MapInfo MapX v5.0

Support Notes

November 2002

MapInfo Corporation
Troy, NY
Introduction

The following items are currently unresolved issues and common misunderstandings in MapInfo MapX 5.0. If there is a known workaround, it is listed after the issue. The five-digit number that precedes each listing is that incident's bug number. If you need to contact MapInfo Technical Support regarding any of these issues, provide the MapX Support Specialist with the bug number.

Before installing MapX 5.0, you should completely uninstall any previous version (beta, trial version) from your desktop, then reboot.

Current Bugs and Workarounds

15654

StreetWorks 3.0 tables cannot be opened with MapX 4.x or MapX 5.x.

Workaround

Use Streetworks 5.0 or change the paths in seamless layer’s component tables to include the full (rather than relative) path.

17420

Seamless layers do not display when the seamless table’s and component table’s coordinate systems have different bounds.

Workaround

MapInfo Professional has a tool “Check and Set CoordSys Bounds.” With this tool, you can set the coordinate system bounds of the seamless table to match the bounds of the underlying component tables. You will then have to click the “Save Table As...with Coordsys Bounds” button and save the seamless table to a different name.

19003

Calls to Layer.BeginAccess fail on SpatialWare for SQL Server layers. The same call does not fail for other SpatialWare layers (IUS, Oracle).

Workaround

Do not use Layer.BeginAccess or EndAccess with remote layers – the performance advantage provided by these methods is limited to native layers.

19040

Feature.Update does not always work for SpatialWare 4.5 for IUS.

Workaround

A change was made to MapX to use the last record in the subsequent select statement rather than the first. This results in better behavior, but is never a guarantee. The bug was changed to system limitation. You can handle this problem two ways:
Option #1

First, you can specify some data value(s) during the initial AddFeature statement (line 5). The data specified only needs to uniquely identify the new feature from the original feature (e.g. it doesn’t need to be unique throughout the entire table). Of course, if it is possible that there are other records with features at that same location, then the data value(s) specified when adding the new feature must uniquely identify the new record from all of the overlapping records. A possibility would be to use a dummy value for Capital (since the code changes its value anyway). Consider the following code:

```
For Each fCurrent In ftrs
    MsgBox i
    fCurrent.KeyValue = i
    sName = fCurrent.Name

    Dim newrvs As RowValues
    Set newrvs = ds.RowValues(fCurrent).Clone
    newrvs.Remove "province_name"
    newrvs.Remove "pop_1991"
    newrvs.Remove "sw_member"
    newrvs.Item("Capital").Value = "XXUNIQUEXX"

    Set ftr = newLyr.AddFeature(fCurrent, newrvs)
    ds.Refresh
    Set rvSet = ds.RowValues(ftr)
    rvSet.Item("capital").Value = sName

    ftr.Update False, rvSet
    ds.Refresh
    i = i + 1
Next
```

This particular code is rather fragile but is for illustrative purposes only.

Option #2

A better approach would be to fetch the feature using auto-increment as follows:

```
For Each fCurrent In ftrs
    MsgBox i
    fCurrent.KeyValue = i
    sName = fCurrent.Name

    newLyr.AddFeature fCurrent
    Set ftr = newLyr.Search("sw_member=" & i).Item(1)
    ds.Refresh
    Set rvSet = ds.RowValues(ftr)
    rvSet.Item("capital").Value = sName

    ftr.Update False, rvSet
    ds.Refresh
    i = i + 1
Next
```
The feature that is returned from the AddFeature method is irrelevant. But because the feature you want is the one whose sw_member value is 'i', you can now fetch the feature yourself.

15914

SpatialWare 4.6 for SQL Server 2000 does not create a layer if a query returns no records.

Server layers that access SpatialWare for SQL Server using the sp.spatial_query stored procedure (e.g., do not use the UDF forms introduced in SpatialWare 4.6) have a limitation. The query that is passed in to MapX must contain at least one row in the result set. The reason for this is that queries passed to the stored procedure that return no records do not supply any column information, which prevents MapX from being able to properly initialize the layer. If this condition occurs, this error will be returned on the call to Layers.Add: “Unable to access SQL Server dataset. Possible cause: unable to process spatial query on SQL Server which returns no rows.” The only known workarounds are to test for and handle the exception or for the client application to create its own connection and test the query first. This is, however, not necessary since catching the exception will accomplish the same thing.

17975

Calls to the Map.Refresh inside ToolUsed event cause distortion of the map image when the pan tool is used.

Workaround

It is not necessary to refresh the map after panning. Check the value of the ToolNum parameter to the ToolUsed event. If it is 1001 (miPanTool) then do not call Map.Refresh.

