

# WindowHelper 3.6

## Requirements

This version of WindowHelper takes advantages of some new features of MapInfo Pro 17.0.1. This means that you **need MapInfo Pro 17.0.1** or newer to be able to use WindowHelper 3.6.

## About WindowHelper

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

WindowHelper is designed for the new ribbon based interface of MapInfo Pro 64 bit.

WindowHelper 3.6 is consider a new tool even though it has been built on top of the earlier WindowHelper 2.1. But moving the tool to the 64-bit has made big parts of the tool obsolete as these features now are supported directly in the MapInfo Pro interface.

To get the most out of this tool it is recommended that you add this tool to the Tool Manager and sets it to Autoload.

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

If you have any requests or feedback on the WindowHelper, please send your feedback thru the Ideas Community: <http://ideas.pb.com/>

Add your feedback to the MapInfo Pro product and mark the subject with "WindowHelper tool".

## Install WindowHelper

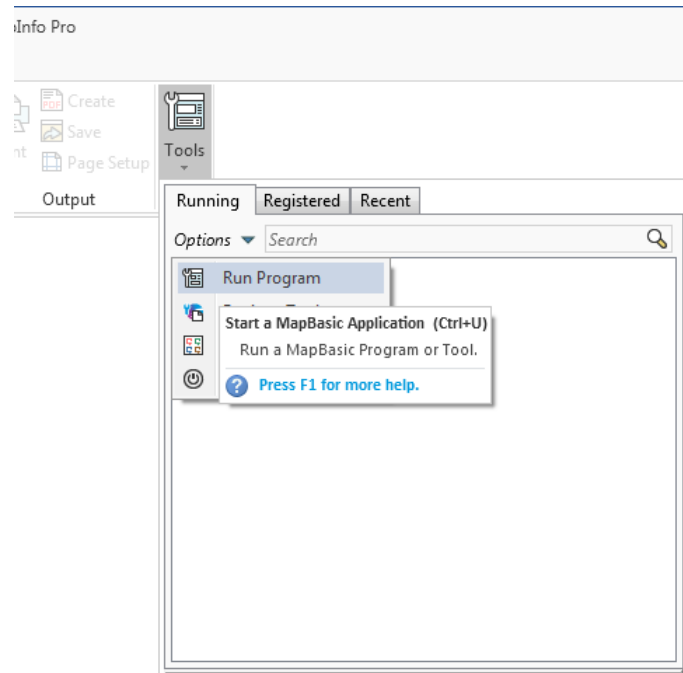
You install WindowHelper by downloading the WindowHelper.zip file. Extract the files into a new folder or into the folder with your other MapInfo tools.

It is not recommended to put this tool on a network drive as it is using a number of .NET resources that not easily can be run from a network drive.

So make sure you put the tool on your local hard drive.

## Add tool to Tool Manager

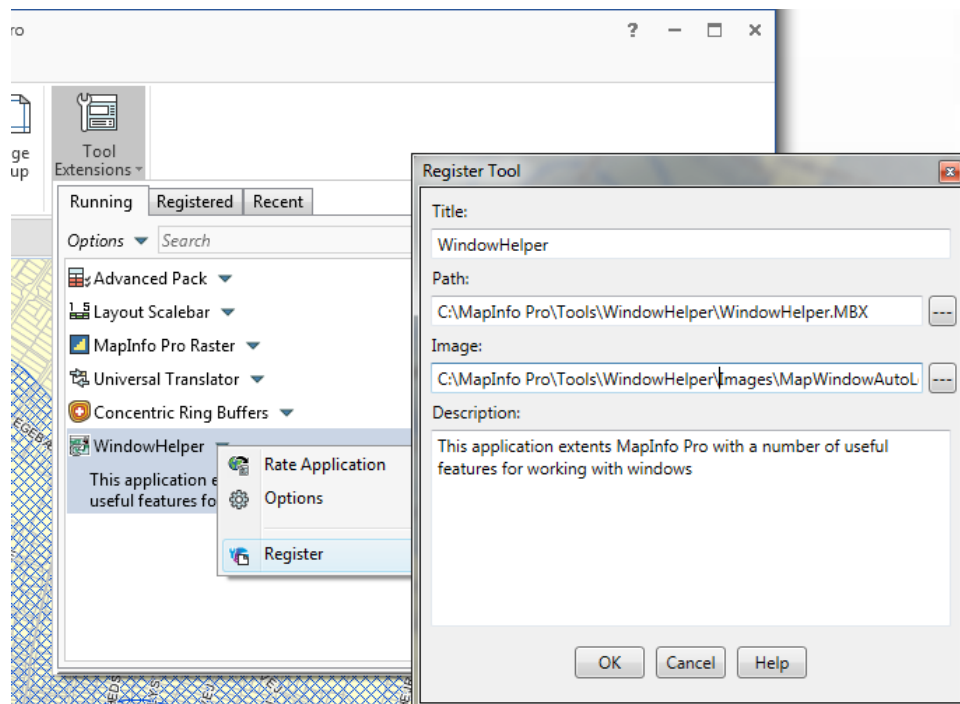
Go to Home > *Tools* and click on the *Run Program* in the *Options* menu.



