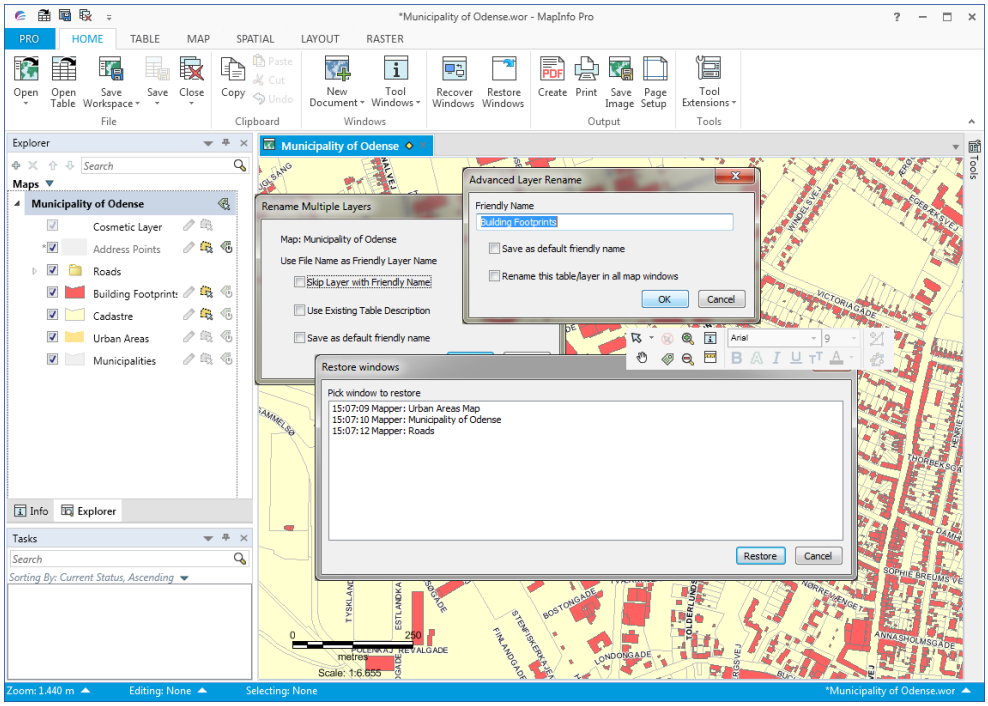
WindowHelper

For MapInfo Pro

December 2024



Contents

Installation ... 2
About WindowHelper ... 2
Features of WindowHelper ... 3
Windows in General ... 3
Modify Window ... 3
Restore Closed Windows ... 3
Save Window to Workspace ... 3
Tables, Selections and Files ... 5
Locate File in Windows Explorer ... 5
Show Info on Selection ... 5
Open File In Favorite Text Editor ... 5
Close Table ... 5
Map Windows ... 6
Open Workspace Explorer Window ... 6
Creating Map Extent ... 6
Add Smart Text to Map ... 7
Map Windows - Layers ... 8
Advanced Rename Layer ... 8
Rename Multiple Layers ... 8
Set Default View ... 8
View Entire Layer(s) ... 9
Apply Styles ... 9
Copy and Paste Layer Properties ... 10
Move Layer to Top/Bottom ... 11
Map Windows – Thematic Layers ... 12
Static Theme ... 12
Quick Ranged Theme ... 13
Quick Individual Theme ... 14
Save Theme to Object Styles ... 15
Browser Window ... 16
Browse in Floating Window ... 16
Pan and Zoom to Selection ... 16
Filter By Cell Value ... 17
Group By Current Column ... 17
Clear Selection ... 18
Layout Window ... 20
Clear Selection in Layout window ... 20
Edit Text in Dialog ... 20
Function List ... 21
Published Custom MapBasic Functions ... 22
Options ... 25

Installation

WindowHelper is to be downloaded from the MapInfo Marketplace.

This distribution is automated so that you only have to click a button to get the add-in installed and loaded into MapInfo Pro.

If any updates to the tool gets published, you will see the small ***Notification*** symbol in the lower right corner of the MapInfo Pro window turn red. Double-click on the symbol to open the ***Notification*** window and from here access the updates from the MapInfo Marketplace where you easily can install the updates.

Please note that WindowHelper requires MapInfo Pro v17.0.3 or newer to work.

About WindowHelper

WindowHelper is a tool that adds additional features to MapInfo Pro which helps you while working in especially map and browser windows.

To get the most out of this tool it is recommended that you set it to Autoload.

The features of this tool have been made in response to requests from customers.

If you have any requests or feedback on the WindowHelper, please send your feedback through this article on the MapInfo Pro Community: WindowHelper

Features of WindowHelper

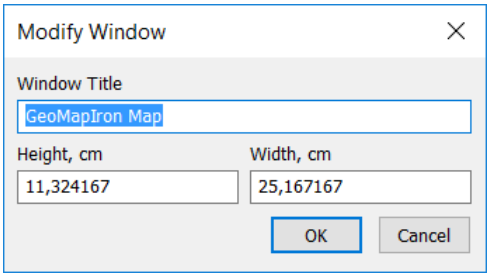
When you run the WindowHelper tool, it will place buttons on the ribbon, menu items in several context menus and embed itself in the Tools window.

The *WindowHelper* tool embeds itself in several places. Some features are placed directly on tabs of the ribbon and others can be found in context menu/right click menus.

Windows in General

Modify Window

The ***Modify Window option*** can be found on the ***HOME tab*** in the ***Windows group***. It lets you specify the floating size of document window. This is useful if you need to export your map window to an image of a certain size.



The ***Modify Window dialog*** lets you specify a title for the window and set a height and width of the window using the current paper units.

Click *OK* to apply the values entered in the dialog.

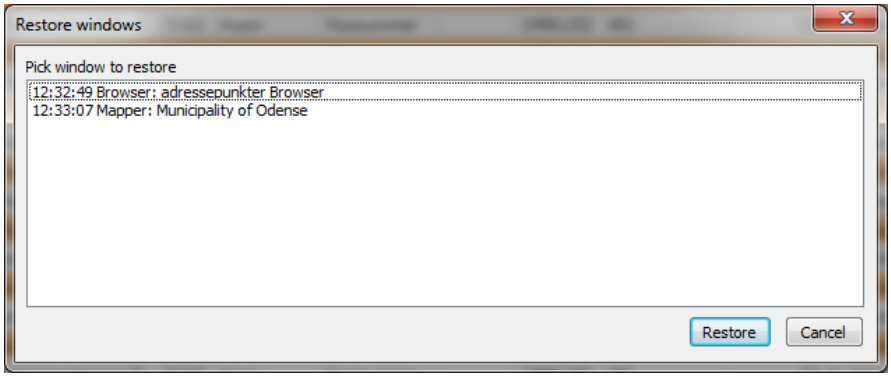
If your window is not floating, you will not see the size change until you make the window floating. You can do this by right clicking on the tab of the window and select *Floating* from the context menu.

Restore Closed Windows

On the ***MAP tab*** in the ***Options group*** you also find the feature ***Restore Windows***.

This option lets you restore windows that have been closed during the current session. This can help you recreate a map that you closed by accident.

Click on the ***Restore Windows button*** and select the window that you want to restore. Then click on ***Restore***.



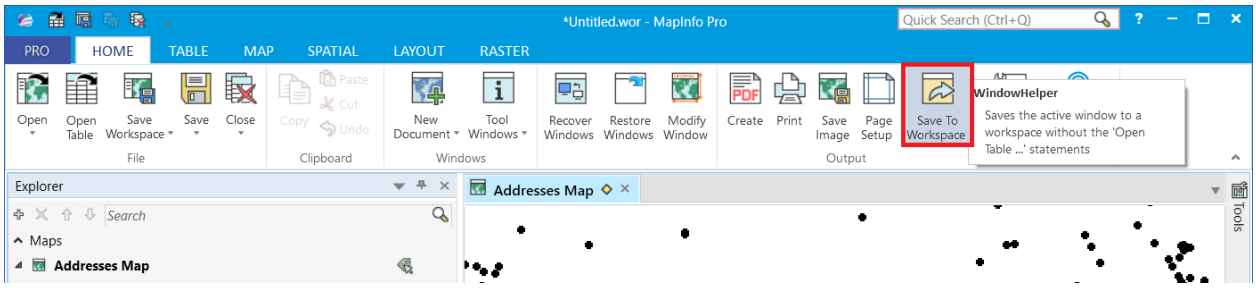
When you have restored all the windows, you want to, close the ***Restore Windows dialog*** via the ***Cancel button***.

What happens is that WindowHelper takes a snap shot of the window just before the window is being closed. This snap shot is similar to the statement that you will find in a workspace.

There is however one difference: WindowHelper does not "capture" which tables needs to be open. All it captures is the Window statement. So if you have closed some tables since you closed the window, the restore window might not work.

Save Window to Workspace

The ***Save Window to Workspace control*** can be found on the ***HOME tab*** in the ***Output group***. It saves the currently active window to a workspace file without saving any of the *Open Table* statements.



It can be used for creating templates for layouts or even map windows. It's up to you as a user to make sure that the tables used in the window are open when you reopen this workspace.

Tables, Selections and Files

Locate File in Windows Explorer

In the right click menu of the layers in the *Layer List* and the tables in *Table List* in the *Explorer window* you find this menu item, which lets you locate and select the file of a table/layer in Windows Explorer.

This feature works on one selected table or layer.

If you have selected a raster layer/table, you will be asked if you want to locate the tab file or the raster file.

This has been disabled on the *Table List* for MapInfo Pro v2021 or newer where this feature has been built into the core product. Here it is called *Open Containing Folder*.

Show Info on Selection

When activated, the tool shows the attributes of the selected (and deselected) record in the *Info window*.

You can turn the tool on and off via the *Show Info control* in the *Options group* on the *MAP tab*.

This has been enhanced to also show the number of selected records in the statusbar. It shows the number of selected records and the total number of records in the base table. This is always active - also when *Show Info on Selection* has been deactivated.

Open File In Favorite Text Editor

In the right click menu of the tables in the *Table List* and the right click menu of the layers in the *Layer List* in the *Explorer window* you find this menu item, which lets you open the tab file of a table/layer in your preferred text editor, for example Notepad.

This feature works on one selected table or layer.

This gives you quick access to the content of the tab file and an overview of the table structure as well as the metadata of the tab file.

Close Table

In the right click menu of the layers in the *Layer List* you find the *Close table menu item*, that makes it easy to close the table for a specific layer.

