Grid Query

Grid queries are used to build new grids from existing grids where the new grid values are derived according to whether specific queries imposed on the existing grid files have been met. For example, the user can create an expression that will query a set of grid files and identify all coincident cells that:

in Grid1 are > 1049m   AND

in Grid2 are <= $250,754.50   AND

in Grid3 are = “Dense Urban”   OR

in Grid4 are = “Leeds County”.

Therefore, by examining each grid and “drilling down” through the four grid layers on a cell-by-cell basis, a fifth grid can be built from the overlying and spatially coincident cells that identifies all cells that meet (or do not meet) all of the query conditions.

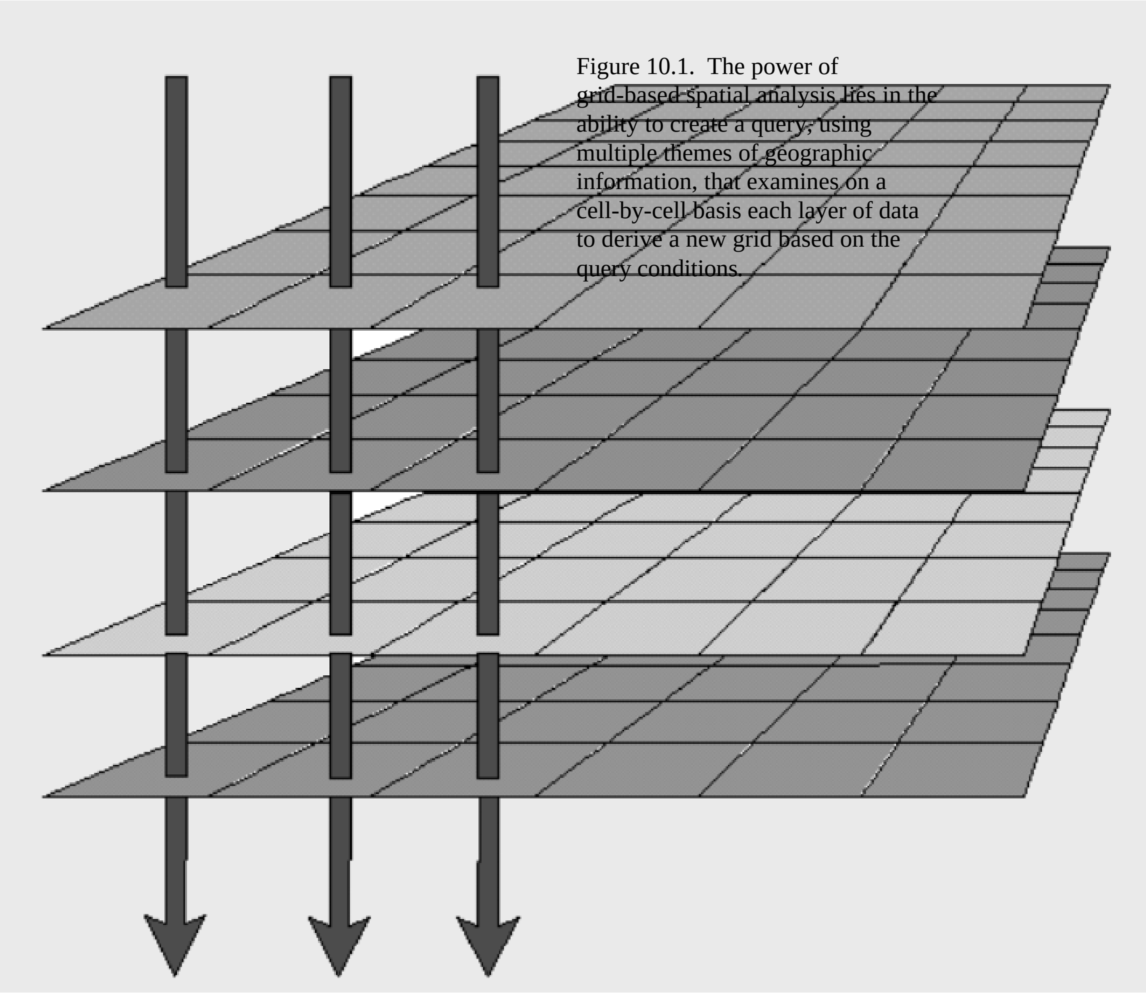


Figure 10.1. The power of grid-based spatial analysis lies in the ability to create a query, using multiple themes of geographic information, that examines on a cell-by-cell basis each layer of data to derive a new grid based on the query conditions.

