



MapInfo Pro

Version 2019.3

MapInfo Pro Release Notes

Contents:

Introduction.....	2
Notes about this Release.....	2
System Requirements.....	65
Dependencies and Prerequisites.....	65
MapInfo Pro Database Connectivity and Support.....	68
Microsoft Office Support.....	69
MapInfo Pro Web Server Support.....	69
Installing MapInfo Pro.....	70
Repairing MapInfo Pro from the .MSI file.....	76
Support Notices.....	78
Downloading Tools and Applications.....	78
Locating Your Documentation.....	78
Sample Data Enhancements.....	79
Open Source Attribution.....	79



Introduction

This document gives you a list of the new and enhanced features introduced in this release. For details on these features, see *What's New in MapInfo Pro* chapter in the MapInfo Pro *Help System*. It also provides information about resolved issues and known issues that are important to MapInfo® Pro users.

Notes about this Release

MapInfo Pro

What's New in MapInfo Pro

Thank you for upgrading to the most advanced computer mapping product in the Precisely software family! As the field of computer mapping continues to expand, Precisely leads the way with new products that are designed to fulfill your computer mapping needs from the most basic to the most specialized. This is a list of the most important features and enhancements scheduled to be released with MapInfo Pro 2019.3.

Here are the highlights:

New Features

MapInfo Pro version 2019.3

Python Console Window

MapInfo Pro supports writing add-ins in the Python language. This allows developers to take advantage of the Python language and its existing libraries/modules when writing MapInfo Pro add-ins.

Like MapBasic, you can now edit and execute Python code interactively from within MapInfo Pro using the **Python** console window. On the **HOME** tab, open the **Tools Windows** gallery in the **Windows** group and click **Python** to open the **Python** console window. You can also open this window using the **Ctrl+Shift+Y** short-cut key.

You can also use the **SQL** window to create and manage Python scripts in the same way as SQL statements or MapBasic scripts. In the **Scripts** pane, click **Create new script** and select **Python** to start writing a Python script. All the helper drop-down options from the **Assist** section, except **Joins** and **Variables**, are available for use with the Python scripts.

LiDAR Surface

The **LiDAR Surface** tool is somewhat similar to the standard **Triangulation** tool available via the **Create Raster** Gallery in the **Interpolate** group. This tool provides limited triangulation capabilities to a standard MapInfo Pro user. **LiDAR Surface** uses only LiDAR data loaded from the LAS or LAZ format files.

For more details, see [Surface Analysis](#).

How do I access LiDAR Surface

- On the **MAP** tab, click on **LiDAR Surface** in the **Content** group to open the **LiDAR Analysis** dialog.

GetMessageWinText() function

Purpose

Returns the text present in the Message window.

Syntax

```
GetMessageWinText$( )
```

property is the requested property of the script dialog.

Example

```
Dim sRet as string
Print "This is a Message"      'send this to Message window
sRet = GetMessageWinText$( )
sRet will contain "This is a Message"
```

Return Value

String.

Description

This function returns the text present in the Message window.

MapBasic Silent Install

You can now install MapBasic silently in the background, just like MapInfo Pro. To install MapBasic silently, use the following commands:

For English

```
Setup.exe /L1033 /S /v"INSTALLDIR="C:\Test" /qb"
```

For Russian

```
Setup.exe /L1049 /S /v"INSTALLDIR="C:\Test" /qb"
```

For Japanese

```
Setup.exe /L1041 /S /v"INSTALLDIR="C:\Test" /qb"
```

Remove the `INSTALLDIR` parameter to install at the default location: `.C:\Program Files\MapInfo\MapBasic`.

To make it an entirely silent install without any progress bar, use the `/qn` switch instead of `/qb`

New Object Model

A new object model with a collection of Maps, including themes, has been added with all layers and layer properties.

This would make it very easy for Python and .NET developers to automate tasks and easily create add-ins. For more details, see the Extensibility Reference Help.

Input() function

Returns the requested input of a user from dialog. You can call this function from the MapBasic window in MapInfo Pro.

Syntax

```
Input(caption, prompt [, defaultValueExpr] [, type] [, properties]
[, valuesExpr])
```

caption is a string that will appear as the title in the **Input** dialog.

prompt is a string that displays text directly above the input edit box, showing what is expected for input. You can use an underscore (`_`) as an accelerator key.

defaultValue is an optional value returned if nothing is entered. It is shown as a value in the input edit box. It should be of the same variable type as expected for the return value.

type is an optional value described below to determine which type of input is requested. The default value is a **string**.

properties is an optional string parameter that can specify key-value pairs of items such as **HelpText**, **MinimumValue for a number**, **ValidationPattern**, etc.

Note: Multiple properties may be specified in the same string separated by a semi-colon (;). For example,

```
"MinimumValue=1;MaximumValue=10;HelpText=Enter a value between 1 and 10"
```

property	Description
MinimumValue	MinimumValue for a number.
MaximumValue	MaximumValue for a number.
HelpText	Used as a tool-tip on a control
CancelledValue	Value to be set if you cancel a dialog.
ValidationPattern and ValidationMessage	Applies to: string, any Combo-Box type but only when <i>IsEditable=True</i> ValidationPattern: regular expression against which the input is validated. ValidationMessage: message to show when input does not match the pattern.
NumberFormat	Applies to numeric types - <i>integer, shortint, largeint, float, decimal</i> . For example, <pre>print input ("MapInfo", "test", 12345.678, "float", "NumberFormat=" "\$, #.##" ") print input ("MapInfo", "test", 12345.678, "float", "NumberFormat=" "\$, #.##; ; (, #.##) " ") print input ("MapInfo", "test", 12345, "integer", "NumberFormat=" " , # " ")</pre> For more information on how to use number format strings, see the MapBasic Help Topic for the function Format\$()
AllowNoSelection	Applies to Combo-boxes. <i>True</i> or <i>False</i> - if <i>True</i> , then no input(selection) is required on the combo-box. If <i>False</i> (default), then a Validation error is shown, and the user must choose an item from the Combo-box.

valuesExpr is an optional value that generates a list of choices or restrictions on the range of values allowed.

Return Value

Any standard variable type like *String*, *Integer*, *LargeInt*, *Float*, *Logical*, *Date*, *DateTime*, or *Time*.

Example

```
Dim RetVal as Integer
RetVal = Input$("My Input Caption", "Input an integer", 999, "This is a
  help topic")
print RetVal
```

Description

The Input() function provides a simple way to request input from user for use in an application.

type	valuesExpr	Description	properties	Return Value	Examples
integer		The return value will be of type <i>integer</i> . Only valid integers can be entered. The <i>DefaultExpr</i> is evaluated and if it can be converted to an integer, it is used as the initial value.	MinimumValue MaximumValue CancelledValue HelpText NumberFormat	Integer	MinimumValue=1; MaximumValue=10 print input("MapInfo","test",12345,"integer","NumberFormat=","#")
smallint		The return value will be of type <i>smallint</i> . The <i>DefaultExpr</i> is evaluated and if it can be converted to a <i>smallint</i> , it is used as the initial value.	MinimumValue MaximumValue CancelledValue HelpText NumberFormat	SmallInt	
largeint		The return value will be of type <i>largeint</i> . The <i>DefaultExpr</i> is evaluated and if it can be converted to a <i>largeint</i> , it is used as the initial value.	MinimumValue MaximumValue CancelledValue HelpText NumberFormat	LargeInt	

type	valuesExpr	Description	properties	Return Value	Examples
float		<p>The return value will be of type <i>float</i>.</p> <p>The <i>DefaultExpr</i> is evaluated and if it can be converted to a <i>float</i> it is used as the initial value.</p>	MinimumValue MaximumValue CancelledValue HelpText NumberFormat	Float	<pre>print input ("MapInfo", "test", 12345.678, "float", "NumberFormat=""\$, #.##;,(#.##)""")</pre>
file		<p>Uses the MapBasic FileOpenDlg() function.</p> <p>.EXT is the type of file extension.</p> <p>The <i>DefaultExpr</i> is evaluated and is used as the initial value.</p>	Extension CancelledValue HelpText	String	<pre>Extension=TAB Extension=WOR Extension=CSV</pre>
table	expr	<p><i>expr</i> is a logical MapBasic expression that is evaluated</p> <p>for each open table. If it returns <i>TRUE</i>, the table is included</p> <p>in the drop-down list.</p> <p>ScriptContext(2) is substituted with the name (alias) of the table being evaluated. If no <i>expr</i> is given, then all open tables are listed. If there is an error evaluating the expression, then the table is included.</p> <p>The <i>DefaultExpr</i> is evaluated and if it matches the alias of one of the listed tables, that table is pre-selected.</p> <p>The return value is the alias of the selected table or empty string if none.</p>	CancelledValue HelpText AllowNoSelection	String	<pre>TableInfo(ScriptContext(2), TAB_INFO_DOMINANT_OBJECT_TYPE)=2 Or TableInfo(ScriptContext(2), TAB_INFO_DOMINANT_OBJECT_TYPE)=3 Like(ScriptContext(2), "%U%", "") dim t as string t = Input("Input", "_Table:", "usa", "table", "HelpText=Pick a dang table", "TableInfo(ScriptContext(2), TAB_INFO_DOMINANT_OBJECT_TYPE)=2") select * from t t = Input("Input", "_Table:", "usa", "table", "HelpText=Pick a dang table", "Like(ScriptContext(2), ""%U%"" , """"))") select * from t</pre>

type	valuesExpr	Description	properties	Return Value	Examples
column	tableColExpr	<p><i>tableColExpr</i> is a logical MapBasic expression that is evaluated</p> <p>for each open table and each column. If it returns <i>TRUE</i>, the table.column is included in the drop-down list.</p> <p>ScriptContext(2) is substituted with the alias of the table being evaluated and ScriptContext(3) is substituted with the column alias,</p> <p>If no <i>expr</i> is given, then the columns from all open tables are listed. If there is an error evaluating the expression, then the table is included.</p> <p>The <i>DefaultExpr</i> is evaluated and if it matches the alias of one of the listed columns, that column is pre-selected.</p> <p>The return value is the alias of the selected column or empty string if none.</p>	<p>CancelledValue</p> <p>HelpText</p> <p>AllowNoSelection</p>	Alias	<p>Open Table "C:\work\Data\USA\USA.TAB"</p> <p>dim col as Alias</p> <p>col = Input("Input", "_Column:", "URL", "column", "", "TableInfo(ScriptContext(2), TAB_INFO_NAME)=""USA""</p> <p>And ColumnInfo(ScriptContext(2), ScriptContext(3),COL_INFO_TYPE)=1")</p> <p>browse col from usa</p>
logical		<p>A check-box is displayed. The <i>DefaultExpr</i> is evaluated and if it can be converted to a logical, it is used as the initial value.</p>	<p>CancelledValue</p> <p>HelpText</p>	Logical	

type	valuesExpr	Description	properties	Return Value	Examples
date		<p>An editable Date Picker is displayed. The <i>DefaultExpr</i> is evaluated and if it can be converted to a date, it is used as the initial value.</p> <p>A range can be specified via a string in the <i>ValesExpr</i></p>	<p>MinimumValue</p> <p>MaximumValue</p>	Date	<p>MinimumValue=19960101;</p> <p>MaximumValue=19960531</p>
string		<p>This is an editable text-box. The <i>DefaultExpr</i> is evaluated and is used as the initial value.</p> <p>If a MaximumLength is specified, it is the number of characters you can enter.</p> <p>The default REGEX pattern is ".*"</p>	<p>ValidationPattern</p> <p>ValidationMessage</p> <p>MaximumLength</p> <p>CancelledValue</p> <p>HelpText</p>	String	<p>MaximumLength=5</p> <p>ValidationPattern=10:[^A-Z]+</p> <p>Dim ret as String*3</p> <p>ret = Input("Input", "Limit 3", "", "String", "MaximumLength=3")</p> <p>Print ret</p>
readonly		<p>This is a read-only text-block.</p> <p>The <i>DefaultExpr</i> is evaluated and is used as the value.</p>	<p>HelpText</p>	String	

type	valuesExpr	Description	properties	Return Value	Examples
list	valuesExpr	<p>A Combo-box is displayed. The text portion of the combo-box is not editable. To make it Editable, use the property <i>IsEditable=True</i>.</p> <p>The <i>DefaultExpr</i> is evaluated and is used as the initial value. If the <i>defaultvalue</i> matches a value in the list, that value is selected.</p> <p><i>valuesExpr</i> is needed to specify the choices:</p> <p><i>valuesExpr</i> is assumed to contain a semi-colon (;) delimited list of choices.</p> <p>A semi-colon can be represented in the value by (;:). A quote can be represented in the value by (").</p> <p>If the <i>AllowMultiSelect</i> property is True, then multiple choices can be made and the result is a ';' delimited string.</p>	<p>AllowMultiSelect</p> <p>IsEditable</p> <p>CancelledValue</p> <p>HelpText</p> <p>ValidationPattern</p> <p>ValidationMessage</p> <p>AllowNoSelection</p>	String	<pre>"george;ringo;paul;peter;"fred";;" print Input("MapInfo", "_ Select a Beatle: ", "", "list", "IsEditable=True;ValidationPattern=.;+", "george;ringo;paul;peter") print Input("MapInfo", "_ Select a Beatle: ", "", "list", "IsEditable=False", "george;ringo;paul;peter") print Input("MapInfo", "_ Select a Beatle: ", "george;ringo", "list", "AllowMultiSelect=True;ValidationPattern=.;+; CancelledValue=None", "george;ringo;paul;peter;"fred"")</pre>

type	valuesExpr	Description	properties	Return Value	Examples
values		<p>A Combo-box is displayed. The text portion of the combo-box is not editable. To make it editable, use the property <i>IsEditable=True</i>.</p> <p>The <i>DefaultExpr</i> is evaluated and is used as the initial value. If the <i>defaultvalue</i> matches a value in the list, that value is selected.</p> <p><i>valuesExpr</i> is needed to specify the choices:</p> <p><i>valuesExpr</i> it is assumed to be a MapBasic expression containing at least one table.column or column reference to an open table. It is evaluated against the rows in the table to get unique values until either 1000 unique values are found, the end of the table is reached, or two seconds have passed.</p>	<p>CancelledValue</p> <p>HelpText</p> <p>AllowMultiSelect</p> <p>AllowNoSelection</p>	String	<pre>UCASE\$(usa.state_name) Open Table "C:\work\Data\USA\USA.TAB" Map From usa Dim val as string val = Input("MapInfo", "Search Value:", "", "values", "IsEditable=True;ValidationPattern=.*", "UCASE\$(usa.State_Name)") select * from usa where USA.State_Name like "%" + val + "%" Multiselect: Dim val as string val = Input("MapInfo", "Search Value:", "", "values", "AllowMultiSelect=True;ValidationPattern=.*", "UCASE\$(states.State_Name)") print val</pre>

type	valuesExpr	Description	properties	Return Value	Examples
color		<p>A Color drop-down is displayed. The <i>DefaultExpr</i> is evaluated and is used as the initial value. The value is converted to an integer and is assumed to be in RGB format. If not specified, black is the default color.</p> <p>Note: The standard definitions file, MAPBASIC.DEF, includes Define statements for several common colors (BLACK, WHITE, RED, GREEN, BLUE, CYAN, MAGENTA, and YELLOW). If you want to specify a Red color, you can simply use the identifier RED instead of calling RGB() function.</p>	CancelledValue HelpText	Integer	<p>defaultExpr: RGB(255,0,0)</p> <p>defaultExpr: 255</p>

