

# Spectrum™ Technology Platform

Version 2018.2.0

Spectrum Spatial Map Uploader Guide



# Table of Contents

## 1 - Map Uploader

---

About the Spectrum Spatial Map Uploader	4
Installing and Configuring the Map Uploader	10
Running the Map Uploader	12
Why Does My Uploaded Map Look Different in MapInfo Pro?	13
Troubleshooting	17
Copyright	20

# 1 - Map Uploader

Welcome to Spectrum Spatial Map Uploader for use with Spectrum™ Technology Platform and MapInfo® Pro version 16.0 and higher.

Rather have a PDF? [Download the Map Uploader Guide PDF](#)

**Note:** See [Map Uploader \(Pre-MapInfo Pro v16\)](#) if you are using the previous version of the tool.

## In this section

About the Spectrum Spatial Map Uploader	4
Installing and Configuring the Map Uploader	10
Running the Map Uploader	12
Why Does My Uploaded Map Look Different in MapInfo Pro?	13
Troubleshooting	17
Copyright	20

## About the Spectrum Spatial Map Uploader

The Spectrum Spatial Map Uploader is a tool that copies the properties of the layers in a MapInfo Pro map to a Spectrum Spatial server. It takes the information of one map stored in MapInfo Pro and stores the equivalent information in a set of named resources on the server. This tool can be downloaded from the Spectrum Spatial section of the Welcome Page, under **Map Uploader** on the Utilities tab.

The map created on the Spectrum Spatial server will consist of a number of distinct parts known as named resources.

- One named map whose name and location will be specified by the user.
- A named layer for each layer in the map plus a label layer. These will be created in their own subfolder with a name is determined by the name of the map. Each resource will have the same name as used in MapInfo Pro.
- A named label source for each layer that can possibly be labeled will be created in the same location as the layers.
- A named table which will point to the data source used for each layer. This will be created in the same location as the map and will use the MapInfo Pro table "alias" name as its name. The named table frequently has the same name as the TAB file that was used to open the table in MapInfo Pro.

The named resources that are created are used by Spectrum Spatial to draw the map and query the data. Any application using Spectrum Spatial, including Spectrum Spatial Analyst and Spectrum Spatial for Business Intelligence, will depend on these resources being created correctly.

Named resources contain different types of information.

- The named map contains the default geographic location, zoom level, and the ordered list of named layers of the map as they were viewed in MapInfo Pro.
- A named layer contains all the style and rendering properties of that layer. These can vary from nothing (when all the properties are stored with the data), to ranged zoom constraints, or to one or more thematic definitions with many bins and styles or multiple styles overrides.
- A named label layer contains the list of named label sources. The order of these determines the order of label processing while rendering the map. This order is identical to the order in MapInfo Pro.
- A named label source contains all the label properties used for the set of labels from one table.
- A named table points to the data from the perspective of the server. If the data is in a file (Native TAB, shapefile, GeoPackage), the named table must point to that data on the server. If the data is in a database there must be a valid connection to that database.

Spectrum Spatial uses a named connection to help with this process. These connections must be created before you run the Map Uploader so that the tool can use them to ensure that the data in the map can be found.

**Note:** The tool does not copy data to the server. Any file based data must exist on the server and be placed under a connection that is created in Spatial Manager before the Map Uploader is used. The Workspace Packager tool that ships with MapInfo Pro can be used for this purpose.

## How the Map Uploader Works

The Map Uploader behaves in an interactive, wizard-like fashion. This allows the tool to :

- Clearly show data sources that are not supported in Spectrum Spatial and therefore will not exist in the uploaded map.
- Clearly show layers that have settings that are not fully supported. You can decide whether to allow the layer or labels to be uploaded without the unsupported settings or to remove the layer from the uploaded map.
- Choose the exact location of the map on the server repository, to create a new folder in the repository if desired, and to select an existing map for overwriting.

Based on whether a new map is being created or an existing map is being created, existing named tables will be reused with different rules. An existing map has a list of tables it uses and these will all be reused wherever they are in the repository. A new map will only search for tables in the same folder of the repository.

