```
Sat.UTCGregorian      Sat.EarthMJ2000Eq.X   Sat.EarthMJ2000Eq.Y   Sat.EarthMJ2000Eq.Z  
22 Jul 2014 11:29:10.811  137380                -75679.6              21487.6  
22 Jul 2014 11:30:10.811  137394                -75652.8              21492.7  
22 Jul 2014 11:33:00.242  137433                -75577                21507.1  
....
```

Using Excel to convert the data file to JavaScript Object Notation

Copy and paste the data into an Excel worksheet named Trajectory. Add an empty worksheet named JSON. If your copy of Excel does not display a command ribbon for Developer, use Bing to search for tutorials on how to add the Developer ribbon. Within the Visual Basic for Applications (VBA) editor, add a module and copy and paste the following code:

```
Sub parseData()  
    Dim rgDat As Range          ' Range object for the trajectory data sheet  
    Dim aRecord As String      ' A string variable the next line of data.  
    Dim aDate As String        ' String variable for the date  
    Dim aTime As String        ' String variable for the time  
    Dim aX As String           ' String variable for the X coordinate  
    Dim aY As String           ' String variable for the Y coordinate  
    Dim aZ As String           ' String variable for the Z coordinate  
    Dim i As Integer           ' Row counter  
  
    Set rgDat = ThisWorkbook.Worksheets("Trajectory").Range("A2")  
  
    i = 1  
    While rgDat.Cells(i).Value <> ""  
        aRecord = ""  
        aDate = ""  
        aTime = ""            ' Reset variables.  
        aX = ""  
        aY = ""  
        aZ = ""  
  
        aRecord = rgDat.Cells(i).Value  
        aDate = Mid(aRecord, 1, 12)  
        aTime = Mid(aRecord, 13, 12)    ' Parse the data.  
        aX = Mid(aRecord, 27, 12)  
        aY = Mid(aRecord, 49, 12)  
        aZ = Mid(aRecord, 69, 12)  
  
        Call writeJSON(aDate, aTime, aX, aY, aZ, i)  
  
        i = i + 1  
    Wend  
  
End Sub  
-----  
Sub writeJSON(aDate, aTime, x, y, z As String, n As Integer)  
    Dim rgJSON As Range  
  
    Set rgJSON = ThisWorkbook.Worksheets("JSON").Range("A2")  
  
    rgJSON.Cells(n).Value = Chr(39) + "{" & """" & "Date" & """" & ":" & """" &  
-----  
                                Trim(aDate) & """" & "," &  
                                & """" & "Time" & """" & ":" & """" & Trim(aTime) &  
                                & """" & "," &  
                                & """" & "X" & """" & ":" & Trim(x) & "," &  
                                & """" & "Y" & """" & ":" & Trim(y) & "," &  
                                & """" & "Z" & """" & ":" & Trim(z) & "},'" +  
  
End Sub
```

The following data snippet presents the JSON formatted data on the JSON spreadsheet:

```
{"Date":"22 Jul 2014","Time":"11:29:10.811","X":137380,"Y":-75679.6,"Z":21487.6}, ' +  
{ "Date":"22 Jul 2014", "Time":"11:30:10.811", "X":137394, "Y":-75652.8, "Z":21492.7}, ' +  
{ "Date":"22 Jul 2014", "Time":"11:33:00.381", "X":137433, "Y":-75576.9, "Z":21507.2}, ' +  
{ "Date":"22 Jul 2014", "Time":"11:41:06.513", "X":137542, "Y":-75358.2, "Z":21548.3}, ' +
```

Notice the single quotes between the commas and the plus signs at the end of the lines; these quotes are unmatched at the beginning of the lines. Excel drops a single quote at the beginning of a line. To fix this problem, perform a search and replace in a text editor. Search for {“Date and replace it with {“Date. To complete the conversion to JSON and to declare a JavaScript variable, add *var trajectory = '{"coordinates":['* + at the top of the text file and *']}'*; at the end of the text file. Be sure to delete the last comma, otherwise the JSON parser will throw an unexpected character error. Now, the trajectory data object can be incorporated into a JavaScript + WebGL visualization.

Web Graphics Language (WebGL)

A consortium, named the Khronos Group, develops and maintains standards for languages and file formats related to 3D computer graphics. According to the organization’s website, “WebGL is a royalty-free, cross-platform” Application Programming Interface (API) that enables 3D graphics within the HyperText Markup Language (HTML). With a WebGL compliant code library, a developer can incorporate 3D objects into the HTML code. In addition to WebGL, the Khronos Group defines the COLLABORATIVE Design Activity (COLLADA) standard for 3D models. Several 3D graphics modeling programs can export COLLADA files and a few JavaScript code libraries can import COLLADA models.