type	valuesExpr	Description	properties	Return Value	Examples
window	:expr	<p><i>expr</i> is a logical MapBasic expression that is evaluated</p> <p>for each open window. If it returns TRUE the window is included in the drop-down list.</p> <p>ScriptContext(1) is substituted with the ID of the window being evaluated. If no expr is given, then all open windows are listed. If there is an error evaluating the expression, then the window is included.</p> <p>The <i>DefaultExpr</i> is evaluated and if it matches the ID of one of the listed windows, that window is pre-selected.</p> <p>The return value is the ID of the selected window or 0 if none.</p>	<p>CancelledValue</p> <p>HelpText</p> <p>AllowNoSelection</p>	Integer	<pre>WindowInfo(ScriptContext(2), 3)=1 WindowInfo(ScriptContext(2), WIN_INFO_TYPE)=1 Or WindowInfo(ScriptContext(2), WIN_INFO_TYPE)=2 Dim window as Integer window = Input("MapInfo", "Window:", "", "window") print window + ":" + WindowInfo(window, WIN_INFO_NAME) window = Input("MapInfo", "Window:", frontwindow(), "window") print window + ":" + WindowInfo(window, WIN_INFO_NAME) Select a Map Window print Input("Choose Window", "_Select a Window:",frontwindow(), "", "window","windowinfo(ScriptContext(1), 3)=1")</pre>

type	valuesExpr	Description	properties	Return Value	Examples
layer	:expr	<p><i>expr</i> is a logical MapBasic expression that is evaluated</p> <p>for each layer in each open map. If it returns TRUE, the layer is included in the drop-down list. ScriptContext(1) is substituted with the ID of the map being evaluated. ScriptContext(4) is substituted with the ID of the layer being evaluated.</p> <p>If no <i>expr</i> is given, then all layers in the last used map are listed. If there is an error evaluating the expression then the layer is included. The <i>DefaultExpr</i> is evaluated and if it matches the ID of one of the listed layers, that layer is pre-selected.</p> <p>The return value is the ID of the selected layer or 0 if no <i>CancelledValue</i> property is provided.</p> <p>Note: If multiple maps are included, then the return value is ambiguous. So it is expected that the expression is used to pick only the layers from one map.</p>	<p>CancelledValue</p> <p>HelpText</p> <p>AllowNoSelection</p>	Integer	<p>pick a selectable layer from the current map</p> <p>Dim layer as integer</p> <pre>layer = Input("Choose Layer", "_ Select a Layer:", "1", "layer", "CancelledValue=-1", "ScriptContext(1)=FrontWindow() and LayerInfo(ScriptContext(1), ScriptContext(4), 3)=1")</pre> <p>print layer</p> <p>pick a selectable Point layer from the current map</p> <pre>layer = Input("Choose Layer", "_ Select a Layer:", "1", "layer", "CancelledValue=-1;HelpText=Selectable Point Layers", "ScriptContext(1)=FrontWindow() and LayerInfo(ScriptContext(1), ScriptContext(4), 3)=1 and TableInfo(LayerInfo(ScriptContext(1), ScriptContext(4), 1),60)=2")</pre> <p>print layer</p>

type	valuesExpr	Description	properties	Return Value	Examples
coordsys		Choose Projection dialog is displayed. if <i>DefaultValueExpr</i> is provided then, the Projection list will initialize to the matching projection member in the Choose Projection dialog.	CancelledValue HelpText	string	Print Input("Projection Picker", "Choose Projection", "coordsys", "Coordsys Earth Projection 1, 104")
brush		Choose Brush dialog is displayed.	CancelledValue HelpText	brush	dim b as brush b = Input("Brush", "Choose _Brush", "", "brush") print b
pen		Choose Pen dialog is displayed.	CancelledValue HelpText	pen	dim p as pen p = Input("Pen", "Choose _Pen", "", "pen") print p
font		Choose Font dialog is displayed.	CancelledValue HelpText	font	dim f as font f = MakeFont ("Arial Baltic", 512, 9, 11599792, WHITE) f = Input("Font", "Choose _Font", f, "font") print f
symbol		Choose Symbol dialog is displayed.	CancelledValue HelpText	symbol	dim s as symbol s = Input("Font", "Choose _Font", "Symbol (34, 16711680, 12)", "symbol") print s

ScriptContext() function

Purpose

Returns the requested property of a script dialog that is being displayed.

Syntax

```
ret = ScriptContext(property, [, variable])
```

property is the requested property of the script dialog.

Property	Value	Return Type	Remarks
SCRIPT_CONTEXT_WINDOWID	1	Integer	<p>Used when prompting for a window variable or a layer variable. This returns the current window id being evaluated.</p> <p>To choose a map window</p> <pre>print Input("Choose Window", "_Select a Window:", frontwindow(), "", "window", "windowinfo (ScriptContext (1), 3)=1")</pre>
SCRIPT_CONTEXT_TABLEALIAS	2	String	<p>Used when prompting for a table or column variable. This returns the current table alias being evaluated.</p> <p>To choose a point table</p> <pre>dim t as string t = Input("Input", "_Table:", "usa", "table", "HelpText=Pick a dang table", "TableInfo (ScriptContext (2), TAB_INFO_DOMINANT_OBJECT_TYPE)=2") select * from t</pre>

Property	Value	Return Type	Remarks
SCRIPT_CONTEXT_COLUMNALIAS	3	String	<p>Used when prompting for a column variable. This returns the current column alias being evaluated.</p> <p>To choose a character column from usa table:</p> <pre> Open Table "C:\work\Data\USA\USA.TAB" dim col as Alias col = Input("Input", "_Column:", "URL", "column", "", "TableInfo (ScriptContext (2), TAB_INFO_NAME)=""USA"" And ColumnInfo (ScriptContext (2), ScriptContext (3),COL_INFO_TYPE)=1") browse col from usa </pre>
SCRIPT_CONTEXT_LAYER_NUM	4	Integer	Used when prompting for a Layer variable. This returns the current layer number being evaluated.

Return Value

Any standard variable type like *String*, *Integer*, *LargeInt*, *Float*, *Logical*, *Date*, *DateTime*, or *Time*.

Description

Returns the requested property of a script dialog that is being displayed.

MapInfo Pro version 2019.1

Python Language Support

MapInfo Pro now supports writing add-ins in the Python language. This allows developers to take advantage of the Python language and its existing libraries/modules when writing MapInfo Pro add-ins. Python, with its feature rich set of libraries, is a very popular platform for Data Analysis and Visualization. We have provided Python developers and data analysts an easier way to work with MapInfo data by providing access to MapInfo Pro table data. While you can make use of any of the libraries and functionality available to Python, the ability to perform data analysis on MapInfo Pro table attribute data using the rich set of available data analysis tools (for example, Pandas, SciPy, Matplotlib, etc.) adds more capabilities and functionalities to existing ones.

For details, see **Using Python Language** under **Creating an Add-in** in the MapInfo Pro Extensibility Reference Guide.

Installation

Python 3.7.6 x64 is installed with MapInfo Pro in the installation directory in the Python37 folder. MapInfo Pro uses the PythonNet implementation to interact with the Python environment and allow interop between .NET and Python.

Python installation bundled with MapInfo Pro has the following modules pre-loaded:

EFAL is installed with MapInfo Pro in the installation directory in the EFAL folder. Install Additional Python modules:

1. numpy
2. scipy
3. seaborn
4. pandas
5. matplotlib
6. osgeo (Access to GDAL library with MapInfo EFAL driver for NativeX format)
7. ptvsd (Python Tools for Visual Studio debug server)
8. mi_py_debugger (Custom module enable debugging of python script in MapInfo Pro)

To install additional python modules, open the script prompt.bat in %INSTALLDIR%\Python37 folder and execute the following command:

```
python -m pip install module_name
```

Running Python Scripts and Add-ins

The "run application" MapBasic statement now supports running *.py files. Any valid python will work as long as the required modules are installed. The Run Program dialog (Ctrl+U) now lists both .mbx and .py files. You can also drag a .py file onto MapInfo Pro to run it.

There are two ways in which MapInfo Pro runs Python files:

1. Add-in Scope - If the .py file is recognized as a MapInfo Pro add-in, then it is run in its own scope and app domain. Note that when the add-in is run, a .mbx file will still be created next to it, which means that the location must be write-able.
2. Global Scope - If the .py file is not recognized as a MapInfo Pro add-in then it is run in the Global scope. This is a special python environment that is initialized the first time any python code is run. The following setup is done:
 - sys, clr, subprocess, shlex, matplotlib python modules are already imported in this special python environment.
 - matplotlib module uses a Non-GUI backend called agg which will not allow call to plt.show() from within MapInfo Pro in global scope.
 - Reference to WindowsBase .Net framework assembly is added using clr module.

- PythonProUtil c# utility class is imported as proUtil from MapInfo.Types assembly.
- MessageOutput class is imported from MapInfo.Types assembly.
- MessageOutput class overrides the default python stdout, stdin, stderr and prints to MapInfo Pro message window.

```
import clr
import subprocess
import shlex
import matplotlib
# Non-GUI backend for matplotlib. use plt.savefig(fname) instead of
plt.show()
matplotlib.use('agg')
clr.AddReference('WindowsBase')
from MapInfo.Types import PythonProUtil as proUtil
from MapInfo.Types import MessageOutput
sys.stdout = sys.stderr = sys.stdin = MessageOutput()
```

- RASTER folder is added to path.
- A python variable named pro is initialized to point to the IMapInfoPro interface. It can be used to access MapInfo Pro's object model and to execute MapBasic commands.
- The following help methods are imported in global scope, which can be used directly in Python scripts:

```
end_mapinfo(interactive) -> Halts MapInfo Pro.
Parameters:
  interactive (bool): prompt to save changes or not when exiting.

do(command) -> Execute an interpreted MapBasic command.
Parameters:
  command (str): the command to run.

eval(command) -> Evaluate an interpreted MapBasic function.
Parameters:
  command (str): the command to evaluate.

get_current_module_path() -> Gets the current executing module path
for standalone python script. (In global scope __file__ is not
available,
therefore use this method.)
```

- If a startup.py file is found in one of the following locations it is executed in the global scope setup. Similar to the way startup.wor is found.
 - FOLDER_MI_APPDATA (-1)
 - FOLDER_MI_LOCAL_APPDATA (-2)
 - FOLDER_MI_PREFERENCE (-3)
 - FOLDER_MI_COMMON_APPDATA (-4)
 - Folder where MapInfoPro.exe is located

See GetFolderPath\$() MapBasic function for more help on these locations.

Once the global scope has been initialized as above, every time the Python code is run, a copy of the global scope is made and used to execute the code.

Creating a Python Add-in

To create a MapInfo Pro add-in in python:

1. Use one of the existing python add-in templates to start with your add-in. The MapBasic version 2019 installation contains two python add-in templates for creating a simple MapInfo Pro add-in or a layout custom frame add-in in python.
 - SAMPLES\RIBBONINTERFACE\Python\py_addin_templates\Simple
 - SAMPLES\RIBBONINTERFACE\Python\py_addin_templates\CustomFrame
2. Copy one of the add-in templates listed above and start modifying it.
 - Start with renaming the files (PY and MB) to your add-in name.
 - Rename module reference as per your new file names.
 - Check for all TODO in python code and modify the code as per your requirement.
3. Once the code is completed, you can run the main python file directly in MapInfo Pro. If your add-in does not have a corresponding .mbx file, a new one with the same base filename as the .py file will be created next to it. This means that the location must be write-able. You can also create, customize and compile the template .mb file.

Note: The name of the MapBasic MB file should be the same as the main python module file. Also, the class below should be present in the main Python module file to load the add-in.

```
# this class is needed with same name in order to load the python add-in
# and can be copied
# as it is when creating another add-in.
class main():
    def __init__(self, imapinfo):
        self._pro = imapinfo

    def load(self):
        try:
            # uncomment these lines to debug the python script using
            VSCODE
            # Debug in VSCODE with Python: Attach configuration

            #ptvsd.enable_attach()
            #ptvsd.wait_for_attach()
            #ptvsd.break_into_debugger()

            if self._pro:
                # obtain the handle to current application if needed
                self.thisApplication =
                self._pro.GetMapBasicApplication(os.path.splitext(__file__)[0] + ".mbx")
```

```

        except Exception as e:
            print("Failed to load: {}".format(e))

    def unload(self):
        try:
            print("Unloading")

self._pro.Ribbon.Tabs['TabMap'].Groups[0].Controls.Remove(self._button)

        except Exception as e:
            print("Failed to unload: {}".format(e))

    def __del__(self):
        self._pro = None
        pass

    def addin_name(self) -> str:
        return "Hello Python Add-in"

    def addin_description(self) -> str:
        return "Hello Python Add-in Description"

    def addin_defaultcommandtext(self) -> str:
        return "Hello Python Add-in Default Command"

    def addin_defaultcommand(self):
        self.on_hello_button_clicked(self)

    def addin_imageuri(self) -> str:
        return join(dirname(__file__), "images/copyright16.png")

```

Loading a MapInfo Pro Python Add-in

To Load a python addin in MapInfo Pro, you need to create a startup MapBasic script which will load the Python module and start executing it. The following sample mapbasic script templates can be used as is or can be modified for loading a Python add-in in MapInfo Pro.

```

'MBX Template for Simple Python Addin
'*****
'      Copyright 2020 Precisely
'      All rights reserved.
'*****/
Include "mapbasic.def"
Include "IMapInfoPro.def"
Include "ResStrng.def"

Declare Method PyAddinAbout Class "MapInfo.Types.PythonAddinLoader" Lib
"MapInfo.Types.dll" Alias AddIn_About(ByVal p1 as This)

```

```

Declare Method PyAddinHelp Class "MapInfo.Types.PythonAddinLoader" Lib
"MapInfo.Types.dll" Alias AddIn_Help(ByVal p1 as This)
Declare Method PyAddinDefaultCommand Class
"MapInfo.Types.PythonAddinLoader" Lib "MapInfo.Types.dll" Alias
AddIn_DefaultCommand(ByVal p1 as This)
Declare Method PyAddinVersion Class "MapInfo.Types.PythonAddinLoader"
Lib "MapInfo.Types.dll" Alias AddIn_Version(ByVal p1 as This) As String
Declare Method PyAddinName Class "MapInfo.Types.PythonAddinLoader" Lib
"MapInfo.Types.dll" Alias AddIn_Name(ByVal p1 as This) As String
Declare Method PyAddinDescription Class "MapInfo.Types.PythonAddinLoader"
Lib "MapInfo.Types.dll" Alias AddIn_Description(ByVal p1 as This) As
String
Declare Method PyAddinImageUri Class "MapInfo.Types.PythonAddinLoader"
Lib "MapInfo.Types.dll" Alias AddIn_ImageUri(ByVal p1 as This) As String
Declare Method PyAddinHideRunningProgram Class
"MapInfo.Types.PythonAddinLoader" Lib "MapInfo.Types.dll" Alias
AddIn_HideRunningProgram(ByVal p1 as This) As Logical
Declare Method PyAddinProcessCommandUri Class
"MapInfo.Types.PythonAddinLoader" Lib "MapInfo.Types.dll" Alias
AddIn_ProcessCommandUri(ByVal p1 as This, ByVal commandUri as String)
As Logical
Declare Method PyAddinDefaultCommandText Class
"MapInfo.Types.PythonAddinLoader" Lib "MapInfo.Types.dll" Alias
AddIn_DefaultCommandText(ByVal p1 as This) As String

Declare Sub EndHandler
Declare Sub Main

Declare Sub AddIn_About
Declare Sub AddIn_Help
Declare Sub AddIn_DefaultCommand
Declare Function AddIn_Name() As String
Declare Function AddIn_Description() As String
Declare Function AddIn_ImageUri() As String
Declare Function AddIn_Version() As String
Declare Function AddIn_AssemblyName() As String
Declare Function AddIn_ClassName() As String
Declare Function AddIn_HideRunningProgram() As Logical
Declare Function AddIn_ProcessCommandUri(ByVal commandUri as String) As
Logical
Declare Function AddIn_DefaultCommandText() As String

Global mapinfoApplication as This
Global mbAddIn as This

Sub Main
    onerror goto ErrorTrap

        if (SystemInfo(SYS_INFO_MIPLATFORM) = MIPLATFORM_WIN64) then
            if (SystemInfo(SYS_INFO_MAPINFO_INTERFACE) =
MIINTERFACE_RIBBON) then
                goto ContinueSample
            end if

