



MapXtreme 2008 v7.0.0 Release Notes

Americas:
Phone: 518 285 6000
Fax: 518 285 6070
Sales: 800 327 8627
Government Sales: 800 619 2333
Technical Support: 518 285 7283
www.pbinsight.com

UK and EMEA:
Phone: 44 1753 848200
Fax: 44 1753 621140
Technical Support: 44 1753 848229
www.pbinsight.co.uk

Asia Pacific:
Phone: 61. 2.9437.6255
Fax: 61.2.9439. 1773
Technical Support: 1800 648 899
www.pbinsight.com.au

© 2009 Pitney Bowes Software Inc. All rights reserved. MapInfo, the Pitney Bowes Software Inc. logo, and MapXtreme 2008 are trademarks of Pitney Bowes Software Inc. and/or its affiliates.

December 2009

These Release Notes provide information on enhancements and updates made to MapXtreme 2008 7.0.0. It also covers bug fixes, known issues, and other important information for this release.

This document is a supplement to the MapXtreme 2008 Developer Guide. Refer to this document for release specific information and to the Developer Guide for overall product information and usage. The Developer Guide is available in three formats and from several locations, including:

- HTML version integrated into Visual Studio (in the Help Contents pane)
- PDF version accessible from the Start > All Programs menu under MapInfo > MapXtreme 2008 7.0.0 > Learning Resources browser
- **PDF version** on the MapInfo website

If you would like a printed version of the Developer Guide, you may purchase one for a nominal charge from Pitney Bowes Business Insight's **Print Site**.

List of Topics:

♦ What's New	2
♦ What's Changed	13
♦ API Changes	13
♦ Bug Fixes	21
♦ Known Issues	26
♦ System Limitations	32
♦ Documentation Updates	32

What's New

MapXtreme 2008 includes the following new features in version 7.0.0.

- **Grid Enhancements**
- **SQL Server 2008 Support**
- **Linear Referencing and Dynamic Segmentation Operations**
- **Geometry Model Extended to Support M Values and Z Values**
- **Oracle Support for M Values and Z Values**
- **Support for Oracle 11g**
- **ShapeFiles Support**
- **MapXtreme Tile Handler**
- **Popular Visualization CRS / Mercator (EPSG:3857)**
- **Coordinate System Support**
- **Extensible Data Provider**
- **Updated ECW Raster Handler**
- **Geocoding U.S. Addresses**
- **ToolStripButtons and Icons**
- **New Snap Mode Events**
- **Localization Kit**
- **Workspace Manager Enhancements**
- **ISessionEventHandler for Loading Custom Assemblies**
- **Default View for Remote Tables**

Grid Enhancements

This release of MapXtreme 2008 provides a number of new capabilities and enhancements for working with grids, including grid creation, grid interpolators, Grid API and grid style.

Grid Creation

New with this release of MapXtreme 2008 is the ability to create continuous grids using a writable grid handler and interpolators. These grids are created programmatically using the `MapInfo.Raster.CreateGridFromFeatures` class, the `Mig.ghl` grid handler and one of two provided interpolators. Grids can be created from tables of data points or from selections.

A continuous grid is a map that is divided into a rectangular grid of cells, where each cell contains a data value representing either a measured data point, or an interpolated value based on the surrounding data points. The continuous grid displays the changing data values using a continuous gradation of color across the map.

Previously MapXtreme was limited to reading a grid and getting back a list of its grid cell values. It also could read information about a grid, such as hillshading and styles. The `ContinuousGridWrite` class was available, but lacked the ability to use an interpolator which would have created grids that better reflect its data points.

The new `MapInfo.Raster.CreateGridFromFeatures` is the main class to be called when creating a continuous grid. It uses the `MapInfo` writable grid handler and an interpolator to produce the continuous grid map.

Grid Interpolators

MapXtreme 2008 provides two grid interpolators for creating a continuous grid: IDW and TIN. The IDW and TIN interpolators are included in a separate namespace called `MapInfo.Raster.Interpolators`.

The interpolators provide algorithms for determining grid cell values according to their particular formulas. The IDW, or Inverse Distance Weighted interpolator, is best suited for data values such as population, or any data that yields arbitrary values over the grid rather than be related to or influenced by neighboring values. This interpolation method also works well for sparse data. TIN, the triangulated irregular networks interpolator, works best for terrain data and for data points that have a linear progression or relationship to each other across the grid, such as temperature.

See *Chapter 18: Working with Rasters and Grids* in the Developer Guide.

Grid API

If one of the provided interpolators does not meet your needs, you can create your own by deriving it from the new `Interpolator` interface. See the `MapInfo.Raster.Interpolators` namespace in the Developer Reference.

Other enhancements to the Grid API include the ability to clip the grid to match the outline of the map boundary. Grids are created based on the minimum bounding rectangle of the source data. If you wish your grid boundary to follow the outline of your map boundary, you would set the `CreateGridFromFeatures.ClippingGeometry` property to the `Geometry` you wish to clip against.

Grid Style

Once you have created a grid from your data points, you can modify characteristics such as hill shading, styles and inflection points using the new Grid Style dialog and Workspace Manager.

Grid style support is provided programmatically through the `MapInfo.Styles.GridStyle` class, via a `GridStyleControl`, and incorporated into Workspace Manager. A sample application called `GridForm` also provides this capability. Changes to inflection values and colors can be persisted to a workspace and loaded at a later time.

MapXtreme provides C# sample applications that incorporate the `GridInflectionCalculator`. Find these samples under the `\Samples\<VisualStudio2005>\<VisualStudio2008>\Desktop\Features\GridForm`.

SQL Server 2008 Support

This release of MapXtreme 2008 includes support for reading and writing data from/to Microsoft's SQL Server 2008, including the spatial data types `GEOMETRY` and `GEOGRAPHY`, along with M and Z value support for both spatial formats. For information on SQL Server 2008 data types, see the [documentation](#) for SQL Server 2008.

To access data from SQL Server 2008, MapXtreme requires SQL Server Native Client 10. Data is then handled like data from other remote database management system that MapXtreme supports. Use the `TableInfoServer` class to define the connection string and an SQL statement to execute on the remote table. Internally, MapXtreme uses ODBC to access the remote database.

Spatial tables from SQL Server 2008 must be registered in the `MapInfo_MapCatalog` so that MapXtreme understands what it reads. The `MapCatalog` provides four new `spatialcolumn` values to represent SQL Server 2008 tables:

- 17.x for `GEOMETRY` without M and Z values
- 18.x for `GEOGRAPHY` without M and Z values
- 20.x for `GEOMETRY` with M and Z values
- 21.x for `GEOGRAPHY` with M and Z values.

The x in the above values refers to the type of spatial object: 0 for points only; 1 for lines only; 2 for regions only, and 3 for all spatial object types.

Data can be uploaded using MapInfo Professional or EasyLoader or you can use MapInfo Professional to make existing data mappable, which will create the entry in the MapCatalog.

MapXtreme supports SQL Server 2008 tables created in MapInfo Professional (table versions 900, 950 and 1000) and EasyLoader.

See *Chapter 12: Accessing Data from a DBMS* in the Developer Guide for more information on MapXtreme's support for SQL Server 2008.

Linear Referencing and Dynamic Segmentation Operations

MapXtreme 2008 offers a number of new operations for carrying out linear referencing and dynamic segmentation. With MapXtreme 2008's new support for M and Z values, this is accomplished using the measure (M) value on MultiCurve feature geometries. The methods are defined as part of an instantiable MapInfo.Geometry.LinearReferencingOperations class, which provides flexibility for adding properties or extending behaviors as needed.

Linear referencing operations include the ability to set and manage measure values on a MultiCurve, locate points at a specific measure value or extract sub-curves between two measure values. Practical uses for these operations include determining the location of a stranded motorist when given a nearby mile marker, or identifying sections of highways that are in various stages of construction or repair.

Sample Application

Provided in the Samples folder of your MapXtreme installation is a linear referencing sample application that generates multiple parallel line segments representing different characteristics of the road. In this example, road data that indicates pavement conditions, maximum speed zones and number of lanes are displayed as offset MultiCurves along the actual roadway to show the condition or characteristic of a particular road section. Each characteristic or condition is shown in a different line style.

For more information, see *Chapter 22: Linear Referencing* in the Developer Guide and the MapInfo.LinearReferencing namespace in the Developer Reference.

Geometry Model Extended to Support M Values and Z Values

The MapXtreme Geometry model has been extended to support reading and writing M and Z values and batch geometry creation.

Support for M values and Z values

MapXtreme 2008 now includes support for M and Z values for points along a line. These attributes allow you to assign measured (M) and vertical dimension (Z) information to the nodes to give you greater display, querying and analytical capabilities beyond the traditional two-dimensional X, Y framework.

Read/write support is available for Oracle, SQL Server 2008 and MapXtreme tables in memory (MemTable).

Additionally, MapXtreme 2008 reads M and Z values from ESRI Shapefiles. Previously M and Z values from Shapefiles were ignored. This matches the read capabilities of Shapefiles in MapInfo Professional.