Now navigate to the location where you unzipped WindowHelper and select the file WindowHelper.mbx.

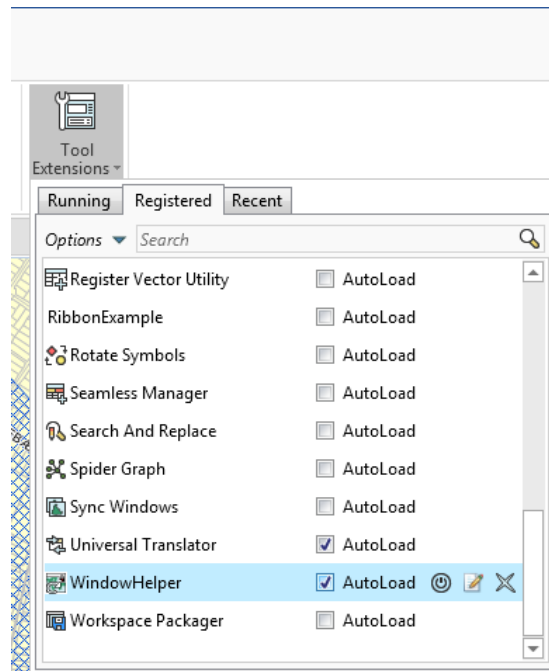
Once the tool has been loaded you can go back to the Tools window and click on the context menu of WindowHelper in the list of *Running* tools.

Then click on *Register* to add WindowHelper to the list of registered tools.



Notice that the values have been prefilled for you. Click *OK* to add WindowHelper to the list of tools.

Go back to the *Tools* window select the *Registered* tab to see the registered tools. Here you can check the *Autoload* checkbox for this tool to load it whenever MapInfo Pro starts up.



## New features in WindowHelper 3.6

**Published functions to the MapInfo Pro interface:** version 3.6 takes advantage of the new option to publish custom MapBasic functions to the MapInfo Pro interface. Currently these functions have been published:

- WINHSetCoordsysUsingTable
- STRINGReplace
- STRINGGetValueBetweenChars
- STRINGConcatenateWithSeparator
- STRINGLeftWord
- COLExists
- COLTypeAsString
- TABIsOpen
- CONFIGReadKey, CONFIGWriteKey
- STLAlterPen, STLAlterBrush, STLAlterSymbol
- STLSymbol2Text, STLBrush2Text, STLPen2Text
- STLText2Symbol, STLText2Brush, STLText2Pen
- OBJDirection
- OBJReverse
- OBJCartesianCreatePointAlong
- OBJCombineFromTable
- MAPGetExtent
- MAPHGetCoordinate
- TABHRegExReplace, TABHRegExIsMatch, TABHRegExFirstMatch

These functions will appear in the *Functions* dropdown in the *SQL Select*, *Label Expression*, *Thematic Expression* and in a number of other dialogs showing the *Functions* dropdown.

**Modify Window dialog:** The Modify Window dialog has been brought back on request.

## New features in WindowHelper 3.5

**Number of selected records** is now shown in the statusbar. This gives you a quick overview over how many records you have selected instead of having to browse the selection.

**Browse in Floating window** is a new feature that lets you browse a table directly into a floating window. You can access this via the TABLE tab, via Ctrl+Shift+F4 and via the context menu of the Table List.

**Static Theme** lets you create a thematic that uses the values from a selected column in combination with the styles from the spatial object in the table.

**New shortcuts** have been added for cloning a map window via the Map Context menu of the Layer List

**The state of the InfoSelHandler** is now stored and restored correctly between sessions.

**Dialog with functions** that can used with the new Exec() function in MapInfo Pro 17.0 such as getting coordinates from a map window to be used in a SmartText.

