



MapXtreme 2005 v6.5 Release Notes

February 2006

In these notes:

◆ Introduction	5
◆ Installation	6
◆ Upgrading to MapXtreme 2005 v6.5	7
◆ Focus on Best Practices	7
◆ Changes to ASP.NET Map Application Template	8
◆ Changes in the Object Model	9
◆ Bug Fixes	15
◆ Known Issues	19
◆ Printing Test Results	23
◆ Documentation Clarifications for MapXtreme 2005 v6.5 . . .	26

Introduction

This document is a supplement to the MapXtreme 2005 Developer Guide. It covers bug fixes, known issues, and other information that was not available at the time the Developer Guide went to press.

For a summary of new features in MapXtreme 2005, see Chapter 1: Introduction to MapXtreme 2005, in the Developer Guide.

Installation

The MapXtreme Developer Guide provides information on system requirements, types of installations (SDK or Deployment) and the steps to install MapXtreme 2005. Be sure to review Chapter 2: Getting Started, in the Developer Guide for complete details. See also the Online Help that is accessible from the installation dialogs.

The following are some key items to be aware of before you install MapXtreme 2005.

Side-By-Side Installations and Use

You can have more than one version of MapXtreme 2005 installed on your system at the same time. Each version of MapXtreme 2005 installs into its own directory.

You can also build a desktop and web application against an earlier version of MapXtreme and run it against a later version.

You cannot, however, run ASP.NET applications built with different versions of MapXtreme in the same process space, as in the case of IIS 5.0 on Windows XP and Windows 2000 operating systems. Each application would require its own version of MapXtreme to be loaded into IIS 5.0's single process space.

IIS 6.0 on Windows 2003 Server, however, can maintain multiple process spaces. To run your ASP.NET applications on the same computer, create an application pool for each version of MapXtreme 2005 and place the appropriate ASP.NET application in it. Restart IIS 6.0. When an application runs in its own process space, it will load the appropriate version of MapXtreme.

This does not affect desktop applications created with different versions on MapXtreme. Each desktop application always runs in its own process space.

.NET Framework SP1 Installation

MapXtreme 2005 v6.5 requires the .NET Framework 1.1 with Service Pack 1 (SP1). The installer will detect if the service pack is not found and prompt you with a Yes/No dialog to install it. Be sure to click Yes. If you click No, the installation will fail.

Install Visual Studio .NET 2003 First

It is recommended that before you install MapXtreme 2005 you should have Visual Studio .NET 2003 installed. If you do not have Visual Studio .NET 2003 or the Microsoft .NET Framework installed, the installer will prompt you to install the Microsoft .NET Framework, and will then install it prior to installing MapXtreme 2005.

Note: If you do not have Visual Studio .NET 2003 installed, some of the sample templates will not be installed.

Windows 2000 Post-Service Pack 4 (SP4) COM+ Hotfix Rollup Package 27

If you are running Windows 2000 and intend to use MapXtreme 2005 in a Web-based application, you must obtain the Microsoft Windows 2000 Post-Service Pack 4 (SP4) COM+ 1.0 rollup package 27. For more information, see the Microsoft KnowledgeBase Article - 822618 at <http://support.microsoft.com/?id=822618>.

Permissions to Temp Directory and other Resources

For deployed web applications, if you are using integrated security, make sure all users who will access the site have permissions on the temp directory and any other necessary resources. MapXtreme 2005 executes in the ASP.NET process space and this process executes using the security token passed from the IIS process. You will have to grant access to any user who will log in access to this directory. If you are using anonymous access then you will have to grant access to the temp directory to the IUSR_ *LocalMachineName* system account. MapXtreme 2005 gets the temp directory from the current TEMP environment setting.

Upgrading to MapXtreme 2005 v6.5

An upgrade of MapXtreme 2005 v6.5 installs into its own directory. It will not overwrite a previous version. This allows you to maintain side-by-side installations of v6.5 with a previous release. A side-by-side installation requires that you copy your old license file to the new v6.5 location.

Note, when using a side-by-side install, you must close Visual Studio .NET before opening a MapXtreme project of a different type. For web applications, you will need to reset IIS (use the iisreset.exe command in a console window or from the Start > Run menu option.)

Refer to the MapXtreme 2005 Developer Guide for more information on updating desktop and web applications.

Focus on Best Practices

This release of MapXtreme 2005 emphasizes best practices for developing enterprise-level scalable and high-performance mapping and data access applications. Chapter 5: Planning an Application, in the Developer Guide, provides an in-depth look at pooling, state management, performance tuning and data access. By understanding these best practice concepts you can make the best design choices before you begin coding.

In addition, we have re-vamped our web application template to follow these best practices. See the next section for a discussion of the changes.

Changes to ASP.NET Map Application Template

The ASP.NET Map Application template for MapXtreme 2005 has been enhanced in two significant ways: for scalability and performance and in its design-time behavior.

Pooling and Manual Session State

The following are the new default settings for the MapXtreme 2005 ASP.NET Web Application template:

```
<add key="MapInfo.Engine.Session.Pooled" value="true" />
  <!--Use this setting to save Session state automatically (HttpSessionState)
or manually (Manual)-->
  <add key="MapInfo.Engine.Session.State" value="Manual" />
<sessionState mode="StateServer" stateConnectionString="tcpip=127.0.0.1:42424"
sqlConnectionString="data source=127.0.0.1;userid=sa;password="
cookieless="false" timeout="20" />
```

Web applications built using values in the template's web.config file will now share session objects and allow the application to manage state on its own. Previously, the template did not use pooling and automatically saved the entire session object. This was only useful for small scale applications, like prototypes, or where the number of users of the application was known, as in an intranet setting. Pooling and manual session state are efficient means of application design that support scalability and performance tuning. See Chapter 5: Planning Your Application in the MapXtreme 2005 Developer Guide for a complete discussion of pooling, state management.

Design-Time Changes

The design-time behavior of the web templates has changed. The MapControl and LayerControl that display on the form when the template is opened, support runtime rendering only. At a minimum, three elements must be provided in order to build a web application with this template: a MapAlias, MapControlID and a pre-loaded workspace. We provide default settings to get you started: `Map1` is the MapAlias, `MapControl1` is the MapControlID and `world.mws` is the pre-loaded workspace.