18573, 3398

Exported map images, the result of a call to Map.ExportMap, are sometimes distorted or the map’s title is truncated.

Workaround

For MapXtreme users, or any time the map object is not instantiated as a control on a form, set the size of the map object first (in terms of pixels) by using the Map.SetSize method, then export the map by calling Map.ExportMap without specifying the optional Height and Width.

If you have a map object embedded on a form, export it at a size whose aspect ratio is correct. Pass some multiple of Map1.MapPaperWidth and Map1.MapPaperHeight to ExportMap as the optional Height and Width parameters.

18588

Selection drawing does not line up with selected features.

Workaround

To fix this and other redraw issues try reducing the hardware acceleration (see below).
Copied from the MSDN:

“*In Windows 95/98/Me/2000, graphics hardware acceleration features can be turned off when system performance indicates incompatibility problems. Specifically, problems can occur when Windows 95/98/Me/2000 assumes a display adapter can support certain functionality that it cannot. In such cases, the side effects might be anything from small irregularities on the screen to system failure. You can disable hardware acceleration features of the display adapter so that the computer can still be used if there is a problem with the display adapter. If changing these settings fixes otherwise unexplained system crashes or performance problems, then the source of the problem is probably the display adapter. Hardware Acceleration is set to Full by default.*”

**22982**

Map symbols not displaying correctly. Occasionally point features will display with the wrong SymbolFontCharacter.

**Workaround**

In the Windows Control Panel fonts applet open (double-click) the Map Symbols font: mapsym__.ttf. Close the file and the display should be corrected.

**23295**

The following error occurs on calls to Datasets.Add with XML Datasets, “Error:1054 Dataset has no fields. No fields were successfully added to the dataset from the source data.”

We recently upgraded the MapX XML dataset driver to use MSXML 4.0. There is a change in this version of XML with respect to using XPath queries with namespaces. Specifically, documents which redefine the default namespace:

```xml
<xml xmlns="#xmldataset">
    ...
</xml>
```

cannot use XPath. Since our driver is dependent upon using XPath to query the document, we cannot support documents which redefine the default namespace. The XML document shown above should be changed to something like:

```xml
<x:xml xmlns:x="#xmldataset">
    ...
</x:xml>
```

or simply remove the default namespace definition:

```xml
<x:xml>
    ...
</x:xml>
```

Refer to [http://support.microsoft.com/default.aspx?scid=kb;en-us;Q288147](http://support.microsoft.com/default.aspx?scid=kb;en-us;Q288147) for more information as well as [http://support.microsoft.com/](http://support.microsoft.com/)
Our help file currently has the wrong information in it relative to namespaces and the structure of the XML document we support (per the above).

3352

When resizing the Map window in such a way that the width of the map is reduced, the width of the map’s title does not change accordingly.

17657

The legend is sometimes not visible when a theme is created.

Workaround

The legend position can be set using the Top and Left properties of the Legend object along with the MapScreenHeight and MapScreenWidth properties of the Map object. See code below for an example.

```vbnet
'position the legend - bottom right
Legend.Left = Map1.MapScreenWidth - Legend.Width
Legend.Top = Map1.MapScreenHeight - Legend.Height

'position the legend - top right
' Legend.Left = Map1.MapScreenWidth - Legend.Width
' Legend.Top = 0

'position the legend - top left
' Legend.Left = 0
' Legend.Top = 0

'position the legend - bottom left
' Legend.Left = 0
' Legend.Top = Map1.MapScreenHeight - Legend.Height
```

10802

Legend moves when zooming in and out on map with pie or bar chart themes.

Workaround

The legend position can be set using the Top and Left properties of the Legend object along with the MapScreenHeight and MapScreenWidth properties of the Map object.

See code sample above in workaround for 17657.

11145

When you attempt to specify the Label with Field option in the Label Properties dialog, the following error occurs: “no object was found using the index you specified.” This happens when the Label Field Dataset is set to a dataset that has more than 31 characters in the name.

Workaround

Make sure dataset names are 31 characters or less.
Note that even with a dataset name of more than 31 characters you can still accomplish this programmatically using LabelProperties.Dataset and LabelProperties.DataField. See code example below.

```vba
Dim ds as MapXlib.Dataset
Set ds = Map1.Datasets.Add(miDataSetLayer, _
    Map1.Layers(1),"veryveryveryverylongdatasetnamex")
Set Map1.Layers(1).LabelProperties.Dataset = ds
Set Map1.Layers(1).LabelProperties.DataField = ds.Fields(3)
```

21852

The FeatureFactory.BufferFeatures method may cause a crash when the feature to be buffered is a text feature not associated with a map layer. Text features created via the FeatureFactory.CreateText method must be added to a layer (Layer.AddFeature) before a buffer can be made based on that feature.