- WebGL API Overview <https://www.khronos.org/webgl/>
- COLLADA 3D Asset Exchange Schema <https://www.khronos.org/collada/>

Several WebGL compliant JavaScript code libraries are available. Two examples are Three.js and X3Dom. The Three.js library includes a COLLADA file loader function. The X3Dom library implements the X3D standard from the Web 3D consortium. A few 3D graphics modeling programs can export X3D. The rest of this document describes the development of a JavaScript + WebGL visualization of the GMAT generated trajectory using the X3Dom code library.

- Three.js <http://threejs.org/>
- Three.js COLLADA Loader <http://threejs.org/docs/>
- X3Dom <http://www.x3dom.org/>
- Web3D Consortium <http://www.web3d.org/>

Developing a JavaScript + WebGL Trajectory Visualization with X3Dom

A scene graph is a schema that defines 3D objects, associated attributes, and relationships among the objects in a scene. Figure 1 depicts a blue ball that represents Earth and a dashed line that represents the trajectory of a satellite orbiting Earth. The X3Dom code library does include functions to texture-map objects, so a future version of the model could include an improved Earth model.

An X3Dom scene graph uses Extended Markup Language (XML) tags to define a scene, background, objects and associated attributes, and a view point. The following code snippet allocates an area of 600 pixels by 800 pixels of web-page real estate, defines a black background, defines a blue ball, and establishes a point of view.

```

<X3D xmlns="http://www.web3d.org/specifications/x3d-namespace"
showStat="false" showLog="false" x="0px" y="0px" width="800px"
height="600px">
  <Scene>
    <background DEF='bgnd' transparency='0'
      skyColor='0.0 0.0 0.0' ></background>
    <Transform id="theEarth" translation="0 0 0">
      <Shape>
        <Appearance>
          <Material diffuseColor='0.2 0.2 1' />
        </Appearance>
        <Sphere radius='0.7' />
      </Shape>
    </Transform>
    <Viewpoint fieldOfView="0.785398"
      position="6 5.5 6.5"
      orientation="1 -1 0 -0.785"
      description="" />
  </Scene>
</X3D>

```

A few lines of JavaScript code steps through the Cartesian coordinate records in the JSON trajectory object and inserts a line into the scene graph. The following code snippet parses the JSON trajectory object and stores the data in a variable named mission. A For loop steps through the coordinates. Code within the For loop creates a X3D Shape, Appearance, and Material nodes. At each iteration, the current set of Cartesian coordinates are added to an IndexedLineSet. The last few lines of the code snippet assemble the line elements and nodes and insert the integrated object into the scene graph as a child of the Earth.

```

var mission = JSON.parse(trajectory) ;

var segIndex = 0 ; // segment counter
var orbitCoords = "" ;

for (var segment in mission.coordinates) {
  var s = document.createElement('Shape'); // Shape Node
  s.setAttribute("id", "segment" + segIndex);
  var app = document.createElement('Appearance'); // Appearance Node
  var mat = document.createElement('Material'); // Material Node
  mat.setAttribute("id", "Mat" + segIndex);
  mat.setAttribute("diffuseColor", 1 + " " + 1 + " " + 0);
  mat.setAttribute("emissiveColor", 0.7 + " " + 1 + " " + 0.3);

  app.appendChild(mat);
  s.appendChild(app);

  var segCoords = [mission.coordinates[segment].X/10000,
mission.coordinates[segment].Y/10000,
mission.coordinates[segment].Z/10000] ;

  orbitCoords = orbitCoords + segCoords[0] + " " + segCoords[1] + " " +
segCoords[2] + " " ;

  var line = document.createElement('IndexedLineSet');

```

```

    line.setAttribute("coordIndex", segIndex);
    var coords = document.createElement('Coordinate');
    coords.setAttribute("point", orbitCoords);

    line.appendChild(coords) ;

    s.appendChild(line);
    var ot = document.getElementById('theEarth');
    ot.appendChild(s);

    segIndex = segIndex + 1 ;
}

```

The appendix provides the complete listing of the web-visualization of the GMAT generated trajectory.