There is no limit to the number of grids that can be used in a *Vertical Mapper Grid Query* expression. However, there are a number of restrictions on the specifications of grids that are used in a multiple grid query.

a) All grids used in a grid query must be in the same MapInfo coordinate system;
b) all grids must be spatially coincident or at least have partial overlap; and,
c) if grids have different cell sizes, the query will be controlled by the grid with the smallest cell size unless the user intervenes.

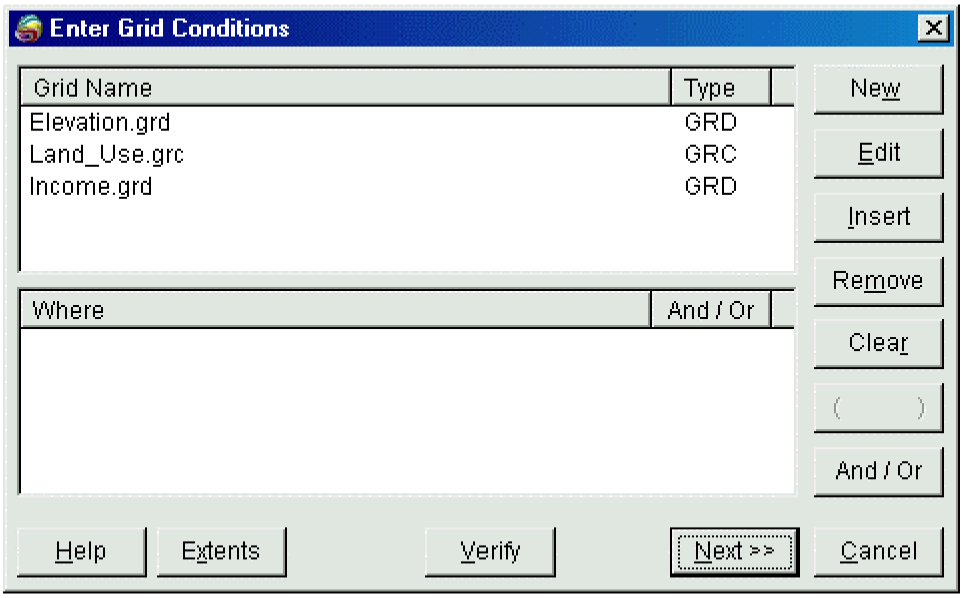
The *Grid Query* tool is comparable to MapInfo's SQL Select command although, in the MapInfo function, the new Query table preserves all of the original data records that meet the SQL conditions. In the *Grid Query* function of *Vertical Mapper*, new cells are created in the derivative grid that are assigned a user-defined value according to whether the cell did or did not meet the conditions of the query expression. This is based on a simple true/false test: Does each cell in the new grid meet or not meet ALL of the query conditions? For an instructional lesson on *Grid Query* refer to Lesson 12 of the *Installation and Tutorials* manual.

Creating and Editing Conditional Queries

A *Grid Query* command in *Vertical Mapper* makes use of a very simple two-dialogue procedure. The first step involves building a query expression from a selection of open grid files as a series of individual conditional clauses. The second stage involves the assignment of values to the new grid, cell by cell, according to whether the query conditions are met (true) or not met (false) for each set of overlying cells.

Select the *Grid Query* command from the drop-down menu in the *Grid Manager*.

1 The ***Enter Grid Conditions*** dialogue is used to build conditional queries from one or more input grids.

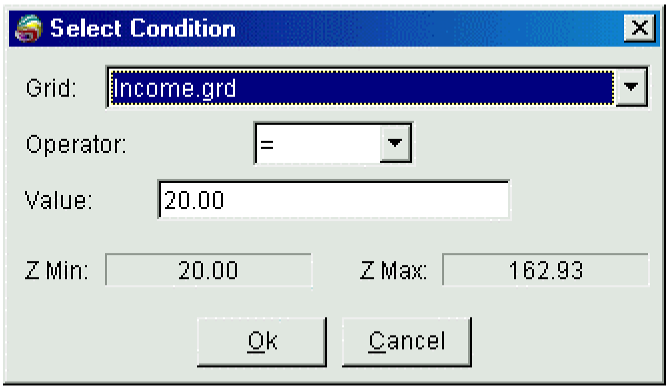


Grid Name	Type
Elevation.grd	GRD
Land_Use.grc	GRC
Income.grd	GRD

Where	And / Or

- The upper window of the *Enter Grid Conditions* dialogue box lists all of the currently open grid files in the *Vertical Mapper Grid Manager*. In order to create a new query, the grid entry must be selected.