Support for reading and writing M and Z values for linear objects was accomplished by extending the MapXtreme Geometry model. The Feature geometries (Point, Multipoint, MultiPolygon, MultiCurve and FeatureGeometryCollection) can now hold values for X, Y, Z and M for each node in the object. The MapInfo.Data.GeometryColumn properties Is3D and IsMeasured allow you to determine whether the object has M or Z

values. The minimum and maximum ranges of M and Z values can be retrieved as well. Additional properties and methods are provided by the various GeometryEditor classes for each geometry type in to read and modify M and Z values at each node.

For more information, see *Chapter 17: Spatial Objects and Coordinate Systems* in the Developer Guide and the MapInfo.Geometry namespace in the Developer Reference.

Support for Batch Geometry Creation

The geometry model has been extended to provide more robust and scalable options for batch geometry creation.

New constructors and Add signatures have been added to the MultiPolygon, MultiCurve, and MultiPoint classes allowing the number of nodes to be provided along with the input arrays. These new provisions allow arrays to be instantiated once and then re-used for constructing new geometries.

A Clear method has been added to the MultiPolygon and MultiCurve classes, and ReplaceAll method overloads are provided with number of nodes along with the input arrays on the MultiPoint class.

An overload to EditingComplete has been added to the MultiPolygon class.

Collectively, these geometry model extensions provide a robust set of options for efficient batch geometry creation. This will be of particular importance to anyone implementing a cursor as part of an extensible data provider. The Extensible Data Provider model is presented in the Developer Guide. See Appendix E: Extensible Data Providers.

See the MapInfo.Geometry namespace in the Developer Reference for the details on these extended classes.

Table Column Metadata Support for M and Z Values

MapXtreme 2008 provides table column metadata support for M and Z values. This feature is useful when you want to know whether geometries of a particular data provider can support 3D and measure values without evaluating its individual geometries. Metadata for a table is accessed from the table's TableInfo property. From there, you can access the GeometryColumn to interrogate if the table supports M or Z values and what the range of values for that table is if the range is known.

For more information, see *Chapter 10: Working With Data* in the Developer Guide and the MapInfo.Data.TableInfo class in the Developer Reference.

Oracle Support for M Values and Z Values

MapXtreme supports reading and writing Oracle GTYPEs with M and Z values. The presence and order of M and Z is determined by inspecting the DIM_INFO array in the USER_SDO_GEOM_METADATA for the table. MapXtreme checks for the following dimension names (case insensitive):

- For M dimension: "M", "Measure", and "M Dimension"
- For Z dimension: "Z", "Elevation", "Depth" and "Z Dimension"

Tables that contain M and/or Z values now return FeatureGeometry objects that contain the data for the dimensions present. Geometries inserted into or updated in an Oracle table will preserve the dimensions of the new geometry that the table is designed to support. If an Oracle table has X, Y, and Z dimensions and a geometry is inserted that has X, Y, Z, and M values, the Z values are persisted to the database and the M values are not. If the Oracle table requires a dimension that is not supplied, a value of NULL will be stored. For example, when inserting a geometry with no M or Z values into a table that is defined with dimensions X, Y, and M, the M values stored in the table will be NULL.

Support for Oracle 11g

MapXtreme 2008 now supports Oracle Database 11g, specifically 11.1.0.6.0 and 11.1.0.7.0. This expands our support for Oracle, including 9i, 10g and 10gR2.

ShapeFiles Support

M and Z Values

MapXtreme 2008 now supports reading three-dimensional geometries stored in ESRI Shapefiles (X, Y, M, and Z).

Caching

MapXtreme now caches Shapefile data for improved performance. The data is cached in a MapInfo MAP file which can be temporary or retained across MapXtreme sessions. If temporary, the cache is created in the temp directory and deleted when the table is closed. If the cache is designated as a persistent, then the cache is located in the same directory as the TAB file and will not be deleted when the table is closed. A persistent cache is shared with MapInfo Professional and is controlled by the PersistentCache property on the TableInfoShapefile class. The user must have write permission to the table path to be able to use the persistent cache option.

See the MapInfo.Data.TableInfoShapefile.PersistentCache property in the Developer Reference for a code example on how to set the cache persistence.

MapXtreme Tile Handler

MapXtreme 2008 provides a REST-based tile handler and public URLs that are used to request map tiles and information. With a REST-based handler, you can embed all the arguments of your map request in a single URL.

MapXtreme 2008 Tile Handler provides support for:

- requesting tiles by their row and column positions in the map
- requesting a list of available maps
- requesting a description of a map
- specifying the cache instructions for better tiling performance

For an example of a web application that takes advantage of the MapXtreme Tile Handler, see the MapXtreme Tile Handler example located on the [MapInfo Developers Code Exchange](#).

The MapXtreme Tile Handler API is included under the MapInfo.WebControls.Tiling namespace. See the Developer Reference for details. Source code for the Tile Handler is provided in the \Samples\WebControlsSourceCode folder.

Popular Visualization CRS / Mercator (EPSG:3857)

MapXtreme 2008 now supports a modified Mercator coordinate reference system that is used in web mapping and visualization applications.

Known as "Popular Visualization" or "Spherical Mercator", this system is a variation on the Mercator projection which uses a new ellipsoid to emulate treating the earth as a sphere. This allows map data to properly align with map tiles in Microsoft Bing Maps, Google Maps and other web and visualization applications. The WGS84-based Mercator projection itself does not provide the proper alignment.

The Popular Visualization coordinate system has been accepted into the EPSG Geodetic Parameter Registry under the code EPSG:3857. Where applicable, projections in GIS are commonly referred to by their EPSG code. For more information, see <http://www.epsg-registry.org> and search on the code number 3857.

MapXtreme 2008 support for EPSG:3857 includes a new WGS84 Sphere ellipsoid and datum in the MapInfo.Geometry namespace: Ellipsoid.PopularVisualization and DatumID.PopularVisualization.

Coordinate System Support

New Zealand Geodetic Datum 1949

MapXtreme 2008 now applies the same NTv2 datum conversion for the 3-parameter New Zealand Geodetic Datum 1949 (enumeration value 31) as it does for the 7-parameter version of the same datum (enumeration value 1010).

The NTv2 grid datum conversion for New Zealand is used when transforming between New Zealand Geodetic Datum 1949 (NZGD49) and New Zealand Geodetic Datum 2000 (NZGD2000). Now the 3-parameter or 7-parameter NZGD49 version will transform properly to NZGD2000 using NTv2.

The National Transformation v. 2 (NTv2) algorithm and grid shift file format, developed by the Geodetic Survey Division of Geomatics Canada, enables you to convert data between the NAD 27 reference system and the NAD 83 reference system. Grid shift files used with the algorithm contain one or more rectangular grids that indicate the coordinate differences between NAD 27 and NAD 83.

The National Transformation was originally designed to convert Canadian data from NAD 27 to NAD 83. The NTv2 grid was an enhancement to compensate for the primary and secondary points that were based on the GRS80 ellipsoid. This format was later adopted for datum conversion in Australia, New Zealand, and Germany.

The NTv2 datum conversion was introduced in MapXtreme v 6.8. *Chapter 17: Spatial Objects and Coordinate Systems* in the Developer Guide includes information on how to modify NTv2 for other datum conversions (adding new grid shift files) and how to turn NTv2 on or off for particular grid shift files.

Irish Transverse Mercator

MapXtreme 2008 now supports the Irish Transverse Mercator coordinate system. This coordinate system was implemented jointly by the Ordnance Survey of Ireland (OSI) and the Ordnance Survey of Northern Ireland (OSNI) in 2001 to address new needs brought on by the introduction and popularity of GPS. As the name suggests, it was patterned after Transverse Mercator and optimized for Ireland. For more information on this coordinate system, see ["Migrating to ITM"](#).

Information about all supported coordinate systems for MapXtreme can be found in the \Common Files\MapInfo\MapXtreme\7.0.0 under \Program File or \Program Files(x86). The following are the values for the Irish Transverse Mercator.

```
<srsName>Irish Transverse Mercator (ITM)</srsName>
  <srsID>
    <code>coordsys 8,115,7,-8,53.5,0.99982,600000,750000</code>
    <codeSpace>mapinfo</codeSpace>
    <remarks>Irish Coordinate Systems</remarks>
  </srsID>
  <baseCRS />
  <definedByConversion />
  <usesCartesianCS />
</ProjectedCRS>
<ProjectedCRS>
```

NADCON Conversion for Hawaii

MapXtreme 2008 now performs a datum transformation of Hawaiian data that results in a shift of less than three meters compared with previous transformations of more than 500 meters. MapXtreme accomplishes this by transforming the Old Hawaiian datum through the North American Datum Conversion Utility (NADCON). Previously this transformation did not utilize NADCON for the Old Hawaiian datum.