Future Work

Potential improvement activities for this model include a new scene graph, a conversion utility, a restructured design of the visualization system, and code to read mission data into a visualization system. A notional list of tasks include:

1. Develop an orrery, i.e., a solar system scene graph
2. Texture map the planets in the orrery, sources of planetary texture maps include:
 - a. JPL Solar System Simulator <http://maps.jpl.nasa.gov/>
 - b. NASA 3D Resources <http://nasa3d.arc.nasa.gov/images>
 - c. Planetary Visions http://www.planetaryvisions.com/TP_intro.php?pid=4201
3. Create a command line utility program to convert trajectory data into JSON objects
4. Write HTML and JavaScript code to access JSON objects that are stored in separate files
5. Produce an animated web-visualization that moves 3D models along a trajectory
 - a. Display the date and time stamps during the animation
 - b. Design widgets to control the animation, e.g., speed, forward, backward, pause
 - c. Include function(s) to load 3D models into the scene graph
6. Implement the HTML5 file reader functions to upload trajectory data files
 - a. W3C File API <http://www.w3.org/TR/FileAPI/>
 - b. Reading files in JavaScript <http://www.html5rocks.com/en/tutorials/file/dndfiles/>
 - c. FileReader <https://developer.mozilla.org/en-US/docs/Web/API/FileReader>

Conclusion

This tutorial provided educational resources for the General Mission Analysis Tool, explained how to export trajectory data and convert it into a JavaScript object, and provided example code for a simple web-based trajectory visualization. Additionally, the tutorial explained the WebGL standard and identified Three.js and X3Dom as two compliant JavaScript code libraries. A section on future work briefly describes tasks that can build upon the GMAT generated trajectory visualization prototype to produce a general purpose web based mission visualization system.

Appendix trajectory visualization code

```
<!DOCTYPE html>
<head>
  <meta http-equiv="X-UA-Compatible" content="chrome=1" />
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Trajectory from GMAT</title>
  <meta name="author" content="Daniel A. O'Neil">
  <meta name="copyright" content="© Daniel A. O'Neil" />
  <style>
    p.case { clear: both; border-top: 0px; solid: #fff; }
  </style>
  <link rel="stylesheet" type="text/css" href="x3dom.css" />
</head>
<body>

<h2>Prototype Web Visualization of a GMAT Generated Trajectory</h2>
  <p class="case">
    <X3D xmlns="http://www.web3d.org/specifications/x3d-namespace"
    showStat="false" showLog="false" x="0px" y="0px" width="800px" height="600px">
      <Scene>
        <background DEF='bgnd' transparency='0' skyColor='0.0 0.0
0.0' ></background>
        <Transform id="theEarth" translation="0 0 0">
          <Shape>
            <Appearance>
              <Material diffuseColor='0.2 0.2 1' />
            </Appearance>
            <Sphere radius='0.7' />
          </Shape>
        </Transform>
        <Viewpoint fieldOfView="0.785398" position="6 5.5 6.5"
orientation="1 -1 0 -0.785" description="" />
      </Scene>
    </X3D>
  </p>
  <p class="case" align="center">
<p id="demo"></p>

<script>
var trajectory = '{"coordinates":[' +
'{"Date":"22 Jul 2014","Time":"11:29:10.811","X":137380,"Y":-75679.6,"Z":21487.6},' +
'{"Date":"22 Jul 2014","Time":"11:30:10.811","X":137394,"Y":-75652.8,"Z":21492.7},' +
'{"Date":"22 Jul 2014","Time":"11:33:00.381","X":137433,"Y":-75576.9,"Z":21507.2},' +
'{"Date":"22 Jul 2014","Time":"11:41:06.513","X":137542,"Y":-75358.2,"Z":21548.3},' +
'{"Date":"22 Jul 2014","Time":"12:07:06.014","X":137873,"Y":-74644.6,"Z":21676.7},' +
'{"Date":"22 Jul 2014","Time":"12:52:06.014","X":138365,"Y":-73366.1,"Z":21886.5},' +
'{"Date":"22 Jul 2014","Time":"13:37:06.014","X":138755,"Y":-72033.8,"Z":22080.1},' +
'{"Date":"22 Jul 2014","Time":"14:22:06.014","X":139043,"Y":-70648.3,"Z":22257.4},' +
'{"Date":"22 Jul 2014","Time":"15:07:06.014","X":139227,"Y":-69210.5,"Z":22418},' +
'{"Date":"22 Jul 2014","Time":"15:52:06.014","X":139307,"Y":-67720.8,"Z":22561.8},' +
'{"Date":"22 Jul 2014","Time":"16:37:06.014","X":139281,"Y":-66179.9,"Z":22688.4},' +
'{"Date":"22 Jul 2014","Time":"17:22:06.014","X":139148,"Y":-64588.2,"Z":22797.5},' +
'{"Date":"22 Jul 2014","Time":"18:07:06.014","X":138907,"Y":-62946.4,"Z":22888.8},' +
'{"Date":"22 Jul 2014","Time":"18:52:06.014","X":138556,"Y":-61254.7,"Z":22961.9},' +
'{"Date":"22 Jul 2014","Time":"19:37:06.014","X":138092,"Y":-59513.6,"Z":23016.3},' +
'{"Date":"22 Jul 2014","Time":"20:22:06.014","X":137515,"Y":-57723.5,"Z":23051.8},' +
'{"Date":"22 Jul 2014","Time":"21:07:06.014","X":136821,"Y":-55884.6,"Z":23067.6},' +
'{"Date":"22 Jul 2014","Time":"21:52:06.014","X":136008,"Y":-53997.3,"Z":23063.4},' +
'{"Date":"22 Jul 2014","Time":"22:37:06.014","X":135074,"Y":-52061.8,"Z":23038.5},' +
'{"Date":"22 Jul 2014","Time":"23:22:06.014","X":134015,"Y":-50078.4,"Z":22992.2},' +
'{"Date":"23 Jul 2014","Time":"00:07:06.014","X":132828,"Y":-48047.1,"Z":22923.9},' +
```