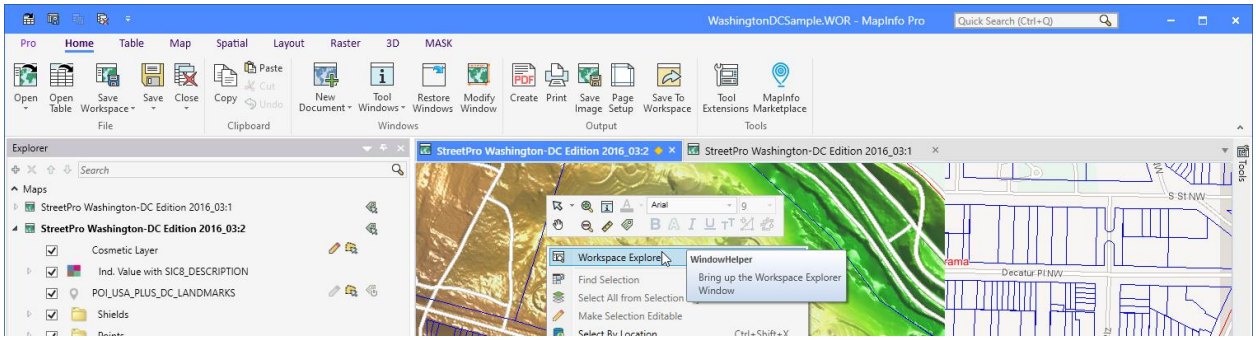
This feature works on one and multiple selected layers.

Notice that when you close the table thru the layer, all windows only showing this table and queries based on this table are also closed.

Map Windows

Open Workspace Explorer Window

From the right-click menu in the map window, you can now open the *Workspace Explorer window*.



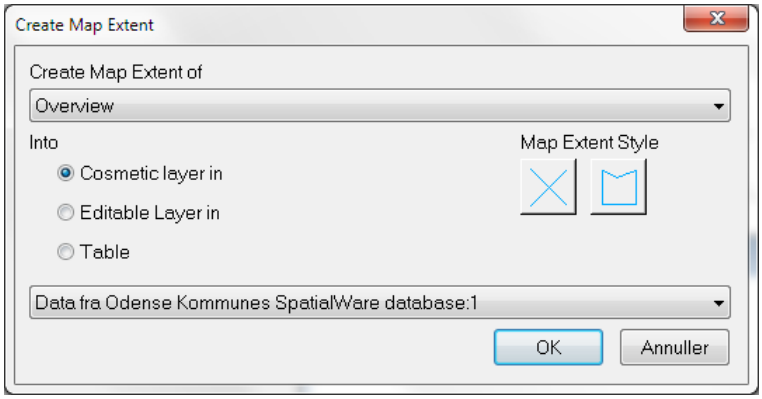
This is one of the menu items that you can control if you want to have via the *Options dialog*.

Creating Map Extent

You can use WindowHelper to create the extent from one map window in another map window. This can be used for printing or simply for seeing the position of your detailed map in an overview map.

You can also create the map extent of the map window it the window itself. This can be useful it you need to find objects within the current map extent.

On the *MAP tab* in the *Options group*, click on *Map Extent* to start the process. You can also access this feature via the right click menu in the map window.



In the *Create Map Extent dialog* you can now pick the window which map extent you want to create.

You can choice three ways of storing the map extent:

- The cosmetic layer of a map
- The editable layer of a map
- A table

If you pick 1 or 2, you must also pick what map window to store the map extent into. If you pick the 3rd option you must pick what table to store the map extent into.

In the dialog, you can also change the style for the map extent created. The style will be written to the configuration file and used again next time you create a map extent.

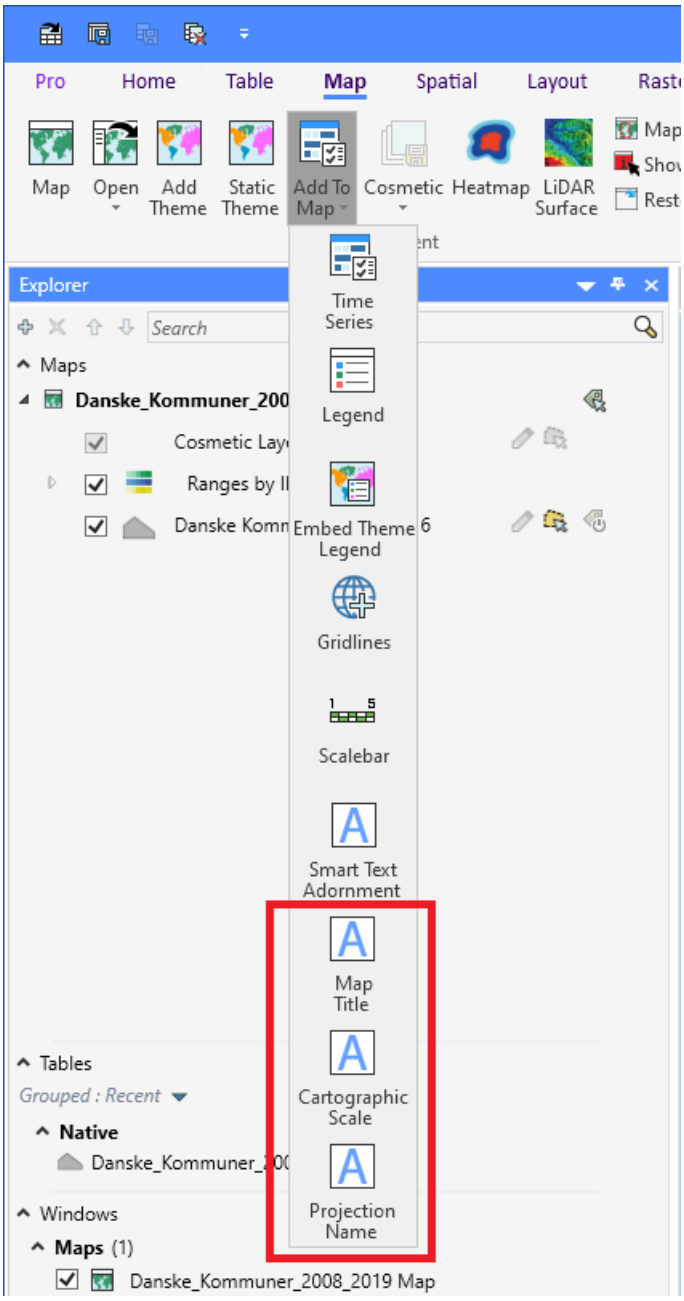
Add Smart Text to Map

MapInfo Pro v2021 came with a new capability: Smart Text Annotations for maps.

WindowHelper can help you add a few commonly used annotations to the map with a single click.

From the Add to Map dropdown, you can choose between three options:

- Map Title: This will show the name of the map in the upper left corner of the map window.
- Cartograhic Scale: This will show the scale in the lower left corner of the map in this form: "1:10,000".
- Projection Name: This will show the name of the projection for the map in the lower right corner of the map.

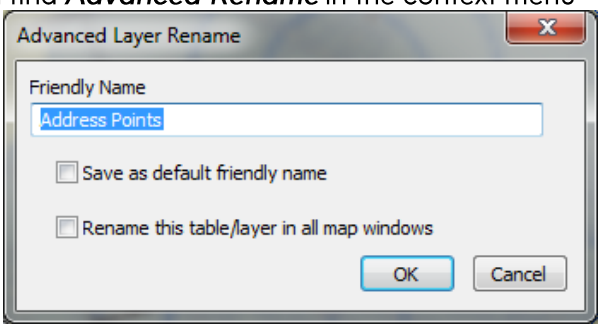


Map Windows - Layers

Advanced Rename Layer

The *Advanced Rename Layer* feature builds on the feature added with MapInfo Pro 15.0 that lets you give a layer a friendly name.

This feature extends the capability slightly. You will find *Advanced Rename* in the context menu of the layers in the *Layer List*.

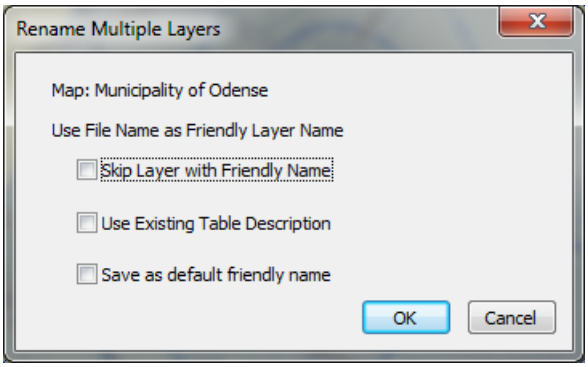


When renaming a layer of a base table, you can also save the friendly name as the default friendly name. This saves the friendly name as the table description to the metadata section of the table and MapInfo Pro can then automatically apply this name when you add the table to a map.

You can also rename the table in other map windows, if it should exist multiple times.

Rename Multiple Layers

Rename Multiple Layer is found in the context menu of the maps in the *Layer List*.



With this feature, you can easily change the friendly name of all the layers in your map.

It will rename the tables to match the file name of the tables. This means that where MapInfo Pro normally would use an underscore instead of a space, you will now see a space and so a nicer looking layer name.

You have a few options when renaming all your layers:

- *Skip Layer with Friendly Name* lets you keep the friendly name of layers that already have been given a different layer name.
- *Use Existing Table Description* will check if the table has a description stored in the metadata section. If there is, this description will be used as the friendly name.
- *Save as default friendly name* will save the friendly name used as the table description in the metadata section of the individual tab file.

Set Default View

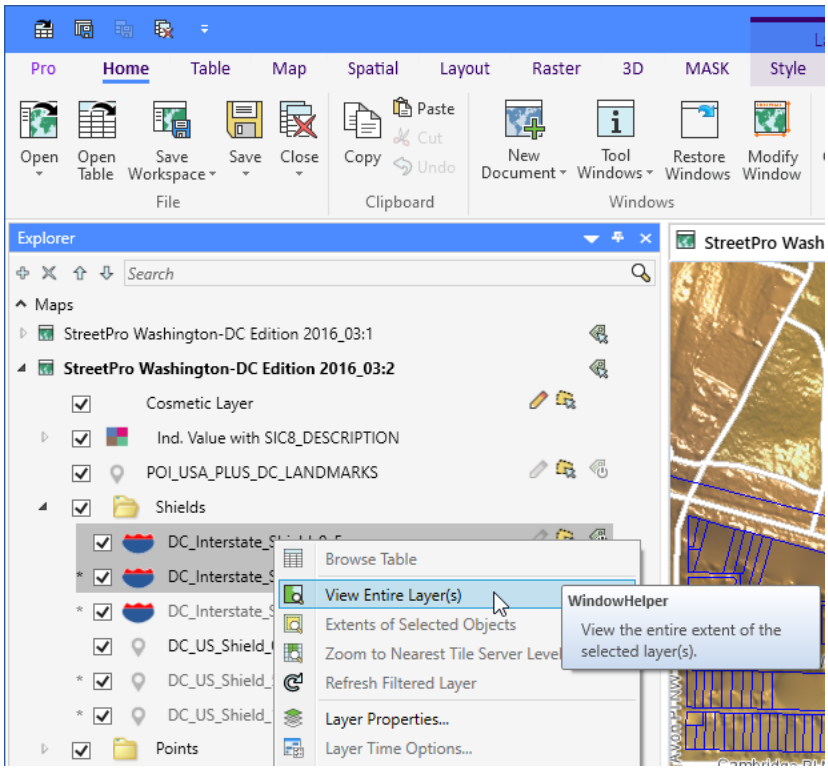
You will find this menu item in the right click menu of the maps and of the layers in the *Layer List*.