23280

Some features may not display for native tables with the CACHE=ON tag in the metadata. This tag may appear in the metadata if you perform the Save Copy As command on a live access table for which the cache is on in MapInfo Professional.

Workaround

Open the .tab file in a text editor and change CACHE=ON to CACHE=OFF. Note that you should not use cache for any native tables, the performance benefit is only realized for remote tables. When saving a remote layer as a native layer in MapInfo Professional the cache settings will be saved in the tab file metadata. If you specified cache on when opening that table in MapInfo Professional, the metadata will reflect that.

23192

A malformed Format$() expression in a call to Datasets.AddField causes a crash when you try to label layer with that field.

Workaround

Use a properly formed Format$() expression to create the new field in the dataset. For example, assume you have numeric field “X” in your dataset

```vba
' this causes a crash when setting autolabel to true for the layer
ds.AddField "test", "format$(X, ","#.#"")"
Set ds.Layer.LabelProperties.Dataset = ds
Ds.Layer.AutoLabel = True
```

The same code works nicely if you change the Format$() expression to the one below

```vba
ds.AddField "test", Format$("X", ",#.#")
```
Selection highlighting for labels is not persisted when the map view changes due to pan or zoom. This only happens when LabelProperties.LabelPartialObjects = True for the label’s layer.

Calls to FeatureFactory.CreateText fail if a layer has not yet been added to the map.

**Workaround**
Add a layer to the map before calling FeatureFactory.CreateText. The layer does not need to be in the map at the time the method is called. You can add and remove a temporary layer before calling CreateText.

A black halo on symbol features displays as white. If you set SymbolFontHalo to true and set SymbolFontBackColor to miColorBlack the halo will appear white. All other colors work as expected.

**Workaround**
Use any color other than miColorBlack, a very dark gray for instance.

SelectionChanged events execute twice when the Selection.SelectByID method is called.

SelectionChanged events are executed when the stock Select tool (miToolSelect) is clicked on an already-selected feature even though the actual selection has not changed. The SelectionChanged event is executed when the Select tool is used to click on edit handles of a selected feature from an editable layer (Layer.Editable = True) even though the actual selection has not changed.

MouseWheel panning only shows the result of the pan operation. Clicking the MouseWheel and moving the cursor away from the click point should gradually pan the map, instead the map is panned and redrawn only when the MouseWheel is released.

The map title is truncated when printing.

**Workaround**
In the call to Map.PrintMap use Map.Width, Map.Height or some multiple of those values with correct aspect ratio as the Width and Height parameters.
Layer.Bounds does not return an accurate rectangle when the Numeric Coordinate System is not the same as the Display Coordinate System (Map.NumericCoordSys, Map.DisplayCoordSys).

Workaround

Sample data needed for some Haht site sample applications is not installed by default. The Haht JavaWorld sample application uses the WORLD_Detail.gst geoset and the Haht Demographics sample application uses the MIDATLAN.gst geoset. If you have not yet installed MapXtreme you can simply select these two geosets in the component selection dialog of the MapX data installer. If you have already installed MapXtreme you can rerun the MapX data installer from the MapXtreme cd by browsing to <CDDrive>:\SERVER\MAPX\Data and running setup.exe. Select these two geosets in the components selection dialog and they will be installed to the current geodictionary directory.

When installing MapXtreme over an existing installation of MapXtreme 3.0, any changes made to any of the four default broker groups (HelloWorld, FindNearest, Themes, Default) will be overwritten and the settings will be put back to their default values. For example, if one or more of the groups is marked as disabled, or the default geoset for that group has been changed, those settings will be set back to what they were when MapXtreme was first installed. If you are using any of the default broker groups for your application you will have to restore your custom settings manually after reinstalling or upgrading MapXtreme.

The MapX data installer installs three Military Grid Reference System sample geosets. There are no options in the components selection dialog to install or not install these geosets. The geosets are MGRS_UPS_NORTH, MGRS_UPS_SOUTH, MGRS_UTM. If you don't wish to use them, delete the geosets and their component files from the data install directory after you complete the installation.