## New features in WindowHelper 3.0

**Advanced Rename Layer** is one new feature in WindowHelper. This features lets you rename a layer or cosmetic layer in a map window. You can also save this new Friendly name as the default friendly layer name for the table so that the table always will use it. And you can ask to have the table renamed in all map windows.

**Rename Multiple Layers** is another new feature. This lets you rename all the layers of a map to match the file name of the table. It means that the underscores of the table name can be removed with spaces.

**Filter Using Cell Value** lets you quickly filter the records in a browser using the cell value of the cell you right click on.

**Show Info on Selection** shows the attributes of the selected record in the Info window.

## Features of WindowHelper

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

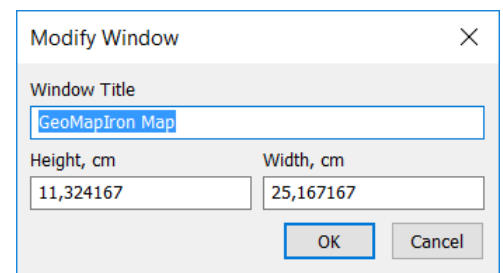
The *WindowHelper* tool embeds itself in a number of places. Some features are placed directly on tabs of the ribbon and you will find many feature in context menu/right click menus in a number of places.

### Modify Window

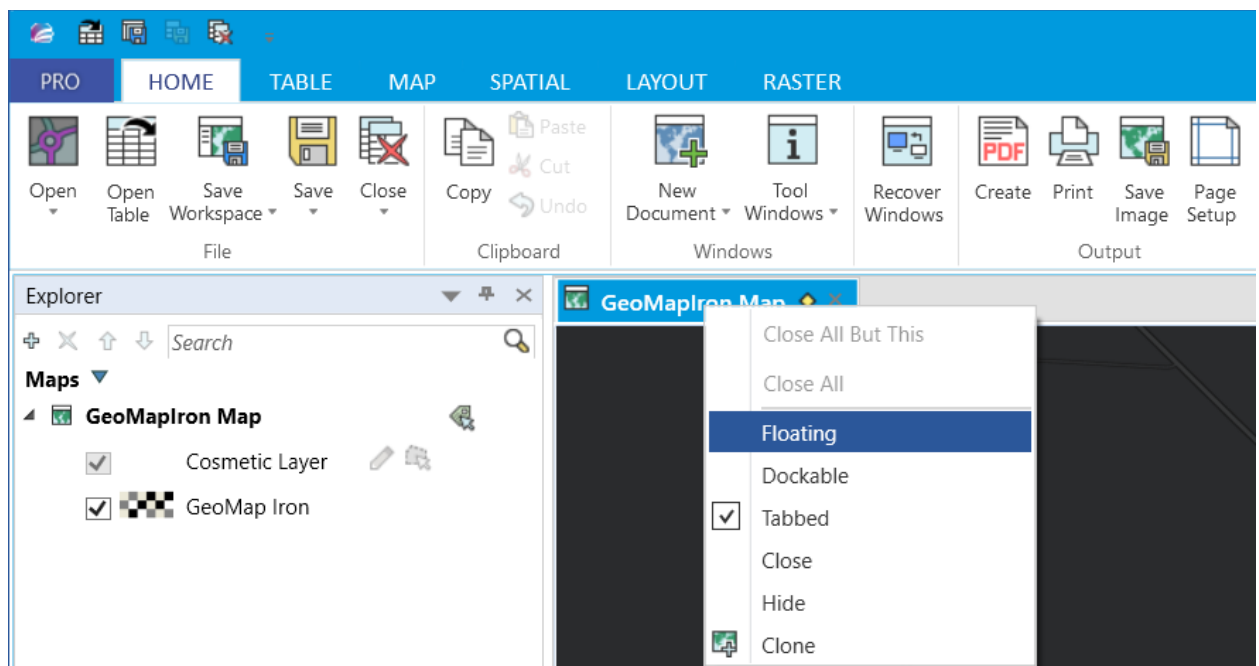
The *Modify Window* option can be found on the *HOME* tab in the *Windows* group. It lets you specify the floating size of document window. This is useful if you need to export your map window to an image of a certain size.

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.



