



# MapXtreme 2004 Release 6.1 Developer Guide Supplement

January 2005

## In these notes:

---

♦ <a href="#">Introduction</a> . . . . .	1
♦ <a href="#">Upgrading to MapXtreme 2004 Release 6.1</a> . . . . .	2
♦ <a href="#">Managing MapInfo Session State</a> . . . . .	3
♦ <a href="#">Bug Fixes in MapXtreme 2004 Release 6.1</a> . . . . .	17
♦ <a href="#">Known Issues</a> . . . . .	29

## Introduction

---

### Who Should Read This Document?

This document is a Supplement to the MapXtreme 2004 Developer Guide. This Supplement includes not only upgrade instructions, but also covers new features and describes important techniques that are *not* described in the existing MapXtreme 2004 Developer Guide. As such, this Supplement is required reading for all MapXtreme 2004 administrators.

This document also incorporates the Release Notes from all releases of MapXtreme 2004:

- version 6.0 April 30, 2004 (Build 75)
- limited release of June 30, 2004 (Build 78)
- version 6.1 December 3, 2004 (Build 104)

The latest information pertaining to Release 6.1 is contained in the section [Bug Fixes in MapXtreme 2004 Release 6.1 on page 17](#). Issues that are currently under investigation or in development seeking resolution are listed in [Known Issues on page 29](#).

---

## Upgrading to MapXtreme 2004 Release 6.1

---

Uninstall any trial or beta version of MapXtreme that you may have on your computer. To upgrade to MapXtreme 2004 Release 6.1, run Setup.exe and follow the prompts. You can install Release 6.1 side-by-side with Build 75 of Release 6.0. If you are uncertain which build number you have installed, select **Start > Run** and type **assembly**.

**Note:** If you had installed the limited release (Build 78), you *must* uninstall that build before installing Release 6.1.

### Build 75 Desktop Applications

Desktop applications created with Build 75 can be re-compiled if you are upgrading to Release 6.1. Controls may have to be re-added to the form.

Alternatively, you can also redirect your applications to use the new MapInfo assemblies without recompiling. This technique instructs your application to look for an upgraded version of a referenced assembly. For more complete instructions on this application configuration procedure, see [Redirecting Assembly Versions on page 9](#).

### Build 75 Web Applications

For web applications created with Build 75, you must change the project version numbers from the old version number to 1.1.0.104 to develop your application with Release 6.1. For more complete instructions on changing project version numbers, see [Creating or Recompiling Web Applications on page 7](#).

### Relocating License Files

Current MapXtreme 2004 users (Release 6.0) must copy their license files to the corresponding 6.1 folder (C:\Program Files\Common Files\MapInfo\MapXtreme\6.1).

### Web Sample Applications

For Web sample applications, you must create an appropriate virtual directory. The virtual directory is created for C# sample applications. For VB, you can change the virtual directory created for the C# application to point to the VB application.

For example: In a default installation, the C# sample application ThematicsWeb located in C:\Program Files\MapInfo\MapXtreme\6.1\Samples\Features\ThematicsWeb\cs has a virtual directory already installed. You should change this virtual directory to use it for the Visual Basic application ThematicsWeb.

1. From the Control Panel, open IIS. Locate the ThematicsWeb folder in the default Web sites.
2. Right click the folder and change the local path from:  
C:\Program Files\MapInfo\MapXtreme\6.1\Samples\Features\ThematicsWeb\cs to  
C:\Program Files\MapInfo\MapXtreme\6.1\Samples\Features\ThematicsWeb\vb.
3. Restart IIS.

4. Use Task Manager to stop and restart aspnet\_wp.exe; in Windows 2003-specific operating systems, the aspnet process is named w3wp.exe.

Follow the same basic procedure for other Visual Basic sample applications.

## Sample Data Installer

If you installed sample data with MapXtreme 2004 Release 6.0, you should uninstall this data before attempting to install new sample data provided with the Release 6.1 free sample data installer. The uninstallation will also remove any data that was changed by the customer.

**Note:** The data uninstaller will remove data even though it has been altered by the customer. Customers that have changed the sample data and want to keep their changes should move the data to another location before uninstalling.

The 6.1 Data Installer will fail if it detects old sample data.

## Windows XP Service Pack 2

Microsoft Windows XP Service Pack 2 (SP2) causes some known issues with the MapXtreme 2004. The issues are mostly related to the added security features. By default Windows XP SP2 closes many ports that were previously used to communicate with other computers. Also when Windows Firewall is enabled, it blocks unsolicited connections to your computer and this behavior causes problems with MapXtreme 2004.

Microsoft has a Knowledge Base article that describes how to selectively enable programs to run as they did before Windows XP SP2 was installed. By adding MapXtreme 2004 to the of exceptions, you can circumvent the difficulties you may encounter when using MapXtreme 2004 with Windows XP SP2.

See the Microsoft Knowledge Base article at:

<http://support.microsoft.com/default.aspx?kbid=842242>

For an FAQ How Windows XP Service Pack 2 (SP2) Affects SQL Server and MSDE, see:

<http://www.microsoft.com/sql/techinfo/administration/2000/security/>

---

## Managing MapInfo Session State

### ISession Interface

The MapXtreme 2004 MapInfo.Session Namespace includes the ISession interface, which defines access to most of the information that is central to an application (for example, maps and tables). ISession also manages the initialization of resources needed for a MapXtreme application.

In an ASP.NET application, the state of the ISession is unique to each ASP.NET session because a session represents a client and each client has its own state. Every time the client makes a request, the request retrieves the last state of the session, executes and then places the new state back into storage.

The WebSessionActivator is responsible for automatically setting up the ISession instance so that it is accessible through the Session.Current static property and it has the proper state specific to the ASP.NET session being processed. The WebSessionActivator is an IHttpModule and has access to ASP.NET application-specific events that define the lifetime of a request. The following is a description of what the WebSessionActivator does during these events:

- BeginRequest

The WebSessionActivator creates a new ISession instance (or acquires one from the pool if MapInfo.Engine.Session.Pooled setting is true in the web.config file) and stores it in the request-specific storage so that ISession is accessible through the Session.Current property.

If sessionState mode setting is InProc and MapInfo.Engine.Session.State setting is HttpSessionState in the web.config file, then this step is deferred until AcquireRequestState because the ISession instance is stored in memory and it must be retrieved from memory storage.

- ...

- AcquireRequestState

If MapInfo.Engine.Session.State setting is HttpSessionState and sessionState mode setting is StateServer/SQLServer, then state deserialization is performed against the current ISession instance in the request-specific storage.

If sessionState mode setting is InProc, the WebSessionActivator retrieves the existing ISession instance from memory (ASP.NET session state) and stores it in the request-specific storage so that it is accessible through the Session.Current property.

- ...

- IHttpHandler execution

This is the step in which your web page or web service-specific code executes. The ISession instance in the request-specific storage is accessible through the Session.Current property.

- ...

- PostRequestHandlerExecute

If MapInfo.Engine.Session.State setting is HttpSessionState, this is when the WebSessionActivator stores the ISession instance in the ASP.NET session state (regardless of the sessionState mode setting).

- ...

- ReleaseSessionState

ASP.NET engine saves the ASP.NET session state in memory or serializes into server storage depending on the sessionState mode setting.

- ...