'{"Date":"23 Jul 2014","Time":"00:52:06.014","X":131509,"Y":-45968.3,"Z":22832.9},' +
'{"Date":"23 Jul 2014","Time":"01:37:06.014","X":130054,"Y":-43842.1,"Z":22718.2},' +
'{"Date":"23 Jul 2014","Time":"02:22:06.014","X":128459,"Y":-41668.6,"Z":22578.9},' +
'{"Date":"23 Jul 2014","Time":"03:07:06.014","X":126718,"Y":-39448.1,"Z":22413.9},' +
'{"Date":"23 Jul 2014","Time":"03:52:06.014","X":124826,"Y":-37180.5,"Z":22222.2},' +
'{"Date":"23 Jul 2014","Time":"04:37:06.014","X":122776,"Y":-34866.2,"Z":22002.5},' +
'{"Date":"23 Jul 2014","Time":"05:22:06.014","X":120562,"Y":-32505.3,"Z":21753.2},' +
'{"Date":"23 Jul 2014","Time":"06:07:06.014","X":118176,"Y":-30098.1,"Z":21472.9},' +
'{"Date":"23 Jul 2014","Time":"06:52:06.014","X":115609,"Y":-27644.9,"Z":21159.6},' +
'{"Date":"23 Jul 2014","Time":"07:37:06.014","X":112851,"Y":-25146.1,"Z":20811.3},' +
'{"Date":"23 Jul 2014","Time":"08:22:06.014","X":109891,"Y":-22602.3,"Z":20425.6},' +
'{"Date":"23 Jul 2014","Time":"09:07:06.014","X":106715,"Y":-20014.3,"Z":19999.8},' +
'{"Date":"23 Jul 2014","Time":"09:52:06.014","X":103308,"Y":-17383.1,"Z":19530.8},' +
'{"Date":"23 Jul 2014","Time":"10:37:06.014","X":99652.6,"Y":-14710.3,"Z":19014.7},' +
'{"Date":"23 Jul 2014","Time":"11:22:06.014","X":95728.2,"Y":-11997.9,"Z":18447.3},' +
'{"Date":"23 Jul 2014","Time":"12:07:06.014","X":91509.8,"Y":-9248.86,"Z":17823.2},' +
'{"Date":"23 Jul 2014","Time":"12:52:06.014","X":86967.4,"Y":-6467.29,"Z":17135.9},' +
'{"Date":"23 Jul 2014","Time":"13:37:06.014","X":82064.3,"Y":-3659.21,"Z":16377.5},' +
'{"Date":"23 Jul 2014","Time":"14:22:06.014","X":76754.2,"Y":-833.407,"Z":15537.9},' +
'{"Date":"23 Jul 2014","Time":"15:07:06.014","X":70978.2,"Y":1996.83,"Z":14603.8},' +
'{"Date":"23 Jul 2014","Time":"15:42:47.351","X":66015.5,"Y":4231.28,"Z":13784.5},' +
'{"Date":"23 Jul 2014","Time":"16:14:45.146","X":61239.5,"Y":6211.26,"Z":12981.6},' +
'{"Date":"23 Jul 2014","Time":"16:43:20.220","X":56663.5,"Y":7951.03,"Z":12198.9},' +
'{"Date":"23 Jul 2014","Time":"17:08:54.363","X":52289.8,"Y":9467.93,"Z":11438.4},' +
'{"Date":"23 Jul 2014","Time":"17:31:47.084","X":48117.9,"Y":10778.4,"Z":10701.4},' +
'{"Date":"23 Jul 2014","Time":"17:52:15.897","X":44145.3,"Y":11898.2,"Z":9988.66},' +
'{"Date":"23 Jul 2014","Time":"18:12:38.357","X":39935.3,"Y":12942.2,"Z":9221.22},' +
'{"Date":"23 Jul 2014","Time":"18:28:50.962","X":36373.4,"Y":13704.4,"Z":8561.65},' +
'{"Date":"23 Jul 2014","Time":"18:44:59.590","X":32609.3,"Y":14381.3,"Z":7853.71},' +
'{"Date":"23 Jul 2014","Time":"18:57:54.044","X":29420.7,"Y":14843.9,"Z":7244.57},' +
'{"Date":"23 Jul 2014","Time":"19:10:45.090","X":26063.7,"Y":15212,"Z":6593.16},' +
'{"Date":"23 Jul 2014","Time":"19:20:43.564","X":23314.9,"Y":15414.6,"Z":6051.37},' +
'{"Date":"23 Jul 2014","Time":"19:31:00.241","X":20333.1,"Y":15523.9,"Z":5454.26},' +