### Creating Map Extent

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

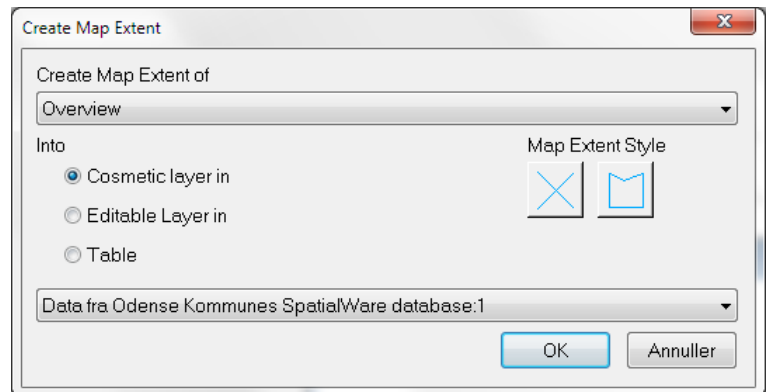
You can also create the map extent of the map window in the window itself. This can be useful if you need to find objects within the current map extent.

On the *MAP* tab in the group *Options*, 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 dialog *Create Map Extent* you can now pick the window which map extent you want to create.

You can choose 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 3<sup>rd</sup> 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.

## Show Info on Selection

The feature has been moved into WindowHelper from a stand-alone tool called InfoSelHandler.

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. This is always active also when Show Info on Selection has been deactivated.



## Locate File In Windows Explorer

In the right click menu of the *Tables* window and the right click menu of the layers in the *Layers* and 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.

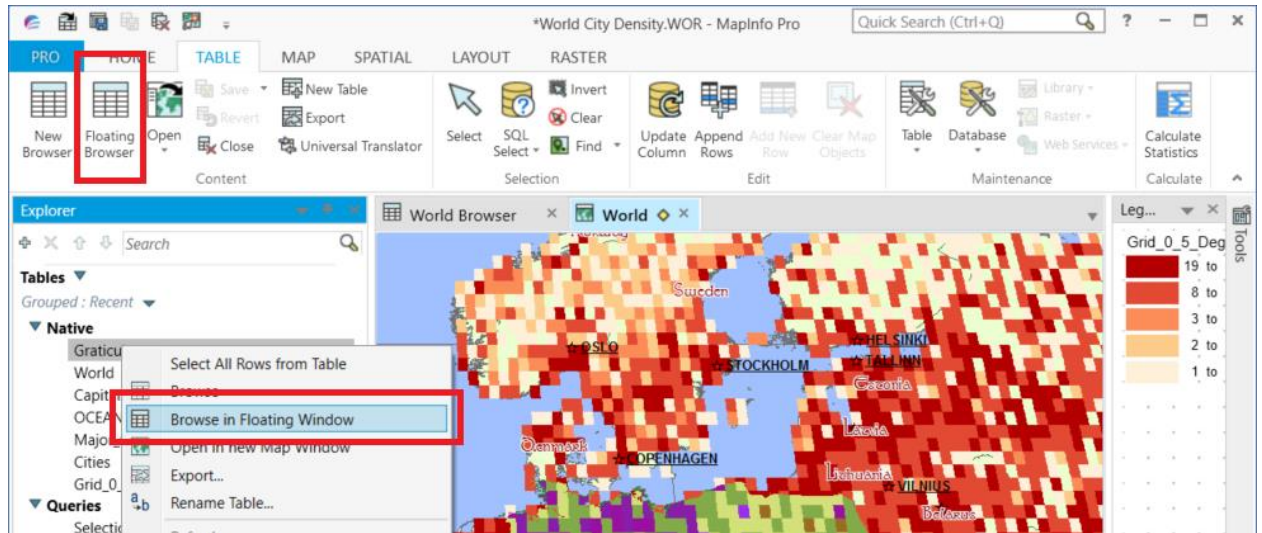
If you have selected a raster layer/table, you will be asked if you want to locate the tab file or the raster file.

## 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*.



## Open File In Favorite Text Editor

In the right click menu of the *Tables* window and the right click menu of the layers in the *Layers* and 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 *Layers* and the *Explorer* window you find the *Close table* menu item, that makes it easy to close the table for a specific layer.

This feature works on one selected layer.

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.

## Set Default View/Set Default View on All Layers

You will find this menu item in the right click menu of the Maps and of the Layers in the Layer Control.

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.