```

```

        end if

        InvalidScenario:
            Note "This sample can only work with 64-bit Ribbon Interface
UX."
            goto Done
        ContinueSample:

        dim app_name as string
        app_name = ApplicationName$()
        app_name = Left$(app_name, len(app_name) - 4)

        dim str_file as string
        str_file = ApplicationDirectory$() + app_name + ".str"
        If FileExists(str_file) Then
            If LoadStrings(str_file) = FALSE Then
                Note GetResString(IDS_Res_Load_Error) +
ApplicationDirectory$() + app_name + ".str"
                Exit Sub
            End If
        End If

        mapinfoApplication = SYSTEMINFO(SYS_INFO_IMAPINFOAPPLICATION)
        mbAddIn = New_PythonAddin()

        if mbAddIn <> 0 Then
            call InitializePyAddin(mbAddIn, mapinfoApplication,
ApplicationDirectory$() + ApplicationName$())
        End if

        Done:
        Exit Sub

        ErrorTrap:
            Note "Main: " + Str$(err()) + ": " + Error$()
            Resume done

    End Sub

Sub EndHandler
    If mapinfoApplication <> 0 and mbAddIn <> 0 Then
        call UnloadPyAddin(mbAddIn)
    End If
    mbAddIn = NULL_PTR
    mapinfoApplication = NULL_PTR
End Sub

Sub AddIn_About
    call PyAddinAbout(mbAddIn)
End Sub

Sub AddIn_Help
    call PyAddinHelp(mbAddIn)

```

```

End Sub

Sub AddIn_DefaultCommand
    call PyAddinDefaultCommand(mbAddIn)
End Sub

Function AddIn_Version() As String
    AddIn_Version = PyAddinVersion(mbAddIn)
End Function

Function AddIn_Name() As String
    AddIn_Name = PyAddinName(mbAddIn)
End Function

Function AddIn_Description() As String
    AddIn_Description = PyAddinDescription(mbAddIn)
End Function

Function AddIn_ImageUri() As String
    AddIn_ImageUri = PyAddinImageUri(mbAddIn)
End Function

Function AddIn_HideRunningProgram() As Logical
    AddIn_HideRunningProgram = PyAddinHideRunningProgram(mbAddIn)
End Function

Function AddIn_DefaultCommandText() As String
    Addin_DefaultCommandText= PyAddinDefaultCommandText(mbAddIn)
End Function

Function AddIn_ProcessCommandUri(ByVal commandUri as String) As Logical

    AddIn_ProcessCommandUri = PyAddinProcessCommandUri(mbAddIn,
commandUri)
End Function

```

```

'MBX Template for Layout Custom Frame Python Addin
'*****
'      Copyright 2020 Precisely.
'      All rights reserved.
'*****/
Include "mapbasic.def"
Include "IMapInfoPro.def"
Include "ResStrng.def"

Declare Method PyAddinAbout Class "MapInfo.Types.PythonAddinLoader" Lib
"MapInfo.Types.dll" Alias AddIn_About(ByVal p1 as This)
Declare Method PyAddinHelp Class "MapInfo.Types.PythonAddinLoader" Lib
"MapInfo.Types.dll" Alias AddIn_Help(ByVal p1 as This)
Declare Method PyAddinDefaultCommand Class
"MapInfo.Types.PythonAddinLoader" Lib "MapInfo.Types.dll" Alias
AddIn_DefaultCommand(ByVal p1 as This)

```



```

Declare Method PyAddinVersion Class "MapInfo.Types.PythonAddinLoader"
Lib "MapInfo.Types.dll" Alias AddIn_Version(ByVal p1 as This) As String
Declare Method PyAddinName Class "MapInfo.Types.PythonAddinLoader" Lib
"MapInfo.Types.dll" Alias AddIn_Name(ByVal p1 as This) As String
Declare Method PyAddinDescription Class "MapInfo.Types.PythonAddinLoader"
Lib "MapInfo.Types.dll" Alias AddIn_Description(ByVal p1 as This) As
String
Declare Method PyAddinImageUri Class "MapInfo.Types.PythonAddinLoader"
Lib "MapInfo.Types.dll" Alias AddIn_ImageUri(ByVal p1 as This) As String
Declare Method PyAddinHideRunningProgram Class
"MapInfo.Types.PythonAddinLoader" Lib "MapInfo.Types.dll" Alias
AddIn_HideRunningProgram(ByVal p1 as This) As Logical
Declare Method PyAddinProcessCommandUri Class
"MapInfo.Types.PythonAddinLoader" Lib "MapInfo.Types.dll" Alias
AddIn_ProcessCommandUri(ByVal p1 as This, ByVal commandUri as String)
As Logical
Declare Method PyAddinDefaultCommandText Class
"MapInfo.Types.PythonAddinLoader" Lib "MapInfo.Types.dll" Alias
AddIn_DefaultCommandText(ByVal p1 as This) As String

Declare Sub EndHandler
Declare Sub Main

Declare Sub AddIn_About
Declare Sub AddIn_Help
Declare Sub AddIn_DefaultCommand
Declare Function AddIn_Name() As String
Declare Function AddIn_Description() As String
Declare Function AddIn_ImageUri() As String
Declare Function AddIn_Version() As String
Declare Function AddIn_AssemblyName() As String
Declare Function AddIn_ClassName() As String
Declare Function AddIn_HideRunningProgram() As Logical
Declare Function AddIn_ProcessCommandUri(ByVal commandUri as String) As
Logical
Declare Function AddIn_DefaultCommandText() As String

Global mapinfoApplication as This
Global mbAddIn as This

Sub Main
    onerror goto ErrorTrap

        if (SystemInfo(SYS_INFO_MIPLATFORM) = MIPLATFORM_WIN64) then
            if (SystemInfo(SYS_INFO_MAPINFO_INTERFACE) =
MIINTERFACE_RIBBON) then
                goto ContinueSample
            end if
        end if

        InvalidScenario:
            Note "This sample can only work with 64-bit Ribbon Interface
UX."

```

```

        goto Done
ContinueSample:

dim app_name as string
app_name = ApplicationName$()
app_name = Left$(app_name, len(app_name) - 4)

dim str_file as string
str_file = ApplicationDirectory$() + app_name + ".str"
If FileExists(str_file) Then
    If LoadStrings(str_file) = FALSE Then
        Note GetResString(IDS_Res_Load_Error) +
ApplicationDirectory$() + app_name + ".str"
        Exit Sub
    End If
End If

mapinfoApplication = SYSTEMINFO(SYS_INFO_IMAPINFOAPPLICATION)
mbAddIn = New_PythonCustomFrameAddin()

if mbAddIn <> 0 Then
    call InitializePyAddin(mbAddIn, mapinfoApplication,
ApplicationDirectory$() + ApplicationName$())
End if

Done:
Exit Sub

ErrorTrap:
    Note "Main: " + Str$(err()) + ": " + Error$()
    Resume done

End Sub

Sub EndHandler
    If mapinfoApplication <> 0 and mbAddIn <> 0 Then
        call UnloadPyAddin(mbAddIn)
    End If
    mbAddIn = NULL_PTR
    mapinfoApplication = NULL_PTR
End Sub

Sub AddIn_About
    call PyAddinAbout(mbAddIn)
End Sub

Sub AddIn_Help
    call PyAddinHelp(mbAddIn)
End Sub

Sub AddIn_DefaultCommand
    call PyAddinDefaultCommand(mbAddIn)
End Sub

```

```

Function AddIn_Version() As String
    AddIn_Version = PyAddinVersion(mbAddIn)
End Function

Function AddIn_Name() As String
    AddIn_Name = PyAddinName(mbAddIn)
End Function

Function AddIn_Description() As String
    AddIn_Description = PyAddinDescription(mbAddIn)
End Function

Function AddIn_ImageUri() As String
    AddIn_ImageUri = PyAddinImageUri(mbAddIn)
End Function

Function AddIn_AssemblyName() As String
    Addin_AssemblyName = "MapInfo.Types"
End Function

Function AddIn_ClassName() As String
    Addin_ClassName = "MapInfo.Types.PythonCustomFrameAddinLoader"
End Function

Function AddIn_HideRunningProgram() As Logical
    AddIn_HideRunningProgram = PyAddinHideRunningProgram(mbAddIn)
End Function

Function AddIn_DefaultCommandText() As String
    Addin_DefaultCommandText= PyAddinDefaultCommandText(mbAddIn)
End Function

Function AddIn_ProcessCommandUri(ByVal commandUri as String) As Logical

    AddIn_ProcessCommandUri = PyAddinProcessCommandUri(mbAddIn,
commandUri)
End Function

```

Registering a Python Add-in

You can now register a .py file as an add-in from the tool manager using the Register Tool dialog. Note that when the add-in is run, a .mbx file will still be created next to it, which means that the location must be write-able.

Setup Python Debugging using Visual Studio Code (VSCODE)

1. Install the Python and pylint extensions and restart VSCODE.
2. Initialize configurations. A configuration drives VS Code's behavior during a debugging session. Configurations are defined in a launch.json file that's stored in a .vscode folder in your workspace.

3. To initialize debug configurations, first select the Debug View in the sidebar. If you don't yet have any configurations defined, you'll see "No Configurations" in the drop-down list, and a dot on the settings icon.
4. To generate a launch.json file with Python configurations:
 - Select the Settings button or use the Debug > Open configurations menu command.
 - In the Select Environment drop-down list that appears, select Python.

The Python extension then creates and opens a launch.json file that contains number of pre-defined configurations. You can modify configurations (to add arguments, for example), and also add custom configurations.

```
Python: Attach or Python: Remote Attach configuration must be defined
like this.
```

```
{
  "name": "Python: Remote Attach",
  "type": "python",
  "request": "attach",
  "port": 5678,
  "host": "localhost",
  "pathMappings": [
    {
      "localRoot": "${workspaceFolder}",
      "remoteRoot": "${workspaceFolder}"
    }
  ]
}
```

5. After the initial configuration is done, setup VSCODE for remote debugging.
 - Install ptvsd using `python -m pip install --upgrade ptvsd` into your environment.
 - For debugging the default Port for the ptvsd use is 5678. You may need to open the port for debugging in the appropriate firewall or other networking configuration.

For debugging the default Port for the ptvsd use is 5678. You may need to open the port for debugging in the appropriate firewall or other networking configuration.

For more details refer to VSCODE python debugging :

<https://code.visualstudio.com/docs/python/debugging>

Debugging Python Scripts in MapInfo Pro

1. In VSCODE open the folder where your python script is located and then open the python script file from VSCODE Explorer.
2. Add the following lines near the top of your script. It is fine to have comments above them.

```
from mi_py_debugger import *
pro = attach_debugger()
```

3. Run the python script (.py) in MapInfo Pro. You will notice a wait cursor denoting that the process is waiting for some debugger to attach to it.
4. Switch to Debug View in VS Code, and select Python: Attach configuration.
5. Start the VS Code debugger using the modified Python Attach configuration. VS Code should stop on your locally-set breakpoints, allowing you to step through the code, examine variables, and perform all other debugging actions. Expressions that you enter in the Debug Console are run on the remote computer as well.
6. Start the VS Code debugger using the modified Python Attach configuration. VS Code should stop on your locally-set breakpoints, allowing you to step through the code, examine variables, and perform all other debugging actions. Expressions that you enter in the Debug Console are run on the remote computer as well.
7. During remote debugging, the debugging toolbar appears. On this toolbar, the disconnect button (Shift+F5) stops the debugger and allows the remote program to run to completion.

Debugging a Python Add-in for MapInfo Pro

1. In VSCODE, open the folder of your python add-in or any python add-in sample (for example, .\SAMPLES\RIBBONINTERFACE\Python\HelloPython).
2. Compile the add-in *.mbx using MapBasic IDE or Notepad++, etc.
3. In your python add-in source code, add the following lines:

```
import ptvsd
# Allow other computers to attach to ptvsd default port.
ptvsd.enable_attach()
# Pause the program until a remote debugger is attached
ptvsd.wait_for_attach()
```

4. Run the add-in MBX in MapInfo Pro v17.0.3 or above. Now once MBX is run in MapInfo Pro, you will notice a wait cursor denoting that the process is waiting for some debugger to attach to it.
5. Switch to Debug View in VS Code, and select Python: Attach configuration.
6. Set a breakpoint in the code where you want to start debugging. Setting a single breakpoint on the statement immediately following the ptvsd.wait_for_attach() line may not work. Set at least one other breakpoint on another statement.
7. Start the VS Code debugger using the modified Python Attach configuration. VS Code should stop on your locally-set breakpoints, allowing you to step through the code, examine variables, and perform all other debugging actions. Expressions that you enter in the Debug Console are run on the remote computer as well.
8. During remote debugging, the debugging toolbar appears. On this toolbar, the disconnect button (Shift+F5) stops the debugger and allows the remote program to run to completion.

MapInfo Pro version 2019

Select By Location

The new **Select By Location** dialog lets you select objects from one table based on their location relative to objects in another table. In other words, it enables you to create a selection using a spatial join between two tables.

SQL Window

Use the new **SQL** window to build and execute SQL queries and statements. The SQL window is essentially a script editor wherein you can write and execute one or more SQL statements along with pre and post script processing snippets. The scripts are run using the *immediate* mode interpreted MapBasic and are not compiled.

Enhancements and Updates

MapInfo Pro version 2019.3

Enable/Disable Workspace Explorer Content

You can now close and open the content of the explorer window without actually closing the window container. This allows you to execute big tasks without affecting the performance and refreshing the explorer window with each table or map open.

```
set window workspaceexplorer disable
```

Invoking this command closes the content of the explorer window and disables any events being listened to. The Map/Layer object model will not work in case explorer window content is disabled.

```
set window workspaceexplorer enable
```

Invoking this command enables the disabled content of the explorer window. Any old reference to Map/Layer objects is disposed of, and new references need to be obtained to get the Map/Layer object model to work.

Note: In case the explorer is disabled, and you ask for the Maps collection, the explorer window is automatically enabled.

Converting Strings to Font, Pen, Brush, and Symbol

You can now directly assign a string-based style clause into a variable of type Font, Pen, Brush, or Symbol. For example,

```
dim fnt as Font
fnt = "Font (\""Arial\"",0,9,0)"
'this assignment will work now, and string-based font clause will convert
```

```
Font variable.
```

This works for Brush, Pen, and Symbol clauses and their respective variable types as well.

SQL Window Enhancements

Two new helper drop-downs have been added to the **Assist** pane in the **SQL** window.

Variables

You can now define one or more variables that can be used in your scripts. When a script is executed, you would be prompted to enter the values.

Help

There is a new **Help** drop-down available in the SQL Window:

1. Help on the SQL Window.
2. The MapBasic Reference - Launches the MapBasic desktop help if MapBasic is installed, or a PDF version, if not.
3. The MapBasic User Guide (PDF version).
4. The Extensibility Reference - Launches the Extensibility Reference desktop help.

SQL Query Improvement

You can now create joins between more than two dynamic query tables. Earlier, this was restricted to only joins between two dynamic query tables. For example,

```
select * from ehitis where ehitis.systeem_kood = "1106800020020" AND
ehitis.kood = "001" into query1
SELECT systeemi_eesvoolu_osa.id, systeemi_eesvoolu_osa.rajatis_id
FROM systeemi_eesvoolu_osa, rajatis, query1
WHERE systeemi_eesvoolu_osa.rajatis_id = rajatis.id
AND rajatis.ehitis_systeem_kood = query1.systeem_kood
```

Python Global Scope Changes

A new python global scope method is now available to attach a command URI to a button created in python script.

```
attach_command_uri(button, uri) ' Attaches a MapInfo Pro command uri to a
button.
' Parameters:
' button: Button control to which MapInfo Pro command uri will be attached.
' uri: Uri to process
```

Example:

```

from MapInfo.Types import IMapInfoPro, ControlType, PaperUnits,
DimensionedValue,
NotificationObject, NotificationType
from System import Uri, UriKind
tab = pro.Ribbon.Tabs.Add("Python")
group = tab.Groups.Add('pytab', 'Custom')
button = group.Controls.Add("ButtonEdit", "Edit Script", ControlType.Button)
button.IsLarge = True
button.LargeIcon =
Uri("pack://application:,,,/MapInfo.StyleResources;component/Images/Mapping/infoTool_32x32.png",
UriKind.RelativeOrAbsolute)
# attach command uri to the button
attach_command_uri(button,
"MapInfoPro://application:PythonConsole.mbx;edit/{0}".format(get_current_module_path()))

```

New Shortcut Keys

The following new shortcut keys are available for the Layer control buttons. Remember to keep the cursor focus inside the Layer control when moving a layer up or down.