MapAlias

The properties `Map1` and `MapControl1` are stored in the template's `WebForm1.aspx`. To change the MapAlias or MapControlID use the Properties window in Visual Studio .NET. The MapAlias identifies the map to display at run-time. The MapControlID is used by the LayerControl and other map tools to ensure they are associated with the correct MapControl.

If a MapAlias is not provided or if the MapAlias provided does not exist, the MapControl will display the first available map listed in your pre-loaded workspace. During design-time, if you remove and replace the MapControl, the template does not assume the MapAlias remains the same. It may also be NULL. In this case, MapControl still displays the first available map in the preloaded workspace.

Because of the new state management code in the template, it is important that the MapAlias remain consistent at run-time. The MapAlias is the key to tying a user's settings to its session. If you need to change the MapControl's MapAlias at run-time, you must provide additional code for your specific needs.

Pre-loaded Workspace

The world.mws and its related sample data files are automatically installed on your system when you install MapXtreme 2005. The workspace is identified in the web.config file for this template, as shown in bold below.

```
<!--Use this setting to preload a workspace on Session creation-->
  <add key="MapInfo.Engine.Session.Workspace" value="C:\Program
Files\MapInfo\MapXtreme\6.5\Samples\Data\world.mws" />
```

To use a different workspace with this template, create the workspace using Workspace Manager and edit the web.config file to point to the new workspace.

For more information see Chapter 2: Getting Started, in the MapXtreme 2005 Developer Guide.

Changes in the Object Model

This section alerts you to new classes, methods, properties and behavior in the MapXtreme 2005 Object Model.

Web Controls

Web controls have been re-architected for MapXtreme 2005 for easier use. The new web controls and tools are provided in a new namespace called MapInfo.WebControls. The old web controls in MapInfo.Web.UI.WebControls are still available for use.

New controls include 8 pre-set navigational tools for panning (N, NE, E, SE, S, SW, W, NW) and a zoombar control for zooming at pre-set levels.

Source code and sample applications are provided so you can learn how they work and modify them for your own needs. Find the source code in ..\MapInfo\MapXtreme\6.5\Samples\Web\WebControls.

The following compare and contrast the old architecture of MapXtreme 2004 with the new architecture of MapXtreme 2005 web controls.

MapXtreme 2004 Web Controls

- Used a posted page callback that called Page_load and Page_unload every time a tool was used.
- The MapControl loaded the workspace at design time and at runtime.
- State Management was done in Page_load and Page_Unload.
- Error processing was done by ASP.NET. Since the page was re-rendered every time, an exception would be caught by ASP.NET and an error page was generated.
- Event handlers for selection and distance tool controls were handled in the Page class.

MapXtreme 2005 Web Controls

- The first time a page is rendered, Page_load and Page_unload are called. Every time a tool is used there is no postback.
- A preloaded workspace is used. At design time, you can set the MapControl MapAlias property to the map alias of the map defined in the pre-loaded workspace. This allows the corresponding map to be loaded into the MapControl at runtime.
- The StateManager is implemented in an application as a class, and an instance is put in the ASP.NET session. SaveState and RestoreState methods are called from this object. The SaveState and RestoreState methods are called every time a tool is used and the LayerControl is updated only if the ASP.NET session includes a StateManager class instance. If Manual mode is used for state management, a StateManager class instance must be in the session. (MapInfo.Engine.Session.State is set to Manual in your web.config file.)
- You have control over how error processing is performed. The error processing is done in the global_asax.cs/.vb file in the application_error event handler.
- Since a Page is not reloaded when a map tool is used, the event handler (as implemented in MapXtreme 2004) cannot be used. For MapXtreme 2005 Web controls, you must write a client-side Javascript command to send a request to the server, and to get and process the response. You must also write your own server-side command class to do server-side processing for the tool. And at design time, you must set your tool's ClientCommand property to the function name in your Javascript and set the Command property to the server-side command class name.

All controls are rendered when the page is rendered at initialization time. After initialization time, when a tool is used, only the map image is rendered. For example, the zoom tool rectangle is drawn, then after collecting the data, a URL is formed using **img.src = url syntax** and only the map image is updated. When the URL is used in the source, the request is sent to the server. On the server side, the request is parsed and the operation is performed. The image is exported and written to the response as a stream that is returned to the client. This way, you can write visually more appealing Web applications.

Customizing Web Tools

New properties are available so that it is easier to customize a web tool. Start by modifying one of our existing map tools, or the tool named "WebTool", which is designed for creating custom tools. You only need to:

- Write a server command class (if the existing one is not adequate) to process server-side behavior for your tool.
- Write a Javascript client command (if existing one is not adequate) to process client-side behavior for your tool.
- Write client interaction, if needed.
- Then just drag the tool or WebTool to your webform, and set its Command, ClientCommand, ClientInteraction, InActive/ActiveImageUrl, CursorImageUrl, and any other properties for your custom tool.

For a complete discussion of the new web control architecture, see Chapter 6: Working with Web Controls, in the MapXtreme 2005 Developer Guide.

Routing Client

The MapXtreme 2005 routing client now includes the ability to calculate drive-time and drive-distance isograms, and the ability to use transient updates to recalculate routes while avoiding, or changing the priority for certain road types, points, or segments. Matrix routing is now supported, allowing you to find the shortest or fastest paths between any number of start points and end points. To support delivery system type routing, time-based routing has been added, allowing you to specify start, stop, and end times. The routing client now includes a number of preferences that allow you to customize your route analysis, driving directions, and route geometries. The changes to the routing object model are many. For more information, refer to the MapInfo.Routing namespace in the Developer Reference and Chapter 19 in the Developer Guide.

Geocoding Client

MapXtreme 2005's geocoding client now supports the MapInfo world geocoder, providing city, postal code, and street level data coverage for numerous countries. A batch request property, geocoding constraints and address level options have been added, such as address ranges, resulting in more customized and accurate geocoding candidates. Included in these constraints is the addition of CASS certification, a process by which a table of mailing addresses is standardized to meet U.S. Postal Service® (USPS) requirements for bulk mailing discounts.

The object model changes include:

- The geocoding client now supports address unit properties returned from the MapMarker Java Server. MapInfo.Geocoding.AddressUnit.Type and AddressUnit.Value are part of the address returned from MapMarker if the address contains a Unit. This typically includes units such as apartment numbers.
- MapInfo.Geocoding.IGeocodeClient.BatchRequests is a new property that sets the number of addresses that will be sent to the server during a single operation. The default value is 25. The value must be greater than 0 and less than 500.