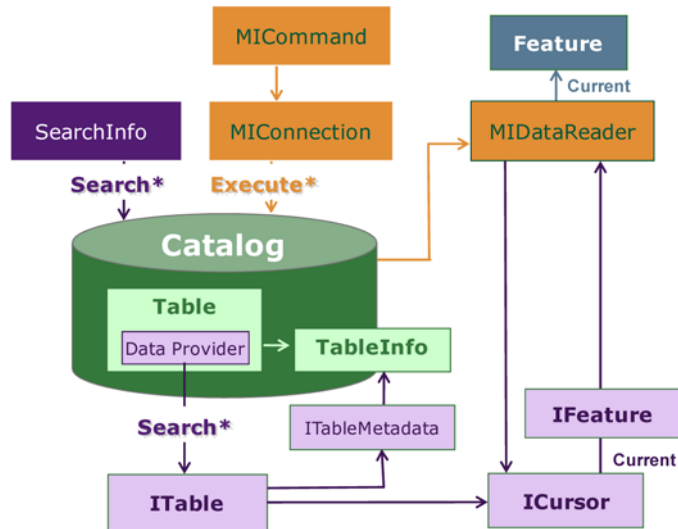
Hawaiian data typically uses the Old Hawaiian datum as a reference. As a result of this conversion change, a map containing Hawaiian data can now be re-projected from Lat/Long NAD27 to Lat/Long NAD83 or WGS84 with a much smaller shift in point locations than was previously possible.

Extensible Data Provider

MapXtreme 2008 introduces an extensible data provider model for developers who wish to integrate their own data formats into MapXtreme. This data provider represents a family of interfaces and supporting classes that implement capabilities of MapXtreme's core data access engine, such as opening a table or using a cursor to read through the rows of a search result.

MapXtreme 2008 supports a wide variety of data sources through its TableInfo class. MapXtreme also supports Microsoft's ADO.NET with a TableInfoAdoNet class for data that cannot be directly accessed through another TableInfo data source.

The addition of an extensible data provider to the supported data sources is available for the experienced .NET developer who wishes to tap directly into MapXtreme's core data access engine rather than indirectly through ADO.NET.



New Namespace: MapInfo.Data.Provider

The MapXtreme 2008 extensible data provider consists of a new namespace, MapInfo.Data.Provider, which contains a number of new interfaces to be used to build your own data provider:

- **IDataProvider** - provides the basis for a custom data provider implementation by exposing the capabilities for opening the table formats serviced by this provider, as well as the data sources managing those tables (when applicable).
- **IDataSource** - defines the interface for a data source, an entity that manages access to data tables.
- **IDataSourceDefinition** - defines the properties necessary to instantiate an **IDataSource** for a specific data provider.
- **ITable** - defines the interface for a table, the basic container of information that MapXtreme can read, query and display in a map.
- **ITableDefinition** - provides the properties necessary to instantiate an **ITable** for a specific data provider.
- **ITableMetadata** - used to relay important information about the properties and supported capabilities of an open table.
- **IFeature** - provides the properties and methods used to define features.
- **ICursor** - defines the class that is used to iterate over a list of features returned from a query.

In addition, the MapInfo.Data.Provider namespace includes several abstract class implementations for use as an optional starting point for building a new data provider.

The MapXtreme 2008 extensible data provider currently supports opening a table, reading the table contents and associated metadata, searching the table contents using several methods, and modifying table content through insert, update, and delete operations.

Additional support for the MapInfo.Data.Provider namespace is provided by new interfaces in the MapInfo.Data namespace. Consult the MapXtreme Developer Reference for additional information.

Extensible Data Provider SDK

MapXtreme 2008 includes a software development kit (SDK) for the Extensible Data Provider containing required and optional interfaces, building blocks of abstract base classes and utilities to get you started, and the guidance to help you along the way. See *Appendix E: Extensible Data Providers* in the Developer Guide.

Included in the Extensible Data Provider SDK is a sample implementation of a data provider that highlights many of the capabilities of the Extensible Data Provider architecture. MapInfo.Spatialite.sln is a Visual Studio solution that uses an SQLite3 ADO.NET data provider to establish a connection to an underlying SQLite3 database and open tables in a MapXtreme application.

This sample implementation also highlights other extensible aspects of MapXtreme, including autoloading custom code at session initialization, and extending the capabilities of Workspace Manager. This sample is located in the \Samples\DataProviders folder under your MapXtreme installation directory. It is further discussed in the Developer Guide, Appendix E.

Updated ECW Raster Handler

This release of MapXtreme 2008 includes an updated ECW raster handler that reads ECW images as well as images in JPEG2000 format. The handler filename, ECW.RHY, remains the same.

Note however, JPEG2000 images will still be read first by LeadTools raster handler LEADTOOL.RHX since MapXtreme evaluates the suitability of each raster handler in alphabetical order according to the file extension (.RHX before .RHY).

Only those JPEG2000 images that LeadTools cannot read will be read in by ECW.RHY. If you wish to affect the order of the raster handlers, you can rename the last character in the file extension. For example, to have the ECW raster handler try to load images before the LeadTools raster handler, use the filenames ECW.RHW and LEADTOOL.RHX. Raster handlers are located by default in the \Program Files\Common\MapInfo\MapXtreme\7.x\RasterGridHandlers folder.

For more information see *Chapter 18: Working with Rasters and Grids* in the Developer Guide.

Geocoding U.S. Addresses

This release includes support for the latest version of MapMarker US (v. 22) for geocoding U.S. addresses. All supported versions of MapMarker Plus for U.S. geocoding can still be accessed via MapXtreme 2008.

There is no change to the MapMarker version for addresses outside the U.S.

MapMarker is a separate purchase from MapXtreme 2008.

ToolStripButtons and Icons

This release of MapXtreme 2008 includes new tool buttons that can be used with a MapControl in a MapXtreme 2008 MapForm application. These buttons, known as ToolStripButtons, are added to a .NET ToolStrip or StatusStrip to give you flexibility with the design of your MapForm.

The ToolStripButtons do not reside in a MapToolBar, which in previous versions of MapXtreme 2008, had been responsible for managing the current tool for the MapControl. The logic that controls the activity and properties of the tools is now included in each ToolStripButton. The MapToolBar, with a default collection of basic mapping tools, has been retained for backward compatibility and future use if you choose, and is still provided in the Visual Studio toolbox.

Note The change to ToolStripButtons from a Toolbar is a **breaking change** for any desktop application that you wish to upgrade to v 7.0.0.

New icons

To support the new ToolStripButtons, we ship a collection of icons that you can use however you choose in your application. These 90 images represent mapping and data access capabilities, drawing operations, file management and more. They are located in the \Samples\Icons folder under your MapXtreme 2008 installation. Each image comes in two sizes (16x16 pixels and 24x24 pixels). For a thumbnail view, see Appendix G in the Developer Guide. Some of these icons are also used in LayerControl and Workspace Manager.

New Snap Mode Events

This release of MapXtreme 2008 includes two new snap mode events for desktop applications that are useful when you need to know the exact map location of your tool cursor as you move in the vicinity of the snap nodes.

MapInfo.Tools.MapTools.**SnappedToNode** event is fired when the tool cursor snaps to the node of an object when snap mode is enabled and is within the node's snap tolerance. MapInfo.Tools.MapTools.**UnsnappedFromNode** event is fired when the cursor leaves the snap tolerance region for a given node.

Both events return map coordinates, as well as screen coordinates, that identify the location of the cursor. By returning map coordinates from each event, you can precisely tell where your cursor is. This is useful when you need these coordinates for other map calculations, such as distance to other features.

In the case of overlapping snap tolerance regions, as you move the cursor across both regions, both events will fire in succession to yield the cursor's location.

Snap mode is enabled either by setting the MapInfo.Tools.IMapToolProperties.**SnapEnabled** Property or from the Tools tab in the Workspace Manager when a map is selected in the tree view. Additionally, to view the nodes for a layer, set the **ShowNodes** property on MapInfo.Mapping.FeatureLayer.

Localization Kit

MapXtreme 2008 provides a Visual Studio solution for developers who wish to translate error messages and dialog/control text elements for use in their own MapXtreme-based applications.

This "Localization Kit" contains resource projects for all runtime components of MapXtreme 2008. It is organized as a Visual Studio solution to make it convenient to edit the resource strings and build the assemblies while much of the behind-the-scenes resource management is taken care of. The solution can be used in Visual Studio 2008, Visual C# 2008 Express Edition, and by the MSBuild command line build utility.

Included in each project are the English resource strings for translating and a Strong Name Key (.snk) file that will compile into a satellite assembly that can be incorporated into your MapXtreme application.

The satellite assemblies are signed with a strong name public key (included in the kit). MapInfo.CoreEngine.Public.Snk is the public key for all assemblies except for MapInfo.Ogc.resources and MapInfo.WebControls.resources, which have their own public keys. Signing your satellite assemblies with these public keys allows you to use the assemblies for testing and debugging. In order to run MapXtreme using these assemblies, they must be put in the global assembly cache (GAC). Only signed assemblies can be placed in the GAC. When you are ready to deploy the localized assemblies, they must be signed with a private key by Pitney Bowes Business Insight. See *Appendix K: Localization Kit* in the Developer Guide.

Workspace Manager Enhancements

Workspace Manager has been enhanced with new features such as multiple map and legend tabs and a new capability to add custom extensions. The user interface has been updated with fresh toolbar icons and some menus have been reconfigured to make using the tools and commands easier.