ADO dataset performance is very slow when using a server-side cursor. The performance improves dramatically when a client-side cursor is used. An ADO recordset is passed as the “Source” parameter to Datasets.Add when creating a dataset of type miDatasetADO. To avoid performance problems, set the CursorLocation property of the ADO Recordset object to 3 (adUseClient) before passing the recordset to Datasets.Add as shown below.

g_ADORs.CursorLocation = 3 'adUseClient
g_ADORs.Open
Session(SESSION_MAPPER).Datasets.Add miDatasetADO, g_ADORs
The MapX wrappers (mapx.cpp and mapx.h) that ship with the MapX C++ sample applications do not include many of the objects, methods and properties new to MapX 5.0. Updated wrappers can be gotten from the MapX website: www.mapx.com.

When saving and restoring MapX datasets using the State object, dataset columns built by using Datasets.AddField with an expression are sometimes not restored properly.

Workaround

Restore the dataset with the State object, then call Datasets.AddField again to regenerate the expression column.

The MapX runtime installer and data installer make a start menu program group for MapX.

A State object does not restore Annotations.Graphic.Style properties correctly.

The MapX runtime installer creates start menu program group and shortcuts that it should not.

Workaround

Use an easy InstallScript call to remove an existing program group:

```
DeleteProgramFolder(szProgramFolder);
```

where szProgramFolder is a STRING variable. If the program group to be deleted is the default for the MapX/Data install, then the code [which should be called _after_ all of our installers have completed running] could look something like this:

```
STRING szProgramFolder;

szProgramFolder = "MapInfo MapX";
DeleteProgramFolder(szProgramFolder);
```

You will want to capture the return value from DeleteProgramFolder() just to make sure the delete actually worked.

Differences between MapX 4.51 and MapX 5.0

In MapX 4.51 when a new table was created using a LayerInfo object of type miLayerInfoTypeNewTable, the coordinate system used to create the new table was the map’s current numeric coordinate system. In MapX 5.0 the map’s numeric coordinate system is used only if you do not specify a features collection as the “Features” LayerInfo parameter. If you do specify a features collection the coordinate system of the layer from which the features collection was built is used as the new table’s coordinate system.
Bugs Fixed in this MapX Release

23985

Cannot use miAggregationCount on a string column. The Fields.Add method allows the user to specify how values in the resulting Field will be aggregated. In MapX 4.51 a user could use miAggregationCount on a string column. This functionality did not work in MapX 5.02.19 but is fixed in this release.

23881

MapX is not reading the “LABEL\PARALLEL” = “TRUE” key in some geosets. This worked in MapX 4.51 but was not working properly in MapX 5.02.19. MapX misinterprets the value of the key. The labels display opposite of the way that they should, e.g., straight labels display as curved, and curved labels display as straight. This problem has been corrected.

23853

There is a new section in the MapX help entitled “MapX Field Naming Conventions”. The sections explains what characters can and cannot be used when naming a MapX Field object. It also explains when these rules are enforced e.g., when an error is thrown when an invalid Field name is used.

23840

SnapToNode now works properly with the AddFeature tools.

23781, 15291

Geosets with multiple raster layers now center properly when opened.

23745

Performance (speed) for panning, zooming and other map operations on seamless raster layers has been improved.

23492, 23493, 23468

Memory leaks in the OCI, DAO and XML datasets have been fixed.

23450

The Geodictionary Manager now displays the geodictionary path when you log on as a restricted user.

The new behavior in MapX 5.0 is described below.

Geodictionary under a restricted user:

The geodictionary will allow restricted users to see the geodictionary file path and registered files. No write access to registry and geodictionary file and path is given by default.

To allow restricted users to change the geodictionary file path and/or additional search paths, registry access to the \HKEY_LOCAL_MACHINE\Software\MapInfo\MapX\5.0 key must to be granted to that user via regedit.
To allow restricted users to register tab files to the geodictionary file path, access has to be given to write to that file and path in addition to the registry access.

**GeosetManager under a restricted user:**

Make a new dword key “NoGeosetToolsMenu” key under \HKEY_LOCAL_MACHINE\Software\MapInfo\MapX\5.0.

The warning (not an error) that comes up “unable to update the registry” has nothing to do with MapX. It is an error that occurs in MFC when the application attempts to register itself as a server. It fails under restricted use. This is harmless.

**23033**

The redraw issue of multiple extraneous lines appearing on the map at certain zoom levels has been corrected.

**23029, 20599**

The redraw problems caused by panning the map when there is an object in the animation layer have been corrected.

**22475**

Setting LabelProperties.LabelPartialObjects to true now produces the expected result when LabelProperties.LabelAlong is set to miLabelAlongMultiSegment.

**21259**

MapX now creates valid geometries when editing objects in Oracle Spatial.

**11800**

You can now bind data (create a dataset) on tables on read-only media, and you can perform Layer.Search on such tables.