Please refer to the MapInfo.Geocoding namespace in the Developer Reference and Chapter 18 in the Developer Guide.

Transient Feature Support

Transient FeatureGeometry objects are now supported in the object model. The Boolean property MapInfo.Geometry.FeatureGeometry.IsTransient can be used to determine if a FeatureGeometry is transient.

FeatureGeometry objects obtained through some data access methods may be transient. These are short-lived objects where the memory may be reclaimed and used during subsequent fetch operations. The use of transient Geometry objects can improve performance. You should make a copy of the object if you need to keep it beyond its short life.

As a result of the new Transient feature capability, the SearchResultProcessor class has been updated to support it. This base class, for which you can define your own post-processor of search results, now returns transient features. Note, however, any returned transient features are only valid while on the current feature. Once you move to the next feature, the previous feature is invalid.

ISession.Reload and ISession.Clear methods

MapInfo.Engine.ISession.Reload is a new method that clears the state of the Session and reloads a workspace that is listed in the application's configuration file. ISession.Reload replaces ISession.Reset.

MapInfo.Engine.ISession.Clear is a new method that resets the state of the session without reloading a pre-loaded workspace.

Geometry Changes

Point in MultiPolygon Behavior

MapXtreme 2005 provides improved Point selection in MultiPolygons that have overlapping Polygons. Previously, a Contains query would fail because the application was treating the overlapping Polygons as one aggregate object. Now when MapXtreme 2005 determines that the Point is within any Polygon, it satisfies the Contains query for the MultiPolygon and concludes the process.

This behavior also pertains to MultiPolygon selection. Previously, if you tried to select the MultiPolygon in the overlapping area, the application would consider the Point selection outside the MultiPolygon and would not select it. MapXtreme 2005 now evaluates that Point to be inside the MultiPolygon and correctly selects it.

These changes are related to the more general issue of how MapXtreme performs a Point in Polygon test. For a Polygon consisting of a single exterior boundary (Ring) and any number interior boundaries, there is no change in the algorithm. But for MultiPolygons, previous versions of MapXtreme could produce erroneous results if a Point was in an area of overlapping Polygons. To correct this, MapXtreme 2005 now treats a MultiPolygon as a series of Polygons (a single exterior Ring boundary and all of its interior Ring boundaries). It checks each Polygon independently to see if the Point is inside the Polygon. If it is inside any of the Polygons, then it recognized as inside the MultiPolygon.

This also produces a performance improvement, since MapXtreme 2005 stops the processing as soon as it knows the Point is inside a Polygon.

SearchWithinDistance method

MapInfo.Data.SearchInfoFactory.SearchWithinDistance now uses the MapInfo.Geometry.FeatureGeometry.Distance() method instead of buffering a Geometry with a given distance, and then searching within the buffer.

Previously, MapXtreme checked if the object was within the buffer. So the object had to be completely inside the distance. (That is, if the farthest point on the object was more than the distance value away, then the object would *not* be retrieved.) This was similar to asking if the object *intersects* the buffer, rather than being within a given Distance.

This change provides more accurate SearchWithinDistance results, and does not suffer from the approximations of the buffering method.

SelfIntersections

For MapXtreme 2005, the SelfIntersects property and SelfIntersection method are now defined on the Geometry class. These methods were previously only available for LineString, Curve, MultiCurve, Ring, Polygon, and MultiPolygon classes.

MapInfo.Geometry.SelfIntersection now returns a FeatureGeometry instead of a MultiPoint. This change was necessary as the method may now return a MultiPoint, a MultiPolygon, or a FeatureGeometryCollection which contains a MultiPoint and a MultiPolygon.

A self-intersection is a Geometry object that crosses itself. With these properties available to all objects in the Geometry class, you can search your data for these objects and remove or fix them. If they remain in your data, operations such as aggregations based on their Geometry, may return incorrect results.

Geometry Editors

New Get methods have been added to the Geometry API to replace the functionality provided by the properties of the Geometry.GeometryEditor and related Geometry classes.

The new Geometry.GetGeometryEditor() returns a reference to a class that implements the IGeometryEdit interface.

The old properties will now produce a compile error and throw a System.NotSupportedException. A message will indicate which new method should be used in place of the old properties.

Object Editing Interfaces

New methods have been added to the Edit interfaces to allow you to access the base object. For example, the IsGeometryPart property of a MapInfo.Geometry.Geometry object will return a boolean indicating if it is part of another geometry object.

With these new methods you can know, for example, if an editable Curve is part of a MultiCurve by calling:

```
curveEdit.Curve.IsGeometryPart
```

Curve is the new property on ICurveEdit which returns the base Curve object.

CoordSys.Equivalent method

The MapInfo.Geometry.CoordSys class provides a new overload on the Equivalent method that allows a concerned CoordSys object to be compared to a target CoordSys object, without regard for the bounds of the coordinate systems themselves. Previously, the method would compare both CoordSys objects and expect to find the same elements in both coordinate systems, which wasn't always the case.

LayerControl for WinForms

MapXtreme 2005 includes several enhancements for `MapInfo.Windows.Controls.LayerControl`.

- A `StyleOverride` node can be dragged and dropped onto another layer to copy the style to that layer. The layer must be of comparable type (i.e., you cannot drag a raster style override onto a vector layer).
- Right-clicking on a node in the layer tree now selects the node. Your subsequent actions will be applied to that layer. Previously it was unclear whether the layer was selected.
- ToolTips can now display information on a per-layer basis instead of one generic string for all map layers.

ExcludeLegacyText

`ExcludeLegacyText` is a new boolean property on `MapInfo.Data.ClosestSearchResultProcessor`. When set to true (default), `LegacyText` objects are not included in a `Data.SearchInfo.Factory.SearchNearest()`.

If `ExcludeLegacyText` is set to false, then `LegacyText` objects are included in the search. The `LegacyText.TextBounds` will be used to search against.

By contrast, the `Mapping.SearchInfoFactory.SearchNearest()` method, by default, does include `LegacyText` objects in its search (`ExcludeLegacyText` is false). This allows such things as text selection in web applications. Note, however, that the selected text may not be what you would expect as the closest text to the selection point, especially if the text is rotated.

Drawing Tools

The `MapInfo.Mapping.LayerHelper` class contains new methods `HasInsertable`, `GetInsertable` and `SetInsertable` that are used to check whether a drawing tool has an insertion layer set. If `HasInsertable` returns true, then the drawing tool can add features to the layer.