Among the changes to the Workspace Manager:

- Each map is now contained in a tabbed map window so you can move easily from map to map. If your map includes legends, they can display in their own tab.
- A new Extensions capability is provided for loading and managing extensions that you or others write to enhance the functionality of Workspace Manager. Workspace Manager extensions are .NET assemblies. The Extensions > Load Extensions command displays an Open dialog with the default file of type Extensions (*.dll). Extensions > Manage Extensions displays a Manage Extensions dialog where you can load and unload extensions and set properties.

- A new Manage Workspace Maps dialog is provided for adding, deleting, and controlling the visibility of maps.

Workspace Manager is discussed in *Chapter 24: Workspace Manager* in the Developer Guide.

ISessionEventHandler for Loading Custom Assemblies

MapXtreme provides a `MapInfo.Engine.ISessionEventHandler` interface for loading custom DLLs that extend MapXtreme's functionality. Use this interface to autoload any extension DLLs that you need for your application, such as extensible data providers, persistence providers, and styles that a default workspace may use. When MapXtreme starts up, it will initialize these assemblies and carry out the required instructions.

In order for `SessionEventHandler` assemblies to be initialized on startup, they must be located in the `\Common Files\MapInfo\MapXtreme\7.x.x\SessionEventHandlers` folder. In fact, any assembly in this folder with the file suffix of `.SessionEventHandler.DLL` will be loaded.

For more information, see *Chapter 9: Working With Core MapXtreme Classes* in the Developer Guide.

Default View for Remote Tables

MapXtreme 2008 now provides a configurable default view for a remote database table so that only the data you are interested in will be opened in a map or browser. This applies when the remote table is the first table opened. Previously MapXtreme 2008 retrieved the entire bounds of the table for the view, making data access slow for large tables.

The default view is stored in the `MAPINFO_MAPCATALOG` in four new columns that hold the bounds (`VIEW_X_LL`, `VIEW_Y_LL`, `VIEW_X_UR` AND `VIEW_Y_UR`). The `MapCatalog` is a server table that contains metadata about remote tables. If there is no entry in the `MapCatalog` for the table, MapXtreme 2008 will retrieve the entire bounds.

If the remote table has no default view entry in the `MAPINFO_MAPCATALOG`, the remote database user (as identified in the database connection) must have `ALTER` permission to the `MapCatalog` in order to change the default view from the original setting. Without this permission, a warning message displays and the default view change fails.

The ability to set a default view is also available in `EasyLoader` and `MapInfo Professional v 10.0`.

What's Changed

- [Breaking Changes](#)
- [API Changes](#)

Breaking Changes

The following changes to MapXtreme 2008 will result in breaking changes to applications built with previous versions:

- [Modifier Theme Serialization/Deserialization Change](#)
- [LayerControl and the ToolStrip Class](#)

Modifier Theme Serialization/Deserialization Change

For MapXtreme 2008 v7.0, there has been a fundamental change in how modifier themes are serialized and deserialized in order to improve performance.

The affected themes are those that implement the IModifierTheme interface, namely RangedTheme, RangedLabelTheme, IndividualValueTheme, and IndividualValueLabelTheme.

The changes are during theme serialization, when the theme's bin or category record counts are serialized. Upon deserialization, the record counts are no longer recomputed, rather the serialized record counts are applied to the theme. This change was put in place to improve serialization/deserialization performance.

It is important to note that for applications that are expecting to have their theme record counts updated upon deserialization will have to add logic to update the theme bin or category record counts after deserialization. For RangeTheme and RangeLabelTheme, this is a call to the MapInfo.Mapping.Thematic.IRangedTheme.Recompute method. For IndividualValueTheme and IndividualValueLabelTheme, this is a call to MapInfo.Mapping.Thematic.IModifierTheme.RecomputeBins.

LayerControl and the ToolStrip Class

MapInfo.Windows.Dialog.LayerControl has been changed to take advantage of Microsoft's ToolStrip, whereby tools are added to a ToolStrip instead of a ToolBar. See [ToolStripButtons and Icons on page 10](#).

API Changes

The following are changes to the MapXtreme 2008 API that you should be aware of:

- [MapInfo.Data.Feature Attribute Null Value Handling](#)
- [IndividualValueTheme Constructor](#)
- [IFeatureEnumerator](#)
- [WFS Client](#)
- [MapInfo.Geometry.Ring.IsClockwise Property](#)
- [Obsolete Namespace: MapInfo.Web.UI.WebControls](#)
- [Obsolete Methods and Properties](#)

MapInfo.Data.Feature Attribute Null Value Handling

Potential Breaking Change in MapInfo.Data.Feature Class

Beginning with v 7.0.0, the MapInfo.Data.Feature attribute value of null will be treated as **System.DBValue.Null** objects, rather than as null references. This represents a change of behavior for the return value of the Feature getter index operator on uninitialized attribute values and has the potential to change the return value for explicitly null-ed values. Client code that checks for null as a return from the MapInfo.Data.Feature index operators should be modified to support a check for DBValue.Null.

The Feature index setter operator will continue to accept a null reference, in addition to DBValue.Null.

The Feature.Geometry property will continue to return a null reference if the Feature does not have a Geometry column definition. Otherwise, a DBNull.Value return indicates a null value on the defined geometry column.

The Feature.Style property will continue to return a null reference if the Feature does not have a Style column definition. Otherwise, a DBNull.Value return indicates a null value on the defined style column.

MIDataReader.GetValue() continues to return DBNull.Value representation for null.

IndividualValueTheme Constructor

The MapInfo.Mapping.Thematics.IndividualValueTheme class provides a new public constructor containing a parameter *categoryCount* for specifying the number of categories to create for the theme. This is in addition to the original constructor that did not provide that capability.

See also [Modifier Theme Serialization/Deserialization Change on page 13](#) for information related to serializing category counts for modifier themes, including IndividualValue themes.

IFeatureEnumerator

MapInfo.Data.IFeatureEnumerator now inherits from System.IDisposable. This means you must make sure that Dispose() is called when using any object that implements IFeatureEnumerator.

This change was made to fix an issue with using seamless tables and to improve performance of FeatureCollections. Going forward, you should always make sure Dispose() is called.

When you use the foreach syntax of C# or VB.NET, Dispose() is called automatically.

Calling the IDisposable Interface

When calling a class that implements the **System.IDisposable** interface, use the try/finally pattern to make sure that unmanaged resources are disposed of even if an exception interrupts your application.

You can also use the using statement instead of the try/finally pattern.

WFS Client

The MapInfo.Wfs.Client PropertyName and TypeName constructors now provide additional parameters that allow the user to specify the namespace the feature resides in, as well as an xmlns prefix for that namespace.

MapInfo.Geometry.Ring.IsClockwise Property

We have added new property on the MapInfo.Geometry.Ring class called IsClockwise. This allows MapXtreme to create SQL geometries using the proper orientation from the FeatureGeometry.

Obsolete Namespace: MapInfo.Web.UI.WebControls

In the process of deprecating older web controls within the MapInfo.Web.UI.WebControls namespace, the default installation (SDK, Runtime, merge modules) no longer creates the virtual directory MapXtremeWebResources_7_0_0. The MapXtreme Web Site templates have the resources within a subdirectory, MapXtremeWebResources.

If you are upgrading older web applications using the deprecated web controls, you will have to manually create the virtual directory pointing to Program Files\Common Files\MapInfo\MapXtreme\7.0.0\MapXtremeWebResources with the alias MapXtremeWebResources_7_0_0. If you have modified the resources then you will need to create the IIS alias, MapXtremeWebResources_7_0_0 and point to your existing modified webresources directory.

Obsolete Methods and Properties

The methods and properties listed in the table below in the left column have been removed from the MapXtreme 2008 object model. The current method or property to use is listed in the second column.

Methods and properties are identified as obsolete two years prior to being removed from the API. The list below contains methods and properties that were marked obsolete in MapXtreme 2008 v 6.8 and earlier.