'{"Date":"23 Jul 2014","Time":"19:40:19.768","X":17479,"Y":15509.4,"Z":4872.58},' +
'{"Date":"23 Jul 2014","Time":"19:47:54.609","X":15041.5,"Y":15393.8,"Z":4367.03},' +
'{"Date":"23 Jul 2014","Time":"19:54:26.674","X":12846.8,"Y":15198.5,"Z":3904.09},' +
'{"Date":"23 Jul 2014","Time":"20:00:08.125","X":10858.8,"Y":14937.9,"Z":3477.63},' +
'{"Date":"23 Jul 2014","Time":"20:05:11.727","X":9027.54,"Y":14618.6,"Z":3078.04},' +
'{"Date":"23 Jul 2014","Time":"20:10:13.770","X":7143.92,"Y":14200.2,"Z":2659.39},' +
'{"Date":"23 Jul 2014","Time":"20:14:47.726","X":5381.99,"Y":13714.4,"Z":2259.74},' +
'{"Date":"23 Jul 2014","Time":"20:18:56.205","X":3742.03,"Y":13166.8,"Z":1879.65},' +
'{"Date":"23 Jul 2014","Time":"20:22:43.925","X":2208.59,"Y":12557.3,"Z":1515.96},' +
'{"Date":"23 Jul 2014","Time":"20:25:46.115","X":965.985,"Y":11982.2,"Z":1214.33},' +
'{"Date":"23 Jul 2014","Time":"20:28:47.087","X":-275.232,"Y":11321.9,"Z":905.742},' +
'{"Date":"23 Jul 2014","Time":"20:31:30.966","X":-1397.15,"Y":10637.3,"Z":619.35},' +
'{"Date":"23 Jul 2014","Time":"20:33:59.684","X":-2404.93,"Y":9936.78,"Z":354.826},' +
'{"Date":"23 Jul 2014","Time":"20:36:17.646","X":-3322.16,"Y":9213.97,"Z":106.822},' +
'{"Date":"23 Jul 2014","Time":"20:38:28.799","X":-4169.28,"Y":8458.01,"Z":-129.734},' +
'{"Date":"23 Jul 2014","Time":"20:40:36.645","X":-4962.21,"Y":7654.68,"Z":-359.289},' +
'{"Date":"23 Jul 2014","Time":"20:42:44.536","X":-5712.84,"Y":6785.65,"Z":-585.814},' +
'{"Date":"23 Jul 2014","Time":"20:44:55.140","X":-6424.59,"Y":5833.41,"Z":-811.489},' +
'{"Date":"23 Jul 2014","Time":"20:47:05.204","X":-7067.92,"Y":4825.77,"Z":-1027.94},' +
'{"Date":"23 Jul 2014","Time":"20:49:10.008","X":-7615.52,"Y":3811.11,"Z":-1225.49},' +
'{"Date":"23 Jul 2014","Time":"20:51:12.320","X":-8080.33,"Y":2780.67,"Z":-1407.52},' +
'{"Date":"23 Jul 2014","Time":"20:53:12.395","X":-8464.65,"Y":1744.18,"Z":-1573.67},' +
'{"Date":"23 Jul 2014","Time":"20:55:11.875","X":-8775.96,"Y":698.077,"Z":-1725.76},' +
'{"Date":"23 Jul 2014","Time":"20:57:16.831","X":-9027.72,"Y":-401.824,"Z":-1870.28},' +
'{"Date":"23 Jul 2014","Time":"20:59:36.631","X":-9224.96,"Y":-1628.4,"Z":-2014.43},' +
'{"Date":"23 Jul 2014","Time":"21:01:54.390","X":-9339.41,"Y":-2823.22,"Z":-2139.03},' +
'{"Date":"23 Jul 2014","Time":"21:04:06.968","X":-9382.49,"Y":-3953.1,"Z":-2243.58},' +
'{"Date":"23 Jul 2014","Time":"21:06:20.802","X":-9367.1,"Y":-5069.01,"Z":-2335.03},' +
'{"Date":"23 Jul 2014","Time":"21:08:39.894","X":-9296.47,"Y":-6199.26,"Z":-2416.45},' +
'{"Date":"23 Jul 2014","Time":"21:11:07.144","X":-9169.45,"Y":-7361.01,"Z":-2489.06},' +
'{"Date":"23 Jul 2014","Time":"21:13:45.296","X":-8982.27,"Y":-8568.04,"Z":-2553.25},' +
'{"Date":"23 Jul 2014","Time":"21:16:37.225","X":-8728.98,"Y":-9832.71,"Z":-2608.87},' +
'{"Date":"23 Jul 2014","Time":"21:19:45.879","X":-8402.04,"Y":-11165.2,"Z":-2655.23},' +