## Static Theme

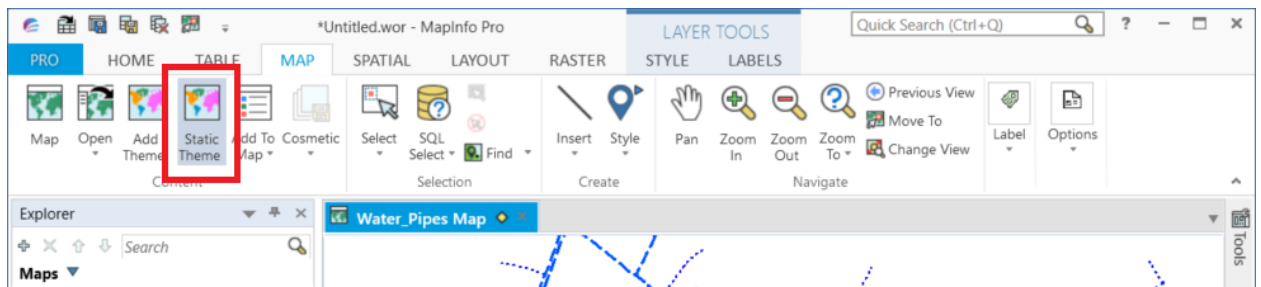
MapInfo Pro allows you to store the styling for a layer in different way: at an object level on the table, as a layer override and as a theme.

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.



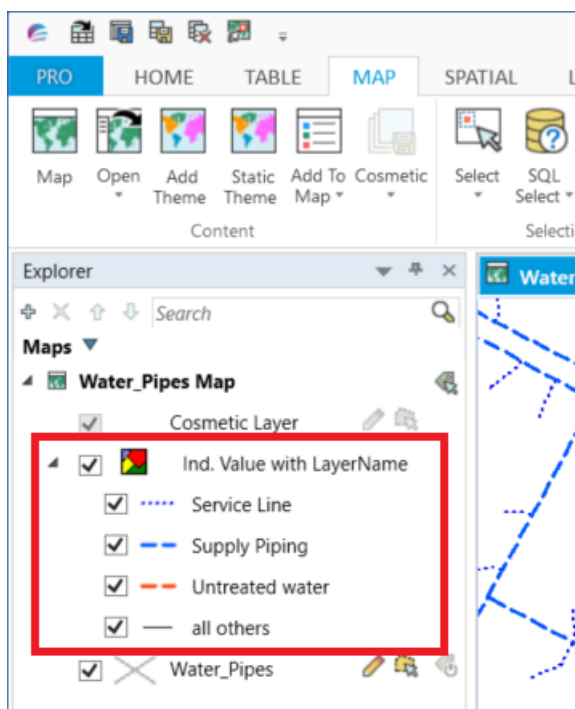
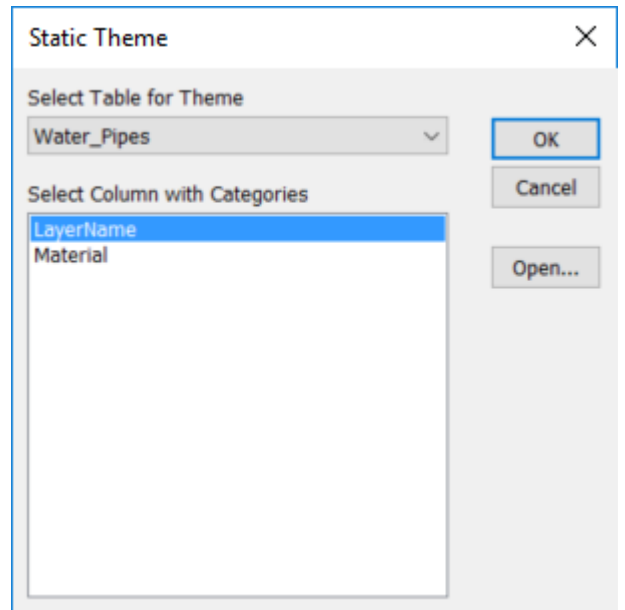
You can find *Static Theme* on the *MAP* tab right next to *Add Theme*.



First you need to 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 from the objects in the table.

