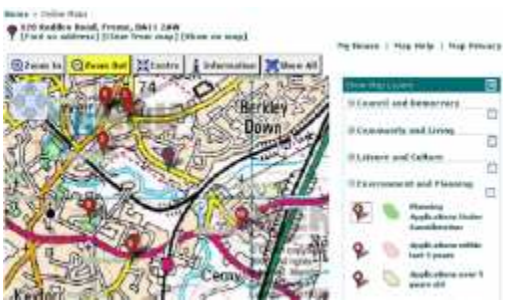
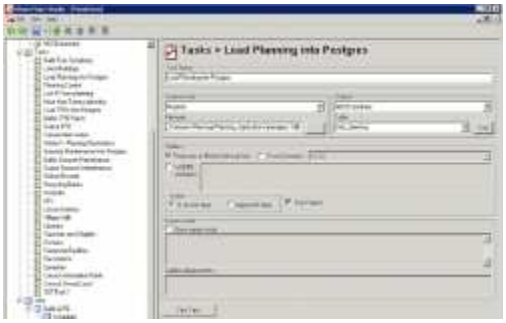
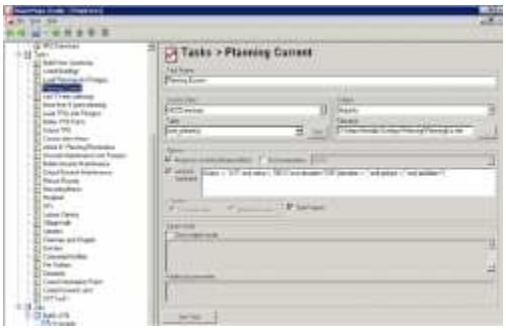
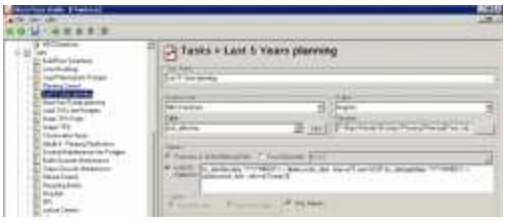
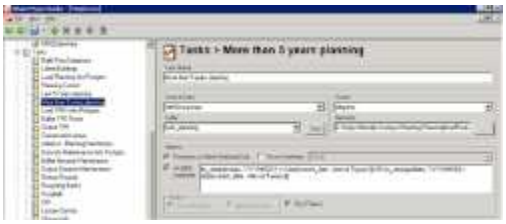
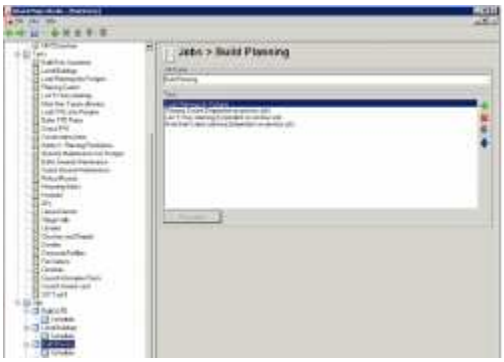
How do I split planning application information?

Steps	Results
You may wish to split your planning applications into the following categories: • **Planning Applications under consideration** • **Applications within last 5 years** • **Applications over 5 years old** This scenario will show you how to achieve this.	
First we have to create a **Connection** in the **Workflow** section of iShareMaps Studio to the **PostgreSQL** database if this does not already exist.	
Then we need to create the individual **Tasks** to massage the data. The first Task that is required is to copy the planning data from MapInfo into the Postgres database. In this screenshot we have created a Task called "**Load Planning into Postgres**" where we are copying the MapInfo .tab file to the Postgres database into a table called **mdc_planning**	
The next task is to create the subset for **Planning Applications under consideration**. Here we are creating a Task called **Planning Current**. Now we need to select the relevant records. To do this we will select the data from the Postgres database using the **WHERE** clause shown in the screenshot and create a new **MapInfo** .tab file called **PlanningCur.tab**	
Now we need to create the next subset for **Applications within last 5 years.** Here we are creating a Task called **Last 5 years planning**. Here the **WHERE** clause is different as we need to select the records by	

date. By selecting records where the **decision date** or **appeal date** is **greater than** or **equal to** the **current_date** this means that this selection will always be up-to-date.	
The last Task that we need to create is for **Applications over 5 years old.** Again we will use the **WHERE** clause which will be the same as for the previous task only the selection will be based on the dates being **less than** the **current_date**.	
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 Planning**. The Tasks that need to be performed are: ● **Load** the Planning information into the Postgres database. ● Create the **Planning Current** subset ● Create the **Last 5 years planning** subset ● Create the **More than 5 years planning** subset. 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 →