Button	Short-cut Key
Add Layer	Shift+A
Remove Layer	Shift+Delete
Move Up	Alt+Up
Move Down	Alt+Down

New Coordinate Systems and Projections

New Bounded GDA2020 Projection Entries for Victoria, Australia.

"--- Victorian Bounded Projections GDA2020 ---"

"VIC VicGrid2020 (GDA2020) Bounded", 2003, 1028, 7, 145, -37, -36, -38, 2500000, 2500000, 1500000, 1500000, 3500000, 3500000

"VIC Zone 54 (GDA2020) Bounded", 2008, 1028, 7, 141, 0, 0.9996, 500000, 10000000, -500000, 5000000, 1500000, 7000000

"VIC Zone 55 (GDA2020) Bounded", 2008, 1028, 7, 147, 0, 0.9996, 500000, 10000000, -500000, 5000000, 1500000, 7000000

"VIC Longitude/Latitude (GDA2020) Bounded", 2001, 1028, 138, -44, 154.666666667, -27.333333333

"Longitude / Latitude (Victoria GDA2020) Bounded", 2001, 1028, 132, -26.8, 154, -43.8

New Russian Projection entries

"--- Russia Longitude/Latitude (GOST 32453-2017) ---"

"Longitude/Latitude (PZ-90)", 1, 9999, 57, -1.43, 0.05, 0.2, 0, 0, -0.13, -0.22, 0

"Longitude/Latitude (PZ-90.02)", 1, 9999, 57, -0.36, 0.08, 0.18, 0, 0, 0, 0, 0

"Longitude/Latitude (PZ-90.11)", 1, 9999, 57, 0.013, -0.106, -0.022, 0.0023, -0.00354, 0.00421, 0.008, 0

"Longitude/Latitude (SK-42)", 1, 9999, 3, 23.57, -140.95, -79.8, 0, -0.35, -0.79, -0.22, 0

"Longitude/Latitude (SK-95)", 1, 9999, 3, 24.47, -130.89, -81.56, 0, 0, -0.13, -0.22, 0

Updates to Set Next Document statement

You can now re-parent the **Legend**, **Statistics**, **Info**, **Ruler**, and **Message** windows using the **Set Next Document** statement. This restores backward compatible functionality with the 32-bit version of MapInfo Pro.

Updates to the End Program statement

While executing a MapBasic script in the interpreted mode (which does not support conditionals like the "if" statement), you need a way to stop the script. To achieve this, you can now use the **End Program** statement in the interpreted mode:

```
End Program [Condition(expr)] ' The Condition clause is optional
```

expr is a boolean (logical) expression. If **True**, the execution of the script text is stopped. If **False**, the execution continues.

Note: This does not apply to compiled MapBasic.

```
print "before"
fetch first from usa
End Program Condition(usa.State="AL")
print "after"
```

If the first state is AL, then the print "after" statement will not be executed.

Updates to the UnDim statement

If you **UnDim** a variable that does not exist, no error occurs now.

Note: This does not apply to compiled MapBasic.

SyncWindows Tool Enhancements

Restored support for:

- Embedding Theme Legend windows in mapper(s).
- Removal of embedded Theme Legend windows from mapper(s).

The **SyncWindows** tool will run in 2019.3 and later only (due to restoration of support for re-parenting Theme legend windows).

MapBasic Auto-complete Improvements

In the MapBasic window, auto-complete is now turned off for listing of parameters.

Now when typing in a method name and entering open parentheses, you will not be automatically prompted with function defines. To prompt, press **Ctrl+Space**.

Double quotes will still be auto-completed.

Addition to IMapInfoPro Interface

The IMapInfoPro interface has the following new additions:

1. OpenWorkpsace method
2. CloseAll method
3. EndMapInfo method
4. CurrentWorkspace property
5. Input() method

For more details, see the Extensibility Reference Help.

MapInfo Pro version 2019.1

Select By Location Dialog

New output options have been added to the **Select By Location** dialog:

- **Into table**

The **Into table** text box allows you to specify the name of the results table. By default, the results table is named **Selection**. If you use **Select By Location** frequently and use **Selection** as your results table name, you may eventually end up with a large number of left-over results tables (for example, Query1, Query2, Query3, etc.). If you want to avoid having a lot of open QueryN tables, enter an alternate table name in the **Into table** field. Each subsequent **Select By Location** operation will use that name as the results table name. That way, you only have to deal with one results table, regardless of how many queries you perform.

Note: The table name that you enter should not be the name of an open table.

- **Display results in new Browser window**

If the button is pressed MapInfo Pro displays the query results in a new Browser window.

- **Add results to topmost Map window**

If the button is pressed MapInfo Pro adds the results as a new layer to the top-most map window and brings that map to the front, or creates a new map window if none exist.

- **Add color override to results added to topmost Map Window split button**

When adding the results of the query to a map window, this option applies a color override with the selected color. Color override is off by default (shows "None" and a Paint Can icon). Click on the split button to open the pop-up to select different modes.

- First radio button turns off color override, the split button icon shows the Paint Can and "None".
- Second radio button switches to "Automatic" mode where colors are initialized to a Color Brewer qualitative color group (8 colors), the split button icon changes to show the next color with a gradient border.
- Third radio button switches to "Manual" mode using the color currently shown next to the third button, the split button icon changes to the selected color with no border. Or you can click directly on the color palette drop down to switch to "Manual" and open the color palette picker and select a different color. This color remains sticky and will not change unless the user selects a different color from the palette or switches to Automatic mode.

When you click directly on the color palette drop down, this opens the color palette picker. The "Automatic Colors" section shows all the colors that are used for automatic mode. Any of these colors can also be selected as the manual color.

- **Find results in Map or Browser windows**

If the button is pressed MapInfo Pro finds results in map and browser windows. In a map window, the view will zoom to the extents of the selection if "Zoom on Find Selection" Map Preference is checked (see the Editing tab in the Map Preferences dialog). The button is disabled when the NoSelect button is pressed.

- **Execute the query without changing the current selection (NoSelect)**

If the button is pressed MapInfo Pro performs the query without changing the pre-existing Selection table. Press this button if you need to perform a query, but you do not want to de-select whatever rows are already selected. The button is disabled when the output table name is "Selection". Change the output table name to something other than "Selection" to enable the button.

Installer

From version 2019.1 **FDODriver** and **CatalogBrowser** are not installed as default components. Use the `ADDLOCAL` parameters to enable the MapInfo Pro silent installer to install these components. Please note that the `ProgramFiles`, `Documentation` and `Tools` parameters are required for a proper MapInfo Pro installation. You can omit other components like `Python37`, `FME`, `FDODriver` and `CatalogBrowser` if you do not want to install them.

For details, see [Installing MapInfo Pro Silently](#) on page 74.

Quick Search

If you are connected to the internet, Software Data and Marketplace content is now searchable from the Quick Search tool.

Note: Toggle this option by right-clicking on the **Quick Search** tool in the **Tools** window and selecting **Search MapInfo Marketplace**.

Save/Restore Add-in Ribbon Controls on QAT

Using the `IRibbon.SaveRibbonState()` and `IRibbon.LoadRibbonState()` methods, you can save or restore a control added to the ribbon to the QAT. The add-in must add these two method calls and also set the `IsQAT` property to **True** to add a control to the QAT.

Here is the description of the two methods from the MapInfo Pro Extensibility Reference Guide:

```
/// <summary>
/// Load add-in's ribbon state.
/// </summary>
/// <param name="identifier">a unique name</param>
/// <param name="path">Add-in path</param>
/// <remarks>An Add-in can call this to load any items a user has added
/// to the Quick Access Toolbar.
/// Only ribbon items with the IsQAT property set to true can be added
/// to the QAT. Gallery controls cannot be added to the QAT.
/// A state file is loaded from the location specified by the path with
/// the name made up of the identifier followed by ".QatState.json"
/// </remarks>
/// <see cref="LoadRibbonState"/>
void LoadRibbonState(string identifier, string path);
```

```
/// <summary>
/// Save add-in's ribbon state.
/// </summary>
/// <param name="identifier">A unique name</param>
/// <param name="path">Add-in path</param>
/// <remarks>An Add-in can call this to save any items a user has added
/// to the Quick Access Toolbar.
/// Only ribbon items with the IsQAT property set to true can be added
/// to the QAT. Gallery controls cannot be added to the QAT.
/// A state file is created in the location specified by the path with
/// the name made up of the identifier followed by ".QatState.json"
/// </remarks>
/// <see cref="LoadRibbonState"/>
void SaveRibbonState(string identifier, string path);
```

Note: See the *ProSampleAddin* that comes with MapBasic for an example of how to do this.

Microsoft Office Access database 2016 engine support

MapInfo Pro now uses MS Office Access 2016 engine for support of MS Access (*.accdb*.mdb) files. MS Access 2.0 and earlier and MS Access 95/97 are not supported in this new engine. All MS Access files in these older versions must be upgraded to version 2000 or later using MS Access 2007 in order to use them with MapInfo Pro.

Trying to open an Access 95 or Access 2.0 database in MapInfo Pro version 2019.1 (using Access 2016 engine) would result in an error. Trying to save a Table to MS Access 95/97 in MapInfo Pro version 2019.1 would also fail.

Import Access 2.0 and Access 95 databases into current versions

Access for Office 365 Access 2019 Access 2016 Access 2013 Access 2010 Access 2007

If you use an existing database in the Access 2.0 format (.mdb), or the Access 95 format (.mdb) in some cases you can import tables, queries, and macros from that database into a more current version of Access that uses the .accdb file format.

To import forms and reports that contain VBA code (and also modules), first convert your Access 2.0 or Access 95 database to the Access 2002 - 2003 or Access 2000 file format by using Microsoft Office Access 2003, Access 2002, or Access 2000. You can then use a more current version such as Access 2007 to convert that file to the .accdb file format which can then be opened by Access 2010, Access 2013, or Access 2016.

If you are using Access 2010, Access 2013, or Access 2016

Access 2007 was the last version to support opening, importing, linking or exporting to Access 2.0 and Access 95. You will need Access 2007 in order to be able to export, import or link data from Access 1.0 and 2.0 (Red 2, or Jet 2) files and Access 95 databases. In this case, you'll have to use Access 2007 as an intermediary step to fully convert an Access 2.0 database to one of these versions.

For details, see

<https://support.microsoft.com/en-us/office/import-access-2-0-and-access-95-databases-into-current-versions-2e9d8851-101d-4407-a881-65d06bb12aa7?ui=en-us&rs=en-us&ad=us>

MapInfo Pro version 2019

Geocode Enhancements

Switch to classic view

The older **Geocode using Server** utility is visible only when the **Geocode** Tool is not loaded. When loaded, the Geocode tool takes precedence and replaces the older version. You can now revert to the legacy **Geocode using Server** Utility without unloading the Geocode tool. Click the new navigation button **Switch to classic view** at the bottom of the **Geocode** Tool to revert to the legacy **Geocode using Server** Utility.

Geocoding Credits

All MapInfo Pro license holders, Basic or Premium will get 2000 free credits a month now.

Pack Tool Enhancements

The name of the **Advance Pack** Tool has been changed to **Pack** now. Along with the name change, you can now access the old **Pack Table** functionality and the enhanced **Pack** Tool from the same interface. The **Pack Table** option from the Table list under the **TABLE** tab has been removed, and its functionality added with-in the **Pack** Tool interface.

A new drop-down list Mode is now available in the **Pack** dialog box. This drop-down contains two options:

1. Simple - Select to pack a table using the old **Pack Table** functionality.
2. Advanced - Select to pack a table using the enhanced **Pack** Tool.

About Tab Enhancements

On the **About** tab in Backstage, you now have easy to find information on current settings used by MapInfo Pro (such as the UILocale, AppLocale, and System Character set in use). The current Number and Date/Time formatting in effect is displayed as well.

At the top of the **About** section, there is an additional **Copy** control to allow for copying this information into an email and other documents for support information if needed.

Bing Maps Update

Bing Maps has a limited license period, which is specific to each version of MapInfo Pro. Bing Maps and the Add Bing Roads to Map and the Move Map to features cease working after this date:

- MapInfo Pro 2019 on January 1st, 2021
- MapInfo Pro 17.0 on January 1st, 2019
- MapInfo Pro 16.0 on January 1st, 2018
- MapInfo Pro 15.0 and 15.2 on January 1st, 2017

New Coordinate Systems and Projections

New Projection entries added to MapInfow.prj

New Polar Stereographic entries added for use in the Universal Polar Stereographic projection category. These systems can apply either a standard parallel OR scale factor at the pole.

MapInfo MapCad Update

The following updates are available in the latest version of MapCad available with MapInfo Pro 2019.

Rollback Editable Layer

A new tool designed specifically for the SPATIAL Tab that does a Rollback of all unsaved changes in the editable layer (for example, Revert table). If multiple edits are made in the editable layer using

various spatial tools, then users may need to reset this editable layer frequently. It is much easier to revert changes to layer without leaving the Spatial tab.

Rotate Text

A new tool to rotate all selected text objects using an additional or absolute angle. This function is also available using the MapBasic Exec()command.

Triangle

A new tool to create triangular regions. It works the same way as creating rectangles and is often needed to create triangular signatures or arrows. Equilateral and isosceles triangles can also be generated.

Cell (Insert)

This tool has been modified in this release. It is used to insert a new Cell to a map. There were no changes in its usage, except for interaction with the cell styles. A new icon is added to make the function more easily recognizable.

Cell Style

This tool is used to specify the current cell and manage the cell library. It was modified significantly to make the dialog easier to use. The cell list can now be sorted by name and ID, and the preview window can be turned off to improve performance when many, and or large, tables are currently open in MapInfo Pro. The use of Cartesian projections has been improved, and in addition, NonEarth coordinate systems are supported. For Cartesian projections, an automatic re-projection between the projection of the library and the layer now takes place. The former option Apply cell ID in the table column is removed due to errors with the DoubleClickMouseHandler, and it was no longer useful. A new icon is added to make the function more easily recognizable.

Cell From Table

This new tool creates cell objects using Coordinates from table columns. This tool works much like the standard tool create points; however, instead of points, cells are created.

Flanked Cells

This new tool creates an accompanying signature to the line and region objects. These flanked cells nestle against the edges.

Ruler

This is the existing standard Ruler tool from the MAP tab placed in the Spatial tab as its frequently needed from this tab when editing objects.

Disaggregate To Lines

This tool has been removed. The functionality is already included in the Convert To Lines tool.

Syntax Highlighting Style for Query Builder

Syntax highlighting in Query builder can be modified from MapInfoProStyle.xml, located in styles folder in the installation directory.

These are the colors for query builder syntax highlighting.

```
<SyntaxHighlightingStyle>
  <Style Key="KeywordSyntaxHighlightBrush" Value="Blue"/>
  <Style Key="OperatorSyntaxHighlightBrush" Value="Gray"/>
  <Style Key="LiteralSyntaxHighlightBrush" Value="IndianRed"/>
  <Style Key="DefineSyntaxHighlightBrush" Value="Red"/>
  <Style Key="FunctionSyntaxHighlightBrush" Value="Orange"/>
</SyntaxHighlightingStyle>
```

Note: Set `OverrideDefaultStyle` value to true, so that the colors updated in this file get applied.

```
<Setting Key="OverrideDefaultStyle" Value="true"/>
```

Known Issues

Mapper Window with an Embedded Theme Legend

Mapper windows containing an embedded theme legend cannot be added to a Layout designer map frame. The Legend needs to be in its own frame within the Layout Designer window.