- The ***New*** button brings up the ***Select Condition*** dialogue which is used to build a specific conditional query. Alternately, the user may highlight the appropriate grid file from the list and click on the right mouse button to bring up the *New Clause* command.


Select Condition

Grid: Income.grd

Operator: =

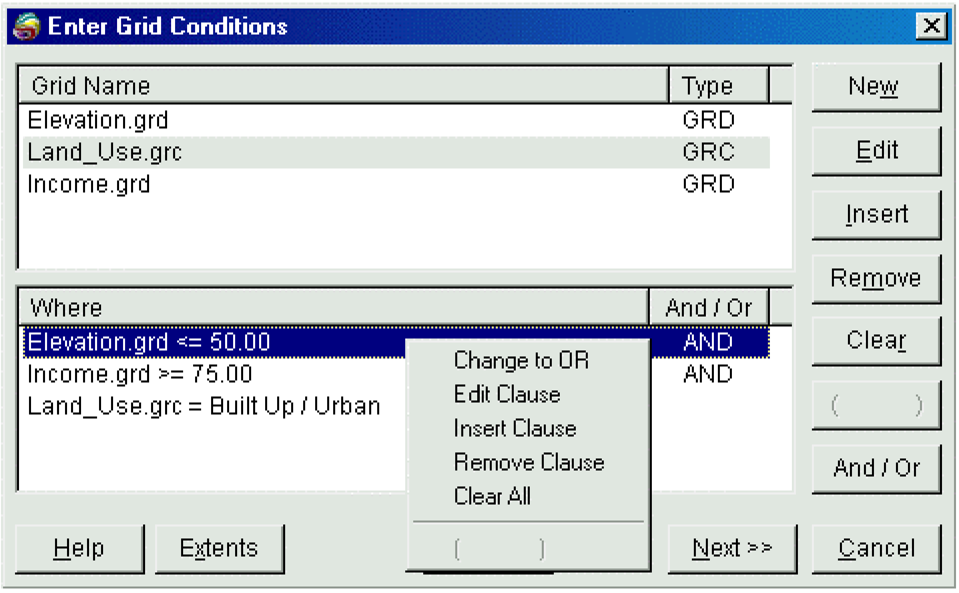
Value: 20.00

Z Min: 20.00

Z Max: 162.93

1. The grid file name drop-down list is used to select the grid to be queried.
2. The conditional operator drop-down list is used to select the appropriate operator for the query. The choices available vary depending upon whether the grid is numeric or character-based. In both cases, the user may also choose to select "**Any Value**" or "**Null Value**".
3. If the condition requires the entry of a specific user-defined value, enter that value in the edit box. If the query is being created for a character-based grid, this box becomes a drop-down list since there are only a specific number of unique classes in any .GRC file.
4. The Z-value range of any numeric grid file being queried is recorded as ***Z Min:*** and ***Z Max:*** at the bottom of the dialogue.

1 The query expression itself is built interactively by the user in the ***Where*** box of the dialogue with individual query clauses separated, by default, with an "AND" Boolean operator. The following commands are available to modify individual highlighted clauses in the *Where* list and are found both as buttons on the right side of the dialogue or as a list of commands initiated as a right mouse button click on a selected clause.



Grid Name	Type
Elevation.grd	GRD
Land_Use.grc	GRC
Income.grd	GRD

Where	And / Or
Elevation.grd <= 50.00	AND
Income.grd >= 75.00	AND
Land_Use.grc = Built Up / Urban	

1. The ***Edit*** button and ***Edit Clause*** command will re-open a *Select Condition* dialogue for any conditional clause and allow the user to make appropriate changes.

2. The ***Insert*** button and the ***Insert Clause*** command will open the *Select Condition* dialogue box and allow the user to insert a new clause before the selected one.