Bug Fixes

Coordinate Systems

The coordinates for the bounding box of a raster layer in Latitude/Longitude referenced in Gauss-Krüger DHDN Zone 2 now display the correct coordinates in the capabilities.xml file returned from a WMS server.

The MapInfo.Geometry.CoordSysFactory overloaded constructor now correctly handles a projection file provided to it in xml format. For example, if you pass in the path to the MICoordinateSystemSet.xml that ships with MapXtreme 2005 (c:\Program Files\Common Files\MapInfo\MapXtreme\6.5), you will no longer get an error message.

Data Access

The MapInfo.Data.AddColumn method will now successfully add a large column of data from an MS Access table to a table in native .TAB format.

TableInfoFactory.CreateFromFeatureCollection method has been modified to set ReadOnly property of columns is false. Columns in a view from a join are read only by default.

The alias from a CreateFeatureCollection is now handled properly to allow a table insert.

The MapInfo.Data.TableInfo.WriteTabfile method is now working properly when using CreateFromFeatureCollection.

The problem with opening a MI Pro Live table on SQL Server has been corrected.

The problem with opening a View table that was created from component tables has been corrected.

The issue when MIDataReader returned a wrong key that was returned by a Feature has been corrected.

The IResultFeatureCollection Add and Remove methods that accept a Feature are now working properly.

In the situation where a data query produces candidates for a MultiResultSetFeatureCollection, and these candidates are filtered out by the SearchResultProcessor to yield no results, the MultiResultSetFeatureCollection will also be empty. This now works as expected.

Using the Select tool to select a Feature on a layer that was added as a result of an MICommand now works properly. The ResultSet was closing after it was added to the map. It now remains open, as long as the base table from which it was created, stays open.

Geometry

An issue involving FeatureProcessor.Combine method has been resolved.

Layers

Animation layers that contain regions or lines are now visible when set.

Removing a Layer from a GroupLayer now works properly.

AutoPosition

AutoPosition and StartPosition properties on the MapInfo.Mapping.MapLoader class now operate correctly when adding layers from a workspace, geoset or table.

When AutoPosition is true, the Layers class will determine if the layer should be automatically positioned or kept in their original order. (note: GroupLayer will be always added on the bottom of layers, because Layers can not smartly arrange it.)

If AutoPosition is false, layers are inserted at the location specified by the StartPosition property.

Legends

The CustomLegendFrameRow class is now serializable, meaning that you can restore a custom row in your legend after saving it, just like a theme or cartographic legend. To determine whether a class is serializable, look for the ISerializable property in the Developer Reference help.

AutoSizeLegend property is true by default and will properly size the legend.

Map Control

RightToLeft Property

The RightToLeft property on the MapControl now works properly for all valid conditions (Yes, No, Inherit) when you marquee select with zoom tools.

Refreshing Layers in a MapControl

Calling MapControl1.Map.Invalidate() fails to refresh the layers after an update is made. To refresh a label layer, you must call LabelLayer.Invalidate(). To refresh other layers, call MapLayer.Invalidate(). The following is expected behavior when using Invalidate and Refresh.

- LabelLayer.Invalidate() = Will not update the layer until you click in the map causing the map to redraw.
- LabelLayer.Refresh() = Immediately updates the map.
- MapControl1.Map.Invalidate() = Will not update the layer until you click in the map causing the map to redraw.
- MapControl1.Refresh() = Will not update the layer until you click in the map causing the map to redraw.

MI SQL Language

MI_Buffer resolution is between 3 and 100 inclusive.

MapInfo SQL function NumberToString is now returning correct results.

An issue with an SQL query running under MICommand based on an MI_Area calculation that used MI_Intersection as the geometry parameter has been resolved.

AreaOverlap

The documentation for the MI SQL aggregate functions ProportionSum, ProportionAvg and ProportionWtAvg clarifies the use of an expression for AreaOverlap in the remarks section. AreaOverlap is not a defined function in the MI SQL. It is provided strictly as an example of use for the proportion functions.

Routing

Routing with MapMarker EMEA (non-North American) applications now returns result codes.

Scalebar

ScaleBarAdornment.Rounding property has been added to the object model to indicate whether the numbers displayed in the scalebar should be rounded. If this property is set to true, the scalebar numbers are rounded up to the nearest power of 10, which is closest to the map width value.

Note that the default exponent is never calculated as a negative value. This will cause the values to be whole numbers. However, when the map width rounds to an odd number, then the middle value will have a .5 (i.e. 0 - 11.50 - 23). This is expected behavior.

ScaleBarAdornment.PaperWidth property now works correctly for values less than 0.5.

The scalebar adornment now retains its values (does not reset to 0.0) when calling MapInfo.Mapping.Map.SetView() and then setting it back to its previous view.

Searches

SearchWithinDistance now uses MapInfo.Geometry.FeatureGeometry.Distance instead of a buffer resolution to ensure points within the distance are captured.

SearchNearest now works properly for layers containing LegacyText objects.

Selections

Selections now re-draw correctly when the display unit of the map is changed. Selection drawing code now correctly converts units when zoom min and max values are being compared to the current zoom.

MapXtreme completes the operation without error when you are attempting to remove all items from the current DefaultSelection. The table must remain open.

MapInfo.Mapping.MapExport.ExportMap.ExportSelection property and MapInfo.Mapping.FeatureViewer.DrawSelections now work properly. The selection highlighting will not display or be exported when these properties are false.

Styles

Workspace Manager's Print Preview and printing a layer of symbols using a custom symbols override style has been fixed (no longer produces black symbols).

Setting the halo color to black for white text objects has been fixed. This has also been corrected when using this effect as a style override.

Font.TextEffect is now working properly when set to Halo, Box or None.

Custom Symbols are now installed with the MapXtreme Software Development Kit (SDK) in the c:\Program Files\Common Files\MapInfo\MapXtreme\6.5\CustSymb folder. For deployed applications, you must add the MapInfoCustSymb.msm to your package if you wish to use these custom symbols in your application.