Progress and Resolution of Outstanding Issues

MapInfo Pro 2019.3

Issue Number	Description and Resolution
MIPRO-90421	IndividualThemeItemInfo() and IndividualThemeInfo() MapBasic commands ignore 'All Others' bin item causing an issue with custom order. Resolution: Fixed.
MIPRO-118362	Run Menu command with MB_TOOL_HANDLER not working with multiple add-ins using it. Resolution: Fixed.
MIPRO-118327	Legacy Create Frame for a map window is not including themes based on queries. Resolution: Fixed.
MIPRO-118281	Create ButtonPad statement uses the wrong title for the last button. Resolution: Fixed.

Issue Number	Description and Resolution
MIPRO-117695	While using the WMS service, the request URL appends the user-name and password fields twice. Resolution: Fixed.
MIPRO-117232	A newly created table is not shown as mappable in Table List and in the SQL Window. Resolution: Fixed.
MIPRO-117204	In Quick Access Toolbar, rearranging menu items is not working correctly. Resolution: Fixed.
MIPRO-117203	Looping through a Native table is a lot slower than looping through a NativeX table. Resolution: Fixed.
MIPRO-117076	Window Manager tool window menus have incorrect menu text. Resolution: Fixed.
MIPRO-117055	Certain queries that worked in MapInfo Pro version 17.0 fails in MapInfo Pro version 2019. Resolution: Fixed.
MIPRO-116931	Buttons are disabled in a custom Ribbon and quick access tool bar. Resolution: Fixed.
MIPRO-116910	MBX is being called twice in certain cases. Resolution: Fixed.
MIPRO-116864	Printing to PDF produces a blank PDF file. Resolution: Fixed.
MIPRO-116863	WFS server refresh is not working from the Table List window. Resolution: Fixed.
MIPRO-116861	When using a GeoPackage table, selecting browser columns is not working in certain cases. Resolution: Fixed.

Issue Number	Description and Resolution
MIPRO-116736	For the French locale, the labels and text fields are not correctly aligned in the Find Address window. Resolution: Fixed.
MIPRO-116712	A Legend is not showing correct fill patterns while saving to JPEG files from the layout window. Resolution: Fixed.
MIPRO-116708	MapInfo Pro WMS Service is not working with EPSG 3883. Resolution: Fixed.
MIPRO-116553	Internal error 34623 is reported. Resolution: Fixed.
MIPRO-116478	DriveTime with Spectrum 19.1 results in an Invalid POST payload error. Resolution: Fixed.
MIPRO-116472	Certain GeoPackage files not opening in MapInfo Pro. Resolution: Fixed.
MIPRO-116357	COM Visible property does not work. Resolution: Fixed.
MIPRO-116350	Adding (copy-paste) the contents of the message screen to the layout screen is not working. Resolution: Fixed.
MIPRO-116236	Formatting changes to Legend in a workspace are not preserved. Resolution: Fixed.
MIPRO-116140	Browse is not enabled on the Layer Control Window when the result layer is the topmost Map Window. Resolution: Fixed.
MIPRO-116023	Performance issues with certain SQL queries.. Resolution: Fixed.

Issue Number	Description and Resolution
MIPRO-115991	<p>GDA2020 coordsys string not being written to MapCatalog after Easyloader upload.</p> <p>Resolution: Fixed. EasyLoader now uses the latest EFAL and MFAL updates to support this GDA 2020 CoordSys.</p>
MIPRO-115969	<p>CartesianOffsetXY only allows offsetting up to 100000 meters.</p> <p>Resolution: Fixed.</p>
MIPRO-115968	<p>Window focus shifts to the explorer window while enabling/disabling layers.</p> <p>Resolution: Fixed.</p>
MIPRO-115947	<p>MapInfo Pro closes unexpectedly while executing a selection export the second time.</p> <p>Resolution: Fixed.</p>
MIPRO-115917	<p>While creating a TAB file from a CSV file with UTF-8-BOM encoding, the BOM bytes are included as part of the first field name.</p> <p>Resolution: Fixed.</p>
MIPRO-115415	<p>Custom .pen and .fnt files are not working correctly in the Legend Designer window.</p> <p>Resolution: Fixed.</p>
MIPRO-111655	<p>Error refreshing linked tables when connecting to linked tables using an ODBC connection.</p> <p>Resolution: Fixed.</p>
MIPRO-110823	<p>Using the GetCurrentWorkspacePath\$() command to retrieve the path of a workspace saved using MapBasic returns the path to the previous workspace.</p> <p>Resolution: Fixed.</p>
MIPRO-109378	<p>MapInfo Pro closes unexpectedly while customizing a Mapper MiniToolbar in an Integrated WinForms Application.</p> <p>Resolution: Fixed.</p>
MIPRO-103554	<p>Polyline splitting error with the MapInfo extended format.</p> <p>Resolution: Fixed.</p>

Issue Number	Description and Resolution
MIPRO-101592	While creating an Oracle workspace, certain symbols appear at the end of the .tab extension preventing the file from being saved. Resolution: Fixed.

MapInfo Pro 2019.2

Issue Number	Description and Resolution
MIPRO-116167	At times MapInfo Pro fails to launch and displays a license error. Resolution: Fixed.
MIPRO-116160	MapInfo Pro closes unexpectedly while using the FileSaveAsDlg() and FileOpenDlg() commands with long captions. Resolution: Fixed.
MIPRO-116023	SQL queries containing cross-joins take a long time to execute. Resolution: Fixed.
MIPRO-115917	While creating a TAB file from a CSV file with UTF-8-BOM encoding, the BOM bytes are included as part of the first field name. Resolution: Fixed.
MIPRO-115892	While executing the Run Menu command, the command code M_TOOLS_MAPBASIC is not working. Resolution: Fixed.

MapInfo Pro 2019.1

Issue Number	Description and Resolution
MIPRO-115753	MapInfo Pro closes unexpectedly while appending tables. Resolution: Fixed.

Issue Number	Description and Resolution
MIPRO-115609	Memory leak when processing objects. Resolution: Fixed.
MIPRO-115343	While using the Alter Menu command, you cannot remove an item from the context menu. Resolution: Fixed.
MIPRO-114905	While using the Update command in the SQL window, the Column drop-down is greyed out if the table name is long. Resolution: Fixed.
MIPRO-114556	In MS Excel, certain columns are erroneously formatted as Character instead of Float. Resolution: Fixed.
MIPRO-114380	MapInfo Pro closes unexpectedly while calling Create ButtonPad statement using OLE in the MapBasic window. Resolution: Fixed.
MIPRO-113983	To set the text style from LAYOUT tab text controls, a text frame is required in the Layout window. Resolution: Fixed. Now you can set text styles even without creating a text frame in the Layout window.
MIPRO-113908	MapInfo Pro closes unexpectedly while editing attributes of joined tables. Resolution: Fixed.
MIPRO-113895	The TIN Interpolator is creating incorrect results. Resolution: Fixed.
MIPRO-113346	While using the Geocode Add-in, partial geocoded data is lost in case of exceptions like insufficient credit, network disconnection, etc. Resolution: Fixed.
MIPRO-112786	Data is not correctly updated with Update column using a temporary query table. Resolution: Fixed.

Issue Number	Description and Resolution
MIPRO-110832	MapInfo Pro closes unexpectedly while creating a object type variable in immediate mode MapBasic executed in a custom MapInfo Pro Integrated Mapping application. Resolution: Fixed.
MIPRO-110703	In MS Excel, you cannot open XLSX files as NativeX format. Resolution: Fixed.
MIPRO-105177	Australian Datum MGA 2020 is not being used correctly by FME Quick Translator when creating NATIVE MapInfo tables. Resolution: Fixed.

MapInfo Pro 2019

Issue Number	Description and Resolution
MIPRO-112221	Using the animation layer in a map window with Enhanced Rendering turned on results in a memory leak. Resolution: Fixed.
MIPRO-111594	Incorrect information in log file while geocoding using the PB server. Resolution: Fixed.
MIPRO-111593	When geocoding using server, the result codes are not updated correctly. Resolution: Fixed.
MIPRO-111482	MapInfo Pro User Agent HTTP Requests do not conform to HTTP 1.1 standards. Resolution: Fixed.
MIPRO-111038	Opening a workspace having an older version of the MapInfo PDF Printer (that is no longer installed) assigns the default printer to the workspace. Resolution: Fixed.

Issue Number	Description and Resolution
MIPRO-110998	<p>If you use the Info Tool with two overlapping objects, each with a percent symbol in a column, the percent symbol does not display correctly.</p> <p>Resolution: Fixed.</p>
MIPRO-110592	<p>Closing MapInfo Pro when the Workspace Packager tool is set to AutoLoad displays an error.</p> <p>Resolution: Fixed.</p>
MIPRO-99997	<p>Legend Designer frames with Pie Charts in a Layout Designer window are pixelated</p> <p>Resolution: Fixed.</p>
MIPRO-96573	<p>Using Open Universal Data with ESRI Geodatabase results in incorrect columnar data.</p> <p>Resolution: Fixed.</p>
MIPRO-94617	<p>TIN Interpolator memory leak.</p> <p>Resolution: Fixed.</p>
MIPRO-92843	<p>Performance issues while drawing curved labels when there is an overlap between characters within a label.</p> <p>Resolution: Fixed.</p>
MIPRO-85296	<p>When placing a map and legend on a layout window to print as a PDF, the legend is not displayed correctly.</p> <p>Resolution: Fixed.</p>
MIPRO-80249	<p>If you save a workspace containing a thematic legend (with the theme unchecked in the Layer control) and then re-open it, the legend does not display.</p> <p>Resolution: Fixed.</p>
MIPRO-17074	<p>When opening a PNG raster file, the translucent images do not show up right with the correct background.</p> <p>Resolution: Fixed.</p>

MapInfo Pro Advanced

What's New in MapInfo Pro Advanced

The following sections provide a brief description of important features and enhancements made to MapInfo Pro Advanced.

New Features

MapInfo Pro Advanced version 2019.3

LiDAR Tools

In this release of MapInfo Pro Advanced, we have added two new LiDAR-based gridding and analysis tools to generate rasters from LIDAR data.

Surface Analysis

The **Surface Analysis** LiDAR tool is somewhat similar to the standard **Triangulation** tool available via the **Create Raster** Gallery in the **Interpolate** group. This tool provides limited triangulation capabilities to a standard MapInfo Pro user. The difference between the two is that the **Surface Analysis** tool uses only LiDAR data loaded from the LAS or LAZ format files. This makes the task of triangulation simple, involving the following steps:

1. Select the LAS/LAZ files.
2. Detect or specify the coordinate system.
3. Select the elevation surface.
4. Validate the maximum triangle side length and the raster cell size.
5. Select the clipping options.
6. Specify the output raster file name.
7. Execute operation and display the MRR.

The maximum triangle size, polygon clipping options, and cell size are the only additional options available in the **Surface Analysis** tool.

Note: MapInfo Pro standard users cannot post-process the raster using MapInfo Pro. So the coordinate system needs to be specified, and options like clipping must be available.

Surface Analysis command

What is the purpose of Surface Analysis

This tool provides limited triangulation capabilities to a standard MapInfo Pro user.

How do I access Surface Analysis

The **Surface Analysis** tool can be accessed in the following ways:

- On the **MAP** tab, click on **LiDAR Surface** in the **Content** group to open the **Surface Analysis** dialog.
- On the **RASTER** tab, in the **Interpolate** group, open the **LiDAR Tools** gallery and click on **Surface** to open the **Surface Analysis** dialog.

Exploring the Surface Analysis Dialog Box

Input File

Click **Select Files** to open the **Select Input File** panel. Click **Add Files** to browse and select one or more input data files. You can also use the **Add Folder** option to add all files in a directory.

Interpolation Method

This option allows you to choose between the **Surface Analysis** tool and another LiDAR gridding and analysis tool called the **Tree Canopy Analysis**.

Method Options

The **Method Options** vary according to the selected Interpolation Method. For the Surface Analysis tool, the following method options are available:

Select Standard Surface

Standard surface settings for creating elevation based modeling. The available options are:

1. Ground - Filter for ground surface returns.
2. Building - Filter for building surface returns.
3. Roads - Filter for road surface returns.
4. Water - Filter for water surface returns.
5. Vegetation - Filter for vegetation surface returns.
6. Low Vegetation - Filter for low vegetation surface returns.
7. Medium Vegetation - Filter for medium vegetation surface returns.
8. High Vegetation - Filter for high vegetation surface returns.
9. Built Environment - Filter for built environment surface returns.
10. Hard Surface - Filter for hard surface returns.
11. Top Surface - Filter for the first return representing the top surface.
12. Bottom surface - Filter for the last return representing the bottom surface.

LiDAR Filter settings are chosen to select the last return, representing the deepest penetration of the laser at all points.

Last return.

13. Not Ground - Filter for all first returns, excluding ground classified returns.
14. Not Vegetation - Filter for all last returns, excluding those classified as vegetation.

More Options

The **More Options** button appears depending on the interpolation method selected. Expand **More Options** to set the advanced parameters for the selected interpolation method.

Maximum Triangle Size

The maximum side length of a triangle. Use this to limit the size of the triangles created.

Clipping

Clipping options to constrain interpolation distance around source data points.

Cell Size

Specify the cell size for the output raster in the **Cell Size** box. The cell size defines the width and height of a raster cell in distance units. If the raster cells are square, both width and height are specified with the same value.

Output

Output File

Specify a name, location, and format for the output raster that you are creating. In the **Output File** box, enter the name you want to specify for your output file.

Tree Canopy Analysis

The Tree Canopy tool provides you with three different methods for Canopy cover analysis:

Canopy Coverage

Canopy Coverage is the percentage area of each raster cell that is covered by tree canopy. Simple statistics on an area will tell you what the total percentage and area of tree canopy coverage is. It could also be used as a measurement of the percentage of a cell that receives direct sunlight.

Canopy Density

Tree Canopy Density is a proxy measure of the thickness of the tree canopy within the area of a raster cell. The measurement of how difficult it is for the LiDAR Laser shots to penetrate through the canopy to the ground is a proxy for the canopy density. Each raster cell will be populated with a value from 0 to 1, representing the density of the canopy over the area of that cell.

The above methods work by counting both ground classified and vegetation classified returns that are within each raster cell (or within a defined radius of the center of each cell) and returning the ratio of vegetation returns to the total number of ground and vegetation returns. They differ in the filtering rules that are used to select the returns from the LiDAR survey for processing.

Canopy Height

The Canopy Height method uses classified LiDAR survey data to generate raster output of the height of the tree canopy across the survey area.

Input Filtering

All the three methods above apply filtering rules on the fly when they read the LAS or LAZ files to select certain returns from the total number of returns within the file. Thus the LAS or LAZ data is filtered to include only those returns that are relevant to the processing. These methods largely control this filtering, although you can apply Scan Angle, Z range, and Intensity range filters if desired.

Note: Returns flagged “withheld” are not included. Returns flagged “synthetic” or “key point” or “overlap” will be included (subject to the other filtering rules).

LiDAR Classification

The spatial location of each return, the “Classification” and the “Z” data are used in the processing. If you have version 1.4 LAS files (or later), you can elect to use the “Extended Classification” data in preference. If your LiDAR data has not been classified, then it will not be suitable for use in this tool. As a minimum requirement, the data must have either ground returns or vegetation returns appropriately classified.