Obsolete Method or Property	Current Method or Property
MapInfo.Data.GeoDictionary.GeoDictionary.DefaultMatchThreshold	MapInfo.Data.GeoDictionary.DefaultMatchThreshold
MapInfo.Engine.ISession.Reset	MapInfo.Engine.ISession.Clear
MapInfo.Geocoding.GeocodeClientFactory.GetMiAwareGeocodeClient(string url)	MapInfo.Geocoding.GeocodeClientFactory.GetEnvinsaLocationUtilityService(string)
MapInfo.Geocoding.GeocodeClientFactory.GetMiAwareGeocodeClient(string,string,string)	MapInfo.Geocoding.GeocodeClientFactory.GetEnvinsaLocationUtilityService(string,string,string)
MapInfo.Geometry.Curve.CurveEditor	MapInfo.Geometry.Curve.GetCurveEditor()
MapInfo.Geometry.CurveSegment.CurveSegmentEditor	MapInfo.Geometry.CurveSegment.GetCurveSegmentEditor()
MapInfo.Geometry.Ellipse.EllipseEditor	MapInfo.Geometry.Ellipse.GetEllipseEditor()
MapInfo.Geometry.Envelope.EnvelopeEditor	MapInfo.Geometry.Envelope.GetEnvelopeEditor()
MapInfo.Geometry.FeatureGeometryCollection.FeatureGeometryCollectionEditor	MapInfo.Geometry.FeatureGeometryCollection.GetFeatureGeometryCollectionEditor()
MapInfo.Geometry.Geometry.GeometryEditor	MapInfo.Geometry.Geometry.GetGeometryEditor()
MapInfo.Geometry.LegacyArc.LegacyArcEditor	MapInfo.Geometry.LegacyArc.GetLegacyArcEditor()
MapInfo.Geometry.LegacyText.LegacyTextEditor	MapInfo.Geometry.LegacyText.GetLegacyTextEditor()
MapInfo.Geometry.LineString.LineStringEditor	MapInfo.Geometry.LineString.GetLineStringEditor()
MapInfo.Geometry.MultiCurve.MultiCurveEditor	MapInfo.Geometry.MultiCurve.GetMultiCurveEditor()
MapInfo.Geometry.MultiPoint.MultiPointEditor	MapInfo.Geometry.MultiPoint.GetMultiPointEditor()
MapInfo.Geometry.MultiPolygon.MultiPolygonEditor	MapInfo.Geometry.MultiPolygon.GetMultiPolygonEditor()
MapInfo.Geometry.Point.PointEditor	MapInfo.Geometry.Point.GetPointEditor()

Obsolete Method or Property	Current Method or Property
MapInfo.Geometry.Polygon.PolygonEditor	MapInfo.Geometry.Polygon.GetPolygonEditor()
MapInfo.Geometry.Rectangle.RectangleEditor	MapInfo.Geometry.Rectangle.GetRectangleEditor()
MapInfo.Geometry.Ring.RingEditor	MapInfo.Geometry.Ring.GetRingEditor()
MapInfo.Geometry.RoundedRectangle.RoundedRectangleEditor	MapInfo.Geometry.RoundedRectangle.GetRoundedRectangleEditor()
MapInfo.Routing.RouteClientFactory.GetMiAwareRouteClient(string url)	MapInfo.Routing.RouteClientFactory.GetEnvinsaRoutingClient(string url)
MapInfo.Routing.RouteClientFactory.GetMiAwareRouteClient(string url, string username, string password)	MapInfo.Routing.RouteClientFactory.GetEnvinsaRoutingClient(string url, string username, string password)
MapInfo.Routing.RoutePlan.RoutePreference	MapInfo.Routing.RoutePlan.RoutePreferences.RouteMethod
MapInfo.Routing.WayPointList.WayPointList(DPoint startPoint, DPoint endPoint)	MapInfo.Routing.WayPointList(MapInfo.Geometry.Point, MapInfo.Geometry.Point)
MapInfo.Routing.WayPointList.WayPointList(DPoint startPoint, DPoint endPoint, DPoint[] viaPoints)	MapInfo.Routing.WayPointList(MapInfo.Geometry.Point, MapInfo.Geometry.Point, MultiPoint) or MapInfo.Routing.WayPointList(MapInfo.Geometry.Point, MapInfo.Geometry.Point, ViaPointList)

Upgrading to MapXtreme 2008 v7.0.0

MapXtreme 2008 installs into its own directory using the form `<installdir>\MapInfo\MapXtreme\7.x.x`, where X.x.x is the current release. It will not overwrite a previous version. This allows you to maintain side-by-side installations of different releases of the product.

Because 7.0.0 is a major release, you cannot copy a v6.x.x license to the 7.0.0 installation. You can only copy v6.x.x licenses to other v6.x.x installations. You must get a new license from Pitney Bowes Business Insight Customer Service. For more on Licensing, see *Chapter 2: Getting Started* in the Developer Guide.

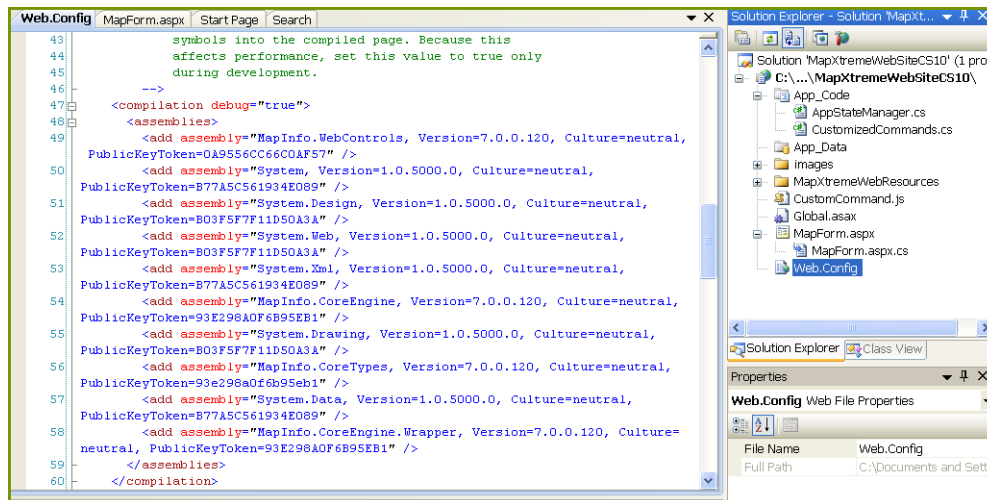
Note When using a side by side install, you must close Visual Studio before opening a project of a different type. For web applications, you will also need to reset Internet Information Services (IIS). Use the `iisreset.exe` command in a console window or from the Start > Run menu option.

This section contains the following topics:

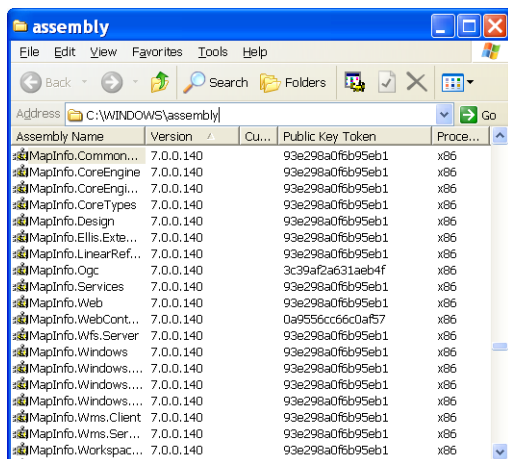
- [Updating Existing Web Sites](#)
- [Updating Existing Desktop Applications](#)
- [Post-back Web Controls](#)

Updating Existing Web Sites

In order to run a web site created with a previous release of MapXtreme, you must edit your application's Web.config file to point to the new versions of the assemblies. The example below shows the four MapXtreme assemblies that must be included. One of the assemblies, MapInfo.CoreEngine.Wrapper, is new beginning with the MapXtreme 2008 v6.8.0 release. Any web application created with MapXtreme v6.7.1 or earlier that you wish to update must include this assembly.



Following the illustration above, edit your Web.config file to include the version number and PublicKey token of the MapXtreme assemblies. To determine version number and PublicKey token, open the Run dialog from the Start menu and type `assembly`. The Global Assembly Cache (GAC) lists all assemblies installed on your system.



You can also reach the Assembly window by navigating from Start > All Programs > Control Panel > Administrative Tools > .NET Framework 2.0 Configuration and choose Manage the Assembly Cache.

When you are finished editing, save your Web.config file and rebuild your web site.

Updating Existing Desktop Applications

Desktop applications created with a previous release of MapXtreme can be recompiled to work in the current release. Controls may have to be re-added to the form.

There are several things you need to do first.

- Set Copy Local property to false
- Add new assemblies to project
- Redirect assemblies to the new assemblies

Set Copy Local property to False

Verify that the Copy Local property for your controls is set to False. There exists a known issue that when you drag and drop a MapXtreme desktop control onto a Windows form, references are loaded that have the Copy Local property set to True. MapXtreme references must point to assemblies residing in the Global Assembly Cache (GAC), not to the local bin path, which is what happens when Copy Local is set to True.

The same behavior exists for ASP.NET Web applications when dragging and dropping MapXtreme web controls onto a form or when building a console application that is not based on a MapXtreme template. The same workaround applies here: Set the Copy Local property to FALSE.

This does not occur when using the MapXtreme Web Site template or sample applications (which are based on the MapXtreme Web Site template).

Add New Assemblies to Project

For applications built using MapXtreme v6.7.1 or earlier, you must also add two assemblies: MapInfo.CoreEngine.Wrapper.dll and MapInfo.CommonDialogExt.dll.

MapInfo.CoreEngineWrapper.dll is required for MapXtreme Web Site applications. Both assemblies are required for desktop applications.

Redirecting MapXtreme Assemblies To Newer Versions

You must redirect your application to use the current version of the assemblies. Microsoft provides several mechanisms to redirect assemblies. For more detailed information, see Microsoft's [.NET Framework Developer Center](#).