FontPointSize.PointSize

To get or set the point size of a MapInfo TrueType font symbol, you must use the FontPointSize.PointSize property instead of the Font.Size property. The online documentation for explains the following:

The PointSize property is used to get (or set) the point size value used by a FontPointSize object. The value of the PointSize property (which is of type "double") takes precedence over any other "point size" settings that may exist. For example, if a Font object having point size 24 is assigned to the Font property of a FontPointSize object, but the PointSize property itself is instead set to be 20, then a point size of 20 is used; i.e. the size of the Font object is ignored.

Tables

MemTable Serialization

Key values are now reset properly when deserializing MemTables in a web application. Previously, the key value would continually increase.

Inserting NULL_STYLE Style Objects

MapXtreme 2005 now properly checks the NULL_STYLE style when determining whether or not an insert into a table will be successful. A successful insert will happen if:

- the object is non-NULL and the style object is NULL
- the object is NULL and the style object is also NULL

You may not insert a row with a NULL object and a non-NULL style.

ResultSets and Views

A View table created from component tables in .TAB format opens correctly now whether or not the table name includes the .TAB file extension.

Table Alias

For table aliases that are named with leading numerals, the Catalog.GetValidTableName method now prepends an underscore '_' so that the alias can be parsed correctly. This affects all table types. In the case of seamless tables, the alias refers to the component table that makes up the seamless table.

Setting the Alias property of a TableInfo object now works correctly.

WriteTabFile

WriteTabFile is not supported for ESRI Shape files. If called, a NotSupportedException displays. This also applies to Server, ResultSet and WMS tables.

The WriteTabFile method for MS Access tables now correctly writes out the TABLE clause.

Columns

MapXtreme now correctly updates an ID column with the Feature's key whether the Feature is from a table or a ResultSet.

MIDataReader now returns the same column names as TableInfo.Columns when using a TableInfoView.

MapXtreme now opens MS Access tables. It ignores any unsupported fields, such as memo fields and numeric fields defined as decimal.

Tables with zero rows on MS SQL Server can now be opened successfully in MapXtreme 2005.

Catalog.OpenTable for a TableInfoAdoNet has been modified so that the key column field is checked in a case-insensitive manner.

Supplying an invalid table name in Catalog.OpenTable will now display a more appropriate error message "Unable to open table: File not found: filename." Previously this action would generate a raster engine error message.

When MapXtreme attempts to open a Shapefile containing an incorrect field specification (such as width), a more specific error message displays now indicating the problem field.

A MapInfo_MapCatalog table is no longer required in a database that contains non-mappable tables.

Themes

AllowEmptyString property for IndividualValueTheme and IndividualValueLabelTheme now works correctly. This property determines if a bin with zero value will display in the theme legend.

Dot density themes now use two display pattern algorithms to represent data values. For layers containing single Polygons or non-regions, the dot display pattern does not change. Multi-Polygons use a new algorithm that calculates the dot density based on smaller polygons with holes rather than on Multi-Polygons as a whole. This allows the data for these objects to be more accurately represented.

The DistanceUnit for an ObjectThemeLayer's VisibleRange property now sets correctly to use the theme's zoom units rather than the default for the layer.

A DistanceUnit of a millimeter for pie size now displays the pie in the legend at a reasonable size.

Known Issues

The issues that follow are those identified since the previous release of this product. They are either currently under investigation or in active development seeking a resolution.

Data Access

MapXtreme returns an error if you attempt to insert a feature into a table when the entry in the MapInfo_MapCatalog for the table has the RenditionType set to 0 and the RenditionColumn set to null.

MapXtreme will not load a remote database table containing an MI_STYLE field if the RenditionType is set to 0 and the RenditionColumn is set to NULL in the MapInfo_MapCatalog.

Executing an update query on an MS Access table fails to insert style and object. For example:

```
insert into custpointswest (Obj, MI_Style) values ( newobject, newstyle))
```

However, if you only insert the Obj field, the query works, as below:

```
insert into custpointswest (Obj) values ( MI_Point  
(-121.331658, 38.546608, 'EPSG:4326'))
```

MapXtreme Web applications that access Microsoft Access databases may experience failed connection attempts. This is due to a limitation in Microsoft's Jet database engine when used in a high-stress, 24 x7 server environment. The Jet database engine, the engine behind MS Access, is limited to 64 concurrent connections. This includes the number of tables across all threads. In high-stress web environments, the actual number of successful concurrent connections may be less. For more information, see the following Microsoft technical publication:

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;q222135>.

Microsoft recommends using MS SQL Server with IIS for web applications that require absolute data integrity or high user concurrency.

Features

Region Features have a white border around them even though null border has been selected. To avoid this, set the border option to be a solid line of the same color.

Labels

The rotation handles that display for labels and legacy text that are selected do not operate. Additionally, the anchor point position cannot be changed.

Label visible range is not inclusive at MaxZoom in some situations. The visible zoom level varies due to the coordinate system, possibly due to rounding calculations.

Legends

The BarTheme legend does not update to show scale after the map zoom level is changed. In place of the scale, a "Not drawn to scale" message displays.

The LegendFrame.Rows.Visible property, when set to 'false', does not turn off the legend row as expected.

Sample Code

In the MapXtreme Developer Guide, the VB code example for customizing the AddPolygonTool does not compile. Use the following code in its place:

```
Dim insertionlayerfilter As IMapLayerFilter
Dim style As MapInfo.Styles.CompositeStyle
Dim addmaptoolproperties As MapInfo.Tools.AddMapToolProperties
Dim maptool As MapInfo.Tools.MapTool
insertionlayerfilter = _
MapLayerFilterFactory.FilterByLayerType(LayerType.Normal)
style = New MapInfo.Styles.CompositeStyle

addmaptoolproperties = New _
MapInfo.Tools.AddMapToolProperties(MapLayerFilterFactory.FilterForTools _
(MapControl1.Map, insertionlayerfilter, _
MapLayerFilterFactory.FilterVisibleLayers(True), _
"CustomPolygonAddMapToolProperties", Nothing), style)

maptool = New MapInfo.Tools.AddPointMapTool(MapControl1.Viewer, _
MapControl1.Handle.ToInt32(), MapControl1.Tools, New _
MapInfo.Tools.MouseToolProperties(Cursors.Default, Cursors.Default, _
Cursors.Default), MapControl1.Tools.MapToolProperties, _
addmaptoolproperties)
```

Serialization

You cannot deserialize a ASCII PointRef table if the reference table is closed. You must first deserialize the pointRef table and then close the reference table.

When using ranged themes, do not clear the theme before deserializing it. This will result in differences in the theme properties between the serialized and deserialized theme.