If you use the menu item from the Maps right click menu, the default view will be set on all layers of the map that support this.

If you use the menu item from the Layers, you will set the default view of one layer at a time.

The default view is the zoom and center of a table used when the table is open as the first table in a new map window.

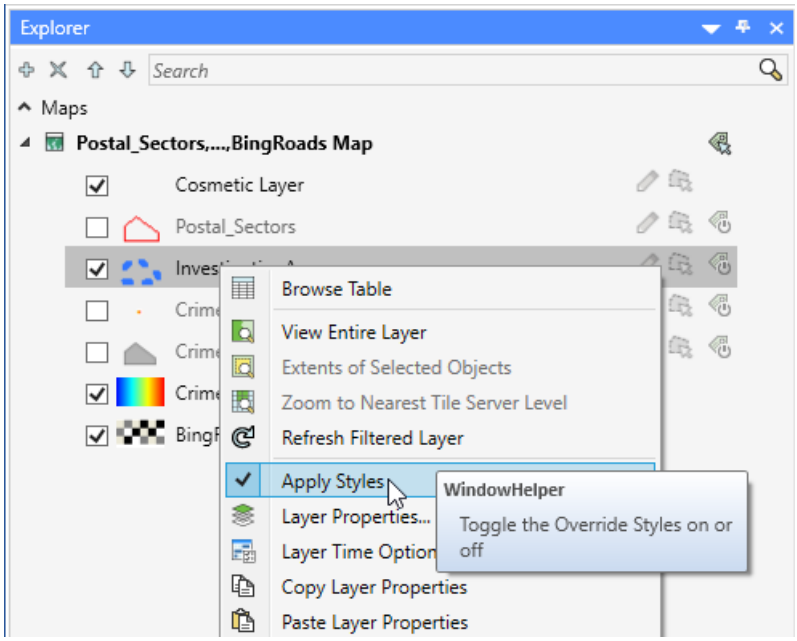
View Entire Layer(s)



You can select one, two or more layers from the context menu of the layers in the *Layer List* and use *View Entire Layer(s)* to zoom to the combined extent of these layers.

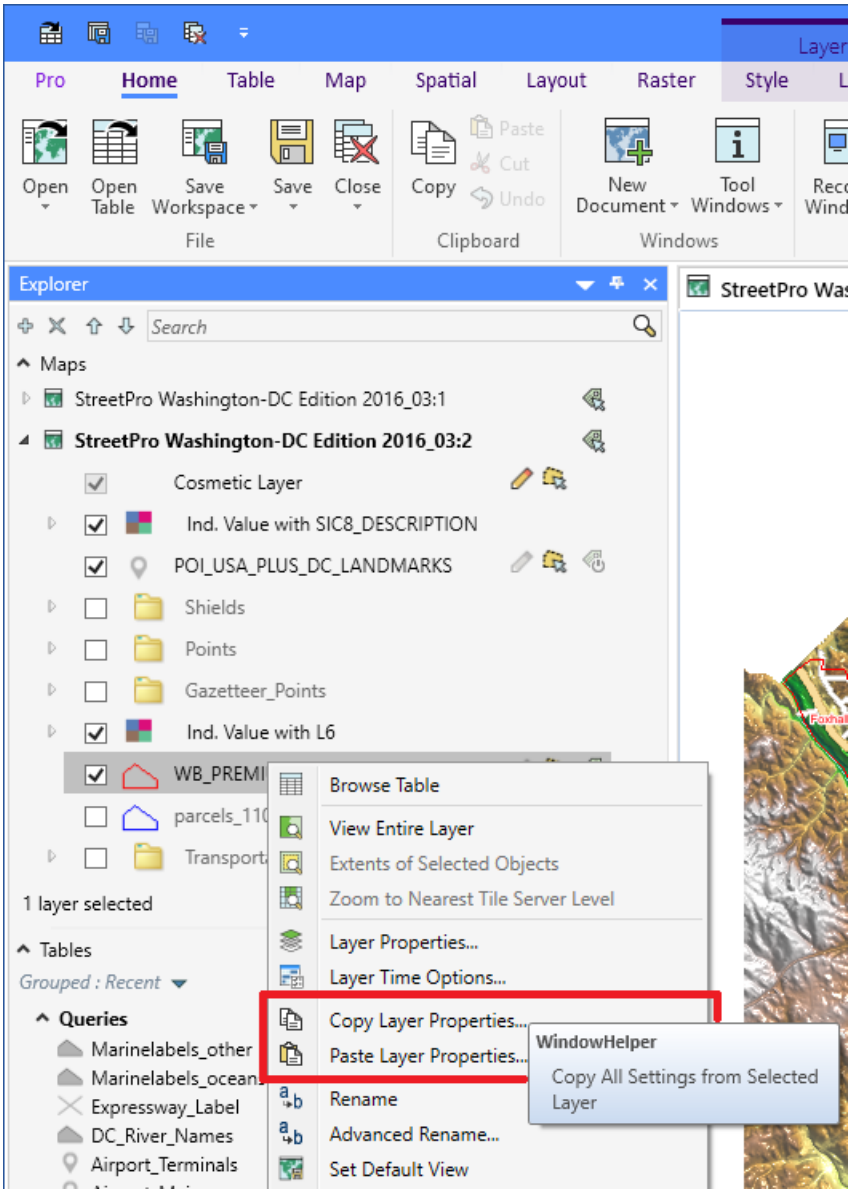
This works as an extended version of the standard *View Entire Layer* which only allowed you to zoom to the extent of a single layer.

Apply Styles



From the context menu of the Layers in the *Layer List*, you can use the *Apply Style menu item* to turn override styles on and off.

Copy and Paste Layer Properties



From the context menu of the layers, you can copy the layer properties of a selected layer. Once you have copied these, you will be able to paste these onto other layers.

This can help you reuse layer properties from one layer on one or more other layers, or on the same layer in a different map window.

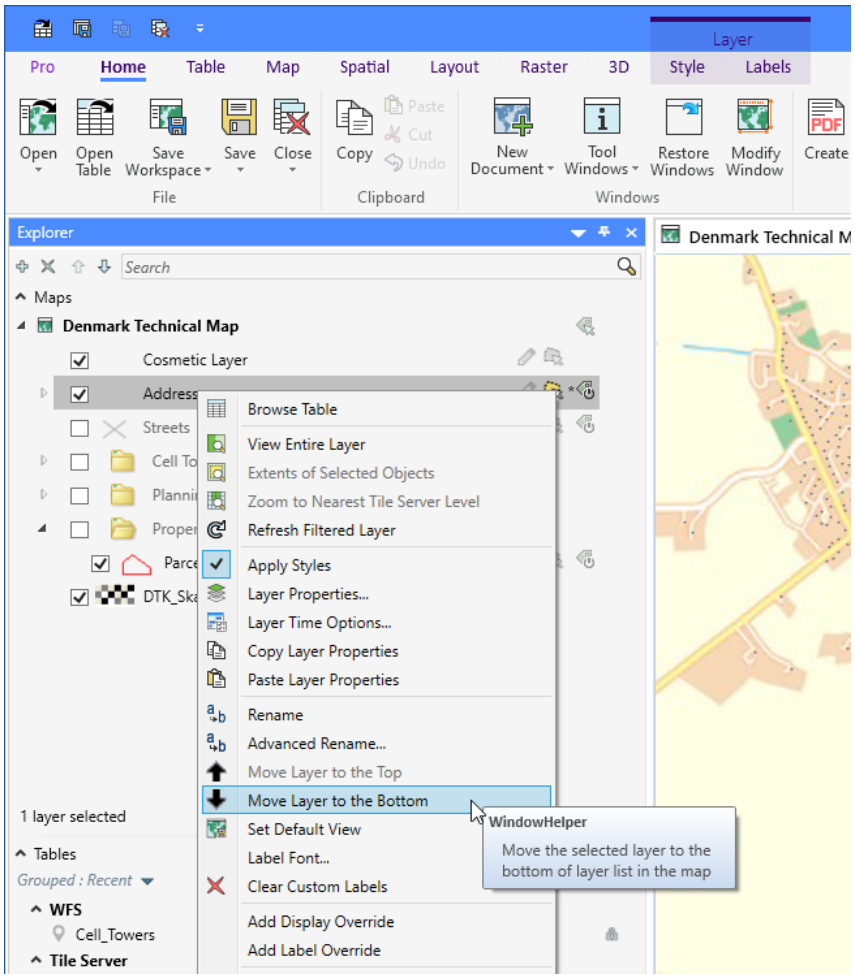
You can paste the copied properties onto most layer types, such as vector and raster layers. WindowHelper currently doesn't support copying from and pasting to cosmetic layers and thematic layers.

When you paste to a different layer type, WindowHelper will ensure that only properties supported by the layer will get pasted. You will for example not be able to paste global styles from a vector layer onto a raster layer.

For label expression, WindowHelper will try to validate that the column exists. If it doesn't, it will use the first column of the layer. If you have used a label expression, WindowHelper will try to apply this onto the layer. This may however fail if the expression is referring to columns that don't exist on that layer.

Currently only layer styles and label properties are copied and pasted. WindowHelper doesn't copy style overrides, label overrides, hotlink settings, time properties, and filters.

Move Layer to Top/Bottom



From the context menu of the layers in ***Layer List***, you can now with a single click move the selected layer to the top or bottom of the layers.

This makes it easier to move a layer from say the bottom of a map with many layers to or near the top. Simply right-click on the layer and choose the ***Move Layer to the Top control***.

Instantly, the layer will get moved to the top.