Application Configuration File

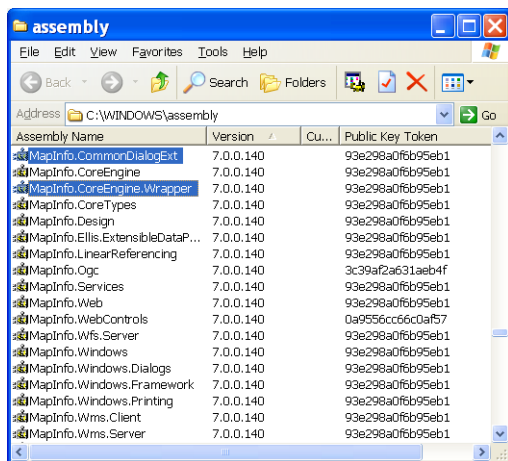
It is recommended that you use an application configuration file to accomplish assembly redirection. The configuration file must be located in the same directory as the application and is named after the application. For example, the configuration file for *myApp.exe* must be named *myApp.exe.config*.

The application configuration file overrides settings in the publisher's policy file.

To redirect assemblies, you must identify the version numbers and PublicKey tokens for the current release and add them to your application configuration file. The version number is in the form X.x.x.x, for example, 7.0.0.532.

To find the information, open the Run dialog from the Start menu and type `assembly`. The Global Assembly Cache (GAC) lists all assemblies installed on your system. You can also reach the Assembly window by navigating from Start > All Programs > Control Panel > Administrative Tools > .NET Framework 2.0 Configuration and choose Manage the Assembly Cache. This tool will also create an application configuration file, if one does not exist.

The illustration below highlights the two assemblies that must be added to desktop applications created with MapXtreme v6.7.1 or earlier.



Publisher Policy File

A publisher policy file containing redirection settings could be installed in the GAC with the assembly. However, this is not a supported MapInfo configuration.

Machine Configuration File

The Microsoft .NET Framework 2.0 Configuration tool can be used to update the machine configuration file, which is located in `\Microsoft.NET\Framework\v2.0.50727\Config` under your Windows installation directory.

Specifying redirection settings here will cause *all* applications referencing the assembly to use the upgraded version. Use this method of redirection carefully, since the machine configuration file overrides settings in both the application configuration file *and* the publisher's policy file.

Post-back Web Controls

The postback-style web controls present in MapXtreme v6.5 and earlier have been marked obsolete in favor of the JavaScript-enabled web controls in the `MapInfo.WebControls` namespace. See [Obsolete Namespace: MapInfo.Web.UI.WebControls on page 15](#) for important information if you are upgrade your web application and wish to continue using the postback web controls.

In the process of deprecating older web controls within the `MapInfo.Web.UI.WebControls` namespace, the default install (SDK, Runtime, merge modules) no longer creates the virtual directory `MapXtremeWebResources_7_0_0`. The MapXtreme Web Site templates have the resources within a subdirectory, `MapXtremeWebResources`. If you're

upgrading older web applications using the deprecated web controls, you will have to manually create the IIS virtual directory, MapXtremeWebResources_7_0_0 with the same content as your existing virtual directory, MapXtremeWebResources_6_x[_x].

Bug Fixes

Issues in the following areas of the product were resolved during the development of MapXtreme 2008 v7.0.0:

- **Data Access**
- **Mapping**
- **Grid**
- **WMS, WFS**
- **Web Tools, Applications**
- **Desktop Tools, Applications**
- **Tables**
- **Object Editing**
- **Persistence**
- **Seamless, Rasters**

Data Access

Oracle Seamless Table (T23931)

The issue of opening a seamless table from an Oracle database has been resolved.

UpdateFeature in MS Access (UC11031)

The UpdateFeature method to update records in a Microsoft Access table has been corrected.

MS Access Table (UC11048)

Table names in MS Access containing spaces will now open properly.

SearchNearest on MultiPolygons (UC11024)

SearchNearest using a point against a FeatureGeometryCollection that contains a MultiPolygon is now returning the collection as expected.

MISQL Count (T27778)

MI SQL select queries using "is not null" now return the correct records.

Memory Leak Due to Specific OrderBy (T27107)

The memory leak that occurred when a specific OrderBy was used on query definition has been resolved.

AccessViolationException (S10041)

An AccessViolationException is no longer thrown when performing multiple updates.

Mapping

Curved Labels in MapInfo Professional (T23821)

Curved labels created in MapInfo Professional and saved as a MapInfo workspace (.MWS) now display in the correct location when opened in MapXtreme 2008.

Rotating LegacyText (T26820)

The description in the Developer Reference related to LegacyText has been clarified. The point of rotation for a LegacyText object is in the upper right.

MapBasic Brush and Fonts (N15523 and N15525)

StyleFactory.FromMBString() method now recognizes MapBasic brush and Font clauses correctly. Additionally the Brush clause now requires a Pen clause to precede it. Support has been added for Font clause and Font + Pen clauses.

MapBasic Symbol with Null Style (N15526)

MapXtreme 2008 now inserts a default symbol style for a feature that is added to the Map when the style is defined by StyleFactory.FromMBString() with a null symbol style. Previously this situation ended in a System.AccessViolationException.

Composite Styles from MapBasic Strings (T21859)

MapBasic strings for composite styles now disregard any associated text style components and render the remaining style components correctly.

LabelSource Alias (UC11051)

The LabelSource alias is now being stored in the workspace as an attribute in the LabelSource tag. For example:

```
<LabelSource maxLabels="2147483647" name="USA" alias="Eif">
```

Anti-Aliasing/Translucency Memory Leak (UC11050)

A reported possible memory leak when interacting with a Raster layer with anti-aliasing and translucency enabled has been corrected.

Legend Font (UC11045)

Overlapping text in a theme legend when using TextEffect.Box and Translucency is on has been resolved.

White Symbol Styles Rendering as Gray (T25277)

The situation where symbols with white fills were rendering as gray has been resolved.

Style ColumnUpdates in RDBMS (T29012)

Style columns in remote database management systems are now being updated correctly. The problem was identified in SQL Server 2008 when the column ordinal of the style column was mis-interpreted as that of the Key column and, therefore, skipped during the update. Style column updates in database management systems is column schema order dependent.

Exporting to JPEG2000, WBMP and PSD (T26580)

When exporting a map or legend to an image in JPEG2000, WBMP or PSD, MapXtreme 2008 now throws an explicit exception that the formats are not supported by the System.Drawing.Image object.

Grid

Grid Table Path (T27616)

Grid tables no longer require an associated .TAB file in order to be opened in MapXtreme 2008.

Grid Inflection in .MWS (T27391)

MapXtreme 2008 now can read grid inflection information that was saved to an .MWS workspace by MapInfo Professional. Previously when the workspace was opened in Workspace Manager and the Grid Style dialog button was clicked, MapXtreme would throw an exception.

WMS, WFS

WMS TableInfo Column type (T27218)

The WmsClient column definition and the attribute data type for features created with an MI DataReader are now synchronized. Previously the MIDataReader would populate the TableInfo column type with attributes created from a RasterInfo.

WMS FeatureLayer Alias (T26571)

A GetFeatureInfo request where the alias for a WMS FeatureLayer is different from the DataSourceRef attribute no longer returns "Object reference not set to an instance of an object." MapXtreme 2008 now sets the table alias to the associated DataSourceName and leaves the FeatureLayer alias to what is specified in the WMS Server configuration file.

WMS and FeatureInfo (UC11049)

MapXtreme 2008 now correctly returns the FeatureInfo for one selected feature within a WMSClient. Previously it returned the FeatureInfo for all the features within the MBR.

MapXtreme WFS Server and MapInfo Professional

This release of MapXtreme 2008 fixes the interoperability issue between the MapXtreme's WFS Server implementation and MapInfo Professional as a client. In the previous version of MapXtreme, changes were made to pass the OGC compliancy test, but MapInfo Professional no longer recognized it as a MapXtreme WFS server. A new xml reader method was introduced to specify the namespace corresponding to the feature. For a GetFeature request to a MapXtreme WFS Server, property names that are missing namespaces will now use the column namespace. Previously an exception was thrown for non specified namespaces.

WFS and Namespaces (T26573 and T27523)

MapXtreme 2008 has added new namespace prefix parameters to the PropertyName and TypeName constructors for a WFSClient. This was in response to an issue where a certain request that worked correctly in MapInfo Professional did not work in MapXtreme v6.8.

Web Tools, Applications

WebTools in Firefox (T24307)

Web tool cursors are now working as expected in Firefox.

JavaScript Updated for Web Controls

The Javascript (*.js) portion of the MapXtreme web controls has been updated to better support cross browser compatibility, namely for IE7, IE8, and Firefox 3.

Pan Tool in Web Applications (T23676)

The Pan tool for web applications has been modified to correct a problem with automatically updating the other controls. It now works as expected.

Panning Performance (N15438)

Panning a map containing data on a file server is now performing as expected. Previously the redrawing operation paused multiple times before the mouse click ended.

Web Sample Applications and State Management (T27953)

Layer settings in the web sample applications are now being maintained correctly in Layer Control by the AppStateManager. This caused users to see changes such as layer visibility, that another user had made. This behavior was seen in the HelloWorld sample application.