Styles

The black halo for a Text style is not displaying correctly. This is observable in the TextStyleDlg and StyleSampleBitmapButton object (button on the Style tab). As a workaround, choose a very dark gray as the halo instead of black.

The Symbol style for a Graduated Symbol theme is not properly reflected in the Layer Control. Graduated Symbols appear correctly on the map but are represented by a different symbol in the Layer Control.

Tables

MapXtreme is unable to insert features from the result of a join. You can save the results as a TAB file by creating a view table (MapInfo.Data.TableInfoView) instead of directly inserting features (MapInfo.Data.Table.InsertFeature). The resulting view table can then be added to the map.

TableInfoView.WriteTabFile() doesn't write out the CharSet property, resulting in the default value (lat1) being used.

Inserting records into a TableInfoServer table using MICommand takes a long time for a large number of records.

Tools

ToolTips do not display when you hover over a feature on the map with the select tool. The InfoTips will display if you select the feature.

Viewing the Learning Resources in Netscape

If you are running Netscape as your default browser, the Learning Resources may not initially display properly. If you launch the Learning Resources and it appears with blue links and missing image icons, right-click anywhere within the Learning Resources and select **DISPLAY LIKE INTERNET EXPLORER** to correct the display issue. Once you select this option, the Learning Resources will always display correctly for you. This option will not affect your viewing preferences for other browser content or web sites.

Web Applications

In a web application, if you zoom in on one area of a map, click the Back button, and zoom in on another area of the map, the previous zoom view displays. Using the new WebControls, the Forward and Back browser buttons are disabled.

The web InfoTool selects features incorrectly and inconsistently. The expected behavior is that the tool selects objects that intersect the select point within a certain pixel tolerance. Currently it selects objects that have a minimum bounding rectangle (MBR) that enclose the clicked point. These MBRs may have larger or smaller tolerances. This feature does not exist in the new Web MapControl.

The ExitProcess key in the web.config file is set to 'true' by default, which may cause an error when debugging your application. A Microsoft issue required that the setting default to be 'true', but that issue has since been resolved. Override the default value for the ExitProcess key by setting it to 'false'.

The Web tools for distance and polygon selection causes MapXtreme to crash after multiple uses of the tools. This is not specific to MapXtreme, but to a bug in Microsoft's vgx.dll. According to Microsoft, when you view a Web page in Microsoft Internet Explorer 6.0, IE stops responding or crashes, and you receive an access violation in vgx.dll. This problem occurs if the Web page renders vector graphics on the screen and if your computer is running Microsoft Windows XP Service Pack 2 (SP2). For more information see <http://support.microsoft.com/?scid=kb;en-us;885932>. A workaround for those who do not want to install the Microsoft KB885932 fix is to increase the COM+ activation time (for example, from 60000msec to 120000msec).

Web Controls

You must set a MapAlias value every time you add a MapControl to a WebForm. If the MapAlias is not set at design time, your application will not run properly, resulting in a crash. Further, if you change the MapAlias value at runtime, an error will result from the application looking for the original MapAlias.

Setting the PointSelection tool's PixelTolerance property to 0 (zero) will result in a crash at runtime when you use the Point Selection tool. The exception that is thrown states "Specified argument was out of the range of valid values." Use a positive integer when setting PixelTolerance.

When using the Pan tool at runtime, right-clicking in the map will cause the shortcut menu to display; however, if you left-click off the menu to hide it, the map will pan from your first to second click, regardless of the first click being a right-click. Similarly, when using the Zoom In or Zoom Out tool at runtime, the map will zoom either in or out using the same right-click scenario, as though both clicks were left-clicks.

WMS/WFS

Workspace Manager encounters an error when you select **MAP > VIEW ENTIRE LAYER** for a WMS TAB file.

Miscellaneous Issues

Putting a message box in a window (form) constructor prevents marquee zoom in and zoom out tools from working. To avoid this situation, put the MessageBox call in the form's Load event handler method.

Making a selection on a remote table and then exiting the application throws an error. This happens only when running the application from within Visual Studio. Your application, when run on its own (either in debug or release mode), will not produce an error. The error stems from the remote table being left open with an active selection when you exit the application.

MapXtreme experiences a slowness when creating Ranged Themes. Loading a range theme on a large table takes longer than expected, and changing range types or bins takes considerably longer to complete. Because there is no progress indicator for this operation, it may appear that your application has locked up, but the operation will complete in time. Creating ranged themes against non-indexed columns appears to take less time than against indexed columns.

MapInfo.Geometry.DisplayTransform.ToDisplay method returns incorrect values.

Printing Test Results

The following are observations made during the development and testing of MapXtreme 2005's new printing feature. These results focus on the behavior when using special drawing attributes. The output may or may not be as expected since many factors are involved (printer driver, print method and drawing attribute). This section will help you make some informed choices.

For an overview of printing from MapXtreme 2005, see Appendix H in the Developer Guide. For the online object model documentation, see the MapInfo.Printing namespace.

Map Drawing Attribute Test Results

Map size, paper orientation, and paper size had no affect on printing results. Drawing attributes, print method (to EMF or direct to device), and choice of printer driver do affect the printed output and the display in Print Preview. The following describes our testing results for these drawing attributes:

- Polygons with Holes

- Transparent Raster
- TrueCcolor
- Bitmap Pattern Scaling
- Transparent Vector Patterns

Polygons With Holes

Setting the value of `DrawingAttributes.SpecialPolygonHoleHandling` property to true and printing with a PCL printer driver will correctly print polygons with holes. If printing with a Postscript driver, you may see lines stretched across the hole, but they are less noticeable than if `SpecialPolygonHoleHandling` is set to false. The lines may be visible in PrintPreview.

Set this value to false when you are drawing to the screen.

Transparent Raster

When printing to a printer using a PCL driver, choose either print method (to EMF or direct to device) and set `DrawingAttribute.SpecialTransparentRasterHandling` property to true. Setting this to false for either print method will print the transparent area of a raster image black.

Tips for Printing Transparent Raster Layers

The following can be tried if your map containing transparent raster layers print black. Note transparent raster printing is printer driver dependent. Not all printer drivers handle the raster operation (ROP) successfully.

- Choose `MapPrinting.Print` method to print to enhanced metafile (EMF).
- Use `MapXtreme's` ROP method to display transparent raster. Turn off special handling of transparent raster (set to false) when printing.
- Print in true color (24-bit) if possible.
- For printing to HP plotters, use HPGL/RTL driver instead of a PostScript driver.