'{"Date":"23 Jul 2014","Time":"21:23:13.639","X":-7994.19,"Y":-12569,"Z":-2691.17},' +
'{"Date":"23 Jul 2014","Time":"21:27:01.532","X":-7501.3,"Y":-14037.4,"Z":-2715.27},' +
'{"Date":"23 Jul 2014","Time":"21:31:09.956","X":-6922.04,"Y":-15559.6,"Z":-2726.34},' +
'{"Date":"23 Jul 2014","Time":"21:35:40.529","X":-6253.4,"Y":-17132.7,"Z":-2723.53},' +
'{"Date":"23 Jul 2014","Time":"21:40:36.622","X":-5488.39,"Y":-18763,"Z":-2705.96},' +
'{"Date":"23 Jul 2014","Time":"21:46:03.170","X":-4615.81,"Y":-20462.4,"Z":-2672.34},' +
'{"Date":"23 Jul 2014","Time":"21:52:37.252","X":-3535.53,"Y":-22391.4,"Z":-2615.91},' +
'{"Date":"23 Jul 2014","Time":"21:59:31.570","X":-2379.75,"Y":-24293.7,"Z":-2541.85},' +
'{"Date":"23 Jul 2014","Time":"22:07:21.455","X":-1055.71,"Y":-26315.5,"Z":-2443.66},' +
'{"Date":"23 Jul 2014","Time":"22:15:56.177","X":400.475,"Y":-28386.8,"Z":-2322.78},' +
'{"Date":"23 Jul 2014","Time":"22:25:19.625","X":1992.47,"Y":-30505.7,"Z":-2178.29},' +
'{"Date":"23 Jul 2014","Time":"22:35:37.509","X":3728.27,"Y":-32675.4,"Z":-2008.84},' +
'{"Date":"23 Jul 2014","Time":"22:46:56.246","X":5616.8,"Y":-34898.8,"Z":-1812.87},' +
'{"Date":"23 Jul 2014","Time":"22:59:22.911","X":7667.53,"Y":-37178.1,"Z":-1588.62},' +
'{"Date":"23 Jul 2014","Time":"23:13:05.305","X":9890.4,"Y":-39514.5,"Z":-1334.18},' +
'{"Date":"23 Jul 2014","Time":"23:28:12.026","X":12295.6,"Y":-41908.4,"Z":-1047.5},' +
'{"Date":"23 Jul 2014","Time":"23:44:52.571","X":14893.8,"Y":-44359.2,"Z":-726.394},' +
'{"Date":"24 Jul 2014","Time":"00:03:17.418","X":17695.6,"Y":-46865.3,"Z":-368.534},' +
'{"Date":"24 Jul 2014","Time":"00:23:38.154","X":20711.7,"Y":-49423.6,"Z":28.5123},' +
'{"Date":"24 Jul 2014","Time":"00:46:07.611","X":23953,"Y":-52030.2,"Z":467.29},' +
'{"Date":"24 Jul 2014","Time":"01:10:59.980","X":27430.1,"Y":-54679.2,"Z":950.426},' +
'{"Date":"24 Jul 2014","Time":"01:38:30.971","X":31153.1,"Y":-57363.4,"Z":1480.61},' +
'{"Date":"24 Jul 2014","Time":"02:08:57.952","X":35131.5,"Y":-60073.1,"Z":2060.57},' +
'{"Date":"24 Jul 2014","Time":"02:42:40.146","X":39373.7,"Y":-62796.9,"Z":2693.02},' +
'{"Date":"24 Jul 2014","Time":"03:19:58.987","X":43887.2,"Y":-65520.4,"Z":3380.66},' +
'{"Date":"24 Jul 2014","Time":"04:01:18.042","X":48677.4,"Y":-68226.2,"Z":4126.03},' +
'{"Date":"24 Jul 2014","Time":"04:46:18.042","X":53665.1,"Y":-70851.7,"Z":4918.35},' +
'{"Date":"24 Jul 2014","Time":"05:31:18.042","X":58431.5,"Y":-73184.7,"Z":5690.38},' +
'{"Date":"24 Jul 2014","Time":"06:16:18.042","X":62992.1,"Y":-75259.7,"Z":6442.33},' +