Map Windows – Thematic Layers

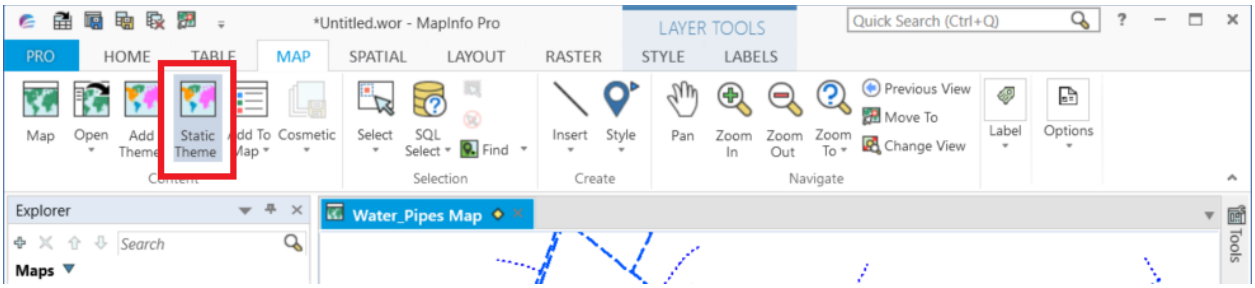
Static Theme

MapInfo Pro allows you to store the styling for a layer in different way: at an object level on the table, as a layer override and as a theme.

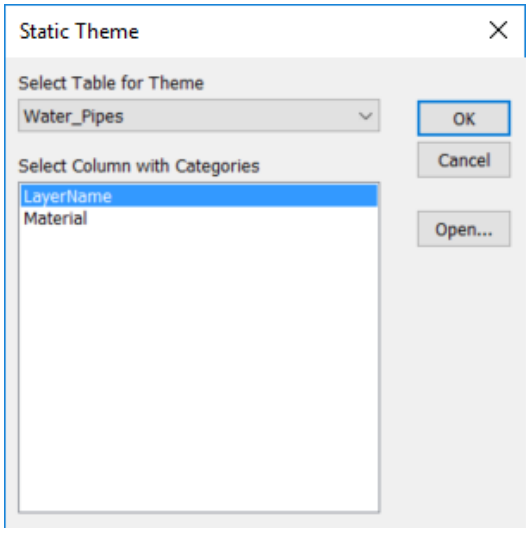
There are times where you might want to switch from the object level to a theme. This is where *Static Theme* can help you.

It lets you create a theme based on values in a selected column and the style matching one record with this value. It will of course work best if all your objects with the same value have the same styling.

You can find *Static Theme* on the *MAP tab* right next to *Add Theme*.

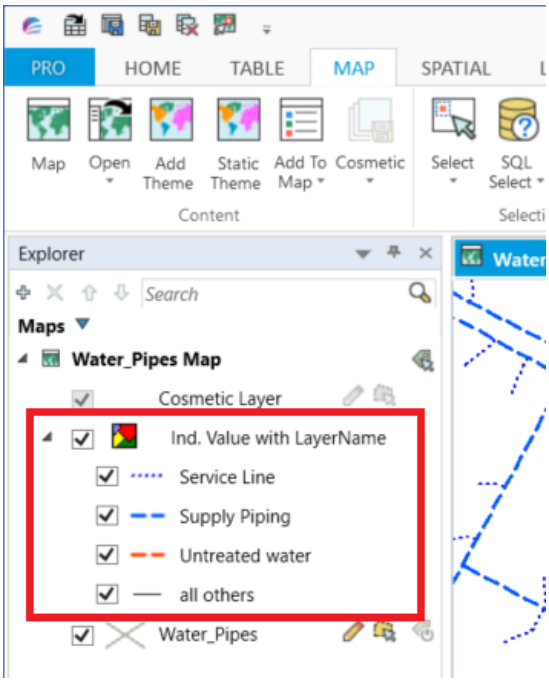


In the *Static Theme dialog*, you must select the table you want to create a Static Theme on and then the column holding the values.



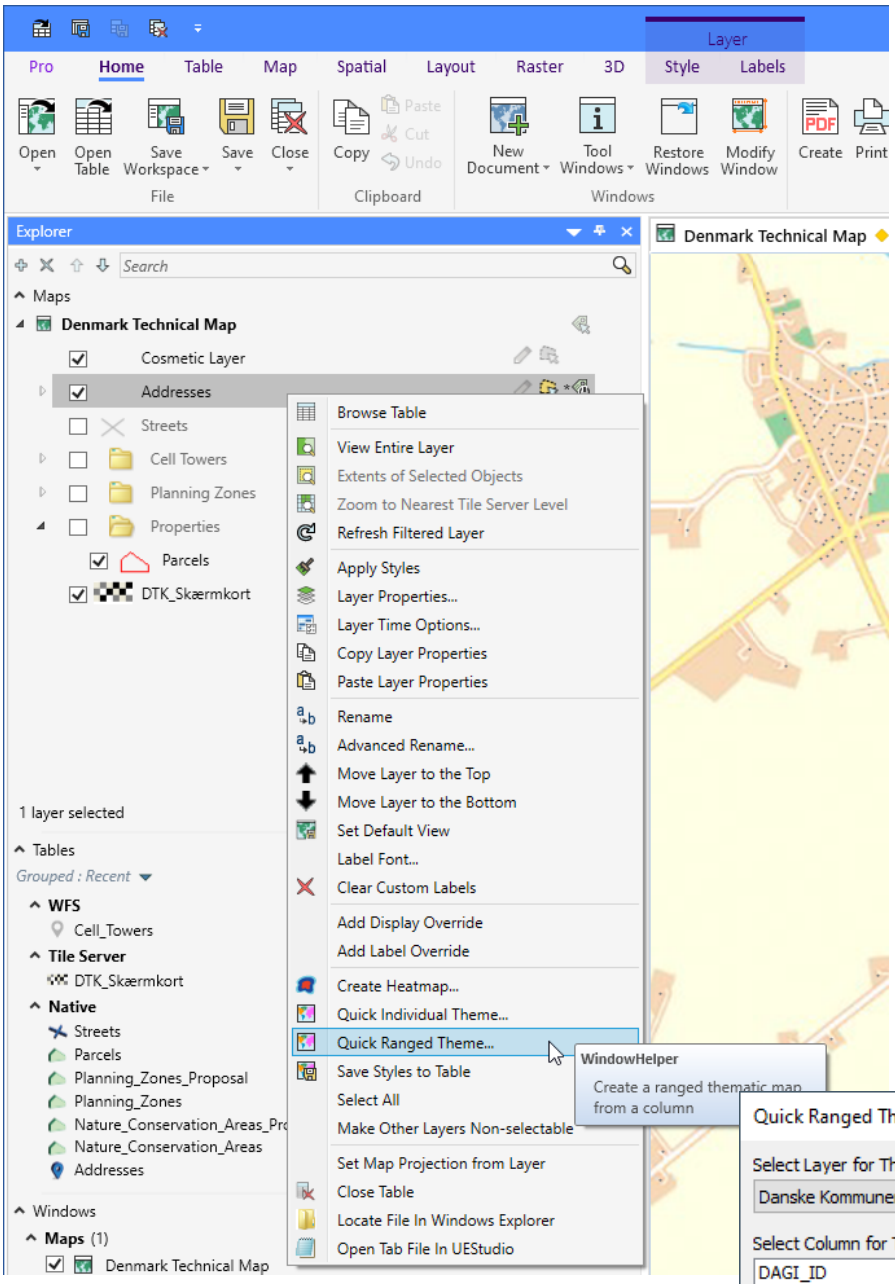
WindowHelper will now create a theme based on the values in the column selected and the style of one of the records with the specific value.

The result can be seen as a new theme in the *Layer List*.

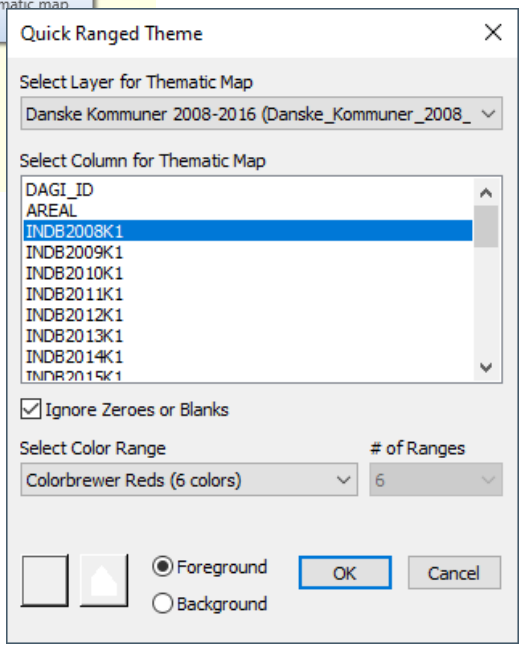


One use case for this is when you want to publish your table to Spectrum Spatial Analyst. If you have a theme on your layer, the legend in Spectrum Spatial Analyst will look a lot better than if the legend is created based on the styling form the objects in the table.

Quick Ranged Theme



From the right-click menu on a layer, you can access *Quick Ranged Theme...* which allows you to create a ranged thematic in a few clicks.



From the *Quick Ranged Theme dialog*, you can select the layer and the column to use. Notice that the layer has been preselected to be the layer you right clicked on in the *Layer List*.

Check the Ignore Zeroes and Blanks if you don't want to use these in your theme.

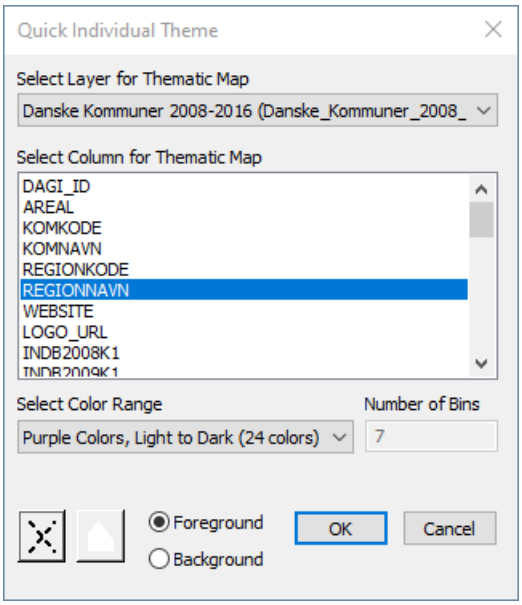
From the *Select Color Range list*, you can select one of multiple color ranges to use, and *# of Ranges* can allow you to set the number of ranges in your thematic map.

The color ranges are read from a configuration file, QuickThematic.ini, stored in the folder where WindowHelper is installed. Some ranges just define a start and an end color. For these you can specify the number of ranges. Others define a fixed

number of colors. For these you can't define the ranges. But remember you can afterward quickly modify your thematic map using the ***Theme tab***.