Desktop Tools, Applications

Desktop Select Tool (UC11033)

NodeChanged event now fires correctly when using the desktop Select Tool.

Snap Cursor (T28357)

Desktop selection tools no longer leave snap cursor artifacts on the screen.

MapFactory and AddTextMapTool (UC11022)

The AddTextMapTool can continue to edit legacy text objects after the MapFactory has been cleared. Now, when the MapControl's associate map changes, the AddTextMapTool's "SelectToolData" is updated as well, to reflect the new map. Previously, it was searching for text to select on a map that no longer existed.

MapControl Memory leak (T29352)

A MapControl memory leak has been resolved for the situation where a Windows form is shown and hidden multiple times.

Tables

Saving View Tables (T26110)

Workspaces containing view tables defined using MI SQL aggregations are now saved correctly.

SpatialWare Join Query (T27411)

SpatialWare join query containing a string literal no longer throws an exception.

CSV Files (T15522)

MapXtreme can now open .CSV files in addition to .TXT files. Other ASCII file types cannot be opened.

Indexed Column and Update (T28099)

The issue surrounding the Update function that uses an indexed column in the where clause has been resolved.

Text TAB Files (T28422)

MapXtreme 2008 can now open a text .TAB file when the filenames for the text file and .TAB are different.

Object Editing

FeatureGeometry.Distance (T28336)

FeatureGeometry.Distance method is now yielding the correct result in the situation where a cutter object contains interior rings. The limitation to this fix is that it does not support Linear cutter objects. An exception will now be thrown if the cutter object is linear.

Persistence

Workspaces (T26915)

MapXtreme 2008 now provides better error messaging when MapXtreme encounters an invalid workspace or geoset using MapWorkspaceLoader or MapGeosetLoader.

MapXtreme accepts any valid workspace or geoset regardless of the file extension used (unless they are invalid file system characters). MapXtreme will not open a .WOR created by MapInfo Professional, however, if the workspace is a valid MapXtreme workspace or geoset with an extension .WOR, it would open in MapXtreme.

Seamless, Rasters

Seamless Layer Pure Virtual Function Call Error (UC11041)

The issue has been resolved when selecting an object from a seamless layer with the Select tool caused a R6025 Virtual Function Call error.

Note As a result of this fix, the public API has been changed. IFeatureEnumerator is now inherited from System.IDisposable. You must call Dispose when you use an object that implements IFeatureEnumerator. See [Obsolete Methods and Properties on page 15](#).

Seamless Raster Base Tables (T26909)

The situation where opening two seamless base tables with the same name (beginning with a numeric) only recognized the raster image from the first base table, has been resolved. The second base table was never completely opened and the seamless component referenced the first table.

Known Issues

The following are issues uncovered during the development and testing of MapXtreme 2008:

- [64-bit Systems and Platform Target x86](#)
- [Switching Between .NET Framework 1.1 and 2.0 on 64-bit Windows Systems](#)
- [ADO.NET Serialization](#)
- [Data Access](#)
- [Enhanced Rendering](#)
- [Licensing](#)
- [Linear Referencing](#)
- [Mapping](#)
- [Object Editing](#)
- [Styles](#)
- [Visual Studio](#)
- [Web Applications](#)
- [Web Controls](#)
- [WMS](#)
- [Workspace Manager](#)

64-bit Systems and Platform Target x86

On a 64-bit machine, you may get a BadImageFormatException when trying to run a MapXtreme 2008 desktop application. Since MapXtreme runs in 32-bit emulation mode on 64-bit systems (WOW64), you must assign the application you are building in Visual Studio to use the Platform Target x86. In the project properties, click on the Build tab and choose x86 for the Platform Target.

Switching Between .NET Framework 1.1 and 2.0 on 64-bit Windows Systems

MapXtreme 2008 is a 32-bit application that runs on 64-bit systems using Microsoft's WOW64 technology. It is built on top of the .NET Framework 2.0. MapXtreme Web Sites or ASP.NET Web Sites or Applications run under ASP.NET 2.0.

If you have a side-by-side installation of MapXtreme that must run under the .NET Framework 1.1 (v6.6 and earlier), follow the instructions below to switch from one framework to the other. See <http://support.microsoft.com/kb/894435> and <http://support.microsoft.com/kb/911720> for more information.

Using MapXtreme (v6.6 or earlier) on the .NET Framework 1.1 on a 64-bit System

1. Execute the following script to enable 32-bit mode:

```
cscript %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET W3SVC/AppPools/Enable32bitAppOnWin64 1
```

2. Execute the following command to install the version of ASP.NET 1.1 and to install the script maps at the IIS root and under:

```
%SYSTEMROOT%\Microsoft.NET\Framework\v1.1.4322\aspnet_regiis.exe -i
```

3. Under Web service extensions in IIS Manager, ensure ASP.NET version 1.1.4322 is set to Allowed.

Using MapXtreme (v6.7 or later) on the .NET Framework v2.0 on a 64-bit System

1. Execute the following script to enable 32-bit mode:

```
cscript %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET W3SVC/AppPools/Enable32bitAppOnWin64 1
```

2. Execute the following command to install the version of ASP.NET 2.0 (32-bit) and to install the script maps at the IIS root and under:

```
%SYSTEMROOT%\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis.exe -i
```

3. Under Web service extensions in IIS Manager, ensure ASP.NET version 2.0.50727 (32-bit) is set to Allowed.

ADO.NET Serialization

Because of situations in which we cannot reliably re-establish references to a shared DataTable instance (resulting in disjoint copies), the automatic serialization of TableInfoAdoNet-based tables is currently unsupported. Chapter 10 in the MapXtreme 2008 Developer Guide outlines a manual procedure that can be used until this capability can be properly and reliably supported.

Data Access

Oracle GroupBy and Views (T26413)

MapXtreme is unable to open an Oracle view that was defined using a DISTINCT or GroupBy clause. The following code suggests a workaround: The KeyType property is set to Explicit, and the StringCollection in the KeyColumns collection is used to provide the name of the column(s) in the resultset that can be used as the key for the table. As an alternative to the code below, the KeyType.Explicit can be persisted in the MapInfo Workspace (.MWS).

```
Catalog c = Session.Current.Catalog;
```

```
    TableInfoServer tis = new TableInfoServer("foo");
    tis.Toolkit = ServerToolkit.Oci;
    tis.ConnectionString = @"srvr=ORA10GR2;uid=at-smipro;pwd=at-smipro";
    tis.Query = @"select * from ATSMIPRO.T26413";
    tis.KeyType = KeyType.Explicit;
    StringCollection sc = new StringCollection();
    sc.Add("STATE");
    tis.KeyColumns = sc;
    tis.CacheSettings.CacheType = CacheOption.Off;
```

```
Table t = c.OpenTable(tis);
```

Data Reader Exception Using Stored Procedure (T24303)

Executing a data reader on a table using a stored procedure on a XY database will throw an error. As a workaround, apply a SpatialSchemaXY to the TableInfoServer.

MS Access Failed Connections

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 Microsoft technical publication ["Using Microsoft Jet with IIS"](#). Microsoft recommends using MS SQL Server with IIS for web applications that require absolute data integrity or high user concurrency.

Enhanced Rendering

Enhanced rendering (anti-alias and translucency) is covered in detail under “Using Workspace Manager Features” in Chapter 24: Workspace Manager, in the MapXtreme 2008 Developer Guide.

Translucency Slider Controls (T22617)

The translucency slider controls in the Style dialogs have no effect unless enhanced rendering (translucency) is enabled. When enhanced rendering is disabled, the translucency slider controls in the styles dialogs appear to be enabled, yet have no effect.

To enable translucency, in Workspace Manager, check the Enable Translucency under the Style tab. Programmatically, set the `MapInfo.MapInfo.DrawingAttributes.EnableTranslucency` property.

Anti-Alias with Translucency (T23343)

Using anti-aliasing with translucency will temporarily distort labeling when panning.

Range Theme and Dot Density Color Picker Dialogs (T22856)

The color picker dialog used by ranged themes and dot density themes does not have support for translucency. This means that, when using this dialog, you cannot choose a translucent color value for the inflection point of a ranged theme nor a translucent dot color for a dot density theme. You can do it programmatically.

Note: The other MapXtreme 2008 style dialogs, such as `LineStyle`, `AreaStyle`, `TextStyle` and `SymbolStyle`, use a different color picker dialog that does support translucency.

Licensing

Web Services (T16917)

The licensing in MapXtreme 2008 is unable to determine if an application is running as a desktop or web application when deployed as a Web Service. For web applications use a desktop runtime license along with your web runtime license.

Linear Referencing

PerpendicularOffset (IN14828 and IN14832)

The linear referencing operation `PerpendicularOffset` does not yield the correct results in the situations where a linear object contains multiple intersections or self intersections.

Mapping

Panning (T19620)

When panning a map that uses a scale range for visibility, white patches display over the map,

Map Center Point Incorrect After Pan and Zoom (T20132)

ASP.NET session state management problems can occur when using `StateServer` and `SqlServer` under .NET 2.0 framework. When performing a zoom or pan operation on the map, the new center point of the map image is offset incorrectly. This problem was encountered only in Visual Studio 2005 using the 2.0 Framework.