- EndRequest

The WebSessionActivator disposes the ISession instance (or returns it to the pool MapInfo.Engine.Session.Pooled setting is true).

If MapInfo.Engine.Session.State setting is HttpSessionState and sessionState mode setting is InProc, the ISession instance is instead deactivated from the request-specific storage but it is not disposed of because it will be reused for the next request of that same session instance.

## Using Session.Dispose(HttpSessionState) Method

This section describes when and how to call the Session.Dispose(HttpSessionState) Method From Session\_End.

All web applications using MapXtreme 2004 web controls default to the following web.config settings (even if not present in your web.config file):

```
<add key="MapInfo.Engine.Session.State" value="HttpSessionState" />
<sessionState mode="InProc" />
```

The first setting is an application-specific setting that controls the mechanism for saving and restoring the state of the MapInfo.Engine.ISession instance. This instance is accessible through the MapInfo.Engine.Session.Current static property. The HttpSessionState setting indicates that the session is saved and restored through the ASP.NET session state. This state is exposed through the current HttpContext and is of type HttpSessionState.

The second setting is an ASP.NET setting that controls how the HttpSessionState is saved and restored. The default (InProc) indicates that the contents of the ASP.NET session state are to be placed in memory and are unique to each ASP.NET session instance.

When you use these settings, there is an ISession instance for each ASP.NET session and it is stored in the HttpSessionState throughout the lifetime of the ASP.NET session. In order for the ISession instance to be properly disposed of when the session times-out or ends, you must add the following statement to your Global.asax Session\_End method.

```
MapInfo.Engine.Session.Dispose(this.Session);
```

in C# or

```
MapInfo.Engine.Session.Dispose(Me.Session)
```

in VB.NET

Making this call will ensure that ISession instance is properly disposed and memory is reclaimed so that the application does not run out of memory.

You do not need this call for any other configuration, since the ISession instance is not stored in memory in any other configuration.

## Using MapInfo.Session.Engine

This section describes when and how to use the MapInfo.Session.Engine.UseCallContext web.config setting.

### Session.Current Issues with Response.Redirect and Server.Transfer

When you use Response.Redirect or Server.Transfer in a web application, there is a possibility that multiple requests for the same ASP.NET session instance will be received by the web server at the same time. This higher load can cause parts of the request execution to occur in different threads. For example, BeginRequest might execute on a different thread than its IHttpHandler execution, which would produce an unhandled exception. The risk is even greater when multiple frames are introduced.

The WebSessionActivator stores the ISession instance in a request-specific storage. By default, this storage, is internally implemented using a thread-specific storage. Given the stated behavior when using redirect/transfer, the following issues can surface:

- The IHttpHandler might not have access to its request specific ISession instance if it executes in a different thread. The following application exception is thrown:  
*The session is not active either because the WebSessionActivator is not registered or you are attempting to access the session in your IHttpHandler's constructor. To register the WebSessionActivator add the following entry to your web.config file: <httpModules><add type="MapInfo.Engine.WebSessionActivator, MapInfo.CoreEngine, Version=<version>, Culture=neutral, PublicKeyToken=93e298a0f6b95eb1" name="WebSessionActivator" /></httpModules>. Replace <version> with the current version of MapInfo.CoreEngine.dll assembly.*
- The WebSessionActivator fails to store the ISession instance in the request-specific storage because it is executing in a thread where another request was executing:  
*There is an active session already.*

To solve this problem, you can switch the type of storage the ISession instance is stored from thread-specific to call context-specific. The call context-specific storage allows the ISession instance to remain the same within the lifetime of the ASP.NET request, regardless of which thread each part of the request executes in. To enable this call context-specific storage feature, add the following application-specific setting in your web.config file:

```
<add key="MapInfo.Engine.Session.UseCallContext" value="true" />
```

You should no longer see the issues described above once you use this setting.

The UseCallContext setting will impact performance and your application may run slower.

### Session.Current Issues Using Asynchronous Calling with WebService Clients

The web services client in Visual Studio .NET support asynchronous invocation of web service methods. You can get information on using the SoapHttpClientProtocol.BeginInvoke method by searching the MSDN web site at <http://www.msdn.com>.

If your server is an ASP.NET web service implementation (asmx) and you asynchronously invoke multiple methods from the client for the same session instance, the following issues can occur:

1. For web service methods that specifically enable ASP.NET session state on the server side (you need to setup a cookie container for it to work properly on the client side):
  - a. Some of the requests are processed under a separate session state rather than using the same state in a synchronous manner. This means that the session state is compromised. This issue is independent of MapXtreme 2004. There is no recommended solution to this issue.
  - b. For those requests that do share the same session state, their deserialization can occur outside of the request call context. If the UseCallContext setting is true, this causes an exception because the session cannot be reached (there is no context available). If UseCallContext is false, deserialization occurs in a separate ISession instance which is not reachable by the rest of the request execution; this causes the state to be lost.

There are no recommended workarounds or solution to these two issues other than to avoid asynchronous calling from the client.
2. For any web service method (regardless of whether or not ASP.NET session state is enabled), parts of the request lifetime might execute in different threads resulting in issues similar to the Response.Redirect/Server.Transfer issues described above. In this case, we recommend you set UseCallContext to true to solve this issue.

---

## Web Configuration Changes

---

### Upgrading WMS/WFS Servers

To upgrade, you must update your web configuration files to use the new WMS and WFS servers.

1. Edit the web.config file to specify MapXtreme 2004 Build 104 for WMS and WFS servers. For example, your current web.config file (for Build 75) might specify Version=1.0.0.75. You must change this to use Version 1.1.0.104. The updated web.config file might appear as:

```
<httpHandlers>
  <add verb="GET,POST" path="*.ashx"
  type="MapInfo.Wfs.Server.HttpHandler, MapInfo.Wfs.Server,
Version=1.1.0.104, Culture=neutral, PublicKeyToken=5539cc02a8223bb3"/>
</httpHandlers>
<httpModules>
  <add type="MapInfo.Engine.WebSessionActivator, MapInfo.CoreEngine,
Version=1.1.0.104, Culture=neutral, PublicKeyToken=93e298a0f6b95eb1"
  name="WebSessionActivator" />
</httpModules>
```
2. Stop and Restart IIS
3. Use Task Manager to stop and restart aspnet\_wp.exe; in Windows 2003-specific operating systems, the aspnet process is named w3wp.exe.

### Creating or Recompiling Web Applications

If you are recompiling an existing web application (or creating a new web application) the ExitProcess key should be omitted or set to false. For example:

```
<add key="MapInfo.Engine.Session.ExitProcess" value="false" />
```

This is a change from Build 75, in which it was necessary to set this key to true.