Under the ***Select Color Range list***, you can configure the default style for the objects. For area-based objects, you can also set the style for the border pen and you can specify if the color should be applied as the foreground or background color.

Quick Individual Theme



From the right-click menu on a layer, you can now access *Quick Individual Theme...* which allows you to create an individual thematic in a few clicks.

From the *Quick Individual Theme* dialog, you can select the layer and the column to use. Notice that the layer has been preselected to be the layer you right clicked on in the *Layer List*.

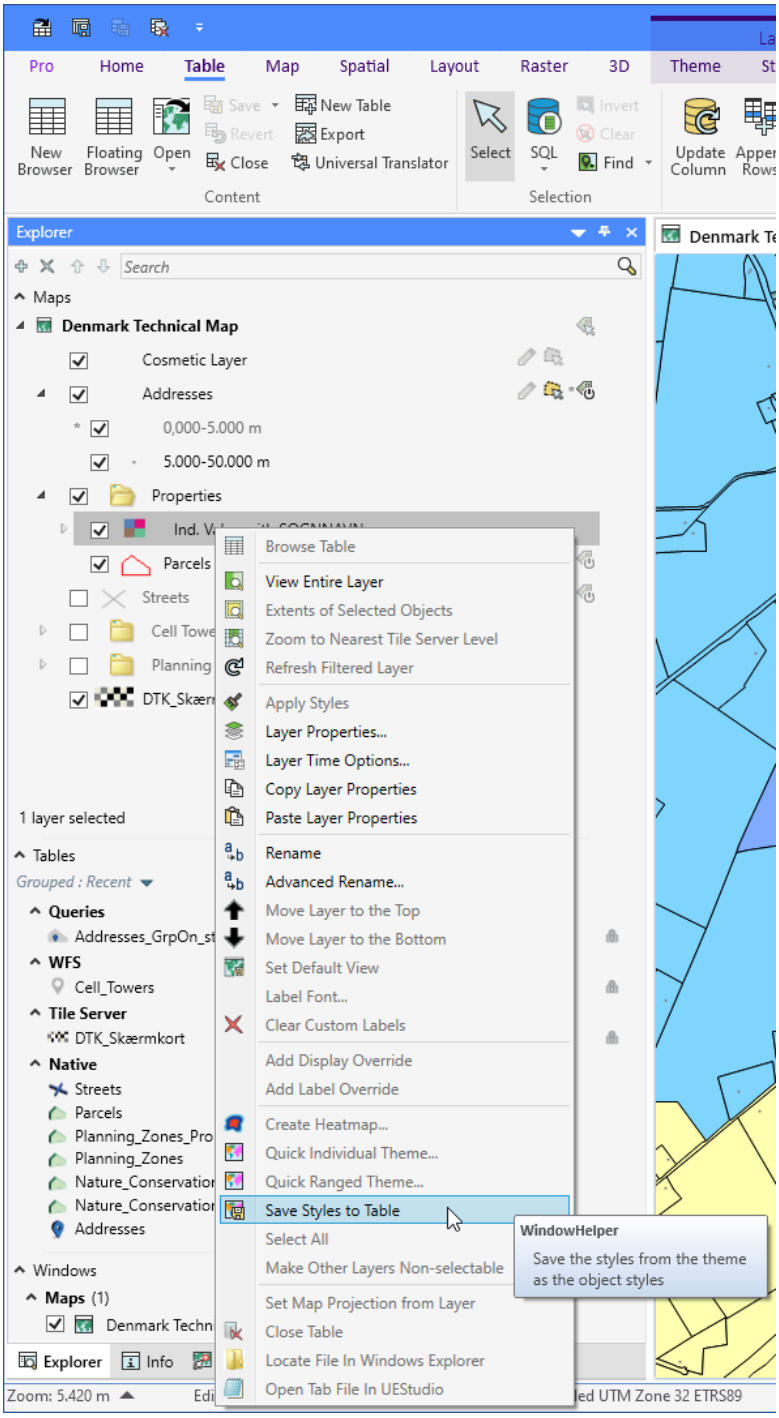
From the *Select Color Range* list, you can select one of multiple color ranges to use, and *# of Bins* tells you how many different values the selected column holds. This will equal to the number of bins in your individual thematic map.

If there are more bins than colors, each color might appear multiple times.

The color ranges are read from a configuration file, QuickThematic.ini, stored in the folder where WindowHelper is installed. Some ranges just defined a start and an end color. For these MapInfo Pro will calculate a total of 24 colors spread between the input colors. Others define a fixed number of colors. But remember you can now quickly modify your thematic map using the *Theme* tab once it has been created.

Under the ***Select Color Range list***, you can configure the default style for the objects. For area-based objects, you can also set the style for the border pen and you can specify if the color should be applied as the foreground or background color.

Save Theme to Object Styles



From the context menu of a layer in the *Layer List*, you can save the styles from an individual thematic layer onto the base layer. This allows you to apply the styles from the theme to the object as their static style.

It works by right-clicking on the thematic layer from where you want to apply the styles.

This must be an individual thematic layer. That's currently the only theme type supported.

Now the tool will run through the theme bins, use the value for each bin to apply the color/style from this bin onto the object in the table.

Once done, remember to save the changes to the table as the table isn't saved automatically by the tool.

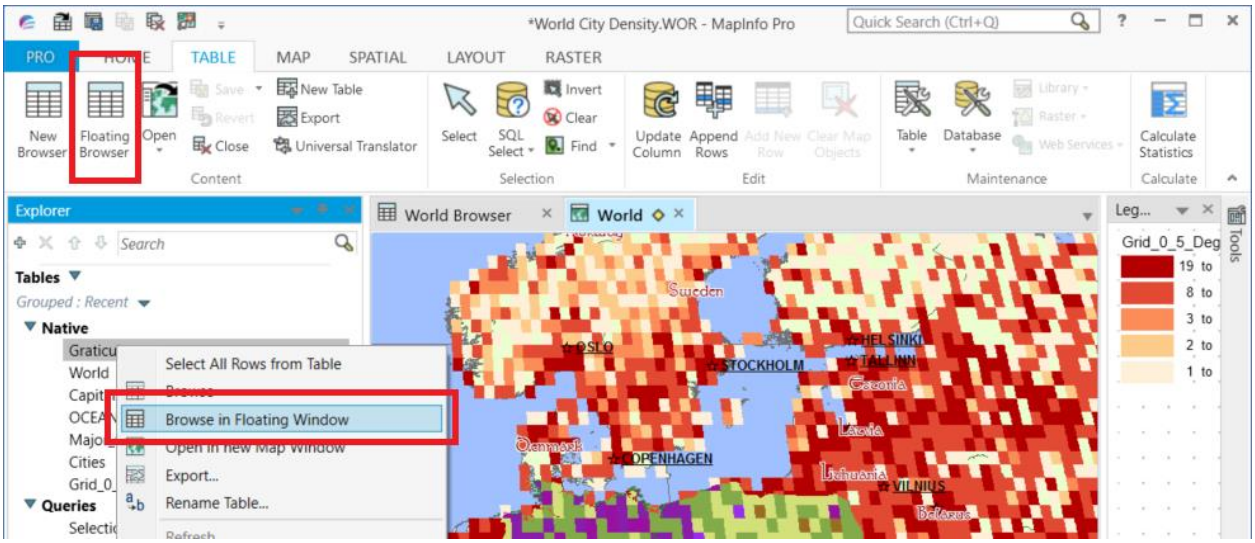
Browser Window

Browse in Floating Window

The *Browse in floating Window* lets you browse a table in a floating browser window instead of a tabbed browser window which is the default behavior.

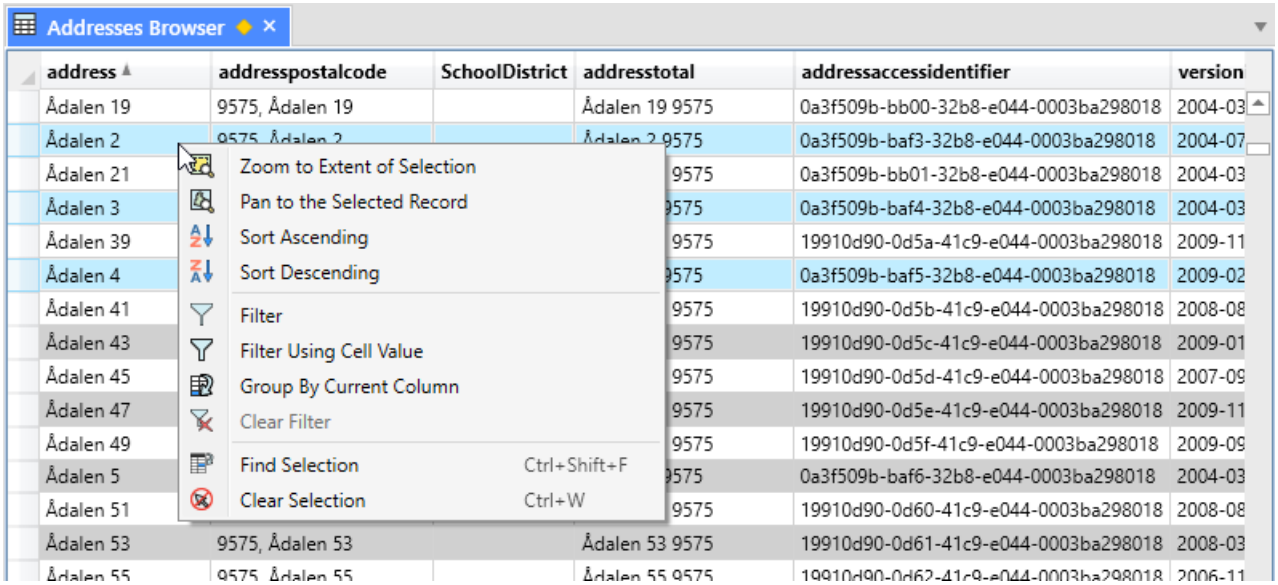
You can find this option on the context menu of the tables in the ***Table List*** and on the ***TABLE tab*** next to ***New Browser***.

You can also use the shortcut *Ctrl+Shift+F4*.



Pan and Zoom to Selection

In the context menu of the Browser window, you can find two options that help you move your map to the currently selected record(s).