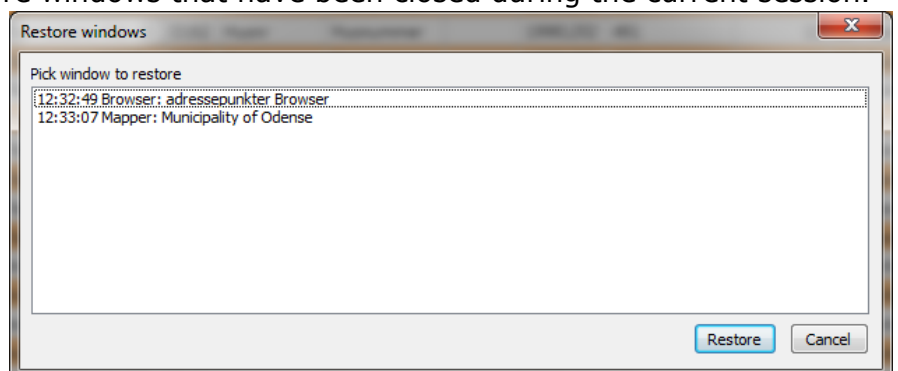
## Restore Closed Windows

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

On the MAP tab in the group Options you also





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

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.

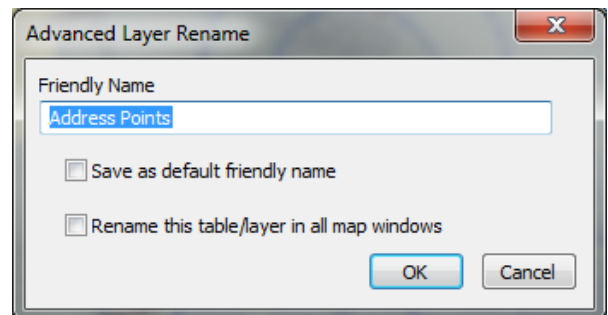
There is however one difference: WindowHelper does not “capture” which tables needs to be open. All it captures is the Window statement. So if you have closed some tables since you closed the window, the restore window might not work.

## Advanced Rename Layer

The Advanced Rename Layer feature builds on the new 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 *Explorer* and then *Layers* window.

When renaming a layer of a base table, you can now 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.



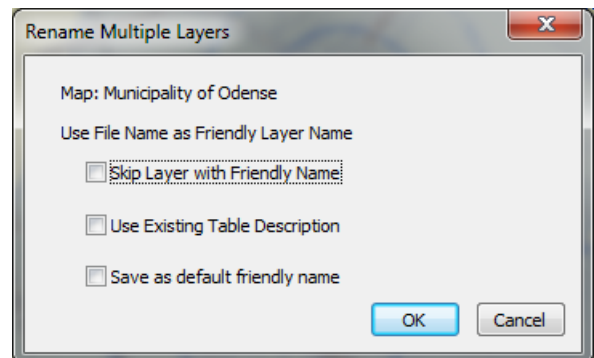
You can also rename the table in other map windows, if it should exist multiple times.

## Rename Multiple Layers

*Rename Multiple Layer* is found in the context menu of the *Map* in the *Explorer* and *Layers* window.

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.

## Filter by Cell Value

This feature can be found in two places:

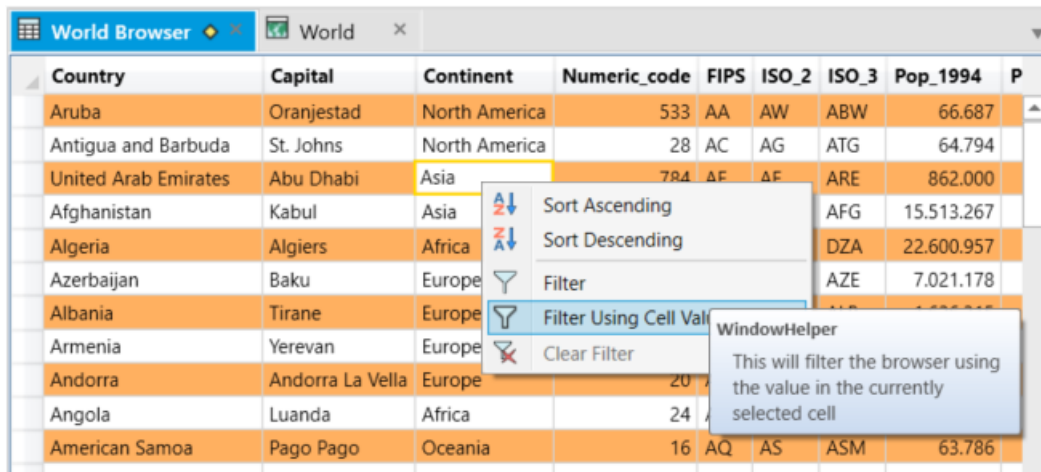
- In the context menu of the browser window

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

The feature gives you a 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.