Translucent Raster

Printing translucent raster images from MapXtreme 2005 is not supported.

True Color

When printing images that have more than 256 colors to a device that only supports 256 colors, dithering is required. There are 2 types of dithering available - halftone and error diffusion. They each produce images that resemble the original image, except that they are using less colors. You'll have to determine which method is more appealing to you.

Bitmap Pattern Scaling

MapXtreme 2005 supports two types of hatch patterns that are used to fill polygons. If you look at the Area Style dialog in Workspace Manager, you will see samples of the available patterns (Right-click on a polygon layer and choose Add Style Override. Click the Area Style button on the Visibility tab to show

the Area Style dialog). The first 7 patterns are Microsoft standard vector hatch patterns. The remaining fill patterns are MapInfo's bitmap hatch patterns. Each type of pattern behaves differently when scaling for print.

Scale is printer-driver dependent; however, not all printer drivers support scaling. You must decide whether MapXtreme or your printer driver will handle it. Choosing not to scale at all results in printed output containing very small bitmap patterns that almost look like a solid color. When MapXtreme or the printer driver handles scaling, the printed result should look like the pattern on screen. If both MapXtreme and the printer scale, the printed output will be abnormally large.

To enable scaling from MapXtreme, programmatically set the `DrawingAttributes.ScaleBitmapPatterns` property to true. Patterns will be scaled up to compensate for the difference between printer and screen resolution.

To disable scaling from MapXtreme, set the `DrawingAttributes.ScaleBitmapPatterns` property to false. Your printer driver will take over if it supports scaling.

Note: This discussion covers scaling for non-transparent bitmap fills. Transparent fill patterns are always scaled.

When you print using a PostScript driver Language Level 1, disable scaling in MapXtreme. You will notice that the Print Preview will display larger than the printed output.

For PCL 6 printer drivers, turn off scaling in MapXtreme. Print Preview and print output will be the same quality and relative scale.

For PCL 5e printer drivers, enable scaling from MapXtreme. The printed output will scale properly.

Printing With Transparent Vector Patterns

When using a PostScript driver, `SpecialTransparentVectorHandling` must be set to true, otherwise MapInfo transparent fill patterns will print black.

When using a PCL 5e driver, `SpecialTransparentVectorHandling` must be set to true, otherwise fill patterns will print very finely.

When printing with a PCL 6, `SpecialTransparentVectorHandling` does not need to be set true or false.

Overall, the PCL 6 printer driver was found to be the most suitable driver to print MapInfo and Microsoft transparent and non-transparent vector fill patterns. The printed output will match PrintPreview. It does not matter if `SpecialTransparentVectorHandling` is set to true or false. However, `ScaleBitmapPattern` must be false (unless your printer driver does not scale patterns).

When printing using a PostScript driver, set it to Language Level 1 when scaling MapInfo bitmap patterns. Additionally, transparent vector handling and scale pattern both need to be set to true. You can print both MapInfo and Microsoft transparent and non-transparent vector fill patterns successfully. However, the Print Preview for non-transparent MapInfo bitmap pattern will not match the printer output.

PCL 5e also requires transparent vector handling and scale pattern set to true. Print Preview for non-transparent MapInfo bitmap pattern will not match the printer output. PCL 5e does not handle non-transparent Microsoft vector fill patterns.

With any of the above testing printer drivers. if ScaleBitmapPattern is true, the Print Preview will not match the print output.

Transparent Vector

The setting DrawingAttributes.SpecialTransparentVectorHandling allows you to manage the printing of transparent vector fill patterns. Factors that affect this are the printer driver and the setting for scaling bitmap patterns.

When using a PCL 6 printer driver, set the SpecialTransparentVectorHandling to false to print good quality transparent Microsoft and MapInfo vector fill patterns. If set to true, the print quality decreases somewhat: the pattern is edgier and not as clear as when the setting is false.

When printing using a PCL 5 printer driver, set SpecialTransparentVectorHandling to false. MapInfo vector patterns do not scale at all. Microsoft's standard vector patterns print well. SpecialTransparentVectorHandling set to true does not print correctly.

When printing with a Postscript driver, setting SpecialTransparentVectorHandling to false prints transparent fill patterns well, but they do not scale. A true setting yields similar output to a PCL 6 driver true setting: edgier and less clear than a false setting yields. There were similar findings for a Plotter using a PostScript driver.

Documentation Clarifications for MapXtreme 2005 v6.5

The following are topics that provide corrections or clarifications to the MapXtreme 2005 documentation set, including the Developer Reference (online help) and the Developer Guide.

Please send feedback to documentation@mapinfo.com if there are topics you feel need additional documentation. You can send email directly from the online Developer Reference in Visual Studio .NET by clicking on the "Send Comments on this Topic" at the bottom of a topic.

MI_Style Column

When inserting a table into a database, note that the MI_STYLE column must be included in the insert statement. See Chapter 10 in the MapXtreme 2005 Developer Guide for a code example.

Layer Selectability

The LayerHelper.SetSelectable method does not affect the selectability of certain layer types, including Labels, WMS/WFS, Raster and Group layers. The documentation for this method does not indicate this limitation. To control selectability of label layers, use the SelectMapToolProperties.LabelsAreEditable property programmatically or via Workspace Manager's Labels are Editable checkbox.

CoordSys.Distance Method

The MapInfo.Geometry.CoordSys class provides two overloaded methods for calculating the distance between two points.

CoordSys.Distance Method (DistanceType, DistanceUnit, CoordSys, DPoint, DPoint) is a static method and must be called directly from the class name itself, and not from a CoordSys object.

The second overloaded method, CoordSys.Distance Method (DistanceType, DistanceUnit, DPoint, DPoint), is an instance method available from any CoordSys object.

The Developer Reference for these two methods does not clearly distinguish between the two.

DisplayCoordSys

There are methods to get and set the coordinate system used to display a map in MapXtreme 2005.

MapInfo.Mapping.Map.SetDisplayCoordSys method is used to set the coordinate system that will be used to draw the map. For example:

```
public static void MapInfo_Mapping_MapSetDisplayCoordSys(Map map)
{
    // This example sets the map's display coordsys
    CoordSysFactory csysFactory = Session.Current.CoordSysFactory;
    CoordSys csys = csysFactory.CreateFromPrjString("1, 62");
    map.SetDisplayCoordSys(csys);
}
```

MapInfo.Mapping.FeatureViewer.GetDisplayCoordsys gets the Coordinate System object for the current Map in the Map Control. It returns the default bounds for that coordinate system.