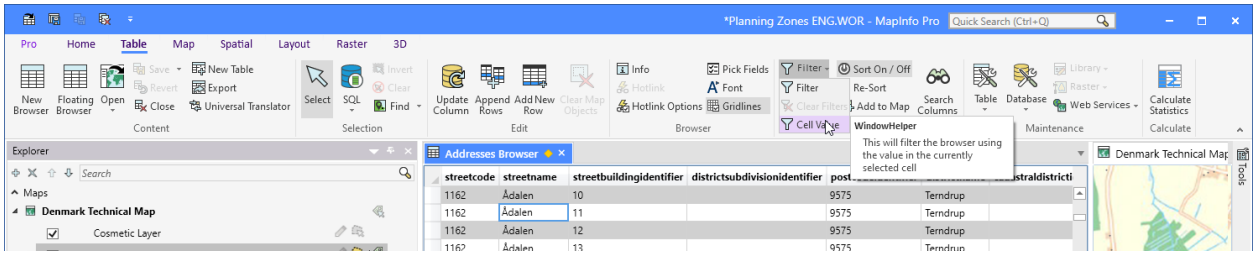
Zoom to Extent of Selecting pans and zooms to the extent of the currently selected records in your Map window. This feature supports a selection with more than one record. If you have selected just one point, the map will only pan, not zoom.

Pan to the Selected Record pans your Map window to the centroid of the selected record. The map will not be zoomed. Also note that this feature only support one selected record.

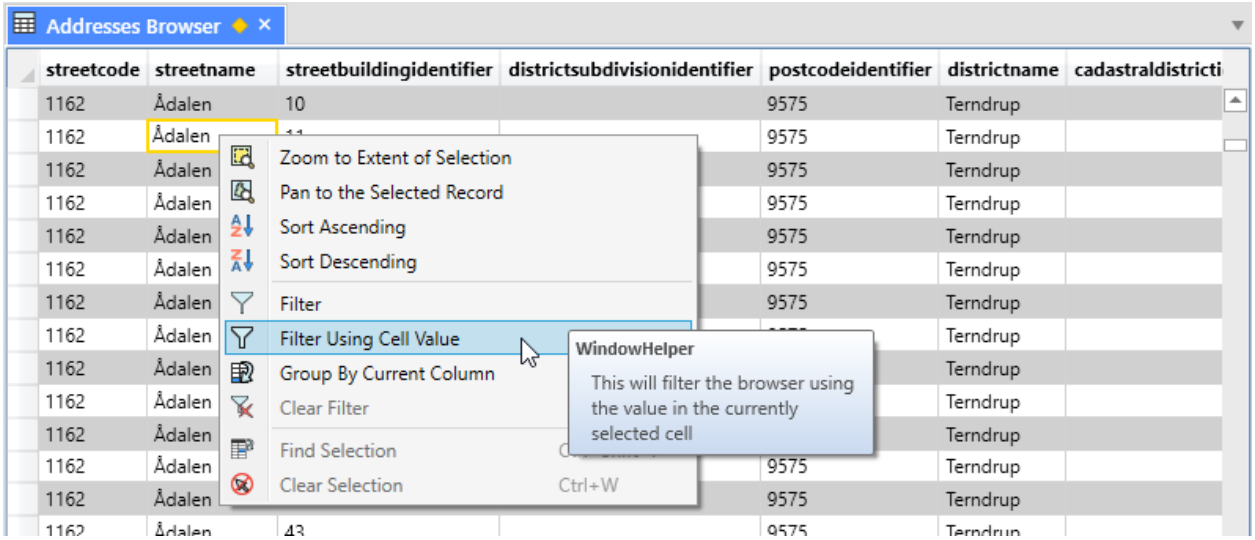
Filter By Cell Value

This feature can be found in two places:

- And in the *Filter* menu in the *Sort and Filter* group on the *TABLE* tab.



- In the context menu of the browser window



The feature gives you are very easy way to filter your browser window using a value in a cell.

Click in the cell with the value that you want to filter the table using. Be aware that the tool by default ignores the top left cell, so avoid using this cell.

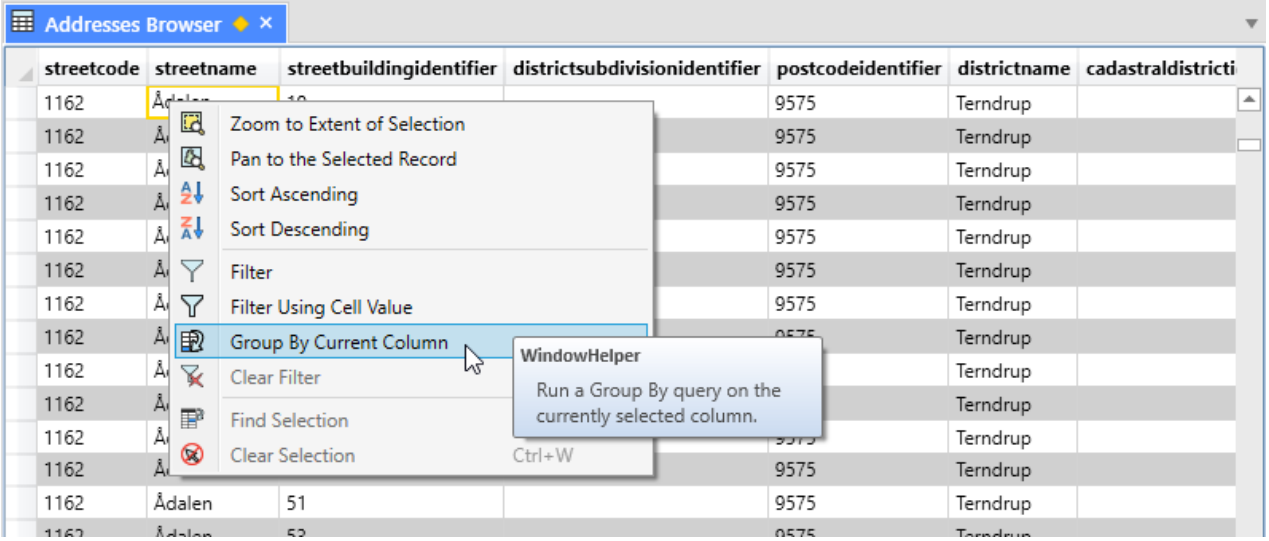
Now click on *Filter By Cell Value* in one of the two places.

The browser window will now only show records matching the value you highlighted.

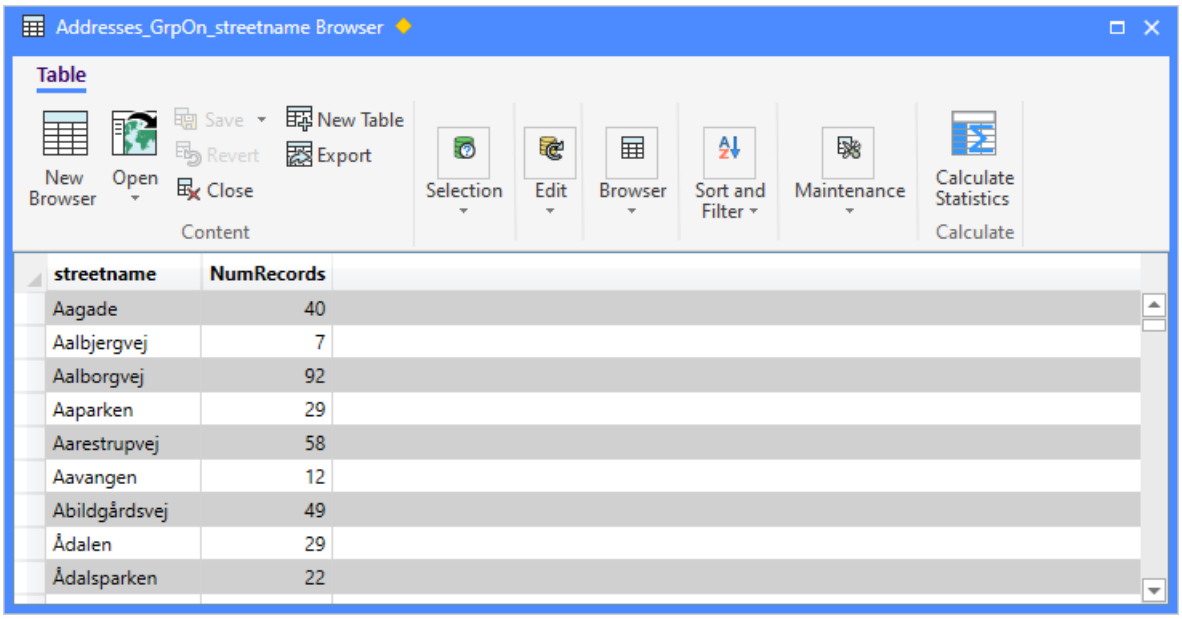
Group By Current Column

This feature can be found in the context menu of the browser window.

Click in a cell in a Browser window and then right-click to access *Group By Current Column*.



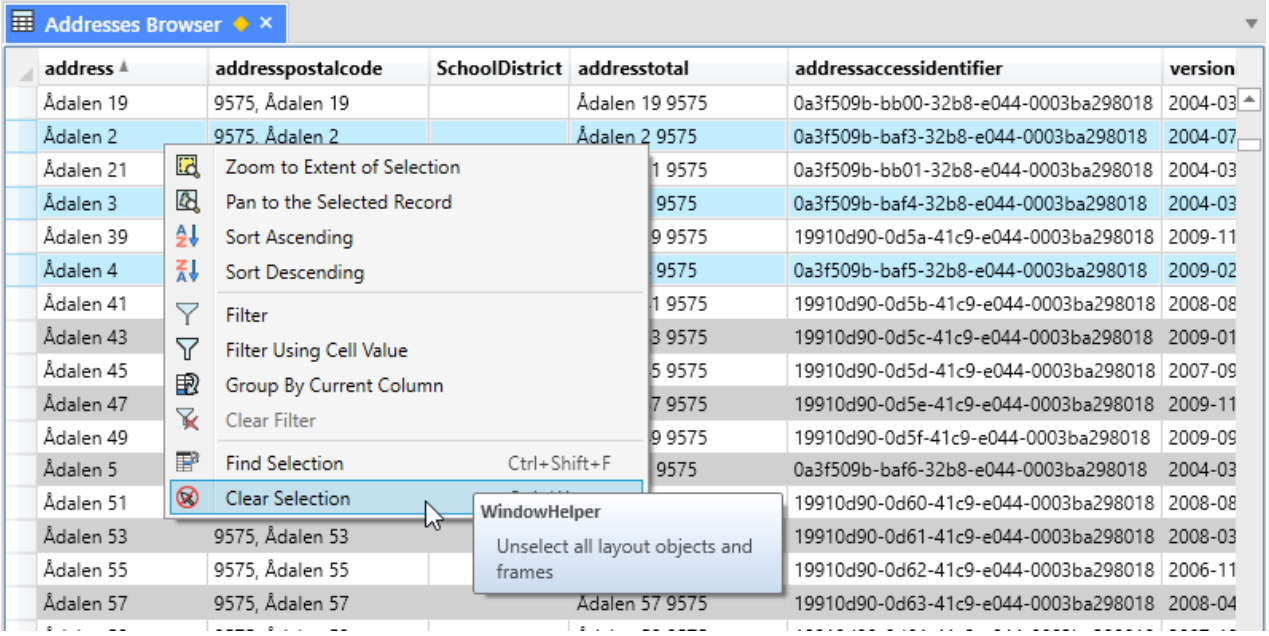
As a result, you will get a floating browser window where the values from the selected column have been grouped and sorted. Next to the values, you can see a count telling you have many times the values appear.