Tables for which existing named resources do not exist will have to be located. If the table in MapInfo Pro uses a database table, database connection on the server will be searched for the table with its schema as a qualifier. File-based tables will be searched by name under any file-based connections on the server.

To accommodate the differences between the capabilities of MapInfo Pro and Spectrum Spatial, the tool works in a number of discrete steps. At any point before the resources are created, the upload can be canceled.

**Step 1** is to verify that the data used in MapInfo Pro can be supported in Spectrum Spatial. If there are no issues, you will not notice that this step occurred. If there are issues with a subset of the data, the map can be created without these data sources.

**Step 2** is to examine the properties of the layers and to warn if any of the style, thematic, or label settings are not an exact match for the server. If there are no issues, you will not notice that this step occurred. If there are issues, you can decide whether to skip that layer or still include it, allowing the tool to create the settings in the most compatible way.

**Step 3** prompts you to give a location and name for the map on the server. You can navigate the Spatial Repository to locate the map and create new folders if desired. If you choose an existing map, you will be prompted whether it should be overwritten. If you choose to overwrite, the map and all the other resources used by that map may be overwritten. Again, if there were no issue in steps 1-2, this will be the first thing you will see.

**Step 4** is to ensure that all the data sources can be located using one or more server connections. File data is located using file paths and database data is located using database connection strings, the database schema and table name. If the data cannot be located, those layers will be skipped. You have the option to override the data that the Map Uploader does locate.

**Step 5** is the upload itself. The named map, layers, label sources, and tables are created in the Spectrum Spatial repository. A progress list will indicate what is being created. When it is done, a link will be provided to Spatial Manager where you can examine and view the map and all the other resources. If there are any server-side or communication errors with the server, they will be displayed. The map may be partially created in that case.

## Preparing for a Successful Map Upload

Not all the features and capabilities of MapInfo Pro are supported in Spectrum Spatial. To ensure a smooth upload from MapInfo Pro to Spectrum Spatial:

- Verify that your map does not use unsupported datasources, or map rendering properties. See [Why Does My Uploaded Map Look Different in MapInfo Pro?](#) on page 13 for detailed information about unsupported features and [Troubleshooting](#) on page 17 for instructions on how to work around some of the limitations.
- Verify that your data is in the location you want it to be from the server perspective and that Spectrum Spatial connections exist that point to it. You must create these connections using Spatial Manager prior to uploading the map. For more information about Spatial Manager, see the [Spatial Manager Guide](#) in the Utilities section of the *Spectrum Spatial Guide*. To copy MapInfo TAB data that is in Native or NativeX format from a MapInfo Pro map to another server, you can use the Workspace Packager tool that ships with MapInfo Pro to copy all the referenced data to a single folder.

## Understanding the Structure of the Named Resources

After uploading, the repository location you specified in the Map Uploader will then contain a named map, named tables, and a folder with "\_Layers" appended to the map name. The "\_Layers" folder will contain the named layers and named label sources.

```

/FolderName
.../Map [named map]
.../TableA [named table]
.../TableB
.../TableC
.../Map_Layers[folder containing named layers and label sources]
...../LayerA [named layer]
...../LayerA_Labels [named label source]

```

```

...../LayerB
...../LayerB_Labels
...../LayerC
...../LayerC_Labels
...../Map_LabelLayer [named label layer]

```

For example, below is the repository view in Spatial Manager after MyFirstMap has been successfully uploaded to /Project1 in the Spectrum Spatial repository. The MyFirstMap\_Layers folder contains all the named layers and named table sources, MyFirstMap is the named map, and the remaining seven items are the named tables.

The screenshot shows the Spatial Manager interface. At the top is a purple header with 'Spatial Manager' and dropdown menus for 'Data' and 'Services'. Below the header is a search bar with a magnifying glass icon and the text 'Search'. To the left is a tree view showing a hierarchy: 'Root' (with a folder icon), 'Configuration' (with a folder icon), 'Project1' (with a folder icon and highlighted in blue), and 'Samples' (with a folder icon). To the right of the tree view is a main content area titled 'Project1'. At the top of this area are three icons: a folder with a plus sign, a folder, and a trash can, followed by a 'Create' button with a dropdown arrow. Below these icons is a table listing items in the Project1 repository. The table has two columns: 'Type' and 'Name'. Each row has a checkbox on the left, an icon representing the item type, and the item name.