To recompile a web application from 6.0 to 6.1:

1. Stop IIS and close all instances of Visual Studio.
2. Open the .cs or .vb project file in a text editor, such as Notepad.
3. Change `AssemblyFolderKey = "hklm\mapinfo.net"` to `AssemblyFolderKey = "hklm\mapinfo.net 6.1"`. If no key is present, add it for any MapInfo Assemblies

There are two ways the Assembly references may be made in the Project file. You may find entries similar to one of the following:

- ```
<Reference
  Name = "MapInfo.CoreTypes"
  AssemblyName = "MapInfo.CoreTypes"
  HintPath = "..\..\Program Files\Common
  Files\MapInfo\MapXtreme\6.0\MapInfo.CoreTypes.dll"
  AssemblyFolderKey = "hklm\mapinfo.net"
/>
```
- ```
<Reference
  Name = "MapInfo.CoreTypes"
  AssemblyName = "MapInfo.CoreTypes"
  HintPath = "..\..\Program Files\Common
  Files\MapInfo\MapXtreme\6.0\MapInfo.CoreTypes.dll"
/>
```

After you make necessary changes to the file, it might look like:

- ```
<Reference
  Name = "MapInfo.CoreTypes"
  AssemblyName = "MapInfo.CoreTypes"
  HintPath = "..\..\Program Files\Common
  Files\MapInfo\MapXtreme\6.1\MapInfo.CoreTypes.dll"
  AssemblyFolderKey = "hklm\mapinfo.net 6.1"
/>
```
4. Open `WebForm1.aspx` in your text editor. Change any references from 1.0.0.75 to 1.1.0.104.
  5. Open `web.config` and change any references to 1.0.0.75 to 1.1.0.104.
  6. Restart IIS.
  7. ReOpen the project in Visual Studio and recompile.
  8. Use Task Manager to stop and restart `aspnet_wp.exe`; in Windows 2003-specific operating systems, the `aspnet` process is named `w3wp.exe`

---

## Redirecting Assembly Versions

---

This section describes how to configure your applications to use the new versions of MapInfo assemblies without recompiling. Redirecting an assembly version means telling your application to look for an upgraded version of a referenced assembly. Microsoft has provided a few different mechanisms for accomplishing this.

For more detailed information, see <http://www.msdn.com> and search for the "Redirecting Assembly Versions" to locate the appropriate topic.

### Redirecting Assembly Versions Without Changing Existing Application

There are several methods for redirecting assembly versions.

#### Application Configuration File

We recommend that customers use an application configuration file to accomplish assembly redirection.

You can use the Microsoft .NET Framework 1.1 Configuration tool located under Control Panel, Administrative Tools to configure an application's redirection settings. This tool will create and/or manipulate the application configuration file for your application.

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

After installing MapXtreme 2004 Release 6.1 follow these steps to redirect your assembly versions.

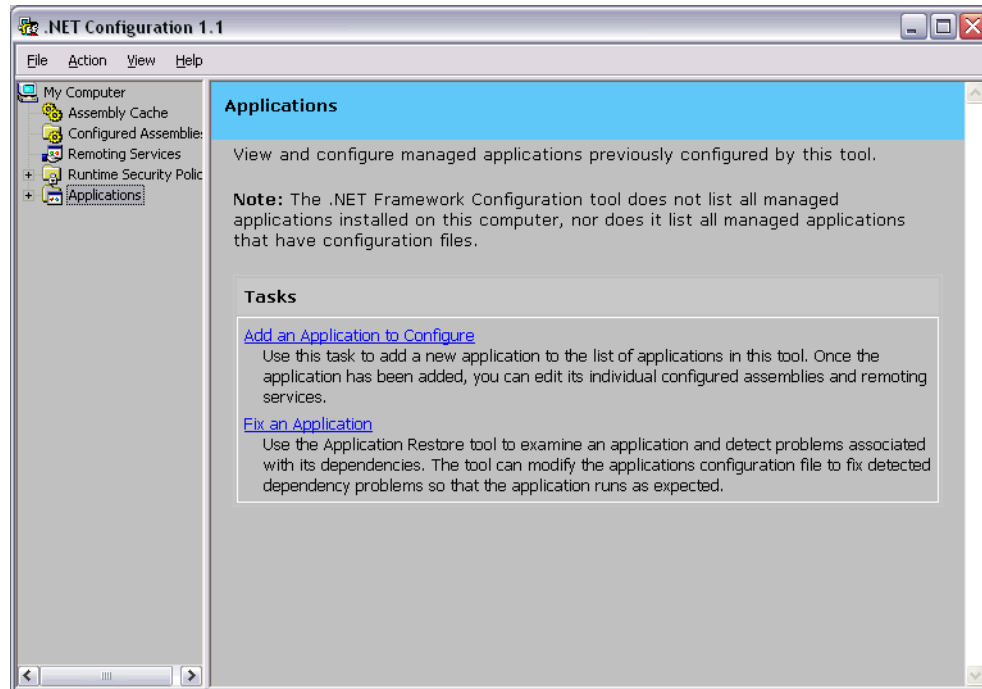
1. Create an application configuration file to redirect your application to load the new assemblies from MapXtreme 6.1.

**Note:** 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*. You can use the configuration tool supplied by Microsoft to accomplish this.