You must provide a list of classification codes, which can be either industry-standard codes or extended codes of your choice, for everything that represents “Ground” and everything that you consider represents “Vegetation”. The best quality result will be obtained if you provide a list of codes for both ground and vegetation. For example, you might supply “Ground” and “Water” as ground classifications and “High vegetation” as the vegetation classification.

If you have both ground and vegetation classified returns, then use the “Ground and Vegetation” mode to allow you to enter lists of both. If you only have ground returns classified, then you can use the “Ground and Not Ground” mode. If you only have vegetation returns classified, then you can use the “Not Vegetation and Vegetation” mode.

Integration Area

The number of returns must be integrated over some areas. By default, we do this over the area of a single raster cell. In this scenario, you ought to choose a cell size that is large enough to ensure that there are a sufficient number of hits in each cell to provide statistical significance. This produces the highest resolution result, which is likely to be best for statistical analysis but can be quite variable and difficult to visualize.

For the Tree Canopy Coverage and Tree Canopy Density methods, you can specify an **Integration Radius**. All returns that are within the **Integration Radius** of the center of the cell will be counted. In this mode, you can also weigh the contributions by distance from the center of the cell using a Quartic weighting function. This is recommended.

Note: The radial method is significantly slower than the simple cell integration method and may take a considerably longer time to execute. To minimize the expected run time, keep the radius of integration as small as possible.

Output raster

The tool will create an output raster containing a floating point band in MRR format. The band will record the Tree Canopy Height above ground.

You can also populate cells that contain no processed LiDAR vegetation returns with zero. Otherwise, these raster cells will be empty by default. If you populate these empty cells, you will usually use some clipping rectangle or clipping polygon to restrict the spatial extent of the output raster.

Tree Canopy Analysis command

What is the purpose of Tree Canopy Analysis

This tool provides limited triangulation capabilities to a standard MapInfo Pro user.

How do I access Surface Analysis

The **Surface Analysis** tool can be accessed in the following ways:

- On the **RASTER** tab, in the **Interpolate** group, open the **LiDAR Tools** gallery and click on **Tree Canopy Analysis** to open the **LiDAR Analysis** dialog with the **Tree Canopy Analysis** method selected.

Exploring the Tree Canopy Analysis Dialog Box

Input File

Click **Select Files** to open the **Select Input File** panel. Click **Add Files** to browse and select one or more input data files. You can also use the **Add Folder** option to add all files in a directory.

Note: This tool will only consume LiDAR data in LAS or LAZ format. It will not read data in other formats like TAB or ASCII. You can select one or more files of either type from one or more directories as input.

Interpolation Method

This option allows you to switch between the two available LiDAR gridding and analysis tools.

Method Options

The **Method Options** vary according to the selected Interpolation Method. For the Surface Analysis tool, the following method options are available:

Tree Canopy Methods

Select between Canopy Coverage, Canopy Density, and Canopy Height.

Class Rules

Determine which classification codes will be supplied to differentiate ground from vegetation. Available options are **Ground**, **Vegetation**, and **Ground and Vegetation**.

Ground Class

Classification codes that represent the ground surface.

Vegetation Class

Classification codes that represent the vegetation canopy surface.

The following options are available for both Ground and Vegetation Classes:

- 0 Created, never classified.
- 1 Unclassified.
- 2 Ground.
- 3 Low Vegetation.
- 4 Medium Vegetation.
- 5 High Vegetation.
- 6 Building
- 7 Low Point (Low noise)
- 8 Model Key/Reserved
- 9 Water
- 10 Rail
- 11 Road Surface
- 12 Bridge Deck
- 13 Wire - Guard (Shield)
- 14 Wire - Conductor (Phase)
- 15 Transmission Tower
- 16 Wire-structure Connector (Insulator)
- 17 Bridge
- 18 High Point (High noise)

Extended Classification

If you have version 1.4 LAS files (or later), check the **Extended Classification** channel to use it instead of the standard classification channel.

Make Empty Zero

Check to record empty cells with zero values rather than an invalid cell.

Patch Size (available only with the Tree Canopy Height method)

Use a large patch size to improve performance and a small patch size to minimize memory requirements.

Integration (available only with the Tree Canopy Coverage and Density methods)

Over Cell - Do this over the area of a single raster cell. In this scenario, you ought to choose a cell size that is large enough to ensure that there are a sufficient number of hits in each cell to provide statistical significance.

Over Radius - Alternatively, you can specify an **Integration Radius**.

Smooth Kernel (available only with the Tree Canopy Coverage and Density methods)

Apply a smooth quartic kernel weighting to all samples within the integration radius.

Clipping

Clipping options to constrain interpolation distance around source data points.

Raster Geometry

Cell Size

Specify the cell size for the output raster in the **Cell Size** box.

Output Geometry

Define output raster origin co-ordinate and dimensions.

Output

Output File

Specify a name, location, and format for the output raster that you are creating. In the **Output File** box, enter the name you want to specify for your output file.

Display Output File

Select to display the output raster in the window.

Compression Settings

Choose Compression settings for optimization of Speed, Storage, or Balanced.

MapInfo Pro Advanced version 2019

Heatmap - Analyzing Point Data

In this release of MapInfo Pro Advanced, we have added a new analysis tool for quickly creating Heatmaps using the point data.

Heatmaps are used to visualize distributions of point data and to show locations where a higher frequency of activity occurs. Heatmaps help you quickly identify such areas – for example, a higher than the average number of criminal activities or traffic accidents.


MapInfo Pro Advanced supports the following types of Heatmap.

1. Estimate
2. Weighted Estimate
3. Sample Count
4. Sample Density
5. Advanced


The first four methods use streamlined and simple workflows to generate standard Heatmap products that are suitable for most applications. The fifth method (Advanced), provides you with access to all the advanced features of the Heatmap tool.

How to access the Heatmap tool

The Heatmap tool can be accessed in the following ways:

- On the **RASTER** tab, in the **Interpolate** group, click the **Create Raster**  button to open the gallery. Select a method from **Heatmap** category.

Accessing Heatmap tool from the MapInfo Pro Ribbon

- A **Heatmap** tool has been added on the MapInfo Pro ribbon also. On the **Map** tab, in the **Content** group, click **Heatmap** , to open the **Heatmap** dialog box. This dialog provides predefined styles for the Heatmap.
- Right-click on the **Layer Control** window and click **Heatmap**.

Heatmap Style	Description
Standard	Use this style for smooth Heatmap with medium resolution and detail.
Detailed	Use this style for smooth Heatmap with high resolution and detail.
Smooth	Use this style for smooth Heatmap with low resolution and detail.
Optimal	Use this style for smooth Heatmap with high resolution, detail, and broad coverage. Optimal produces the best image because it combines high resolution with total coverage.

For more information see, **What's New** chapter of MapInfo Pro Advanced Help.

Enhancements and Updates in MapInfo Pro Advanced

This section summarizes enhancements made to the existing raster features.

MapInfo Pro Advanced version 2019.3

Data Conditioning Invalid Control:

The control can now:

1. Distinguish between range separator and negative value.
2. Add validation for invalid values. This will only add valid values keeping the invalid values in the input box and highlighting the invalid values with a red border.
3. Show help text on hover.

Stamp Tool enhancements

In the Stamp tool, the following two new stamping methods have been added:

1. Average
2. Count

Create Raster tools enhancements

All **Create Raster** tools that use the LAS/LAZ files as input are now able to read projection information if available in the selected LAS/LAZ files and select it in the projection dialogue automatically.

Limitation: CRS, if stored as a WKT string inside LAS/LAZ file (Most common for LAS/LAZ files on version 1.4 specification), cannot be read.

Batch Mode enhancements

Batch mode operation now allows single field single band conversion for supported files.

Statistics tool enhancements

The Statistics tool now supports:

1. Boundary polygon export to TAB.
2. ImagePalette and Classified histograms.

Raster Info tool enhancements

The Raster Info tool now reports the following additional properties:

- Raster
 1. XMLMetaData
 2. TileSize
- Field
 1. XMLMetaData
 2. Overview Cell Coverage
 3. Valid flag per band
- Band
 1. XMLMetaData
 2. BandValueType
 3. EncodingType
 4. Scale and Offset
 5. Clip Min and Max

Contour tool enhancements

You can now specify the resolution at which you need to run contour operation.

Cell Value tool enhancements

The Cell Value tool now supports drilling down into MRD and MVR source rasters via an XML report.

Enhancement in the ECW file driver

ECW native driver now has multi-banded support.

Note: When you open an ECW file with more than four bands, it will, by default, display as an imagery file. But it will also show one more field which has all the bands as continuous.

MapInfo Pro Advanced version 2019.1

Enhancements in the Heatmap tool

Ability to create a Heatmap on the Selected Data


A simplified **Heatmap** tool can now be accessed from the **Map** tab on the MapInfo Pro ribbon. This new tool is provided in addition to the existing **Heatmap** operations accessed via the **Create Raster** button on the **Raster** tab.

The simplified Heatmap tool uses the same methodology available in the **Estimate** and **Weighted Estimate** methods on the **Raster** tab. By default, it will compute a smooth heatmap that reflects the density of the input point data locations. In addition, you can now select a data field from the input

dataset which is used calculate the estimation. In the default case, each data point represents a single sample (or event) at that location. When a weighting field is supplied, the value of this field is interpreted to represent the number of samples at that location. This sample count is used to weight the heatmap calculations. The result is a heatmap that is biased towards points that have a high sample count.

On the **Heatmap** dialog box, we have added the **Select Column** option to enable you to specify the column from the input .TAB file. By default, **None** is selected. If you want to create a simple Heatmap based on density estimation, use **None** in the **Select Column** drop-down list.

Accessing Heatmap tool from the MapInfo Pro Ribbon

- On the **Map** tab, in the **Content** group, click **Heatmap** , to open the **Heatmap** dialog box. This dialog provides predefined styles for the Heatmap.
- Right-Click on the **Layer Control** window and click **Heatmap**.

Clipping in the Heatmap tool

Heatmaps are used to visualize distributions of point data and to show locations where a higher frequency of activity occurs. For example, a higher than the average number of criminal activities or traffic accidents.

We have added an option on the **Heatmap** dialog box for **Clipping**. The clipping is applied to the output raster after the heatmap has been computed. For example, it can be used to clip to a country or municipal boundary to prevent extrapolation outside of the area of interest. The area to be clipped can be specified either by:

- drawing a polygon on the map.
- specifying a .TAB file containing a single or multiple polygon objects.

On the **RASTER** tab, in the **Interpolate** group, click the **Create Raster** button to open the gallery. Select a method from the **Heatmap** category. The **Heatmap** dialog opens. The **Clipping** option has been added under the **Method Options**.

How Clipping works

Clipping has the following options.

None - Denotes no clipping, and nothing is selected on the map.

Polygon - Click this option to either select a TAB file containing a single or multiple polygon objects, or draw a polygon on the map window. When you draw a polygon on the map, MapInfo Pro creates a temporary new selection table. The **Selection** appears in the clipping drop-down list and it contains the area that falls within the drawn polygon.

When you process, the raster is clipped to fit the shape of the polygon and saved as an MRR file.

Area of influence

We have improved the way you specify the area of influence. The **Area of Influence** slider controls the relative size of the radius. By default, the scale on the slider is centered on the value of zero. It changes to the negative integers as it moves to the left and positive integers as it moves to the right. You can set the slider between 0 to 10.

If you want to make the color transition smoother, move the slider to the right (plus values), and if you are going to make the color transition less smooth and more granular, move it to the left (minus values).

User Interface Improvements

This section summarizes features and enhancements made to the **Create Raster** tool.

Updates in the Calculator Tool

We have added more functions to the Calculator tool to support bitwise operation. These functions can be applied to unsigned integers. Any values supplied to these functions is converted to unsigned integers before applying the function.

How to access these functions

On the **RASTER** tab, in the **Operations** group, click **Raster Operations** and then click the **Calculator** button to open the **Calculator** dialog box. Under the **Functions** drop-down list, scroll to **Bitwise Operations** to access these functions.

Function	Meaning
bitand(x,y)	Bitwise “and” operation on x and y
bitor(x,y)	Bitwise “or” operation on x and y
bitxor(x,y)	Bitwise “exclusive or” operation on x and y
bitnot8(x)	Bitwise “not” operation on x cast to 8 bit unsigned integer
bitnot16(x)	Bitwise “not” operation on x cast to 16 bit unsigned integer
bitnot32(x)	Bitwise “not” operation on x cast to 32 bit unsigned integer
bitnot64(x)	Bitwise “not” operation on x cast to 64 bit unsigned integer
bitshl(x,y)	Bitwise “shift left” operation on x, by y bits

Function	Meaning
bitshr(x,y)	Bitwise “shift right” operation on x, by y bits

Enhancements in the Contour Dialog Box

We have added **Match Colors to Input** checkbox, on the main panel of the **Contour** dialog box. This checkbox is used for coloring the contour lines/polygons according to the color scheme of the input raster.

MapInfo Pro Advanced version 2019

User Interface Improvements

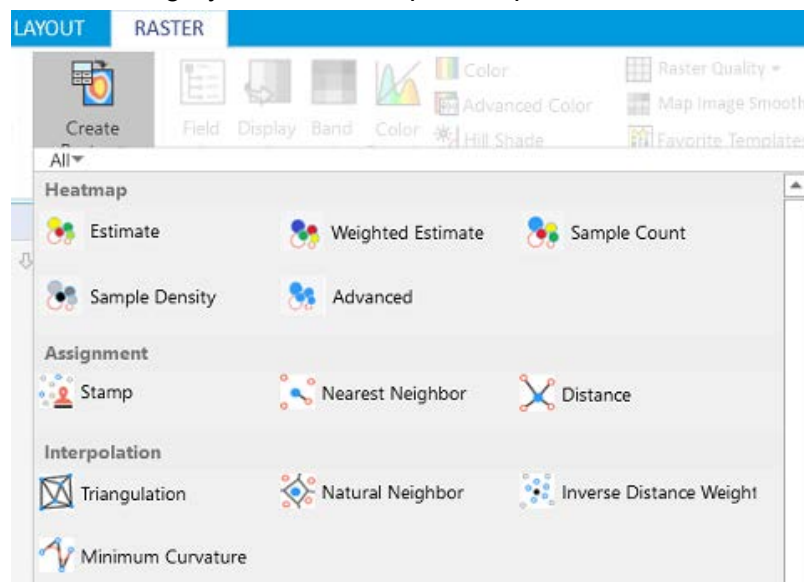
This section summarizes features and enhancements made to the **Create Raster** tool.

Changes in the Raster Ribbon

Previously, there were two separate tools for interpolation **Create Raster** and **Create Raster Multifile** on the ribbon under the **Raster** tab. They are now merged into one tool. Now, you can use the **Create Raster** dialog, to create a raster using single or multiple files.

Categories for Interpolation Methods

In this release, we have reorganized the supported interpolation methods and grouped them into logical categories. There are categories for **Heatmap**, **Assignment**, and **Interpolation** under the **Create Raster** tool. Each category contains multiple interpolation methods, as shown below.



Note: When you select any method from a category, the dialog box title displays the category name instead of the chosen method name. For example, if you select the **Weighted Estimate** method from the **Heatmap** category, the dialog box title displays Heatmap, not **Weighted Estimate**.

A simplified **Heatmap** tool is also available under the **Map** tab on the ribbon.

Input File Selection Panel for Creating a Raster

We have improved the input files selection process for all interpolation methods. Now, you get a **Select Input Files** panel to select the input source files. For example, on the **Heatmap** dialog, click **Select Files**, to open the **Select Input Files** panel. You can then choose single or multiple input files using this panel.

The **Select Input Files** panel has the following options:

- **Group(1)** - Click **Select Rasters** drop-down list to choose files which are currently open in MapInfo Pro.
- **Add Files** - Click this to browse and add one or more input source files.
- **Add Folders** - Click this to browse and select all files with specific file format from a folder.
- **Properties for Group** - Click this to set properties for the selected files and folders.
- **Preview** the selected input source files.
- Add a new group or remove a group

The **Select Input Files** panel enables you to group input files based on the file formats. Adding files in a group helps in settings properties for all files. Instead of setting properties for individual files, you can set properties for a group. The properties of a group get applied to all files in the group

Enhancements in the Create Raster Dialog Box - Interpolate Along Edges

We have added **Interpolate Along Edges** check box in the **Create Raster** dialog box.