<input type="checkbox"/>	Type	Name ▲
<input type="checkbox"/>	Folder	MyFirstMap_Layers
<input type="checkbox"/>	Table	Interstates
<input type="checkbox"/>	Map	MyFirstMap
<input type="checkbox"/>	Table	NY_FiveCounties_Cities
<input type="checkbox"/>	Table	NY_FiveCounties_ZipBdy
<input type="checkbox"/>	Table	NYS_Counties
<input type="checkbox"/>	Table	Urban_Areas
<input type="checkbox"/>	Table	USA
<input type="checkbox"/>	Table	USCTY153

The MyFirstMap\_Layers folder contains named layers, named label sources, and a named label layer:

# Spatial Manager

Data ▾
Services ▾

- Root
  - Configuration
  - Project1
    - MyFirstMap\_Layers
  - Samples

## MyFirstMap\_Layers

Create ▾

<input type="checkbox"/>	Type	Name ▲
<input type="checkbox"/>		Interstates
<input type="checkbox"/>		Interstates_Labels
<input type="checkbox"/>		MyFirstMap_LabelLayer
<input type="checkbox"/>		NY_FiveCounties_Cities
<input type="checkbox"/>		NY_FiveCounties_Cities_Labels
<input type="checkbox"/>		NY_FiveCounties_ZipBdy
<input type="checkbox"/>		NY_FiveCounties_ZipBdy_Labels
<input type="checkbox"/>		NYS_Counties
<input type="checkbox"/>		NYS_Counties_Labels
<input type="checkbox"/>		Urban_Areas
<input type="checkbox"/>		Urban_Areas_Labels
<input type="checkbox"/>		USA
<input type="checkbox"/>		USA_Labels
<input type="checkbox"/>		USCTY153
<input type="checkbox"/>		USCTY153_Labels

## Understanding Reuse - Overwrite versus New Map Behavior

How your named resources are created and appear in the Spectrum Spatial repository depend on the choices you made using the Map Uploader utility.

- If you uploaded a new map, named resources are created in the repository according to the standard file structure (see [Understanding the Structure of the Named Resources](#) on page 6) .
- If you repeat the upload of the same map and choose to overwrite, layer resources will be overwritten in the same folder with the newly uploaded properties. The named tables will be checked and reused. New named tables will only be created if you have added data to the map.
- If you repeat the upload of the same map and choose a different map name in the same folder, a new folder for the layer resources will be created so that the two maps can each have their own styles. Named tables in that folder will be reused if they represent the same data. Any new tables that need to be created will be created, as before, in the same folder as the map.

If you are creating a new map and tables were moved to a "common" folder, the new map won't look at common folder, so there will be duplicates.

## Named Connections

A named connection is a type of named resource that defines the connection details for connecting to data sources. This is where you provide details about your database or file-system location. Named connections are stored in the repository. Named connections are created, viewed, and managed using Spatial Manager.

There are two types of named connections: file-based (file system) and database.

### *Named Connections for Files*

File-based named connections are used for TAB files, shapefiles, and GeoPackage files. The named connection contains a root path that can be updated without affecting the relative hierarchy of the files, such as in the case of relocating files to another drive or system.

### *Named Connections for Databases*

Spectrum Spatial supports JDBC named connections for databases (Oracle, SQL Server, PostgreSQL/PostGIS, and Generic JDBC).

**Note:**

Spectrum Spatial supports the major spatial databases listed above. Nothing needs to be added to support them. If you are trying to access a different database than above, you may need to ensure that the proper database JAR files have been installed in Spectrum™ Technology Platform. Copy the appropriate JAR files to the

<spectrum\_dir>/server/modules/spatial/lib directory. Once you have copied your JAR files, restart the server.

## Installing and Configuring the Map Uploader

The Spectrum Spatial Map Uploader requires some configuration prior to use.