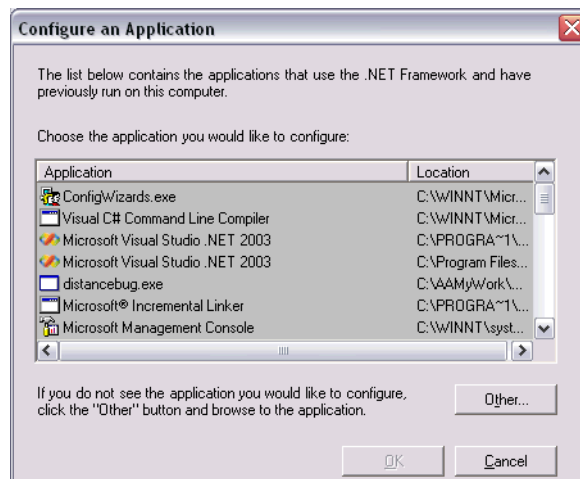
A sample application .config file might appear as follows. The new version is Build 104:

```
<?xml version="1.0"?>
<configuration>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="MapInfo.CoreEngine"
publicKeyToken="93e298a0f6b95eb1" />
        <bindingRedirect oldVersion="1.0.0.75" newVersion="1.1.0.104" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="MapInfo.CoreTypes"
publicKeyToken="f548bcba69d4b8da" />
        <bindingRedirect oldVersion="1.0.0.75" newVersion="1.1.0.104" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

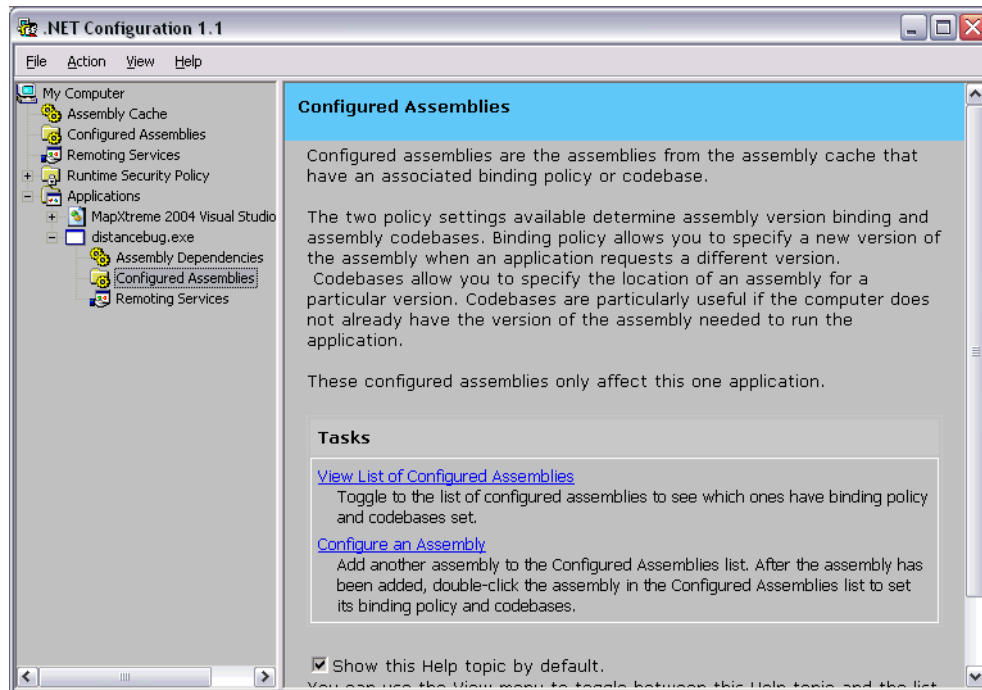
- From the Control Panel choose **ADMINISTRATIVE TOOLS**.
- Select Microsoft .NET Framework 1.1 Configuration. Click on **APPLICATIONS** in the left frame.



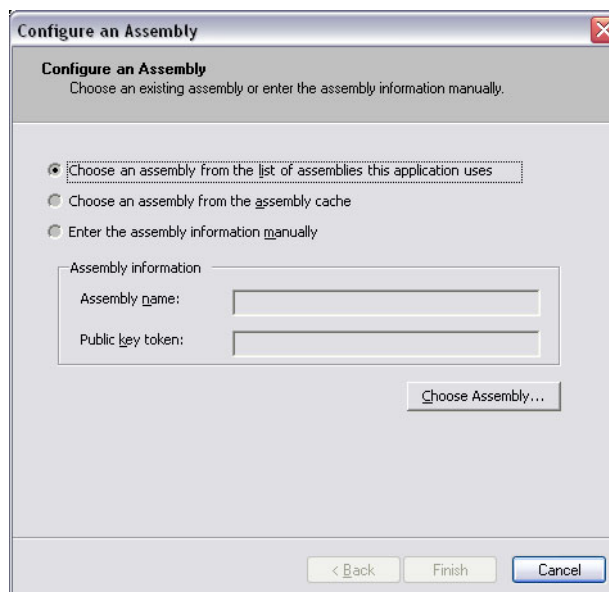
- Select **ADD AN APPLICATION TO CONFIGURE**.



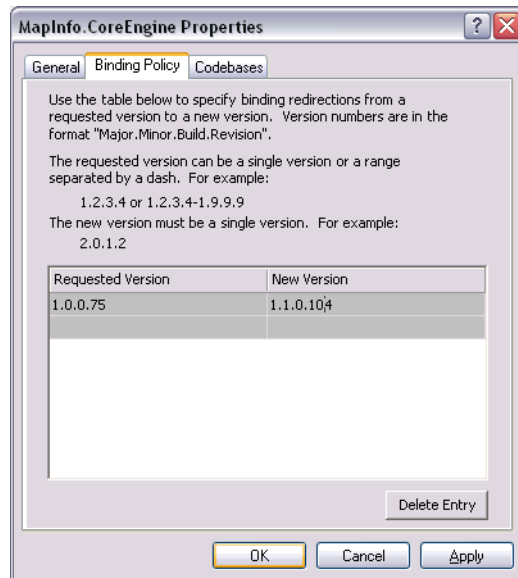
- Click **OTHER...** and browse to the executable name of your application. The entry will appear in the left frame under Applications. In the example below the application is named distancebug.exe. Expand by clicking on the + symbol. Click on **CONFIGURED ASSEMBLIES** in the left frame. You should see the following dialog:



- In the list of Tasks, select **CONFIGURE AND ASSEMBLY**. You can also right-click on Configured Assemblies then choose **ADD**.



7. Select the first radio button to choose an assembly from the list of assemblies, then click on **CHOOSE ASSEMBLY...** Choose an assembly (MapInfo.CoreEngine in this example) and then click **FINISH**. You should now see the following dialog. Choose the **BINDING POLICY** tab.



8. In the MapInfo.CoreEngine Properties dialog, specify the Requested Version and the New Version. The Requested Version is the version that application .exe actually references (such as 1.0.0.75). The New Version is the version to which you are redirecting the application. Specify a New Version of 1.1.0.104 to redirect the application to MapXtreme 2004 Release 6.1. Then click **OK**.
9. Similarly, configure MapInfo.CoreTypes and other assemblies that you want to redirect to the new version. That is, repeat [step 7](#) but this time select MapInfo.CoreTypes (or other appropriate assembly) instead of MapInfo.CoreEngine.

Now if you browse to the location of your application, you should find a file called *myApp.exe.config* or *myApp.dll.config* (where *myApp* is the name of your application).

### 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. See [Application Configuration File on page 9](#) for the recommended way to redirect assembly versions.

### Machine Configuration File

You can use the Microsoft .NET Framework 1.1 Configuration tool located under Control Panel, Administrative Tools to configure an assembly's redirection settings. This tool will manipulate the machine configuration file which is located in:

WINDIR\Microsoft.NET\Framework\v1.1.4322\Config.

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

---

## What's New in MapXtreme 2004

---

Release 6.1 of MapXtreme 2004 includes the following enhancements.

- [Side-by-Side Installer](#)
- [Deploying Web Applications](#)
- [Configuring Custom Raster Handlers](#)
- [Customizing the Custom Symbol Directory](#)
- [Customizing Context Menus](#)
- [Keyboard Shortcut Programmability](#)

**Note:** The number at the beginning of some descriptions is the MapInfo incident tracking number.

### Side-by-Side Installer

The side-by-side installer enables you to install a new version of the MapXtreme SDK without affecting the existing shipping version. Both the original version and newly-installed SDK will function independently. To accomplish this, the side-by-side installer will create unique virtual directories, independent directories, and registry keys.

### Deploying Web Applications

The MapXtreme 2004 Developer Guide outlines the procedure for deploying an application with your own installer. In chapter 2 of the MapXtreme 2004 Developer Guide, see the topic named Deploying with Your Own Installer and the subtopic that outlines the specific steps. The final step refers to creating a virtual directory for your web application. This step is no longer necessary because the virtual directory is now automatically created.

MapInfo also provides a white paper titled: How to Create, Package, and Deploy a MapXtreme 2004 Application. This paper is available for download on the MapInfo web site at:

<http://www.mapinfo.com/support/documentation/manuals.cfm>

From the list of Publications, select MapInfo MapXtreme for Windows then select MapXtreme 2004 Deployment white paper.

### Configuring Custom Raster Handlers

You can configure your MapXtreme 2004 application to use a different raster handler than the one included in the distribution of MapXtreme 2004 or to support an entirely new raster type. You can also change the precedence in which raster handlers are used.

In the default installation of MapXtreme 2004, all raster handlers are placed in *<program files>\Common Files\MapInfo\MapXtreme\6.1\RasterGridHandlers*. This is also the location of the file *mirasteru.dll*. This is the recommended installation location of any other raster handlers that you use in your application. If you use the default location, no other configuration steps are necessary.

If you want to put your custom raster handler in a non-default location, then you must specify the location of your custom raster handle in an application config file for a desktop application, or in the web.config file for a web application. To do this, the developer should define a <Path> or a <SpecialPath> element under <ApplicationDataPaths>, then copy the selected raster handler to that folder.

For example, to configure a desktop application custom raster handler in the non-default MyAppData directory, you could use the following .config file.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <section name="MapInfo.CoreEngine"
      type="MapInfo.Engine.ConfigSectionHandler, MapInfo.CoreEngine,
      Version=1.1.0.104, Culture=neutral,
      PublicKeyToken=93e298a0f6b95eb1" />
  </configSections>
  <MapInfo.CoreEngine>
    <ApplicationDataPaths>
      <SpecialPath>
        <Personal>MyAppData</Personal>
      </SpecialPath>
      <Path>c:\MyAppData</Path>
    </ApplicationDataPaths>
  </MapInfo.CoreEngine>
</configuration>
```

In this example, the <Personal> tag is a special place defined in the user's My Documents. This would refer to a folder called MyAppData in My Documents. Within the <ApplicationDataPaths> tag, use *either* the <SpecialPath> or the <Path> tags. Using both would mean that the raster handler could be put in either "MyAppData" folder. While this would not be an error, it probably is not what you intend to accomplish. Since this is a non-default configuration, you must also place the mirasteru.dll in that same directory.