'{"Date":"24 Jul 2014","Time":"07:01:18.042","X":67360.2,"Y":-77104.6,"Z":7174.56},' +
'{"Date":"24 Jul 2014","Time":"07:46:18.042","X":71547.2,"Y":-78742,"Z":7887.44},' +
'{"Date":"24 Jul 2014","Time":"08:31:18.042","X":75562.8,"Y":-80190.8,"Z":8581.38},' +
'{"Date":"24 Jul 2014","Time":"09:16:18.042","X":79415.6,"Y":-81466.9,"Z":9256.79},' +
'{"Date":"24 Jul 2014","Time":"10:01:18.042","X":83113.3,"Y":-82583.8,"Z":9914.04},' +
'{"Date":"24 Jul 2014","Time":"10:46:18.042","X":86662.3,"Y":-83553.1,"Z":10553.5},' +
'{"Date":"24 Jul 2014","Time":"11:31:18.042","X":90068.5,"Y":-84384.8,"Z":11175.5},' +
'{"Date":"24 Jul 2014","Time":"12:16:18.042","X":93337.2,"Y":-85087.7,"Z":11780.4},' +
'{"Date":"24 Jul 2014","Time":"13:01:18.042","X":96473,"Y":-85669.5,"Z":12368.4},' +
'{"Date":"24 Jul 2014","Time":"13:46:18.042","X":99480,"Y":-86137,"Z":12939.9},' +
'{"Date":"24 Jul 2014","Time":"14:31:18.042","X":102362,"Y":-86496.3,"Z":13495},' +
'{"Date":"24 Jul 2014","Time":"15:16:18.042","X":105122,"Y":-86752.9,"Z":14034},' +
'{"Date":"24 Jul 2014","Time":"16:01:18.042","X":107763,"Y":-86911.5,"Z":14557.1},' +
'{"Date":"24 Jul 2014","Time":"16:46:18.042","X":110289,"Y":-86976.6,"Z":15064.4},' +
'{"Date":"24 Jul 2014","Time":"17:31:18.042","X":112700,"Y":-86952.2,"Z":15556.1},' +
'{"Date":"24 Jul 2014","Time":"18:16:18.042","X":115000,"Y":-86841.8,"Z":16032.3},' +
'{"Date":"24 Jul 2014","Time":"19:01:18.042","X":117191,"Y":-86648.8,"Z":16493.2},' +
'{"Date":"24 Jul 2014","Time":"19:46:18.042","X":119273,"Y":-86376,"Z":16938.7},' +
'{"Date":"24 Jul 2014","Time":"20:31:18.042","X":121249,"Y":-86026.3,"Z":17369.2},' +
'{"Date":"24 Jul 2014","Time":"21:16:18.042","X":123120,"Y":-85602.2,"Z":17784.4},' +
'{"Date":"24 Jul 2014","Time":"22:01:18.042","X":124887,"Y":-85106,"Z":18184.6},' +
'{"Date":"24 Jul 2014","Time":"22:46:18.042","X":126552,"Y":-84539.8,"Z":18569.7},' +
'{"Date":"24 Jul 2014","Time":"23:31:18.042","X":128114,"Y":-83905.6,"Z":18939.8},' +
'{"Date":"25 Jul 2014","Time":"00:16:18.042","X":129575,"Y":-83205.1,"Z":19294.8},' +
'{"Date":"25 Jul 2014","Time":"01:01:18.042","X":130935,"Y":-82440.1,"Z":19634.7},' +
'{"Date":"25 Jul 2014","Time":"01:46:18.042","X":132194,"Y":-81612.1,"Z":19959.5},' +
'{"Date":"25 Jul 2014","Time":"02:31:18.042","X":133354,"Y":-80722.5,"Z":20269.1},' +
'{"Date":"25 Jul 2014","Time":"03:16:18.042","X":134414,"Y":-79772.7,"Z":20563.5},' +
'{"Date":"25 Jul 2014","Time":"04:01:18.042","X":135375,"Y":-78763.8,"Z":20842.6},' +
'{"Date":"25 Jul 2014","Time":"04:46:18.042","X":136235,"Y":-77697.1,"Z":21106.2},' +