## 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.

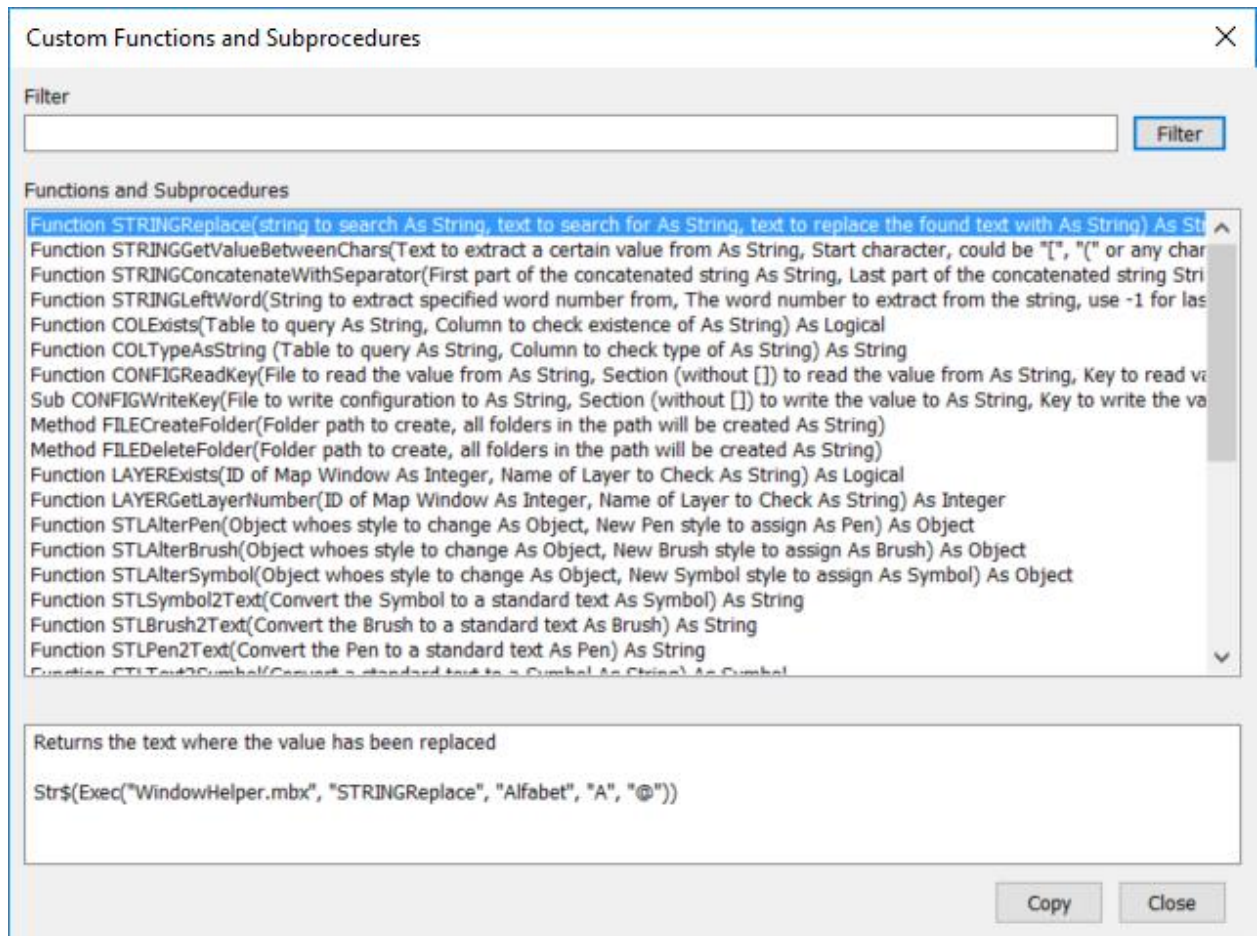
The problem often is to know which functions a MapBasic application provides you access to.

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 version 3.6 takes advantage of the new option to publish custom MapBasic functions to the MapInfo Pro interface.

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:

- **WINHSetCoordsysUsingTable**(*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
- **WINHSetCoordsysUsingMap**(*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
- **STRINGReplace**(*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.
  - Return the *inputString* where the occurrences of *searchFor* have been replaced with *replaceWith*.