To set a new view for the map, use MapInfo.Mapping.FeatureViewer.SetView.

Sample Data with WMS/WFS Layers

Please be aware that some of the sample data that ship with MapXtreme 2005 may refer to third party WMS or WFS servers over which MapInfo has no control. These servers may not be accessible when you attempt to connect to them.

Note, too, the C:\Program Files\MapInfo\MapXtreme\6.5\Samples\Data\WMS Examples\US and HighwaysUSGS_REF&Mapinfo - Us_Roads_States_Capitals.mws contains the JPL_WMS_USA.xml with an incorrect URL to a third party server.

Required Merge Modules for Deployed Applications

MapXtreme 2005 requires only two merge modules to be included in your deployed desktop or web application. This is so you can customize your package to include only those items you need. Required are MapInfoCoreEngine_65.msm and MapInfoMXTConfig_65.msm. (If you are deploying applications for Japanese or Chinese locales see also next section.)

When you package your application using Visual Studio .NET, it will search your project for .NET assemblies and include any installed merge modules that match those assemblies. For example, if your application uses custom symbols, Visual Studio .NET will add MapInfoCustSymb.msm. If your application uses MapXtreme's pre-built web controls, Visual Studio adds MapInfoWeb_6.5.msm to your project.

MIFonts_61.msm is the exception. MapInfoCoreEngine does not require the use of the fonts in this merge module. You must add MIFonts_61.msm to your project separately in order to use any of the 10 MapInfo TrueType symbol sets that are provided with MapXtreme 2005.

For a list of available merge modules and when to include them, see Chapter 2: Getting Started, in the Developer Guide. For a summary of the MapInfo font sets available, see Appendix I: Style Lookups.

Merge Modules for Japanese and Chinese Deployments

When deploying applications for Japanese or Chinese locales, be sure to include the locale-specific merge module in your package. For example, when packaging your application for Japanese, include MapInfoCoreResJPN_6.5.msm. For Chinese, the merge module to include is MapInfoCoreResCHN_6.5.msm.

If you are building an application using the SCP version of MapXtreme 2005, you will need to include MapInfoCoreEngineIntl.msm.

If you are deploying applications for Japanese or Chinese locales and your development installation of MapXtreme 2005 was built using the SCP version, you will need to include both the MapInfoCoreResJPN(CHN)_6.5.msm and MapInfoCoreEngineIntl.msm.

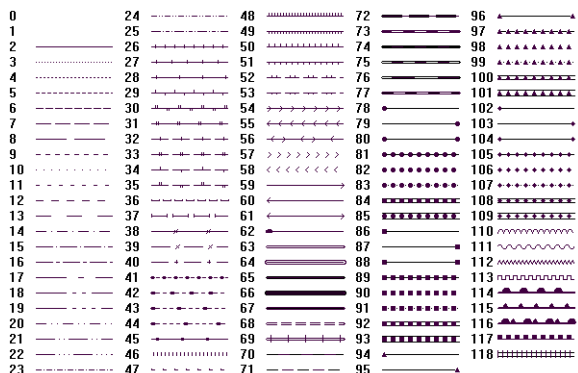
Index Numbering for Interior and Line Styles

As reported in Appendix I of the Developer Guide, there are two different numbering systems used for interior (or fill) styles, depending on how you access them. This also holds true for line styles. This information was omitted from the Line Styles section of the Appendix. The following is a summary of the issue:

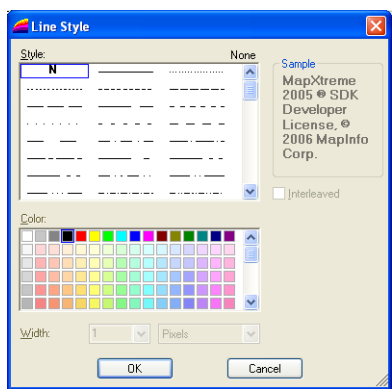
Each fill or line pattern has an associated zero-based index number, which is used for programmatic access into a StyleRepository object: an InteriorStyleRepository for fill patterns, currently holds patterns from 0 to 172 and a LineStyleRepository contains line styles 0 to 118.

Each style also has a 1-based pattern number, which is an internal descriptive name of the fill or line pattern. This pattern number may be used to construct an interior style object, but it cannot be used to access the patterns programmatically from the InteriorStyleRepository or LineStyleRepository. The pattern numbers are also used to indicate fill and line patterns within MapXtreme 2005 workspace files (.mws) as well as in other MapInfo products (for example, MapInfo Professional and MapBasic).

There are 119 available line styles, but only 118 are visible from the MapXtreme Workspace Manager. In the line style image below, the difference is made up because the 0 and 1 styles are both blank.



The Workspace Manager below shows only one blank line style. The zero pattern number is not accessible from Workspace Manager.



MapInfo.Data.IFeatureEnumerator

Retrieving features using an MapInfo.Data.IFeatureEnumerator returns an MIDataReader that must be closed when you are through using it. This information is missing from the remarks for the IFeatureEnumerator Interface topic in the online Developer Reference. Close() is called automatically when MoveNext() returns false.

Location of CoordinateSystem Table

The coordinate system information that MapXtreme 2005 supports is provided in the MapInfoCoordinateSystemSet.xml located in "C:\Program Files\Common Files\MapInfo\MapXtreme\6.x, where 6.x is the release version of MapXtreme 2005.

MapInfo.Data.BindType Enumeration

The explanation for Dynamic binding under the MapInfo.Data.BindType enumeration topic in the Developer Reference online help has been enhanced. Dynamic binding adds expression columns that do not physically store any data values. Expression columns compute their value every time they are accessed. Dynamic binding is currently only supported for creating expression columns which reference other columns currently defined in the Table.

LegendFrame.Size property

MapInfo.Mapping.Legend.LegendFrame.Size property is used to get the size of the LegendFrame. A LegendFrame is a container for legend information that displays in the legend. There can be multiple LegendFrames in a Legend.

Setting the size of the LegendFrame is not supported. The size is controlled by the amount of data in the frame. To adjust the width or height of the Legend, set the Legend.Size property with a new Size object.