

## Ascii Point Import/Export/Report Template files

Carlson Layout uses open source [NLTemplate](#) formatted files to define the format for ascii file point import/export. NLTemplate format is also used for point export to CSV and HTML files.

In Windows default templates are installed into  
C:\Users\"user"\AppData\Local\CarlsonLayout\Config\PredefinedTemplates

In Android default templates are installed into  
Internal Storage/CarlsonLayout/Config/PredefinedTemplates

Users can make copies and modify the default templates to create custom file import/export formats. Any file with the .io extension added to the template folder will be listed in Layout as a possible ASCII Coordinate file format to be imported or exported.

The image displays two side-by-side screenshots of the Carlson Layout application interface. Both screens have a dark blue header with a 'TS Setup' button and several icons. The left screen, titled 'Export Data', features a light gray background with three main sections: 'Export:' with a dropdown menu set to 'Coordinates', 'Format:' with a dropdown menu set to 'ASCII (text)', and 'File Name:' with a text input field containing 'Example Job'. An 'Export' button is located at the bottom right. The right screen, titled 'Export Coordinate File', also has a light gray background and includes a 'Format Description:' dropdown set to 'PXYZD Comma', an 'Example Point:' text field with the value '1,10000.00,5000.00,100.00,Description', a 'Destination Units:' dropdown set to 'Meters', an empty 'Add to Point IDs:' text field, and a 'Points to Export:' dropdown set to 'All'. It includes both 'Cancel' and 'Export' buttons at the bottom.

It is recommended to name the files in a way that communicates the format, as the filename will be displayed as the format identifier in the combo box.

Here is an example 2 line file **PXYZD Comma.io**

```
{% block point %}{ PTID },{ X },{ Y },{ Z },{ DESC }  
{% endblock %}
```

Everything between {% block point %} and {% endblock %} will be repeated for each point in the file during the export. The following tags are available to identify fields:

```
{{ PTID }}  
{{ X }}  
{{ Y }}  
{{ Z }}  
{{ DESC }}  
{{ LAYER }}  
{{ DATE }}  
{{ TIME }}  
{{ HSDEV }}  
{{ VSDEV }}
```

### Tag Notes

Date and Time tags are in UTC format.

In the **PXYZD Comma.io** example you can see that commas separate the fields. Since {% endblock %} is on a second line, each point will be inserted on a separate line.

Text in the file before {% block point %} will be exported before the list of points and text after {% endblock %} will be exported after the list of points. This allows for creating header and footer information in the output. Text in between the {% block point %} and {% endblock %} tags will be repeated for each point.

It is possible to insert another file inside the template file using this syntax:

```
{% include example.txt %}
```

When including another file, it is as if that file were inserted inside the other file. This is helpful if you have a standard header or footer that you want to add and use for all of your templates.

### Export HTML/PDF Coordinates

The NLTemplate format is also used for exporting point coordinates to HTML, PDF and CSV.

Carbon Layout

← TS Setup

2.900m

**Export Data**

**Export:** Coordinates

**Format:** HTML

**File Name:** TS Setup

Export

The Point Coordinate HTML and PDF reports are generated from the file `point_report_html_template.txt` in the `PredefinedTemplates` folder.

Carbon Layout

← TS Setup

2.900m

**Export Data**

**Export:** Coordinates

**Format:** CSV

**File Name:** TS Setup

Export

Point Coordinates CSV reports are generated from the file `point_report_csv_template.txt`.

## Examples:

1)

Point\_report\_html\_template.txt

Output: html file

```
<html><body>

<head>
  <style>
    th {text-align:left; font-size:10pt;}
    tr {text-align:left; font-size:10pt;}
  </style>
</head>

<h2 style="text-align: center; width: 100%;">Point Report</h2>

<table width="100%">
  <tr>
    <th>Pt ID</th>
    <th>X</th>
    <th>Y</th>
    <th>Z</th>
    <th>Desc</th>
  </tr>
  {% block point %}
    <tr>
      <td>{{ PTID }}</td>
      <td>{{ X }}</td>
      <td>{{ Y }}</td>
      <td>{{ Z }}</td>
      <td>{{ DESC }}</td>
    </tr>
  {% endblock %}
</table>

</body></html>
```

2)

Point\_report\_csv\_template.txt

Output: csv file

```
Pt ID,X,Y,Z,Desc  
{% block point %}{{ PTID }},{ X }},{ Y }},{ Z }},{ DESC }}  
{% endblock %}
```