- **STRINGGetValueBetweenChars**(*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 "".
- **STRINGConcatenateWithSeparator**(*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.
- **STRINGLeftWord**(*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.
- **COLExists**(*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).
- **COLTypeAsString**(*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 exist, an empty string is returned.
- **TABIsOpen**(*tableName*) As String
  - *tableName* is the table to look for
  - returns true (1) if the table is open, otherwise it returns false (0)
- **CONFIGReadKey**(*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.
- **CONFIGWriteKey**(*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.
- **STLAlterPen**(*object*, *penNew*), **STLAlterBrush**(*object*, *brushNew*), **STLAlterSymbol**(*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
- **STLSymbol2Text**(*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.
- **STLText2Symbol**(*stringSymbol*), **STLText2Brush**(*stringBrush*), **STLText2Pen**(*stringPen*)

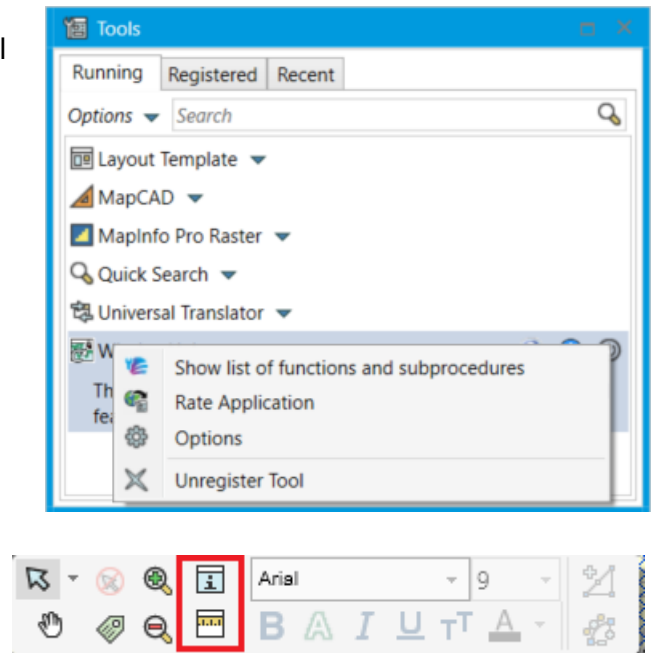
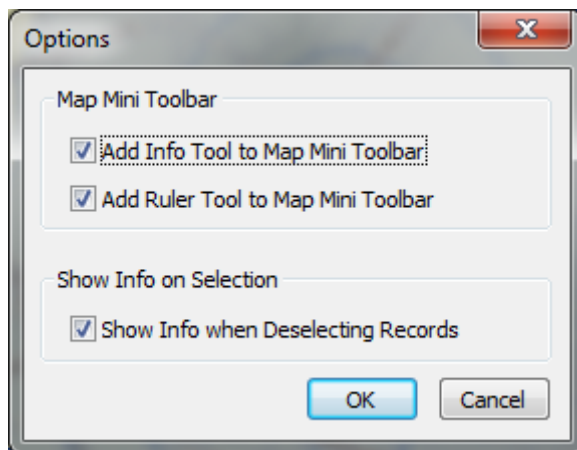


- *stringSymbol*, *stringBrush*, *stringPen* is the string representing the style you want to have converted into a true style.
  - returns the style, pen, symbol or brush, represented by the string entered.
- **OBJDirection(Object)**
  - *object* is the spatial object you want to calculate the direction off. Supports Points, Texts and Lines.
  - returns the direction of the object in degrees, east is zero, counter-clockwise. Unsupported objects return -1. Set the coordinate system using WINHSetCoordsysUsingTable using a projected table.
- **OBJReverse(object)**
  - *object*: The line or Polyline you want to reverse the node order of
  - returns the input object with the nodes order reversed.
- **OBJCartesianCreatePointAlong(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.
- **OBJCombineFromTable(tableName, columnName, columnValue)**
  - *tableName* is the name of the table to combine the objects from
  - *columnName* is the column to find the value given below to select only some records
  - *columnValue* is the group value used when selecting only some records to combine.
  - returns a combined object from the table given, either all records or only those where the value in the column specified equals the value given.
- **MAPGetExtent(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.
- **MAPHGetCoordinate(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.
- **TABHRegExReplace(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.
- **TABHRegExIsMatch(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*
- **TABHRegExFirstMatch(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>

## Options

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



The options lets 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.

## Known issues