### Prerequisites

To use the Spectrum Spatial Map Uploader you need:

- MapInfo Pro 16.0 or above
- Microsoft .NET Framework 4.6.1 or above (this is installed as part of MapInfo Pro)
- Spectrum Spatial 18.2
- A Spectrum user with at least the spatial-admin role
- If you are using Spectrum Spatial Analyst, version 18.2 is required
- Named connections for the data you are using in MapInfo Pro (TAB files, shapefiles, or GeoPackage files must be present on the server)

**Note:** Named connections are created in Spatial Manager. For more information about creating named connections, refer to the Spatial Manager documentation in the *Spectrum Spatial Guide*.

### Adding the Uploader to the MapInfo Pro Tools Menu

The Spectrum Spatial Map Uploader can be downloaded as a MapInfo Pro plugin file from the Spectrum Spatial section of the Welcome Page, under **Map Uploader** on the Utilities tab.

1. Click **Download Zip** to save the `map-uploader-v2.zip` file. Unzip it to a location that is accessible to MapInfo Pro, such as `D:\Spectrum Map Uploader` or `C:\Spectrum Map Uploader`, or where your other MapInfo Pro plugins are installed.
2. Start MapInfo Pro.
3. On the **HOME** tab, in the **Tools** group, click **Tool Extensions**.
4. Click **Options** on any of the tabs then click **Run Program**.
5. In the Run MapBasic Program dialog box, find and select the `NRCreatorAddIn.mbx` then click **Open**.

6. Click **Tool Extensions**.
7. On the **Running** tab, right-click **Map Uploader** and select **Register**.
8. The Register Tool dialog appears, fully populated with the appropriate details.
9. Click **OK**.
10. **Optional:** If you want Map Uploader to always appear on the tools menu when you open MapInfo Pro, select the **AutoLoad** check box on the **Registered** tab.

## Configuring HTTPS

This task explains the steps to configure HTTPS communication with a Spectrum Spatial server that uses a self-signed certificate.

Make sure that the Spectrum JDK is used:

```
C:\>set JAVA_HOME=C:\Program Files\Pitney Bowes\Spectrum\java64
C:\>set path=%JAVA_HOME%\bin;%path%
```

### 1. Create a keystore with a self-signed certificate

```
C:\>keytool -genkeypair -alias client -keystore keystore.p12 -storetype
pkcs12 -keyalg RSA -sigalg SHA256withRSA

Enter keystore password: password
Re-enter new password: password
What is your first and last name?
 [Unknown]: <Spectrum server hostname>
What is the name of your organizational unit?
 [Unknown]: Spectrum
What is the name of your organization?
 [Unknown]: PB
What is the name of your City or Locality?
 [Unknown]:
What is the name of your State or Province?
 [Unknown]:
What is the two-letter country code for this unit?
 [Unknown]:
Is CN=<Spectrum server hostname>, OU=Spectrum, O=PB, L=Unknown,
ST=Unknown, C=Unknown correct?
 [no]: yes
```

### 2. Import the certificate into Spectrum keystore client-keystore.p12

```
C:\>keytool -importkeystore -srckeystore keystore.p12 -destkeystore
<%SpectrumPath%\server\app\conf\certs\client-keystore.p12 -destkeypass
<Spectrum keystorePassword>

Importing keystore keystore.p12 to client-keystore.p12...
Enter destination keystore password:
```

```
Enter source keystore password:
Entry for alias client successfully imported.
Import command completed: 1 entries successfully imported, 0 entries
failed or cancelled
```

### 3. Make the Spectrum configuration changes:

#### a. Modify `spectrum-container.properties` file

```
# HTTPS
spectrum.https.port=8443
spectrum.https.encryption.validateCerts=false
spectrum.https.encryption.trustAll=true

## overrides
spectrum.https.enabled=true
spectrum.https.encryption.selfSignedCert=true
spectrum.https.encryption.trustAllHosts=true
spectrum.http.default.protocol=https
# specify the imported keystore alias
spectrum.https.encryption.keystoreAlias=client (the same as the
alias of keypair in step 1 (it should be in lowercase)).
```