Clear Selection

At the bottom of the context menu in the Browser, you can now find Clear Selection which allows you to remove all records from the current selection.

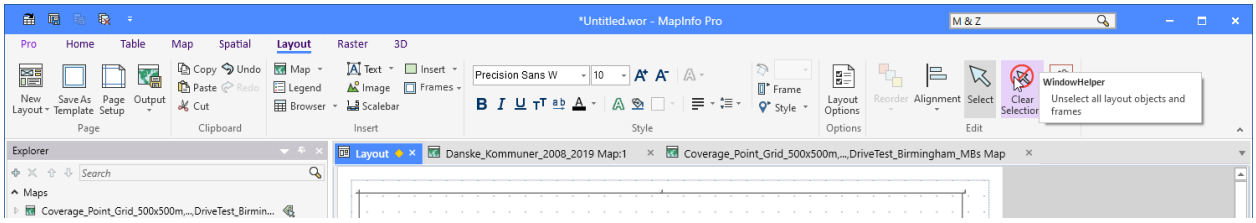
This allows you to clear the selection without having to go to one of the tabs in the ribbon.


Addresses Browser
address
addresspostalcode
SchoolDistrict
addresstotal
addressaccessidentifier
version
Zoom to Extent of Selection
Pan to the Selected Record
Sort Ascending
Sort Descending
Filter
Filter Using Cell Value
Group By Current Column
Clear Filter
Find Selection
Ctrl+Shift+F
Clear Selection
WindowHelper
Unselect all layout objects and frames

Layout Window

Clear Selection in Layout window

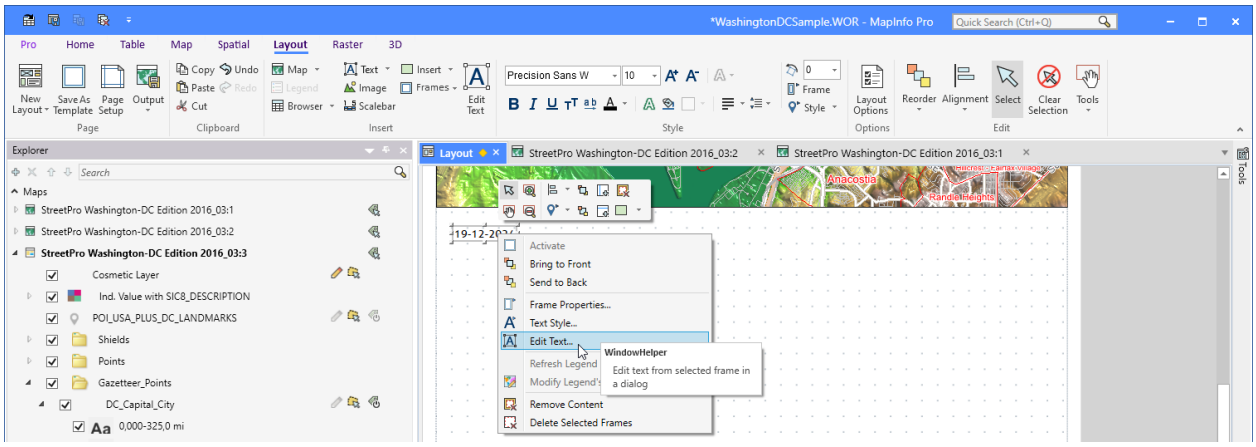
On the *Layout tab*, you can now find *Clear Selection* which will unselect any selected frames or objects in the active layout window.



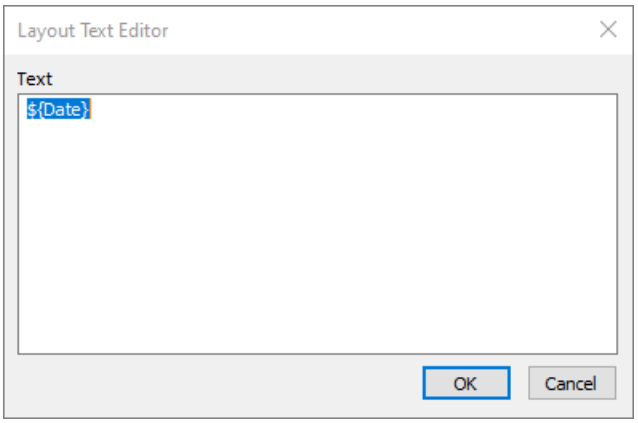
Edit Text in Dialog

You can select a text frame in the layout window and use edit the text in a dialog using the *Edit Text control*.

You will find the *Edit Text control* on the Layout tab in the Insert group and in the context/right-click menu of the Layout window.



The text will be shown in the *Layout Text Editor dialog* allowing you to modify the text and save it back to the text frame on the layout.



Function List

MapInfo Pro 17.0 has a new function that lets you call/use custom subprocedures and functions in a running MapBasic application from multiple places in MapInfo Pro. You can sue these functions from a label expression, in a SmartText in a Layout, in a SQL Select statement, in an Update Column statement and in multiple other places.

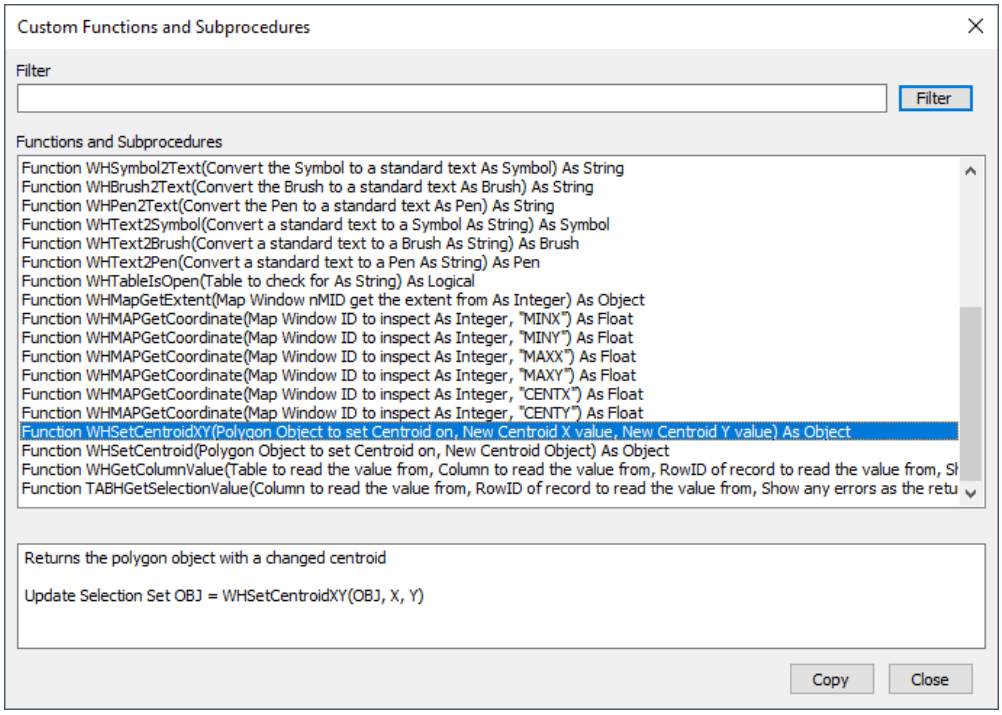
The problem often is to know which functions a MapBasic application provides you access to.

WindowHelper comes with a dialog that lets you view some of the available functions in the tool. You can then quick copy an expression using each function and use these in MapInfo Pro.

You access this list of functions from the context menu of the WindowHelper application in the Tools window, see image under Options further down in this document.

Use the *Filter* option to search for specific functions and click on a function in the list to see a description of the function and an example expression.

Use the *Copy* button or double-click on a function in the list to copy the sample expression to the clipboard.



Published Custom MapBasic Functions

WindowHelper takes advantage of the new option to publish custom MapBasic functions to the MapInfo Pro interface. Do note that these functions have also been published to the MapInfo Pro interface which makes it even easier to use them. You can see a complete list of functions in the **What's New chapter**.

These functions will appear in the Functions dropdown list in SQL Select, in Update Column and in other Expressions dialog thru out the product.

You can also use these functions from the MapBasic and in workspaces if you are using these to perform automated tasks. Just make sure WindowHelper is running when you want to use the functions.

Currently these functions have been published:

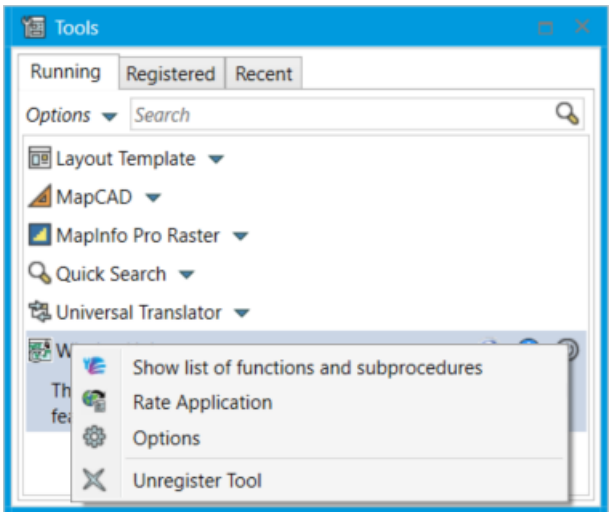
- WHSetCoordsysUsingTable(*tableName*)
  - *tableName* is the name of an open table whose coordsys you want WindowHelper to use. This is useful when using some of the spatial functions of WindowHelper such as MAPGetExtent, OBJCartesianCreatePointAlong and more
- WHSetCoordsysUsingMap(*windowID*)
  - *windowID* is the ID of the map whose coordsys you want WindowHelper to use. This is useful when using some of the spatial functions of WindowHelper such as MAPGetExtent, OBJCartesianCreatePointAlong and more
- **WHGetSelectionValue**(recordNo As Integer, columnName As String)
  - *recordNo* is the record that you want to read the value from
  - *columnName* is the name of the column to read the value from
  - Returns the value from the specified column of the current selection
- **WHGetColumnValue**(tableName, columnName, recordNo, showError)
  - *tableName* is the table to read the value from
  - *columnName* is the column to read the value from
  - *recordNo* is the record number to read the value from
  - *showError* specifies if the function should return the error or an empty string when a value can't be read. Set it to 1 to understand why it doesn't return a string in certain cases.
  - Returns the value from the specified column of the specified table
  - Can be useful as a SmartText to read values from a table and add these to the layout
- **WHReplace$**(inputString, searchFor, replaceWith) As String
  - *inputString* is the string, or column name, you want to find a given string in
  - *searchFor* is the string you are searching for
  - *replaceWith* is the string you want to use instead of the *searchFor* string.
  - Returns the *inputString* where the occurences of *searchFor* have been replaced with *replaceWith*.
- **WHGetValueBetweenChars$**(*inputString*, *startChar*, *endChar*) As String
  - *inputString* is the string, or column name, you want to search
  - *startChar* is the first character you are looking for, for example "{"
  - *endChar* is the end character you are looking for, for example "}"
  - Returns the string between the *startChar* and *endChar* if found, otherwise "".
- **WHCombineWithSeparator$**(*stringFirst*, *stringLast*, *stringSeparator*) As String
  - *stringFirst* and *StringLast* are the two strings you want to combine.
  - *stringSeparator* is the string to put between the *stringFirst* and *stringLast*.
  - Returns a string where the two strings have been combined. If none of them are empty string, stringSeparator is added in between. If one of them are empty, the other string is returned and the *stringSeparator* is not used.
- **WHLeftWord$**(*inputString*, *wordNumber*) As String
  - *inputString* is the string, or column name, you want to extract a word from