Mapping (*continued*)

Exporting to a Higher Resolution (T20744)

Exporting a map with labels to a higher resolution increases the size of the labels.

Long Label Names (T21894)

When label names are very long and a label priority is used to eliminate overlap, the resulting distribution of labels is too sparse. There are large areas of the map where labels could be drawn without overlapping, but did not.

Custom Labels (T23652)

Custom labels cause large memory usage when cloning a map.

Curved Labels (T23790, T23760, T23827, T23856)

When curved labels is on the label tool does not draw the labels curved. This happens when the street layer visibility is turned off.

Curved labels with the default setting of not allowing overlapped labels results in too many curved labels not displaying.

Zooming far in on a curved label will make it cross the street segment.

Clicking and dragging a curved label does not work as expected.

Duplicate Legends (T18421)

Re-opening a workspace with a theme and saving it creates two legends in the workspace.

Appended Custom Legends (T20468)

Saving a map and an appended custom legend to a workspace results in an error.

Theme Legend Serialization (T28211)

Row text changes in a theme legend are not being deserialized. Create the legend with CustomLegendFrameRow to maintain any changed text during the serialization/deserialization process for the legend.

The AllOtherRows.Visible property is not being serialized correctly, thus the row text is getting cut off in the LegendControl.

Legend Border (T29163)

Setting MapInfo.Mapping.Legends.Legend.Border property to True only draws 2 sides of the legend.

Object Editing

Geometry.FeatureGeometry.Difference Method (T29465)

MapInfo.Geometry.FeatureGeometry.Difference method throws null reference exception when attempting to erase a Geometry from a smaller Geometry contained within it.

Styles

Disposing CompositeStyles (T17406)

CompositeStyles do not contain a way to free the resources used, causing memory leaks under some conditions. In the following code example, to avoid a memory leak, set the CompositeStyle to null.

```
private MapInfo.Styles.AreaStyle _style = new MapInfo.Styles.AreaStyle();
for (int i = 0; i < 1000000; i++)
{
    MapInfo.Styles.CompositeStyle cs = new MapInfo.Styles.CompositeStyle(_style);
}
cs.AreaStyle=null;
```

Visual Studio

Desktop Application Deployment (T26033)

An issue exists when you are building your own installer for a desktop application, Visual Studio does not include the following dependencies and their policies:

- Microsoft_VC80_CRT_x86.msm, policy_8_0_Microsoft_VC80_CRT_x86.msm
- Microsoft_VC80_DebugCRT_x86.msm, policy_8_0_Microsoft_VC80_DebugCRT_x86.msm

Visual Studio 2005 Service Pack 1 and Visual Studio 2008 use an updated C runtime library. The custom installer might not work if deployed to a machine without the updated C runtime library.

Web Applications

MapXtreme Assembly References and Microsoft's ASP.NET Web Application Template (T25937)

This issue affects users of MapXtreme 2008 (v7.0.0) who have side-by-side installations of MapXtreme integrated with Visual Studio 2005 SP1 or Visual Studio 2008 or both.

When using the ASP.NET Web Application template (File > New > Project) and you drag and drop MapXtreme web controls onto your form, the Visual Studio designer automatically adds references to the MapXtreme assemblies. These references are added to several files: the web project file (.csproj or .vbproj), the web application web configuration file (Web.config), and the web page itself (.aspx). It is possible that one or more of these assembly references has an incorrect assembly version number.

For example, you have two versions of MapXtreme installed on your system (v6.8.0 and v7.0.0) and you add a MapXtreme LayerControl from v6.8.0 onto your ASP.NET form. Your web project file may reference the v6.8 version of the LayerControl while the Web.config and .aspx files reference the v7.0.0 LayerControl. This will throw errors when you debug your application. There has been no established pattern for this behavior.

This does not occur when you are using the MapXtreme Web Site template (located under File > New > Web Site).

To resolve this assembly mismatch, you must manually correct the version numbers in each file to match the version of the web control you used. To determine the full version number for each installation of MapXtreme, view the assemblies in the Global Assembly Cache (GAC). Assemblies for v6.8.0 display 6.8.0.536. Assemblies for v7.0.0 display 7.0.0.190.

DataAccess Web Sample Application

The DataAccess web sample application does not work in 64-bit systems unless you change an entry in the NamedConnections.xml file to point to your installation folder. The entry is currently:

```
<FilePath>C:\Program Files\MapInfo\MapXtreme\7.0.0\Samples\Data</FilePath>
```

For x64 system with a default installation location for MapXtreme 2008, change the entry to:

```
<FilePath>C:\Program Files (x86)\MapInfo\MapXtreme\7.0.0\Samples\Data</FilePath>
```

If your installation location is something else, change the entry accordingly.

Double-byte Characters in Web Applications (B10104)

The Japanese version of Internet Explorer 6 may incorrectly encode double byte characters in a URL request. This may cause errors in web applications where MapAlias and LayerAlias names have double byte characters (Japanese text). This issue is specific to Internet Explorer 6.

Web Controls

MapAlias Is Required at Design Time (T17815)

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.

The MapXtreme 2008 web template and samples provide a default MapAlias called "Map1".

MapAlias and Session Management (T18518)

Changing to a different MapAlias for a MapControl at runtime restores incorrect session values. The default session management settings in MapXtreme 2008's project templates and sample applications only store changes to the HTTP session, rather than the entire session. As stated in the above known issue, the MapAlias is the key to tying the changes to the correct session. If the MapAlias is changed at runtime, the session will return state that was for the previous MapAlias.

Web tools and Layer Control (IN14935)

The web tools (ZoomIn, ZoomOut, RadiusSelection, RectangleSelection, PolygonSelection, PointSelection, Distance, and Center) are not working when a LayerControl already exists on the web page. As a workaround, add LayerControl after adding the web tools.

WMS

WMS Server (UC11012)

A WMS service created with MapXtreme 2008 is not displaying polygons in .GIF format correctly.

Workspace Manager

Saving Changed Values (T21168)

Changed values are not included when a workspace is saved. For example, if you change one of the zoom values on the Visibility tab then immediately click Save, the change is not preserved upon exiting Workspace Manager and reopening the workspace. However, if you click away from the control containing the changed value (that is, lose focus) prior to clicking Save, the changed value is saved.

System Limitations

Spatialite Extension and Workspace Manager (IN15074)

Workspace Manager fails to load the Spatialite database extension on Windows 7, Windows Vista and Windows Server 2008 unless you run the tool as administrator.

Web Controls Cursor Image Not Changed (IN14781)

If you change the cursor (.cur) or image file (.gif) for a web control in Visual Studio, the changed image will not be visible at runtime. This is due to a Visual Studio limitation in that the absolute path to the image location gets changed to a relative path. MapXtreme expects an absolute path.

To remedy this, after you have chosen the new image via Visual Studio's Select URL dialog, remove the ~/ from the front of the path for the ImageURL you changed. For example, if you changed the CursorImageURL to use a different cursor, change ~/MapXtremeWebResources/MapInfoWebNewCursor.cur to MapXtremeWebResources/MapInfoWebNewCursor.cur. You can also substitute ./ (dot-forward slash) for ~/ . The Select URL dialog displays when you click on the ellipse button for the CursorURL in the Properties window for the tool.

Documentation Updates

State Management and Web Sample Application

Several of the web sample applications contain unclear comments in the code (WebForm1.aspx.cs/vb) about how and when to configure state management. The following is meant to clarify those remarks:

"The code below shows how to configure the application to manage state manually, as recommended by MapXtreme best practices. Manual session state means that the developer controls how much information in the session is to be saved. This is the most efficient manner of saving state.

If you wish to save the entire session with every new request, you must set MapInfo.Engine.Session.State to HttpSessionState in the web.config file."

For more information on state management, see Chapter 6 in the Developer Guide.

FeatureStyle.Modifiers and Layer Ordering

The documentation remarks for the IndividualValueTheme class is missing information related to how MapXtreme handles the ordering of themes and style overrides, and an explanation of when to use Append versus Insert.

Consider the case where a map has several layers, each of which has a style override (opaque color applied to features). Following the code snippet from the Developer Reference (shown below), you add an IndividualValueTheme to one of the layers using the Append method. However, the theme does not display on the map.

```
public static void MapInfo_Mapping_Thematics_IndividualValueTheme(Map map) {
    // Load a map based on one table
    map.Load(new MapTableLoader("Mexico.tab"));
    FeatureLayer fLyr = map.Layers["Mexico"] as FeatureLayer;

    // Create an individual value theme
    IndividualValueTheme thm = new IndividualValueTheme(fLyr, "Pop_90", "Mexico Pop");

    // Add the theme to the FeatureStyleModifiers list
    fLyr.Modifiers.Append(thm);
}
```

Since the layer already contains a style override and you added the theme with the Append method, the theme was underneath the style override and thus was not visible. To display the theme in this case, use `FeatureLayer.Modifier.Insert(0, thm)` where 0 is the first modifier in the collection and `thm` is the theme object.

Order does matter when applying modifiers to a layer. This information is also relevant to other theme types.