#### b. Modify `java.properties` file

```
repository.port=8443
repository.useSecureConnection=true
```

**Note:** To use any Spectrum spatial utilities such as **tilegenerator**, specify the keystore created in step 1 as the `trustStore`

```
JAVA_OPTS=-Djavax.net.ssl.trustStore=.\\keystore.p12
-Djavax.net.ssl.trustStorePassword=password
```

## Running the Map Uploader

There are three ways to run the Map Uploader:

- On the **HOME** tab, in the **Tools** group, click **Tool Extensions**. On the **Running** tab, double-click **Map Uploader**.
- Right-click on an open map and select **Upload Map**.
- From the Explorer window, right-click the name of the map and select **Upload Map**.

# Why Does My Uploaded Map Look Different in MapInfo Pro?

Because several features in MapInfo Pro are either not supported or are supported differently in Spectrum Spatial, maps created in MapInfo Pro will not look identical to the uploaded map in Spatial Manager. In these cases you can follow the guidelines in this section to conform your files to uploading.

## Web Service Layers

MapInfo Pro users can connect to a Bing tile service or other tile service and create map layers from those tiles. Spectrum Spatial and Spectrum Spatial Analyst do work with these services, however those tables and layers cannot be part of a named map. You can either cancel or continue without the unsupported layers.

MapInfo Pro users also can connect to an OGC web service (such as WFS, WMS, or WMTS) and create map layers from the data on those servers. Although Spectrum Spatial and Spatial Analyst also support WFS, WMS, and WMTS, those tables or layers cannot be part of a named map. You can continue without the layer or cancel the operation. If you require a WFS layer in your named map; however, you can convert the file to a native TAB or any other supported datasource (see [Using Data from Unsupported Datasources](#) on page 17).

For all types of service layers (OGC, Bing, tile service), you can set the map config via the Admin Console to enable them in Spectrum Spatial Analyst. For more information, see the *Spectrum Spatial Analyst Administration Guide* at [pitneybowes.com](http://pitneybowes.com).

## Datasources and Queries

MapInfo Pro supports some datasource types that Spectrum Spatial does not. An upload cannot be performed if TAB files point to data stored in one of these formats:

- ASCII
- Microsoft Excel
- Microsoft Access
- FME
- WFS

In any of the above cases, you can either cancel the upload or continue without the unsupported layer. If you require this layer in your named map, however, you will need to convert the table to a

Native TAB or any other supported datasource (see [Using Data from Unsupported Datasources](#) on page 17).

Supported datasources include:

- native TAB
- native extended TAB (NativeX)
- GeoPackage
- Spatial databases (PostGIS, Oracle, SQL Server)

### *Query Layers*

MapInfo Pro allows users to create layers based on the results of a SQL select or any selection in MapInfo Pro. These are layers for which the data is not in a physical table but is instead the result of a query. For example, `select * from CustomerPoints where SALES >= 100000`. This is a query table, not a physical entity but derived from a physical entity. This named view is what is referenced by the named layer.

You can either cancel or continue without the unsupported layer. Otherwise, to use this layer in your named map you must save the query as a native TAB or any other supported datasource (see [Using Data from Unsupported Datasources](#) on page 17).

## Objects in Cosmetic Layers

In MapInfo Pro objects can be drawn in the Cosmetic layer, such as a scalebar. The Map Uploader cannot make a layer out of this, as it is not a permanent table that can be matched to data known to the server. You can either cancel or continue without the unsupported layer. Otherwise, to use this named layer in your named map you must convert the objects in the Cosmetic layer to a native TAB or any other supported datasource (see [Workaround for Cosmetic Layers](#) on page 18).

## Unsupported Label and Style Settings

MapInfo Pro has some label and style settings that are not supported or do not translate readily to Spectrum Spatial. You can continue without the layer or cancel the operation. If you require a layer with stacked styles in your named map, however, you will need to create multiple layers (see [Using Layers with Stacked Styles](#) on page 18). If you require a layer with translucency, you will instead need to use OpenLayers or the transparency settings in Spectrum Spatial Analyst, for example.