Use the <SpecialPath> syntax if the application data is stored in a location relative to a .NET Framework special system folder. For example, if your application data is stored in a directory named MyAppData located under the "My Documents" directory, then the entry in the configuration file could be:

```
<Personal>MyAppData</Personal>
```

where "Personal" is the value of the .NET Framework enumeration Environment.SpecialFolder that represents the "My Documents" directory.

This configuration method can also be used to change the preferred raster handler for file types that can be managed by more than one raster handler. For example, a JPEG file can be handled by either Halo or LEADTOOLS (both of which are bundled in MapXtreme 2004). Normally, Halo has priority because its \*.rhv file extension alphabetically precedes the LEADTOOLS \*.rhx extension. However, because MapXtreme 2004 v6.1 first looks in any <ApplicationDataPaths> defined in the configuration file, it will locate a specified raster handler *before* looking for a handler in the default [CommonFiles] folder. So, for example, you could copy a LEADTOOLS \*.RHX handler in the defined folder and configure MapXtreme 2004 to find and use that handler first.

Alternatively, the MapXtreme 2004 developer can rename file extensions in the [CommonFiles] folder so that the desired raster handler appears first in the alphabetic list. However, this will affect *all* applications developed with MapXtreme 2004 and may produce unintended side-effects. Also, if file extensions are changed in this manner, the renamed raster handlers will *not* be deleted if MapXtreme 2004 is uninstalled. For these reasons, you may want to use the configuration method to change the location and precedence of raster handlers.

## Customizing the Custom Symbol Directory

Custom symbols can be located in a CustSymb folder located in the same folder as your application, or they can be located elsewhere. For information about customizing file locations, see Location of Application Data Files in Appendix B of the MapXtreme 2004 Developer Guide.

If you wish to locate custom bitmap symbols in an alternative CustSymb folder, use a <Path> or a <SpecialPath> element in the application's configuration file. This is very similar to the strategy described for [Configuring Custom Raster Handlers on page 13](#). The following example shows the <SpecialPath> tag:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <section name="MapInfo.CoreEngine"
      type="MapInfo.Engine.ConfigSectionHandler,
      MapInfo.CoreEngine, Version=1.1.0.104, Culture=neutral,
      PublicKeyToken=93e298a0f6b95eb1" />
  </configSections>
  <MapInfo.CoreEngine>
    <ApplicationDataPaths>
      <SpecialPath>
        <LocalApplicationData>Your Corporation\Your
Application<\LocalApplicationData>
          <\SpecialPath>
            </ApplicationDataPaths>
          </MapInfo.CoreEngine>
        </SpecialPath>
      </ApplicationDataPaths>
    </MapInfo.CoreEngine>
  </configuration>
```

This example will cause MapXtreme 2004 to search for custom symbols in a folder like C:\Documents and Settings\username\Application Data\Your Corporation\Your Application\CustSymb. Note that the <SpecialPath> element should *not* include the CustSymb folder name. If it did, the path would be interpreted as ...\Your Application\CustMapXtreme 2004\CustSymb\CustSymb. The duplicated "CustSymb" would be incorrect.

Use the <SpecialPath> syntax if the application data is stored in a location relative to a .NET Framework special system folder. For example, if your application data is stored in a directory named MyAppData located under the "My Documents" directory, then the entry in the configuration file could be:

```
<Personal>MyAppData</Personal>
```

where "Personal" is the value of the .NET Framework enumeration Environment.SpecialFolder that represents the "My Documents" directory.

## Customizing Context Menus

The MapXtreme 2004 developer can create customized LayerControl context menu items that appear when the user right-clicks a node in the layer tree. To support this, the new ContextMenuTargetObject property returns the object that the user right-clicked.

A code sample has been provided that shows how to define a LayerControlEnhancer class that allows a user to add custom items to the LayerControl's context menu. This example can be found in the LayerControl sample application located in the \Samples\Features\LayerControl directory.

## Keyboard Shortcut Programmability

You can create keyboard shortcuts to access everything on the LayerControl toolbar. The PerformDown, PerformUp, and PerformRemove methods already provided programmatic access to the Down, Up, and Remove buttons.

The new AddMenuMnemonic property provides programmatic access to the key associated with displaying the Add menu. The syntax for the AddMenuMnemonic property is:

```
public System.Windows.Forms.Keys AddMenuMnemonic {get; set;}
```

## Adding Custom Buttons to Toolbars

MapXtreme 2004 now has a much simpler procedure for adding custom tool buttons to a toolbar. To support this, the ImageList property has been added to the MapToolBar class. This property is overridden to hide from the design-time property page. The ToolId property of the MapToolBarButton is now visible at design-time so that its value can be set for a custom tool.

To add a custom button to a toolbar, follow this simplified procedure:

1. Add a MapToolBar to your form.
2. In the Visual Studio property window, highlight the Buttons property and press the ... push button to invoke the collection editor dialog
3. Click the **ADD** button to add a new MapToolBarButton.
4. Set the button's ButtonType property to CustomTool (last item the dropdown list). Note that the button's will now appear blank, as there is no image associated with the custom button yet.
5. Set the button's ToolId property to the name of the custom tool you added to the MapControl's Tools collection. If the ToolId value does not match a tool in the collection, then an runtime exception will be thrown when the user clicks the button.
6. Click **OK** to close the button collection editor dialog.
7. In form's constructor, after the InitializeComponent call, add code to add a custom bitmap to the MapToolBar's ImageList. How the custom bitmap is associated with the application is up to the programmer. One option is to add an ImageList to the form at design-time, populate it with the custom images, then write code to transfer its images to the toolbar's ImageList at runtime. That code would look like this:

```
// Add custom tool button's bitmap to toolbar's mage list
foreach (Image image in this.imageList1.Images) {
    this.mapToolBar1.ImageList.Images.Add(image);
}
```

