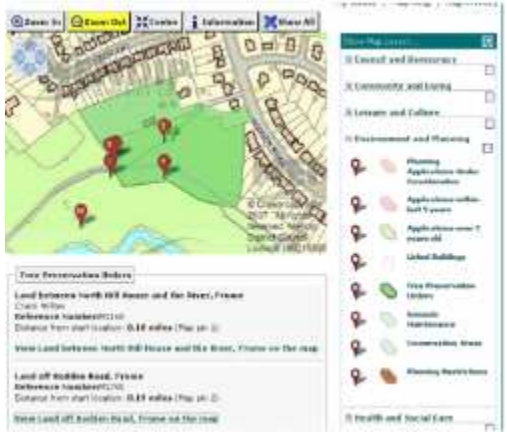

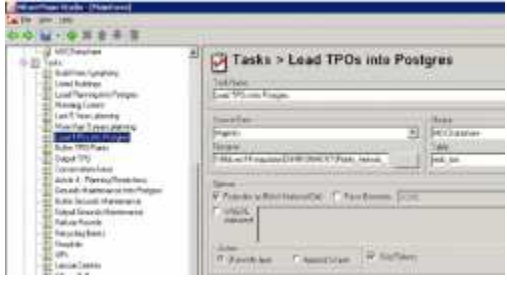
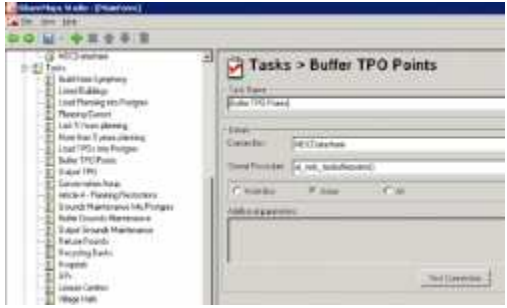


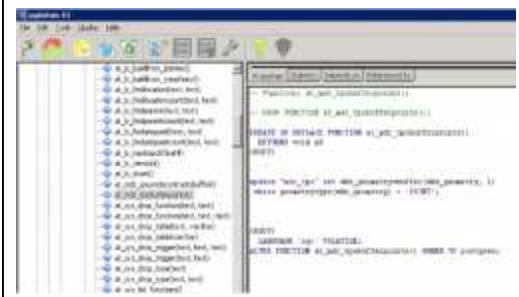


Neither this document nor any part of it should be disclosed to any third party without the prior written consent of Astun Technology Limited. You acknowledge that this document has been provided in reliance upon it being subject to the exemption to disclose. If you disagree that this exemption applies then please return the information and expunge all records of it from your systems, unless we agree to waive this requirement.

How do I add information on TPOs?

Steps	Results
<p>In order to display TPO information on the map the point TPO information needs to be turned into polygons as MapServer cannot show Points and Polygons in the same layer.</p> <p>This is easily achieved by creating Tasks in the Workflow and creating a Job to run them.</p>	
<p>First we have to create a Connection in the Workflow section of iShareMaps Studio to the PostgreSQL database if this does not already exist.</p>	
<p>Then we need to create the individual Tasks to massage the data.</p> <p>The first Task that is required is to copy the TPO data from MapInfo into the Postgres database.</p> <p>In this screenshot we have created a Task called "Load TPOs into Postgres" where we are copying the MapInfo .tab file to the Postgres database into a table called mdc_tpo.</p>	
<p>Now we need to create a Task to add a buffer around all TPO point data to create polygons.</p> <p>This is done using a Stored Procedure supplied by Astun Technology.</p> <p>This Stored Procedure has been created in the Postgres database.</p>	

Here is an example of the Stored Procedure.



The last Task required is to output the data from the **mdc_tpo** table in the Postgres database to a **MapInfo .tab** file. Here it is called **TPO.tab**



Having created the individual tasks we now need to create a Job to run the tasks in the correct sequence.

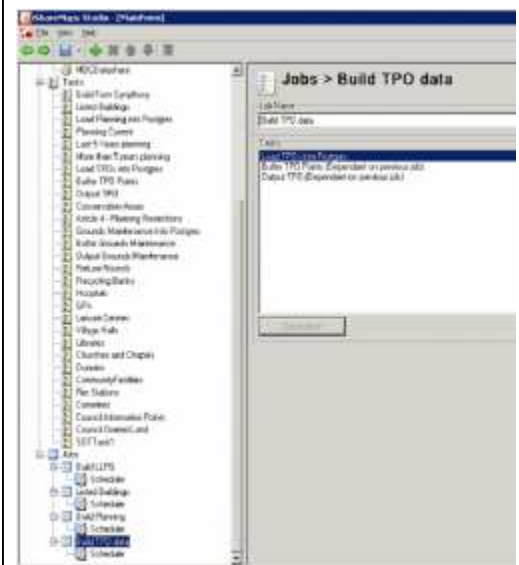
Here we have created a Job called Build TPO data.

The Tasks that need to be performed are:

- **Load** the TPO information into the Postgres database.
- Create a **Buffer** around the TPO Points
- **Output** the TPO information to a MapInfo .tab file

Each of the tasks needs to be dependent upon the first task running successfully. If the task does not run successfully then an email is sent to a predefined address.

The scheduler entry can then be updated to say when and how often this Job should be run.



Here is the result →