### *Unsupported Label Effects*

The list below describes the labeling effects available in MapInfo Pro that are not supported in Spectrum Spatial. When any of these features are used in a map in MapInfo Pro the resulting map created in Spectrum Spatial will look different. In most cases the map will still have the labels but they will appear without these features applied.

- *Allow abbreviated labels.* If these are used in a MapInfo Pro map the primary label column will be used instead in a Spectrum Spatial map. The label placement, which items are labeled and label count may be different in a map in Spectrum Spatial.
- *Label callout lines.* These will not appear and the label placement may be different in a map in Spectrum Spatial
- *Try other positions.* The label placement, which items are labeled and label count may be different in a map in Spectrum Spatial.
- For line object labels certain Curved Label Options are not supported: *Auto position along line, Allow rotated as fallback and Overhang.* The label placement and appearance, which items are labeled and label count may be different in a map in Spectrum Spatial.
- For region object labels certain effects are not supported: *Label at centroid, Auto positions within region, Allow font reduction for fit, Confine labels to region boundary, Place outside if necessary.* The label placement and appearance, which items are labeled and label count may be different in a map in Spectrum Spatial.
- A map in MapInfo Pro that has label priority adjustments will most likely not look the same in Spectrum Spatial because label priority is not supported. Which items are labeled and label count may be different in a map in Spectrum Spatial.
- When a map has had the MapBasic command "*Set Map Layer layerId Label Max number\_of\_labels*" applied in order to limit the number of labels displayed all the labels still appear in the map created in Spectrum Spatial.

### Custom Labels

MapInfo Pro allows users to create custom labels. These are labels that are placed manually placed somewhere other than where auto-labeling rules place them. Spectrum Spatial does not support custom labels and therefore they will be lost.

### Custom Symbols

Any custom bitmap symbols (that is, your own images that do not ship with MapInfo Pro or Spectrum Spatial), need to be uploaded to the server outside of the map upload. Place the images in a jar file in this directory: `install_dir/server/modules/spatial/lib`. MapInfo Pro users can easily create custom symbols by adding any image file to their "Custom Symbols" directory then applying these symbols to a map. The Map Uploader does see that custom symbols are being used, however it has no way of knowing if it is one of the symbols that is shipped with MapInfo Pro and therefore included with Spatial, or if it is a new custom symbol. The Map Uploader therefore creates all the expected named resources; however, when accessing the named map in Spatial Manager, nothing is seen in the preview map window, nor are any errors displayed.

### Unsupported Symbol Styles

The below list includes some known MapInfo Pro point symbols that will not appear the same in Spatial Manager maps due to support limitations. This is not an exhaustive list and some symbols within the folders may appear properly or appear the same:

- Bookshelf Symbol 7
- Custom Symbols PinPoints

- ERS v2 Incidents
- All ERS V2 sets (see screen grab below)
- Marlett
- MS Outlook

### *Unsupported Fonts*

Some fonts used by MapInfo Pro are not supported by Spectrum Spatial or may not be available from the server. In this scenario you may see in the Spectrum Spatial map either a font that is similar but not exactly the same as in MapInfo Pro or the text may get changed.

When using double byte or other languages that contains characters not provided by a font that you are using in Spectrum Spatial, you will see the text change.

When the font Arial Unicode MS is used for the labels then the map in Spectrum Spatial appears correctly. If you encounter label problems with a map in Spectrum Spatial, it may be necessary to change the font used in the map in MapInfo Pro to one that displays correctly in Spectrum Spatial. If a font appears in the character map (charmap in Windows), then it will appear correctly in Spectrum Spatial.

### *Unsupported Label Expressions*

Some theme expressions used in MapInfo Pro are not supported by Spectrum Spatial. When a map is created in Spectrum Spatial using a non-supported theme expression from MapInfo Pro that theme will not be displayed in the map. The layer with the theme will be displayed as if it did not have a theme. Unsupported expressions include ObjectGeography ( object, attribute ), ObjectInfo( object, attribute ), Maximum( num\_expr, num\_expr ), Str\$( expression ), and Val( string\_expr ).