Enhancements in the SDK

We have added the following APIs to create a raster from LiDAR data.

- `InterpolateLiDARAnalysis_TreeCanopyCoverage()`- This method interpolates LiDAR data to find Tree Canopy Coverage.
- `InterpolateLiDARAnalysis_TreeCanopyDensity()`- This method interpolates LiDAR data to find Tree Canopy Density.
- `InterpolateLiDARAnalysis_TreeCanopyHeight()` - This method interpolates LiDAR data to find Tree Canopy Height.

For more information, see the API reference guide `MapInfoProRaster.NET_SDK.chm`, which is installed to `C:\Program Files\MapInfo\Professional\Raster\Documentation\API`.

Known Issues

MapInfo Pro Advanced currently has the following limitations and expected behavior:

- MapInfo Pro Advanced allows you to select one or more files or a folder as input source files. When selecting a .TAB folder as an input source for interpolation, if the selected .TAB folder contains raster referenced .TAB file MapInfo may not give the expected result. To avoid this, please ensure that the selected folder contains only vector .TAB files.
- MapInfo Pro Advanced allows you to select one or more files or a folder as an input source for gridding. When selecting a .TAB folder as an input source, if the selected .TAB folder contains a raster referenced .TAB file, the gridding operation may not give the expected result. It happens only when you choose a .TAB folder, because the gridding operation does not distinguish between a raster and a vector .TAB files.

To avoid this, please ensure that the selected .TAB folder contains only vector .TAB files.

Progress and Resolution of Outstanding Issues

MapInfo Pro Advanced 2019.3

Issue Number	Description and Resolution
MCS-4303	Added support for the world file (j2w) for the ECW/JP2 driver. We now support the following combinations - jp2&j2w, jpx&jxw, jpf and jfw. You must delete the existing TAB files for this to work. Resolution: Fixed.
MIRAST-17292	MapInfo Runtime was not loading GDAL supported Raster files. Resolution: Fixed.
MIRCS-379	Memory leak in Contour Operation. Resolution: Fixed.
MIRCS-426	The Point Inspection tool was reporting incorrect values for some specific scenarios. Resolution: Fixed.

Issue Number	Description and Resolution
MIRCS-438	In the Polygonise tool, the Match Colors to Input option was not working correctly for the ESRI ASCII Grid format. Resolution: Fixed.
MIRCS-439	RasterProcessing.Rename API was not generating renamed TAB files when providing just the new file name. Resolution: Fixed.
MIRCS-450	Added support for multi-band (more than 4 bands) ECW files. Now when you load these files, they will render as imagery by default. They will also show an extra field with each ECW band displayed as continuous. Resolution: Fixed.
MIRCS-453	Overviews were getting corrupted for TIFF imagery files having bit data. Resolution: Fixed.
MIRCS-466	TAB files having DateTime, or Logical type columns at index before the target column were having issues with passing of column information for Create Grid operations. Resolution: Fixed.
MIRCS-468	Cross Section graphs were being truncated when displayed at full screen. Resolution: Fixed.

MapInfo Pro Advanced 2019.1

Issue Number	Description and Resolution
MIRAST-17095	Issue with re-projecting the drone image to the correct latitude and longitude coordinates in MapInfo Pro 17.0.3 with the Discover Bundle. Resolution: Fixed.

Issue Number	Description and Resolution
MIRCS-403	Issue with reading the projection information from the TIFF file in MapInfo Pro 17.0.4 Resolution: Fixed.
MIRCS-435	MapInfo Pro 2019 stops responding while reprojecting large number of raster (a batch of 500) files together. Resolution: Fixed.
MIRCS-438	In the Polygonise tool, the Match Colors to Input option was not working correctly for the ESRI ASCII Grid format. Resolution: Fixed.

MapInfo Pro Advanced 2019

Issue Number	Description and Resolution
MIRCS-415	Issue while merging multiple .grc files in Planet v7.2. Resolution: Fixed.
MIRCS-412	In the Polygonise tool, the Same Value Cells method gives incorrect result for Multi-resolution raster (.MRR) with a single cell value. Resolution: Fixed.
MIRCS-318	The Quick Search tool on the MapInfo Pro ribbon, fails to find the interpolation methods. Resolution: Fixed.
MIRCS-252	Issue with applying the region style in the Contour tool. Resolution: Fixed.
MIRAST-16960	Issue with the Predictive Encoding option in the Create Raster tool. The Predictive Encoding for float data introduces error in the gridding operation. Resolution: Fixed.

System Requirements

This product is tested on the following Microsoft Windows Desktop Operating Systems:

- Windows 8.1 64-bit
- Windows 10 64-bit
- Windows 2012 Server R2 64-bit
- Windows 2012 Server R2 64-bit with XenApp 7.5
- Windows 2016 Server 64-bit
- Windows 2019 Server 64-bit
- Windows 2019 Server 64-bit with XenApp 7.15

Dependencies and Prerequisites

We recommend ensuring that your machine has the latest updates before installing Pro.

The install wizard checks for the following and prompts you if not already on your system; you can choose to have the install wizard install these requirements, or cancel the installation if you do not want to proceed.

- Microsoft Office Access database engine 2016 (x64)
This does not install when the 32-bit office 2016 driver is installed.
- Microsoft .NET Framework 4.7.2
- Microsoft Visual C++ 2015 Update 3 redistributable (x64)
- Microsoft Visual C++ 2017 redistributable (x64)
- Windows fixes and updates:

Operating System	Required Update
Windows 7 and Server 2008 R2	Hotfix KB3154529

Operating System	Required Update
Windows 8 and Server 2012	Hotfix KB3154527
Windows 8.1 and Server 2012 R2	Hotfix KB3154528
Windows 10 *	Cumulative Update KB3156387

* On Windows 10, the install wizard does not install the Cumulative Update. Ensure this update is installed before installing this release.

Windows 8.1 and Server 2012 R2

A Windows 8, 8.1, Server 2012, or Server 2012 R2 system must have Microsoft Update 1 (KB2919355) installed before installing MapInfo Pro. To obtain this update, go to <https://support.microsoft.com/en-us/kb/2919355>. This update may require that you restart the machine.

Installing 32-bit and 64-bit Microsoft Office Drivers

MapInfo Pro is a 64-bit application that can access spreadsheets and tables from both 64-bit and 32-bit editions of Microsoft Excel and Access. However, to use data from Excel and Access in MapInfo Pro, you must have the Microsoft Office 64-bit drivers installed.

As part of the MapInfo Pro installation process, the 64-bit Microsoft Access Database Engine 2016 Redistributable for Office is installed.

If you require the 32-bit edition of Microsoft Office on the same system with MapInfo Pro, the following procedure shows you how to install both versions of Office drivers.

Note: Microsoft does not support both drivers installed on the same computer citing incompatibility between the product editions or their components, see <http://support.microsoft.com/kb/2269468>.

To install Microsoft Office 32- and 64-bit drivers on the same system:

1. Download the Microsoft Office drivers from <http://www.microsoft.com/en-ca/download/details.aspx?id=54920>.
The 32-bit version is `AccessDatabaseEngine.exe`. The 64-bit version is `AccessDatabaseEngine_x64.exe`.
2. Uninstall the 64-bit Office driver, if installed, and reboot the system. You will already have it installed if you have installed MapInfo Pro 64-bit.
3. Install the 32-bit Office 2016 driver.

4. In the System folder (for example C:\Windows\System32), right-click on `cmd.exe` and select **Run as administrator**.
5. At the command prompt navigate to the folder where the 64-bit Office 2016 driver is located.
6. Type the command and press enter: `AccessDatabaseEngine_x64.exe /quiet`
7. If you have a 32-bit version of Microsoft Office 2007, 2010, 2013, or 2016 installed, then delete or rename its `mso.dll` registry key.
 - a) Open the **Registry Editor** window. On the Microsoft **Start** menu, type `regedit` in the **Search** field, and then click **regedit.exe**.
 - b) In the **Registry Editor** window, navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\16.0\Common\FilePaths`.
 - c) If there is a `mso.dll` value, either delete it or rename it.
If you do not delete or rename the file, then may see a reconfigure message when you open a Microsoft Office application.

Both drivers are now installed on your system.

Opening Office 2013 32-bit Excel and Access Files

There is a potential issue when both 32-bit and 64-bit versions of MapInfo Pro are installed on the same machine along with the 32-bit version of Microsoft Office 2013. The 32-bit version of MapInfo Pro can close unexpectedly when opening tables generated in the 32-bit version of Excel 2013 or Access 2013.

The section *Installing 32-bit and 64-bit Microsoft Office Drivers* in the *MapInfo Pro Install Guide* describes how to install the Microsoft Office 32- and 64-bit drivers on the same system. If after following these steps your 32-bit version of MapInfo Pro closes unexpectedly when opening Excel 2013 and Access 2013 tables, then follow the steps outlined below.

1. Uninstall both of the Microsoft Access database engine 2016 drivers.
 - a) From the **Start** menu, select **Control Panel**.
 - b) On the **Control Panel**, select **Programs and Features**.
 - c) Scroll through the list to locate and select **Microsoft Access database engine 2016**. There will be two instances of this application in the list.
 - d) Select **Uninstall** and follow the instructions that the uninstall procedure provides.
 - e) Select the second **Microsoft Access database engine 2016** and uninstall it.
2. Download and install the 32-bit version of Microsoft Access Runtime 2013 (called `AccessRuntime_x86_en-us.exe`) from <https://www.microsoft.com/en-us/download/details.aspx?id=39358>.
3. Download the 64-bit version of Microsoft Access Database Engine 2016 (called `AccessDatabaseEngine_x64.exe`) from

<https://www.microsoft.com/en-us/download/details.aspx?id=54920>. Make note of the download location, such as `C:\Users\myname\Downloads`.

CAUTION: Do not run (install) from your internet browser window, download it first.

4. Install the 64-bit version of Microsoft Access Database Engine 2016 (called `AccessDatabaseEngine_x64.exe`).
 - a) Open a command prompt window with Administrator privileges. In the system folder `C:\Windows\System32`, right-click on `cmd.exe` and select **Run as Administrator**.
 - b) At the command prompt, change directories to the `AccessDatabaseEngine_x64.exe` file (type `cd C:\Users\myname\Downloads` and press **Enter**).
 - c) At the command prompt, type `AccessDatabaseEngine_x64.exe /passive` and press **Enter**.
5. If you have a 32-bit version of Microsoft Office 2007, 2010, 2013, or 2016 installed, then delete or rename its `mso.dll` registry key.
 - a) Open the **Registry Editor** window. On the Microsoft **Start** menu, type `regedit` in the **Search** field, and then click **regedit.exe**.
 - b) In the **Registry Editor** window, navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\16.0\Common\FilePaths`.
 - c) If there is a `mso.dll` value, either delete it or rename it.
If you do not delete or rename the file, then may see a reconfigure message when you open a Microsoft Office application.

Both 32-bit and 64-bit drivers are now installed on your system, and you can open Access 2013 and Excel 2013 files in both 32-bit and 64-bit versions of MapInfo Pro.

MapInfo Pro Database Connectivity and Support

MapInfo Pro supports the following spatial database servers:

- Microsoft SQL Server 2017 (also called SQL Server Spatial)
- Microsoft SQL Server 2016 (also called SQL Server Spatial)
- PostgreSQL 10.3 with PostGIS 2.4
- PostgreSQL 10.1 with PostGIS 2.1
- SQLite 3.8
- Oracle Spatial 18C & 19C
- Oracle Spatial 12c R1 & R2
- Oracle Spatial 11g R2

In order for MapInfo Pro to access a remote database, you must install a client or driver. The following are ODBC drivers that MapInfo Pro supports:

- Microsoft Access ODBC
- Microsoft SQL Server Native Client 11.0
- PostgreSQL Unicode(x64) ODBC driver 9.06.03
- FDO toolkit 3.8 (used by SQLite and installed with MapInfo Pro)
- Oracle Instant Client 12.1.0.2.0

You can also open tables of data from the following and make them mappable, but only for point data:

- Microsoft Access 2007, 2010, 2013, and 2016

For more about working with remote database information, see *Accessing Remote Data* and *Setting your Database Connection Preferences* in the *Help System*.

Microsoft Office Support

MapInfo Pro supports Microsoft Excel (*.xls , *.xlsx) and Microsoft Access *.accdb) formats for versions 2007 to 2016.

For details, see [Installing 32-bit and 64-bit Microsoft Office Drivers](#) on page 66

MapInfo Pro Web Server Support

MapInfo Pro supports the following web servers:

- GML (Geospatial data in XML format) 2.1.2. The Universal Translator (FME Quick Translator) supports versions up to 3.2.
- Web Feature Service (WFS) 1.0, 1.1, and 2.0
- Web Feature Service with transactions (WFS-T) using WFS 1.0
- Web Map Service (WMS) 1.3
- Web Map Tile Service (WMTS) 1.0

MapInfo Pro also supports the mapping tile servers:

- Google Enterprise tile server
- MapXtreme.NET 8.0 tile server
- Microsoft Bing tile server

- OpenStreetMap tile server
- Spectrum Spatial Server 11.x

Installing MapInfo Pro

When MapInfo Pro starts it checks the operating system directory location

`C:\Users\username\AppData\Roaming\MapInfo\MapInfo` for the following files:

- `MapInfoPro.WOR`
- `STARTUP.WOR`
- `MapInfoPro.wox`
- `MapInfoPro.pref`

If these files exist, then MapInfo Pro uses them at startup. If these files do not exist in the location, then MapInfo Pro creates them all except for the `STARTUP.WOR`.

Application data (AppData) files are the non-executable data files that MapInfo Pro uses during operation.

We recommend you to make sure of the following before starting the process to install / upgrade:

- You must have Administrator rights to run the Installer.
- The Installer requires that your TEMP variable be set to a valid directory.
- You must exit from all Windows programs before beginning the installation process.

The installer for MapInfo Pro will install all the files necessary to run Pro in multiple languages. During the installation process the first question will be what language would you like to run Pro in. This will be the default language of the Pro Application. The list of available languages to choose from will depend on what is available in the installer. Eventually all languages will be available as the localization process finishes up.

MapInfo Pro users who register with a Precisely online account are able to run MapInfo Pro in a connected **Viewer** Only (subscription) mode. The MapInfo Pro installer adds a shortcut (Icon) to the Windows Start Menu to start MapInfo Pro in Viewer (subscription) mode. This icon is located next to the regular MapInfo Pro Start Menu shortcut.

Advanced users may wish to run Pro in a different language than was chosen at installation time. This will now be possible by copying the `MapInfoPro.exe.config` from a language sub-folder (such as `fr`, `de`, `ja`, `zh-CN`) under the MapInfo Pro installation location on top of the `MapInfoPro.exe.config` in the MapInfo Pro installation location. Most users will never need to do this, but if you do try it, be sure to make a copy of the file first! Be aware that many Pro settings such as ribbon customization are locale-specific.

These are the language versions of Pro and the sub folders used (Changing `AppLocale` or `System Charset` via `.config` file is for advanced users only).