```
        // Associate the bitmap with the custom tool's button (last image
in the list)
        this.mapToolBarButtonBlueSelect.ImageIndex =
this.mapToolBar1.ImageList.Images.Count-1;
```

## Bug Fixes in MapXtreme 2004 Release 6.1

---

Bug Fixes for MapXtreme 2004 Release 6.1 covers issues in the following areas of the product:

- [Data Access](#)
- [Documentation](#)
- [Selection Class](#)
- [Legends](#)
- [Raster](#)
- [Japanese Legacy Text](#)
- [Installation](#)
- [Deployment](#)
- [Mapping](#)
- [Persistence](#)
- [SDK](#)
- [User Interface](#)
- [Web Applications](#)
- [Desktop Applications](#)
- [General](#)

**Note:** The number at the beginning of each description is the MapInfo incident tracking number.

### Data Access

#### TableInfoServer

31997 TableInfoServer no longer ignores the where condition when adding a table from Oracle.

#### Table.UpdateFeature()

32002 Table.UpdateFeature is now parsing statements correctly. The problem was caused by the underlying ResultSet table that marked the columns read only.

#### AddColumns

T11013 The performance when adding a column with a large number of rows to a TAB file using Table.AddColumns has been improved. It will now be quicker to add a column from a large MS Access, ADO.NET or RDB table to a native TAB file.

**OrderBy Clause in View Tables**

T11023 MapXtreme 2004 ignores the OrderBy clause in queries using TableInfoView(). It does not throw an exception. This works as designed in response to differing behavior in RDBMS databases (e.g., Oracle allows it, SQL Server does not).

**Error Opening Oracle View**

T11188 Opening an Oracle view in Workspace Manager would occasionally produce an error. This has been corrected.

**Catalog.Search Memory Leak**

T11696 After loading data, a Catalog.Search would eventually deplete memory. This memory leak has been corrected.

**MIScrollable Reader**

T11750 MIScrollable reader no longer throws an error when you try to use the current property.

**InsertFeature Method**

T11621 The InsertFeature Method now works properly.

**Table.InsertFeatures Method**

T11160 The Table.InsertFeatures method now correctly inserts features into a table.

**Table.UpdateFeature**

32002 The Table.UpdateFeature method now correctly updates a row in the table.

**Spatialware Loader Performance Enhancement**

T11861 Loading performance of tables from SpatialWare 4.81 into MapXtreme has been improved.

**FeatureProcessor Class**

T12279 Returning all records using SearchALL would produce an error when trying to buffer the resulting IResultFeatureCollection. This has been corrected.

**Memory Management**

T11696, T12007, T11645 Memory management has been improved so that several methods no longer leak memory.

**Joining Access and MapInfo Tables**

T11323 Under some circumstances, joining an Access table with the MapXtreme 2004 USA table would not produce the correct map theme. This has been corrected.

**TableInfoView**

T12076 Some SQL queries would produce inaccurate results when issued through TableInfoView. This has been corrected.

**Find Error with Sample Application**

30578 When Find sample application was used in non-indexed tables, an exception was thrown. A proper error message is now generated.

**Info Tool Control**

T12451 After adding a column with AddColumns, the Info Tool now works properly.

**Documentation****GridRead Constructor Sample Code**

T13262 The GridRead Class example in the online help has been corrected to include the Session.Current call.

**LegendControl Sample Code**

T13242 The LegendControl sample code in the online help has been corrected.

**TableServerInfo**

31857 Added note to the description of TableInfoServer.ConnectionString Property to indicate that "DLG=SQL\_DRIVER\_NOPROMPT" must be specified in connection string for ODBC.

**TableInfoServer Creates CacheParameter Class**

30173 The TableInfoServer class automatically creates a CacheParameter class and sets its type to On. This was not clearly documented previously.

**SpatialSchemaPointRef Sample Code**

32001 The SpatialSchemaPointRef sample code in the online help has been corrected.

**TableInfoNative Class Sample Code**

T11830 The TableInfoNative Class sample code in the online help has been corrected so that it now creates a .TAB file.

**Order By Clause in Query**

T11023 For a TableInfoView query, the Order By clause is ignored and no error message is generated. This is the expected behavior and is now documented properly.

**TableInfoView**

T10899 When using TableInfoView to create a new layer based on a subset of features, the new layer has the same MBR as the base layer. This is by design and is now documented accordingly.

## Selection Class Sample Code

T12120 The Selection Class sample code in the online help has been corrected.

## AllowOverlap and AllowDuplicates

T12473 Duplicates and overlapping between labels is allowed if the label being processed allows it. This is a change from the previous behavior in MapXtreme. This new behavior is consistent between MapXtreme and MapInfo Professional.

## TableInfoAscii Maximum Characters

30955 In ASCII tables, a maximum of 253 characters are displayed. Additional characters are truncated. This is as designed, and is now documented for TableInfoAscii.

## MapExport Class

T12229 The MapExport sample code in the online help has been corrected to indicate that you must clone the map to export it when using the MapControl.

## FeatureProcessor Class

T11923 The sample code for the FeatureProcessor Class has been updated to better illustrate how to use the Combine method.

## Selection Class

### IFeatureCollections Cannot Be Added to a Selection

A Selection is a MultiResultSetFeatureCollection and it only supports adding IResultSetFeatureCollections. It does not support adding other types of IFeatureCollections.

As a result, the error message that displays when attempting to add an IFeatureCollection to the Selection has been corrected to say, "Value must be of type MapInfo.Data.IResultSetFeatureCollection".

### Selection.Add Methods

31995 The Selection.Add(IFeatureCollection) method has been fixed to allow adding an IResultSetFeatureCollection as an IFeatureCollection (its base interface) to a Selection. The following example uses the Selection.Add method to select a feature in an IResultSetFeatureCollection:

```
Dim pFeatureCollection As MapInfo.Data.IFeatureCollection
Dim pTable as MapInfo.Data.Table
Dim pRdr MIDataReader
Dim pFtr As Feature

' Load a map using a table and retrieve the table:
MapControl1.Map.Load(New MapInfo.Mapping.MapTableLoader("C:\Program
Files\MapInfo\MapXtreme\6.0\Samples\Data\USA.TAB"))
pTable = MapInfo.Engine.Session.Current.Catalog.GetTable("USA")

' Get its first feature:
pRdr = pTable.ExecuteReader()
```

```
pRdr.Read()
pFtr = pRdr.Current
pRdr.Close()

' Create a resultset feature collection:
pFeatureCollection =
MapInfo.Data.FeatureCollectionFactory.CreateResultSetFeatureCollection
(pTable, pTable.TableInfo.Columns)

' Add the feature to the collection:
pFeatureCollection.Add(pFtr)