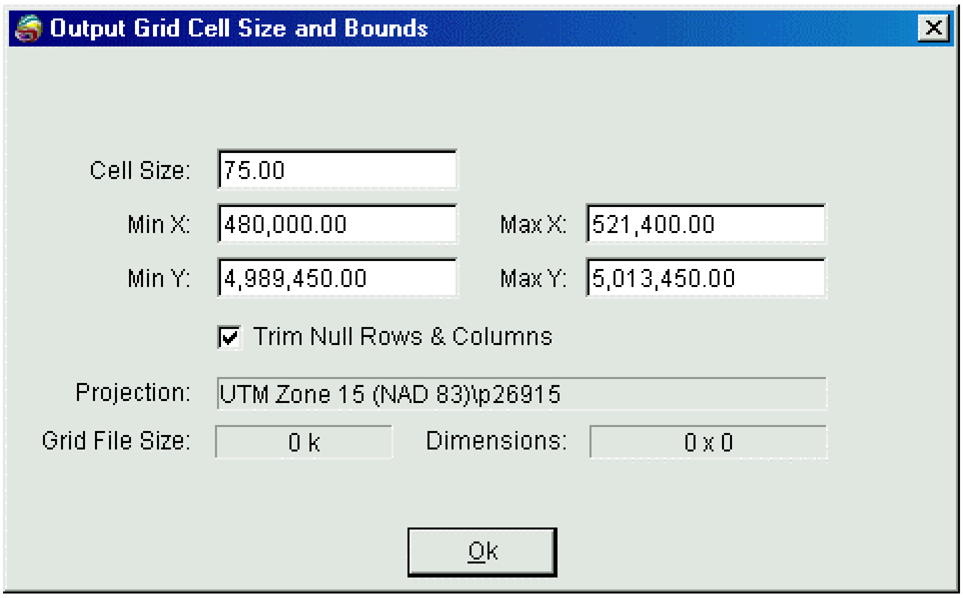
3. The ***Remove*** button and the ***Remove Clause*** command will delete the selected clause.

4. The ***Clear*** button and the ***Clear All*** command will delete all clauses in the *Where* list.

5. The ***(   )*** button and command joins and places brackets around any two or more clauses and converts them into a single line. This most commonly applies to an expression containing clauses that are joined by both "AND" and "OR" operators where the conditional logic may be unclear. For example, without brackets, the process order is unclear in the expression *"GridA = x AND GridY = y OR GridY = z."*. The command is only available if two or more clauses are selected in the *"Where"* list.

6. The ***And/Or*** button and ***Change to AND/Change to OR*** command is used to toggle between the AND and OR operator that joins two clauses.

1 The ***Extents*** button in the *Enter Grid Conditions* dialogue is used to access the ***Output Grid Cell Size and Bounds*** dialogue. This dialogue controls the specifications of the output grid file generated from the *Grid Query* command.



Output Grid Cell Size and Bounds

Cell Size: 75.00

Min X: 480,000.00 Max X: 521,400.00

Min Y: 4,989,450.00 Max Y: 5,013,450.00

☑ Trim Null Rows & Columns

Projection: UTM Zone 15 (NAD 83)\p26915

Grid File Size: 0 k Dimensions: 0 x 0

2. The ***Cell Size*** can be set to any user-defined value but, as with any grid creation technique, cell size is inversely proportional to file size. This value defaults to the smallest cell size found in all of the open grids.
3. The ***Min: X/Y*** and ***Max: X/Y*** can be set to any user-defined values. They default to the maximum extent covered by the grids used in the query expression.
4.
5. The ***Projection*** field records the coordinate system of the open grids and is a non-editable setting for the *Grid Query* command. Remember that all grids included in a *Grid Query* expression must be in the same MapInfo coordinate system that will also be the coordinate system of the output grid.
6. When checked, the ***Trim Null Rows & Columns*** option removes any columns or rows of null value cells from the edges of the grid.