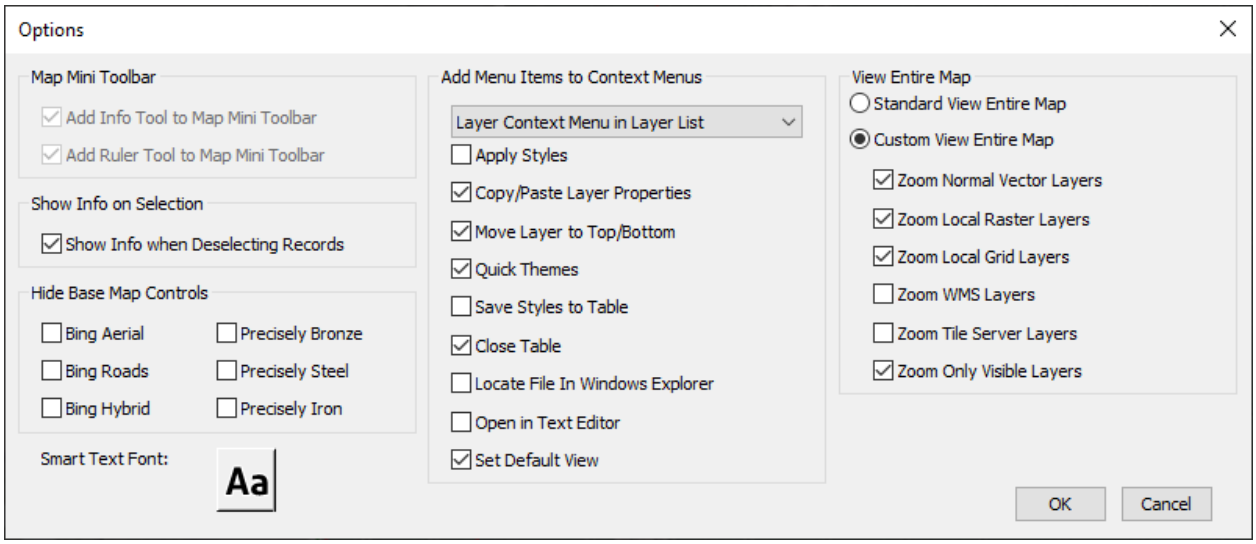
  - *wordNumber* is the word number you want to extract. If you specify -1, it will return the last word
  - Returns the *wordNumber* from the list of in the *inputString* when it is separated by spaces.
- **WHCoulmnExists**(*tableName*, *columnName*) As Logical
  - *tableName* is the table to look for the column
  - *columnName* is the column to look for
  - returns true (1) if the specified column is found in the specified table, otherwise it will return false (0).
- **WHColumnTypeAsString**(*tableName*, *columnName*) As String
  - *tableName* is the table to look for the column
  - *columnName* is the column to look for
  - returns a string representing the column type, including width. If the table isn't open or the column doesn't exists, an empty string is returned.
- **WHTableIsOpen**(*tableName*) As String
  - *tableName* is the table to look for
  - returns true (1) if the table is open, otherwise it returns false (0)
- **WHConfigReadKey**(*fileName*, *sectionName*, *keyName*) As String
  - *Filename* is the config file to read a key value from. Use full path
  - *sectionName* is the section to find the key under. Do not include [] around the section name.
  - *keyName* is the key to get the value from.
  - Returns the value assigned to the specified key under the section in the config file.
- **WHConfigWriteKey**(fileName, sectionName, keyName, stringValue)
  - *Filename* is the config file to read a key value from. Use full path
  - *sectionName* is the section to find the key under. Do not include [] around the section name.
  - *keyName* is the key to get the value from.
  - *stringValue* is the value to assign the specified key.
- **WHAlterPen**(*object*, *penNew*), **STLAlterBrush**(*object*, *brushNew*), **WHAlterSymbol**(*object*, *symbolNew*)
  - *object* is the object to change. Can also be a spatial column, like OBJ
  - *penNew*, *brushNew* or *symbolNew* is the new style to apply on the object
  - returns the object with the new styling applied
- WHSymbol2Text(*symbol*), STLBrush2Text(*brush*), STLPen2Text(*pen*)
  - *symbol*, *brush*, *pen* is the styles you want to have converted into a string representation, like "1,2,255" for a full pattern blue line with a width of 1.
  - returns a string that represents the style.
- **WHText2Symbol**(*stringSymbol*), **STLText2Brush**(*stringBrush*), **WHText2Pen**(*stringPen*)
  - *stringSymbol*, *stringBrush*, *stringPen* is the string representing the style you want to have converted into a true style.
  - returns the style, pen, symbol or brush, represented by the string entered.
- **WHChangeSymbolColour**(*symbol*, *newColorValue*)
  - *symbol* is the symbol that should have a new color assigned
  - *newColorValue* is the new color to assign to the symbol
- WHObjectReverse(*object*)
  - *object*: The line or Polyline you want to reverse the node order of
  - returns the input object with the nodes order reversed.

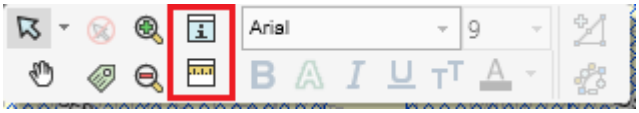
- WHCartesianCreatePointAlong(*object*, *distance*, *offset*)
  - *object* is the line/polyline (single segment) you want to create a point along
  - *distance*, is the distance along the line. A negative distance creates the point before the first node.
  - *offset* is the distance perpendicular to the line you want to offset the point. Negative is left, positive is right
  - returns the points along the input line. Set the coordinate system using WINHSetCoordsysUsingTable using a projected table.
- **WHDebugEnable**(*enableDebug*)
  - *enableDebug* is a logical value that turns debug on or off
- WHMapGetExtent(*windowID*)
  - *windowID* is the ID of the Map window to get the extent of. Can also be FrontWindow() for the current active map window.
  - *returns a spatial object representing the extent of the map window.* Set the coordinate system using WINHSetCoordsysUsingTable using a table.
- **WHMapGetCoordinate**(*windowID*, *coordinateType*)
  - *windowID* is the ID of the map you want to get the coordinate from.
  - coordinateType is the coordinate you want to retrieve. The value can be one of these: MINX, MAXX, MINY, MAXY, CENTX, CENTY
  - returns the coordinate specified from the map. The coordinate will be returned using the current coordinate system of the map.
- **WHRegExReplace**(inputString, patternString, replaceString)
  - *inputstring* is the input string to search using Regular Expression.
  - *patternString* is the pattern to look for in the input string
  - *replaceString* is the text to replace the pattern with in the input string
  - returns *inputString* where the *patternString* has been replaced with the *replaceString* if match was found.
- **WHRegExIsMatch**(inputString, patternString)
  - *inputString* is the input string to search using Regular expression
  - *patternString* is the pattern to search for in the input string
  - returns true if the pattern was found, otherwise false
- **WHRegExFirstMatch**(inputString, patternString)
  - *inputString* is the input string to search using Regular expression
  - *patternString* is the pattern to search for in the input string
  - returns the first substring from inputString that matches the pattern
  - read more about the use of Regular Expressions here: https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference
- WHSetCentroid(polygonobject, pointobject)
  - *polygonobject* is the polygon/region object getting the centroid modified
  - *pointobject* is the new centroid point
  - returns the *polygonobject* with a changed centroid
  - Consider using **WHSetCoordsysUsingTable** to set the coordsys internally in WindowHelper before using this function
- WHSetCentroidXY(*object*, *newX*, *newY*)
  - *polygonobject* is the polygon/region object getting the centroid modified
  - *newx* is the new x coordinate of the centroid
  - *newy* is the new y coordinate of the centroid
  - returns the *polygonobject* with a changed centroid
  - Make sure the current coordsys matches the coordsys of the coordinates given

Options

If you right click on the WindowHelper tool in the *Tools* window, you can access the options for the tool.







The options let you specify if you want to add the *Info Tool* and the *Ruler Tool* to the *Map Mini Toolbar*. These have been disabled if you are using MapInfo Pro 17.0 or newer as you can modify the *Map Mini Toolbar* via the *Customize Ribbon* dialog.

We have in certain cases seen issues when the tool shows information about the deselected features. If you start to see misbehaviour, you can turn of the showing of information when deselecting records.

You can also control the visibility of the Base Maps controls that appear on the *Open* dropdowns on the *HOME*, *TABLE* and *MAP* tabs. Check those base map controls that you want to hide from the dropdowns.

And you can set the default font for the Smart Text options that WindowHelper add to the ***Add to Map*** dropdown.

In the Add Menu Items to Context Menus section, we allow you to control which menu items you want to add to the context menus. We have made this adjustment to avoid adding menu items you never use.

You can control this for the Layer context menu in the ***Layer List***, the Table context menu in the ***Table List***, the context menu in the Browser window, and the context menu in the Mapper window. The number of menu items differs between the context menus.

You can also configure how you want the View Entire Map to work. It can work as it normally does, or you can configure which specific layer types you want to zoom to the extent of. This allows you for example to only zoom to the extent of normal vector tables that have the visibility property checked.