' Add the feature collection to the Selection to select the feature:
MapInfo.Engine.Session.Current.Selections.DefaultSelection.Add
(pFeatureCollection)
```

## Legends

### Legend Border

T11038 The border now correctly draws around the legend, regardless of its vertical or horizontal orientation.

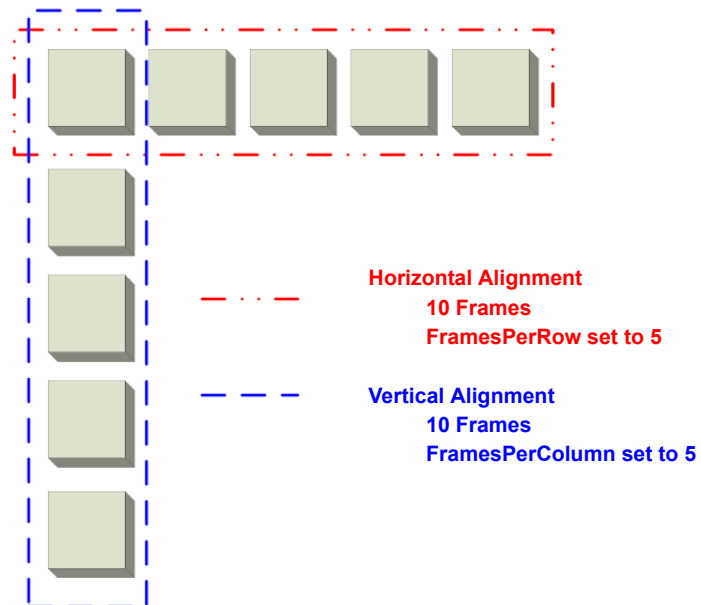
### Legend Frame Alignment Property Changes

T11036 The default values for the LegendFormat.FramesPerRow and LegendFormat.FramesPerColumn properties have been changed from 1 to 0, to be consistent with the behavior of the FramesAlignment property. The FramesPerRow and FramesPerColumn are used in conjunction with FrameAlignment in which you can align frames either horizontally or vertically).

For example, assume that you want to horizontally align a legend containing 10 frames. A horizontal FrameAlignment means the FramesPerRow value will be used. If the FramesPerRow value is 5, the result will be two rows that each contain 5 frames.

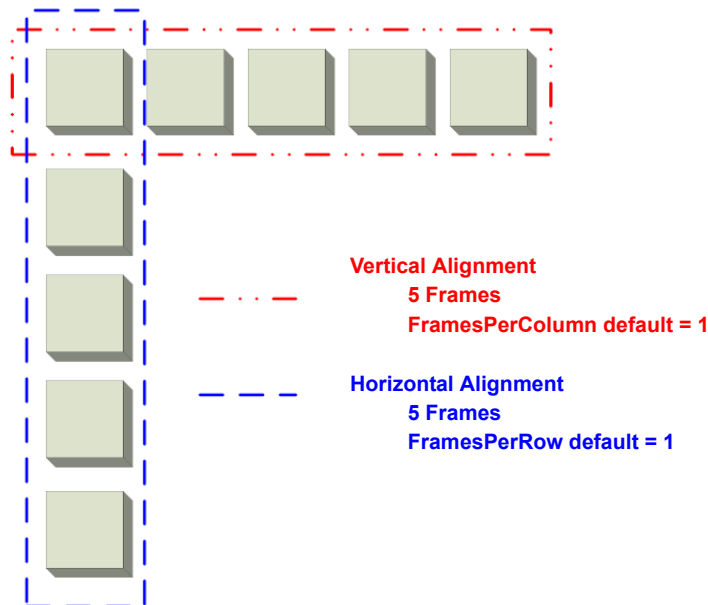
Similarly, to vertically align 10 frames in a legend, setting the FramesPerColumn to 5 will produce 5 rows of 2 frames each (5 frames per column).

The following illustration shows how vertical and horizontal alignments of 10 frames would look.



In Build 75 that used a default value of 1, while it worked as designed, caused the alignment to seem opposite of what it should be. A FramesPerRow default of 1 would look more like a vertical alignment when more than one frame was in the legend (every frame in its own row). See illustration below.

Similarly, a FramesPerColumn of 1 for multiple frames would look like a horizontal alignment (every frame in its own column). With the new default value of zero, only the FrameAlignment value is considered, regardless of the number of frames in the legend.



## Raster

T10973 The Halo JPEG raster handler is now thread safe.

## Japanese Legacy Text

T10922 Japanese LegacyText objects now display non-default fonts correctly.

T11144 Japanese font name is now honored when creating LegacyText objects.

## Installation

T12914 The XML schema for WMS/WFS is now included on MapXtreme 2004 CD and properly installed.

## Deployment

### Build 75 Runtime

If you are upgrading to MapXtreme 2004 Release 6.1, you can install this upgrade side-by-side with Release 6.0, Build 75.

### Packaging With Your Own Installer

Release 6.1 fixes several issues related to using your own application installer with MapXtreme 2004, as follows:

T10905 If you are packaging an application using your own installer, and the application uses COM+ pooling, you must include in it the MapInfo.Web.MSM as well as MapInfo.CoreEngine.MSM. This will register the CoreEngine as a COM+ object.

T10903 Application installers created using merge modules (MSM) now install the Microsoft C Runtime DLLs in the proper place (System32 folder).

T10907 Application installers created to use MSMs now properly create the virtual directory (VDir) MapXtremeWebResources6\_1.

### Deployment with LayerControl

T12671 Deploying a MapXtreme 2004 application with Layer Control now works properly. Previously you may have encountered an unhandled exception.

## Mapping

### Geoset Import

T12443 Geoset import now works properly. This is related to the AllowOverlap and AllowDuplicates behavior as described in Incident T12473.

### Adornments in Workspace Manager

T11702 In some circumstances, adornments were not displayed when opening a Workspace, although the adornments would appear when opening the .TAB file. This has been corrected and adornments will appear, as expected, when opening a workspace.

### ScaleBarAdornment

T11703 The ScaleBarAdornment value is now correctly changed when the Map zoom changed in the Workspace Manager.

### Modified Label Behavior

T11819 A modified label (dragged with the Select tool) now appears properly when Autolabel is enabled on the visible range.

### WMF Export

T12122 An image exported as WMF is now exported in the correct file format.

### **Opening Geosets through Layer Control**

31692 Under some circumstances, opening geosets through Layer Control could cause a crash. This has been corrected.

### **SeachWithinScreenRadius and SearchWithinScreenRect**

T13037 SearchWithinScreenRadius and SearchWithinScreenRect now update correctly when ResultSetCombineMode.Replace is used.

### **Alignment of Labels**

T11590 Default alignment for lines is now Alignment.TopCenter. Default alignment for points is now Alignment.CenterRight. This is consistent with behavior in MapInfo Professional.

### **Callout Lines on Labels**

T11933 Callout lines for labels were still being displayed even when they should have been hidden. This has been corrected so that callout lines are no longer visible when CalloutLine.Use = False.

### **Geoset Zoom Speed**

T10985 Geosets opened in Workspace Manager much faster.

## **Persistence**

### **Workspace Map Units**

T12097, T12043 MapXtreme 2004 can now load a workspace file with map units are set to "survey ft" and "sft".

### **View Tables Saved in Workspace**

T11833 A View table is now properly saved as a Native .tab file when saved in workspace.

### **Label Position**

T12654 Workspace manager will now properly save the label position when Above or Below offsets are used.

### **LabelLayer Persistence**

T11350 LabelLayer visibility is now properly persisted in Workspace Manager.

### **Label Font Persistence**

T12086 Edited fonts are now properly persisted in Workspace Manager.

### **Legend.Format.FramesPerRow**

T11274 Legend.Format.FramesPerRow is now properly saved when FrameAlignment is set to Horizontal.

**Callout Lines on Labels**

31907 Customized callout lines on labels are now being persisted in Workspace Manager.

**GML Attributes**

T12891 GML attributes are now written out according to W3C specifications.

**SDK****Desktop License and Pooling Web Application**

T12909 Previously, a pooling web application could run with a desktop license and without a watermark at runtime. This has been corrected.

**Web License Problem**

T12911 If pooling is enabled in a web application, and if there is only web runtime license, the web application would not run and would throw an exception. This has been corrected.

**Workspace Manager Crash**

T13132 Under some circumstances, Workspace Manager would crash with `ArgumentNullException` when closing all tables. This has been corrected.

**Bar Theme**

T12719 Creating a bar theme and then stacking the bar theme would cause a crash. This has been corrected.

**Add Projections**

T11546 For WFS, a server exception would occur when attempting to load a table that contained a new projection that was added in MapInfo Professional 7.8. This has been fixed.

**Custom Menu Items**

T13021 Adding a custom menu items that causes an item to be added to the layer tree may cause a temporary disturbance of the LayerControl's context menu. This has been corrected.

**User Interface****Info Tool Icon**

31693 When selected, the Info Tool icon now clearly shows that it is active.

**Modifying Theme Styles via Layer Control**

31969 Under some circumstances, modifying theme styles via Layer Control would cause an unhandled exception. This has been corrected.

### **FeatureStyles Sample Application**

31891 In the FeatureStyles sample application, if you shrink the main form so that the MapControl is no longer visible, a System.ArgumentException would occur. This has been corrected.