## Unsupported Theme Settings

### *Unsupported Theme Expressions*

Some theme expressions used in MapInfo Pro are not supported by Spectrum Spatial. When a map is created in Spectrum Spatial using a non-supported theme expression from MapInfo Pro that theme will not be displayed in the map. The layer with the theme will be displayed as if it did not have a theme. Some expressions available in MapInfo Pro are not supported in Spectrum Spatial, including ObjectGeography (object, attribute), ObjectInfo (object, attribute), Maximum ( num\_expr, num\_expr), Str\$ ( expression), and Val ( string\_expr).

Thematic maps with half-pie charts are also not supported at this time.

## Grid Lines

MapInfo Pro users can add grid lines to their maps. However, this is not a permanent layer. It is drawn on top of the map like objects in the Cosmetic Layer, so the grid lines will not be part of the named map.

## Troubleshooting

The Spectrum Spatial Map Uploader does not support all features in MapInfo Pro. In these cases you can follow the guidelines in this section to conform your files to uploading.

### Using Data from Unsupported Datasources

Some types of data cannot be uploaded. To create new named layers that will work with the Map Uploader:

1. Create a version in a supported datasource such as native TAB using the `Save Copy As` command from the **HOME** tab.

**Note:** Supported datasources include:

- native TAB
- extended native TAB (NativeX)
- GeoPackage
- Spatial databases (PostGIS, Oracle, SQL Server)

2. Copy the new native table to the server. Like all other tables, the datasource must be accessible to the server via a named connection. So put it in a location under an existing connection or put it somewhere and create a connection to it. (to find location connections point to open the connection's detail page in Spatial Manager and look at the "Source Folder" property).

**Note:** If Spectrum Spatial is getting its data from a database then a user needs to put the table in their spatial database. Like a flat-file TAB, that database needs to be available to the server via a connection.

3. Remove the unsupported layer from the MapInfo Pro map.
4. Add the newly created layer to the MapInfo Pro map in the same place as the unsupported layer, with the same label and display settings, re-creating any themes.
5. Upload the map.

## Workaround for Cosmetic Layers

Some types of layers, such as query layers or WFS/WMS layers, cannot be uploaded. To create new named layers that will work with the Map Uploader:

1. Create a native TAB version of the table using the `Save Cosmetic Objects` command from the **MAP** tab (**MAP > Cosmetic > Save Cosmetic Objects**).
2. Copy the new native table to the server. Like all other tables, the datasource must be accessible to the server via a named connection. So put it in a location under an existing connection or put it somewhere and create a connection to it. (to find location connections point to open the connection's detail page in Spatial Manager and look at the "Source Folder" property).
3. Add the newly created layer to the MapInfo Pro map.
4. Upload the map.

## Using Layers with Stacked Styles

Layers with stacked styles cannot be uploaded. To create multiple layers that will re-created stacked styles and work with the Map Uploader:

1. Create a version of the first layer in a supported datasource such as native TAB using the `Save Copy As` command from the **HOME** tab.

**Note:** Supported datasources include:

- native TAB
- native extended TAB (NativeX)
- GeoPackage
- Spatial databases (PostGIS, Oracle, SQL Server)

2. Copy the new native table to the server. Like all other tables, the datasource must be accessible to the server via a named connection. So put it in a location under an existing connection or put it somewhere and create a connection to it. (to find location connections point to open the connection's detail page in Spatial Manager and look at the "Source Folder" property).

**Note:** If Spectrum Spatial is getting its data from a database then a user needs to put the table in their spatial database. Like a flat-file TAB, that database needs to be available to the server via a connection.

3. Remove the unsupported layer from the MapInfo Pro map.
4. Add the newly created layer to the MapInfo Pro map in the same place as the unsupported layer, with the same label and display settings, re-creating any themes.
5. Repeat the previous steps for each layer.

6. Upload the map.

## Table Not Found Warning

Map Uploader will try to match the local table with a server table. It does this by getting the tab file or database table name from MapInfo Pro and looking for a match in each connection on the server. If a different table name is used locally versus on the server then a match will not be found, or an incorrect match may be found. To fix this, change one of the table names. To easily move a copy of a MapInfo Pro project to another server, use the Workspace Packager tool to copy all the referenced data to a single folder.