'{"Date":"25 Jul 2014","Time":"05:31:18.042","X":136996,"Y":-76573.5,"Z":21354.2},' +
'{"Date":"25 Jul 2014","Time":"06:16:18.042","X":137657,"Y":-75394,"Z":21586.6},' +
'{"Date":"25 Jul 2014","Time":"07:01:18.042","X":138217,"Y":-74159.6,"Z":21803.1},' +
'{"Date":"25 Jul 2014","Time":"07:46:18.042","X":138676,"Y":-72871.1,"Z":22003.5},' +
'{"Date":"25 Jul 2014","Time":"08:31:18.042","X":139033,"Y":-71529.2,"Z":22187.7},' +
'{"Date":"25 Jul 2014","Time":"09:16:18.042","X":139287,"Y":-70134.7,"Z":22355.5},' +
'{"Date":"25 Jul 2014","Time":"10:01:18.042","X":139438,"Y":-68688.2,"Z":22506.6},' +

```

'{"Date":"25 Jul 2014","Time":"10:46:18.042","X":139484,"Y":-67190.3,"Z":22640.7},' +
'{"Date":"25 Jul 2014","Time":"11:29:10.811","X":139430,"Y":-65715.8,"Z":22752.4}' +
']}';

var mission = JSON.parse(trajjectory) ;

    // Trace the orbits for the following array of objects.
    var segIndex = 0 ;    // segment counter
    var orbitCoords = "" ;
for (var segment in mission.coordinates) {

    var s = document.createElement('Shape');           // Shape Node
    s.setAttribute("id", "segment" + segIndex);
    var app = document.createElement('Appearance');    // Appearance Node
    var mat = document.createElement('Material');      // Material Node
        mat.setAttribute("id", "Mat" + segIndex);
        mat.setAttribute("diffuseColor", 1 + " " + 0 + " " + 0);
        mat.setAttribute("emissiveColor", 1 + " " + 0 + " " + 0.3);
    app.appendChild(mat);
    s.appendChild(app);

    var segCoords = [mission.coordinates[segment].X/10000,
mission.coordinates[segment].Y/10000, mission.coordinates[segment].Z/10000] ;
    console.log("x  " + segCoords[0] + "  y  " + segCoords[1] + "  z  "
+ segCoords[2]) ;

        orbitCoords = orbitCoords + segCoords[0]  + " " + segCoords[1]  + "
" + segCoords[2]  + " " ;

    var line = document.createElement('IndexedLineSet');
    line.setAttribute("coordIndex", segIndex);
    var coords = document.createElement('Coordinate');
    coords.setAttribute("point", orbitCoords);

    line.appendChild(coords) ;

    s.appendChild(line);
    var ot = document.getElementById('theEarth');
    ot.appendChild(s);

    segIndex = segIndex + 1 ;
}

document.getElementById("demo").innerHTML =
"<p>" + "Mission Start Date: " +
mission.coordinates[0].Date + "  " +
mission.coordinates[0].Time + "<p>" +
"Mission End Date: " +
mission.coordinates[154].Date + "  " +
mission.coordinates[154].Time ;

</script>
<script type="text/javascript" src="x3dom.js"></script>
</body>
</html>

```