UI Locale and Sub Folder Name	AppLocale Name	AppLocale code	Language Country/Region	CodePage	SystemCharset Name
zh-CN	zh-cN	2052	Chinese (Simplified)	936	WindowsSimpChinese
cs	cs-CZ	1029	Czech - Czech Republic	1250	WindowsLatin2
da	da-DK	1030	Danish - Denmark	1252	WindowsLatin1
nl	nl-NL	1043	Dutch - The Netherlands	1252	WindowsLatin1
en	en-US	1033	English - United States	1252	WindowsLatin1
fi	fi-FI	1035	Finnish - Finland	1252	WindowsLatin1
fr	fr-FR	1036	French - France	1252	WindowsLatin1
de	de-DE	1031	German - Germany	1252	WindowsLatin1
he	he-IL	1037	Hebrew - Israel	1252	WindowsHebrew
it	it-IT	1040	Italian - Italy	1252	WindowsLatin1
ja	ja-JP	1041	Japanese - Japan	932	WindowsJapanese
pl	pl-PL	1045	Polish - Poland	1250	WindowsLatin2
pt	pt-BR	1046	Portuguese - Brazil	1252	WindowsLatin1
ru	ru-RU	1049	Russian - Russia	1251	WindowsCyrillic
es	es-ES	1034	Spanish - Spain	1252	WindowsLatin1
sv	sv-SE	1053	Swedish - Sweden	1252	WindowsLatin1
tr	tr-TR	1055	Turkish - Turkey	1254	WindowsTurkish

You can install or upgrade MapInfo Pro using the Interactive Installation Wizard or silently through the command line.

Note: For detailed instructions on installing MapInfo Pro, see the *MapInfo Pro Install Guide*. To access the *MapInfo Pro Install Guide* and other documents, launch the MapInfo Pro DVD

auto-start presentation by double clicking the `autostart.exe` in the root folder of the DVD and then clicking **Online Reference** and **MapInfo Pro Install Guide**.

Installing MapInfo Pro Interactively

You must have Administrator rights to run the Installer, and the Installer requires that your TEMP variable be set to a valid directory.

CAUTION: We strongly recommend that you exit from all Windows programs before beginning the installation process. If you are upgrading from an earlier version of MapInfo Pro, we recommend that you uninstall before upgrading.

MapInfo Pro provides its application data files to each user. Called a Per-User install, this functionality runs the first time you run MapInfo Pro or MapInfo Pro client on a machine, and each time the MapInfo Pro Installer is run thereafter. The application data files include, among others, the Pen Styles file, Custom Symbols files, and Thematic Legend templates. These files allow different users to have custom settings.

To install MapInfo Pro:

1. If installing from a DVD, the Installer automatically launches. If the Installer does not automatically launch, then go to the DVD drive and click on `autostart.exe`. If installing from a download, go to the directory into which you downloaded MapInfo Pro and right-click the `setup.exe` file, select **Run as Administrator** from the pop-up menu to install using elevated privileges.

If you see a prompt for permission to continue. Click **Allow** or **Yes** to proceed.

The wizard begins to lead you through the installation process.

2. On the **Launcher** menu, choose **Install Products**.
3. Choose **MapInfo Pro** and then choose **MapInfo Pro Installer**.

You may see a prompt for permission to continue. Click **Allow** or **Yes** to proceed (if you do not respond and the message times-out, the install is unsuccessful).

4. Select the language in which you want to install MapInfo Pro.
In version 2019.3, you have the option of installing MapInfo Pro in sixteen different languages other than English. You can choose between Brazilian, Chinese, Czech, German, Danish, Spanish, Finnish, French (Standard), Hebrew, Italian, Japanese, Dutch, Polish, Russian, Swedish and Turkish.
5. If any of the prerequisites listed under **Dependencies and Prerequisites** are not already installed on your system, a display will prompt users to install the missing prerequisites. Click **Install**. Installing the prerequisites takes several minutes—the status of each prerequisite updates as it installs.
6. If you see a message that MapInfo Pro requires a reboot, click **Yes**. You must reboot to continue with the installation.

7. Click **Next** to continue the process.
8. On the **License Information** dialog box, review the license. Click **Next** to continue.
9. On the **Customer Information** dialog box, do one of the following:
 - Select the **MapInfo Pro - Licensed or Trial version** check-box, type your user name, organization, serial number, and access code in the corresponding fields and click **Next** to continue.

The user name and organization fields are mandatory. If you also enter the serial number and access code, MapInfo Pro Licensed version is installed. If no credentials are entered, the trial version of MapInfo Pro is installed.

This installs MapInfo Pro Licensed or Trial version based on the information provided and creates a shortcut to MapInfo Pro in the **Start** menu.

- Select the **MapInfo Viewer Subscription** check-box, type your user name and organization and click **Next** to continue.

This installs MapInfo Pro in the subscription mode and creates a shortcut to MapInfo Viewer in the **Start** menu.

- Select both the check-boxes and click **Next** to continue. Shortcuts to both MapInfo Pro and MapInfo Viewer are created in the **Start** menu.

Note: If you choose to select both **MapInfo Pro - Licensed or Trial version** and **MapInfo Viewer Subscription** check boxes, MapInfo Pro would be the default application associated with MapInfo Pro registered file types such as workspace (*.wor) or table (*.tab).

The serial number and access code are listed on the Product Activation Information Card or emailed with your license details, and are used to activate your product.

10. Do one of the following:
 - If your organization purchased node-locked licenses, the third character of your serial number is "**N**". Skip this step and go to [step 12](#).
 - If your organization purchased concurrent licenses, the third character of your serial number is "**S**" or a distributable license, the third character of your serial number is "**D**". In this case, the **License Server Name** and **License Server Port Number** fields display on the dialog box.

Note: For more information about node-locked, concurrent and distributable licenses, see **Starting and Activating MapInfo Pro** under Chapter 3 of the MapInfo Pro Install Guide.

If you know the license server name and port number, type them here. If you do not, you can continue with the installation without filling in the license server name and port number. You will have another opportunity to supply this information when you start MapInfo Pro.

11. Click **Next** to continue.
12. Select one of the following options and then click **Next** to continue:

- **Typical** – Choose this option if you will be using MapInfo Pro as a desktop application including remote database access and connectivity.
- **Custom** – Choose this option to restrict the components that are installed by the installation program. For example, you can prevent the installer from adding the documentation and tools.

13. Select the destination folder where you want to install MapInfo Pro and click **Next**.

14. Click **Install** to finish the installation process.

For detailed instructions on installing MapInfo Pro interactively, see **Chapter 2: Installing MapInfo Pro** from the **MapInfo Pro Install Guide**.

Installing MapInfo Pro Silently

You can only perform a completely silent install of MapInfo Pro when your install process has elevated administrative privileges.

The command-line options `/q` and `/qn` do not work when the User Access Control (UAC) is turned on and your install process does not run with elevated administrative privileges. These options turn off the user interface during the installation, which suppresses the Windows UAC. Users must interact with the UAC to provide the necessary credentials to run the installation as an administrator. If the user interface is turned off, the installation is terminated without warning because the UAC cannot gather the appropriate credentials.

To work around the problem, use the `/qb` option in the command line when you run the silent install. This option gives the installation a basic user interface and allows the UAC to display on the dialog box.

To perform a silent install using the `/qb` option:

1. From the command prompt, go to the MapInfo Pro setup directory. The MapInfo Pro installation `setup.exe` indicated in the steps below can be found on the installation DVD (D:\) in this directory: `\Install\MI_PRO\DISK1`.
2. Type the following command:

```
setup.exe /L#### /s /v"/qb USERNAME=\"MyUser\"
COMPANYNAME=\"MyCompanyName\"
PIDKEY=M##### ACCD=#####\" MPRO=##### MVWR=#####
ADDLOCAL=ProgramFiles,Python37,FME,Documentation,Tools,FDODriver,CatalogBrowser"
```

where:

`/L####` is the 4 digit language code identifier. For example, for English type `setup.exe /L1033`, for French type `setup.exe /L1036`, for German type `setup.exe /L1031`. and for Japanese type `setup.exe /L1041`. DO NOT leave a space between the `/L` and the 4 digit language code.

`PIDKEY=M#####` is the product serial number.

`ACCD=#####` is the access code.

`MPRO=True` creates the MapInfo Pro shortcut in the **Start** menu and sets the default file association of registered MapInfo Pro file types such as workspace (*.wor) or table (*.tab) with MapInfo Pro.

`MVWR=True` creates the MapInfo Viewer shortcut in the **Start** menu. It also sets the default file association of registered MapInfo Pro file types such as workspace (*.wor) or table (*.tab) with the new MapInfo Viewer (subscription) if the `MPRO` parameter above is not set to `True`. For details of this new feature, see [MapInfo Pro Viewer](#).

If both the `MPRO` and `MVWR` parameters above are set to `True`, `MPRO` takes precedence and MapInfo Pro would be the default application associated with MapInfo Pro registered file types such as workspace (*.wor) or table (*.tab).

`ADDLOCAL=ProgramFiles,Python37,FME,Documentation,Tools,FDODriver,CatalogBrowser`. From version 2019.1 **FDODriver** and **CatalogBrowser** are not installed as default components. Use the `ADDLOCAL` parameters to enable the MapInfo Pro silent installer to install these components. Please note that the `ProgramFiles`, `Documentation` and `Tools` parameters are required for a proper MapInfo Pro installation. You can omit other components like `Python37`, `FME`, `FDODriver` and `CatalogBrowser` if you do not want to install them.

3. When User Access Control displays on the dialog box, click **Allow** or **Yes**.

For concurrent licenses, the license server name and port number parameters must be included:

```
SNAME="LicenseServerName"
LSPN="LicenseServerPortNumber"
```

Substitute the appropriate license server name and port number when you perform the silent install.

For detailed instructions on installing MapInfo Pro silently, see **Chapter 5: Silent Installation Procedures** from the **MapInfo Pro Install Guide**.

Installing from a DVD Requires a Drive Letter

The Installer must be run from a drive with a letter such as "G" and not from an explicit uniform naming convention (UNC) path. For example, you might have the MapInfo Pro DVD in your computer as `USERSPC`. Other users may share this device as `USERSPC`; however, it would not contain a drive letter. The MapInfo Pro Installation program requires a drive letter. To remedy this situation, map your network drive to a specific drive letter:

1. In Windows Explorer, right-click the shared directory or DVD drive that contains the MapInfo Pro `SETUP.EXE` and select **Map Network Drive**.
2. Choose a drive letter to map.
3. Run the Installation Program again from the newly mapped drive letter.

License Server Utility

If you have a previous version of the License Server Utility (LSU) installed to manage distributable licenses, then you must upgrade to the latest License Server Utility version 5.2. Using a previous version of the License Server Utility with MapInfo Pro 2019.3 will cause MapInfo Pro to become unresponsive.

Using an incompatible version of the License Server Utility might result in the following issues while fetching licenses for MapInfo Pro:

- MapInfo Pro is unable to borrow a license.
- MapInfo Pro is unable to activate a distributed license from the License Server.

If this occurs, contact your License Server administrator and have them upgrade the latest License Server version.

After successfully borrowing or activating a license, if you again roll back to an older version of License Server Utility, you cannot transfer the borrowed/activated license back to the License Server Utility. You will not see an error message as the operation simply hangs.

Repairing MapInfo Pro from the .MSI file

The repair process fails using the .msi file when the Microsoft User Account Control (UAC) is turned on. Depending on the installation conditions the repair stops with the following error message:

You have MapInfo Pro running, please close it and run setup again.

- Shut off the UAC (User Account Control), this requires System Administrative privileges, reboot the machine, and rerun the repair.
- Run the original `setup.exe` file and then run the repair from maintenance mode.
- Run the installed copy of the MapInfo Pro 2019.3 .msi file as an administrator. Go to `C:\Windows\Installer` and search for the MapInfo Pro 2019.3 .msi file. It will have a name similar to `8e95f1.msi`, where the name is different for each machine. Go to `C:\Windows\System32` and find the `cmd.exe` file. Right click this file and select **Run as Administrator** (this may require an administrator password). Run the command `C:\Windows\Installer\nameofMiPro2019.3.msi` and select **Repair** from maintenance mode.

Disabling User Account Control (UAC) on Windows 7 or Server 2008 R2

To disable UAC on Windows 7 or Server 2008 R2:

1. Launch **MSCONFIG** by from the **Run** menu.
2. In the **System Configuration** dialog box, click the **Tools** tab.
3. Click **Change UAC Settings**.
4. Click **Launch**.
5. Move the slider to **Never Notify**.
6. Click **OK** to close the dialog box.
7. Reboot the machine to have this change take effect.

Disabling User Account Control (UAC) on Windows 8.1 or Windows Server 2012 R2

To disable UAC on Windows 8.1 or Windows Server 2012 R2:

1.
 - On Windows 8.1, go to the **Search Charm** and type `Change User Account Control settings`.
 - On Windows 2012 R2, on the **Control Panel**, select **User Accounts**, and click on **Change User Account Control Settings**.
2. In the **User Account Control Settings** console, move the slider to **Never Notify**.
3. Click **OK**.
You may see a prompt to confirm your selection or to enter an administrator password.
4. Reboot the machine to have this change take effect.

Disabling User Account Control (UAC) on Windows 10

To disable UAC on Windows 10:

1. On the **Control Panel**, select **User Accounts**, and click on **Change User Account Control Settings**.
2. In the **User Account Control Settings** console, move the slider to **Never Notify**.
3. Click **OK**.

Support Notices

Bing Maps Expiration Date

Bing Maps has a limited license period, which is specific to each version of MapInfo Pro. Bing Maps and the **Add Bing Roads to Map** and the **Move Map to** features cease working after this date:

- MapInfo Pro 2019 on January 1st, 2021
- MapInfo Pro 17.0 on January 1st, 2019
- MapInfo Pro 16.0 on January 1st, 2018
- MapInfo Pro 15.0 and 15.2 on January 1st, 2017

For details, contact your account representative.

Downloading Tools and Applications

The following web sites give access to MapInfo Pro documents, trials, and supporting materials:

- The MapInfo Pro page: <http://www.precisely.com/product/precisely-mapinfo/mapinfo-pro>
- The MapInfo Pro support page: <http://support.precisely.com>

Locating Your Documentation

MapInfo Pro documentation, in the form of PDF files, installs with MapInfo Pro in to the `Documentation` sub folder (for example, `C:\Program Files\MapInfo\Professional\Documentation`). You must have the Adobe Acrobat reader installed to view PDF files. To download a free copy of the Adobe Acrobat Reader, go to <http://www.adobe.com>.

For instructions on installing MapInfo Pro, see the *MapInfo Pro Install Guide*. To access the *MapInfo Pro Install Guide* and other documents, launch the MapInfo Pro DVD autostart presentation by double clicking the `autostart.exe` in the root folder of the DVD and clicking **Online Reference** and **MapInfo Pro Install Guide**.

Documentation is also available on our website from <http://support.precisely.com>.

Installation Instructions are Available in the Install Guide

For system requirements, installation instructions, and System Administrator notes for performing a work group installation, see the *MapInfo Pro Install Guide*. From the software installer, click **Online Reference** and then **MapInfo Pro Install Guide**.

Instructions for Activating your License are in the Install Guide

Instructions for activating your MapInfo Pro license are in the *MapInfo Pro Install Guide*. The same information is also in the *Activating Your Product*. From the software installer, click **Online Reference** and then **MapInfo Pro Install Guide** or **Activating Your Product**.

The MapInfo Pro Data Directory Document

MapInfo Pro's sample data comes with the *MapInfo Pro Data Directory* document, which describes the sample data. From the software installer, click **Online Reference** and then **MapInfo Pro Data Directory**.

Sample Data Enhancements

To help you get started, Precisely provides you with some United States-based and world-wide maps you can use as a background to your data. To install the free data provided with MapInfo Pro see the section titled *Installing Data* in the *MapInfo Pro Install Guide*. For a description of the sample data supplied with this product, see the *MapInfo Pro Data Directory* document.

Open Source Attribution

This product contains GeoJSON.NET, which is licensed under the MIT license. The license can be downloaded from <https://github.com/GeoJSON-Net/GeoJSON.Net/blob/master/LICENSE.md>. The source code for this software is available from <https://github.com/GeoJSON-Net/GeoJSON.Net>.



2 Blue Hill Plaza, #1563
Pearl River, NY 10965
USA

www.precisely.com

© 1985, 2020 Precisely. All rights reserved.