## SampleData Information Message

The SampleData provided in the `/Samples` folder of the repository points to data in the sample data folder. This data was supplied as sample data for learning about Spectrum Spatial. During an upgrade it could change or be removed. It is the product's data.

Maps that you are using in your organization should use your own data. In the unlikely event that your tables are named the same as the ones in our samples, it is possible that this message would appear. This would occur if you have not copied your own copy to the server or have not created a Spectrum Spatial connection to that data. Other than a temporary demonstration, it is recommended that you do not use the sample data for your own maps.

## Changing Service Request Timeout Value

This section describes the way to change the default `timeOut` value in the Map Uploader.

The map uploader sends multiple service requests to the Spectrum™ Technology Platform server. In some rare cases, the response takes longer than the default time the uploader waits. This can result in the uploader failing to find tables or other resources. To fix this, you can increase the time by editing the default value in the `mapuploader.config` file. This file resides in the `MapInfo\Map Uploader` folder under the user's **AppData** folder.

To change the default timeout setting:

1. Type `%appdata%\MapInfo\Map Uploader` in the Windows explorer.
2. Open the **mapuploader.config** file in a text editor.
3. Search for `requestTimeout`. The default timeout value for Map Uploader is 150 seconds. The setting looks like:

```
<!--Value represents request timeout in seconds-->
  <requesttimeout>150</requesttimeout>
```

```
<setting name="requestTimeout" serializeAs="String">
  <value>150</value>
</setting>
```

4. Change the value and save to change the number of seconds.

## Copyright

Information in this document is subject to change without notice and does not represent a commitment on the part of the vendor or its representatives. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, without the written permission of Pitney Bowes Software Inc., 350 Jordan Road, Troy, New York 12180-8399.

© 2017 Pitney Bowes Software Inc. All rights reserved. Pitney Bowes Software Inc. is a wholly owned subsidiary of Pitney Bowes Inc. Pitney Bowes, the Corporate logo, MapInfo, Group 1 Software, and MapInfo Pro are trademarks of Pitney Bowes Software Inc. All other marks and trademarks are property of their respective holders.

Contact information for all Pitney Bowes Software Inc. offices is located at: <http://www.pitneybowes.com/us/contact-us.html>.

This product contains AutoMapper, version number 4.2.1, which is licensed under the MIT license. The license can be downloaded from <https://github.com/AutoMapper/AutoMapper/blob/develop/LICENSE.txt>. The source code for this software is available from <https://github.com/AutoMapper/AutoMapper>.

This product contains FontAwesome.WPF, version number 4.5.0, which is licensed under the MIT license. The license can be downloaded from <https://opensource.org/licenses/MIT>. The source code for this software is available from <https://github.com/charri/Font-Awesome-WPF>.

This product contains Polly, version number 3.0.0, which is licensed under the BSD 3-Clause license. The license can be downloaded from <https://opensource.org/licenses/BSD-3-Clause>. The source code for this software is available from <https://github.com/App-vNext/Polly>.

This product contains log4net, version 1.2.15, which is licensed under the Apache License. The license can be downloaded from <http://www.apache.org/licenses>. The source code for this software is available from [http://logging.apache.org/log4net/download\\_log4net](http://logging.apache.org/log4net/download_log4net).

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

This product contains AsyncEx 3.0, which is licensed under the MIT license. The license can be downloaded from <https://github.com/StephenCleary/AsyncEx/blob/master/LICENSE>. The source code for this software is available from <https://github.com/StephenCleary/AsyncEx>.

© 2015 Adobe Systems Incorporated. All rights reserved. Adobe, the Adobe logo, Acrobat and the Adobe PDF logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

libtiff © 1988-1997 Sam Leffler, © 2015 Silicon Graphics Inc. All Rights Reserved.

libpng © 2004, 2006-2014 Glenn Randers-Pehrson.

zlib © 1995-2012 Jean-loup Gailly and Mark Adler.



3001 Summer Street  
Stamford CT 06926-0700  
USA

[www.pitneybowes.com](http://www.pitneybowes.com)