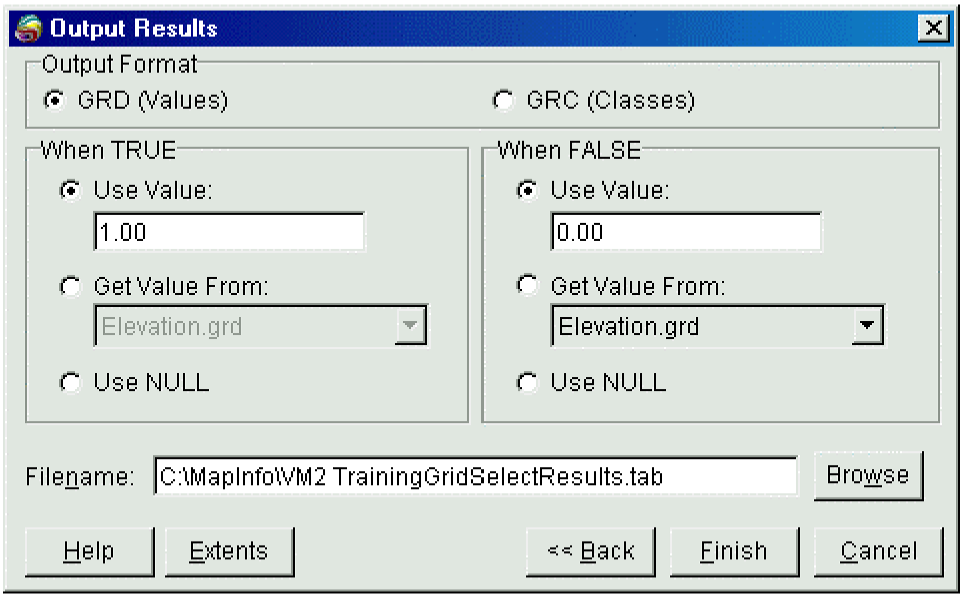
- The ***Verify*** button is used to check the process structure of the *Where* expression as a whole but does not verify the logic of an individual clause. The command will return ***No Errors*** if the expression is structured correctly.
- Once the expression has been verified and checked for logic by the user, the ***Next*** >> button is selected to bring up the ***Output Results*** dialogue.

Structuring the Output Results of a Grid Query

The second stage of a *Grid Query* command in *Vertical Mapper* involves the assignment of values on a cell-by-cell basis to the new grid according to whether the query conditions are met (true) or not met (false) for each set of overlying cells.

1 The ***Output Results*** dialogue box is designed to guide the user in making three basic decisions regarding the form of the output grid file.

1. What type of output grid to build: numeric or character-based.
2. What value to assign to grid cells that DO meet all of the query conditions.
3. What value to assign to grid cells that DO NOT meet the query conditions.



Output Format
● GRD (Values) ○ GRC (Classes)

When TRUE
● Use Value: 1.00
○ Get Value From: Elevation.grd
○ Use NULL

When FALSE
● Use Value: 0.00
○ Get Value From: Elevation.grd
○ Use NULL

Filename: C:\MapInfoVM2 TrainingGridSelectResults.tab

- In the first part of the dialogue, the ***Output Format*** is chosen to be either:

1. a .***GRD*** file where each cell in the grid is assigned a numeric value;

or

2. a .***GRC*** file where each cell in the grid is assigned to a character-based class.

- On the left side of the dialogue, the user must choose the value or class that will be assigned to each cell in the new grid when the grid query returns ***When TRUE***, meaning that all *Where* conditions have been met for that particular cell site. One of three options must be chosen.

    2. The function may assign a specific ***Use Value/Class*** to each cell that is user-defined and entered into the edit box.
    3. The function may ***Get Value From*** a specified grid that is chosen from a drop-down list of all currently open files. The value/class given to each *TRUE* cell in the new grid will thus correspond to the same cell value/class in the specified grid.
    4. The function may be directed to simply ***Use Null*** for all *TRUE* cells in the new grid.

-

- In the same manner, on right side of the dialogue, the user is requested to choose the value or class that will be assigned to each cell in the new grid when the grid query returns ***When FALSE***, meaning that not all *"Where"* conditions have been met for that particular cell site. Again, one of three options must be chosen.

    1. The function may assign a specific ***Use Value/Class*** to each cell that is user-defined and entered into the edit box.
    2. The function may ***Get Value From*** a specified grid that is chosen from a drop-down list of all currently open files. The value/class given to each *FALSE* cell in the new grid will thus

correspond to the same cell value/class in the specified grid.

3. The function may be directed to simply ***Use Null Value*** for all *FALSE* cells in the new grid.

- A file name must be entered for the new grid in the Filename edit box.
- The user may select either the ***Finish*** button to complete the gridding process or, if modifications to the previous dialogue are required, the **<< *Back*** button to return to one dialogue back.
- Once the grid file is created, it appears in a MapInfo Map window with a default colour palette applied.