### **ModifyRangedThemeDlg**

T11279 When creating a new RangedLabelTheme using the Add Theme wizard, you can now adjust the number of ranges successfully.

### **Workspace Manager**

T11704 After opening a workspace that includes adornments, the sequence File > New > No (so as not to save current map) would produce an exception. This has been corrected.

### **Theme Wizard**

31935 After launching the Theme wizard via the Map menu, it was possible to generate and unhandled exception when adding Ranged Label Theme. This has been corrected.

### **Expression Dialog**

T11609, 31864 The Expression Dialog of the Theme Wizard now closes properly.

### **Removing Label Layer**

31926 After creating a label theme via the Theme wizard, removing label layer from map would produce a MapException error. This has been corrected.

### **Label Expressions**

29997 Some proper label expressions were interpreted as invalid. These work correctly now.

### **AddAdornments**

31643 Previously, if you added an adornment to a map that had an existing adornment, the new adornment was added on the bottom instead of on top. This resulted in situations such as a scalebar hidden by a title. This problem has been corrected.

### **Dragging/Moving Adornments**

31641 You can now drag and move Legend, Title, and ScaleBar adornments on the map.

## **Web Applications**

### **SelectionChanged Handler**

T11648 The SelectionChanged handler for the PolygonSelectionTool, RadiusSelectionTool and RectangleSelectionTool now works correctly in web applications.

### **State Sample Application**

31587 The State Sample application accessed from web now works properly.

### **ASP.NET Application**

31273 VB ASP.NET applications now work properly on a postback operation to the server.

### **Response.Redirect and Server.Transfer**

T13463 Under some circumstances, Response.Redirect and Server.Transfer would throw an unhandled exception. This has been corrected. This solution involves a new Session.Dispose method to properly dispose of the ISession instance and a new web.config entry to mandate context-specific storage of the ISession instance. For a more detailed explanation of how to manage the MapInfo.Engine.ISession instance, see [Managing MapInfo Session State on page 3](#).

### **Memory Leak with Pooled Applications**

T11483 Legends were not freeing up memory after they were disposed. This has been corrected.

## **Desktop Applications**

### **Add Tools**

31799 Adding Tools now adds tools at the proper location and proper zoom level.

## **General**

### **License Failure**

A10015 The license failure when installing the Danish localized version of MapXtreme 2004 has been corrected.

### **Installer Failure**

T10905 Applications that use the COM+ to access MapXtreme will fail, if developers use merge modules (MSMs) to distribute MapXtreme 2004 assemblies.

### **AddWebForm Wizard**

T13025 The path to the MapXtreme AddWebForm wizard has been corrected. The wizard now starts up properly.

### **Geoset Selectable Properties**

T13249 Changes to Selectable and Editable properties made in MapXtreme Geoset Manager are now enforced.

### **Workspace Manager TitleAdornment**

T11713 TitleAdornment is now properly displayed when file is loaded from Workspace Manager.

### **Distance Method on Geometry Object**

T12588 The Distance method now works properly when the geometries of the two objects are in different coordinate systems.

### **LoadMapWizard**

31778 The LoadMapWizard return value is now correct. based on whether all tables were opened successfully.

### **Japanese Runtime**

T11536 Workspace Manager now runs properly in the Japanese runtime.

### **Zoom Level**

T11076 The minimum and maximum zoom levels are now consistent with Geoset Manager and Workspace Manager.

### **Search Results**

T13344 When a search retrieves multiple matches, these matches are now displayed properly in the list box.

---

## **Known Issues**

---

The issues that follow were brought forward as Known issues from the Release Notes for Release 6.1. They are either currently under investigation or in active development seeking a resolution.

- [Data Access](#)
- [Desktop Applications](#)
- [Themes](#)
- [Web Applications](#)
- [Workspace Manager](#)
- [Installation](#)
- [Tools](#)
- [General](#)

### **Data Access**

- 24759 Adding an Oracle layer with very small coordinate bounds causes an error.
- 30165 Setting a New MIPParameter to an invalid dbtype should throw an NotSupportedException, but does not.
- 30375 The user is not prompted to enter a password when opening a table that was downloaded from an Oracle database without the password included in the TAB file. To avoid this, add the password to the tab file.

- 30503 You cannot add an Access table through TableInfoServer. Directly opening a \*.mdb file without the column information filled in a TableInfo is not supported. To avoid this, open Access tables with a tab file or include the column information in the TableInfo object.
- 30839 GROUP BY on special characters like Å and Ü do not return the correct results.
- 31278 Loading a linked ODBC table without password information in the tab file into the MapControl causes the web page to time out. To avoid this, add the password to tab file.
- 31529 When the MapTableLoader fails to load a DBMS table, it throws a "TableException" with incomplete information in the message. The name of the table is not included in the message.
- Web Controls do not support Frames.

## Desktop Applications

- 31268 Copying and pasting the desktop MapControl on a windows form leaves some of the properties blank. The following properties are left blank: LeftButtonTool, MiddleButtonTool, and RightButtonTool. To fix this, manually reset the properties of the control in the properties window.
- 30800 Style dialog boxes launch slowly the first time they are launched. Subsequent calls to open a particular dialog box launch faster.
- 30365 Tools with rubber-band outlines can leave 'droppings', or remnants of the rubber-band image, on the map if modifier keys or the mouse wheel are used. To avoid this, turn off mouse wheel support when any tool is being used.
- 29444 When using the Polygon Select tool or custom polygon/polyline tools, the rubber-band image does not always completely disappear when drawing is complete. If redraw speed is not an issue, call the Map.Invalidate() method after the ToolUsed event.
- 30310 Mouse wheel panning causes badly drawn maps if Pan or other tools are being used. If you disable the mouse wheel in your application, you can avoid this problem. You can also write your program so that the map is redrawn periodically.
- 31955 The GridInfoForm sample application only works when a user has a grid layer as the top layer in LayerControl. To make the application work, move the grid layer to the top layer in the LayerControl.

## Themes

- 31915 The CreateThemeWizard allows the user to choose a Label theme when no label sources are present. By doing so, you will receive a cryptic error: "Could not bind to the new display member. Parameter name: newDisplayMember."
- 31555 Colors of pie slices in pie themes will change while zooming if a hatch pattern is used.
- 31234 Pie themes show pies for rows that have no features. Pie themes show up for every row in the table, even rows without features.
- 30036 If you create a Bar chart that contains negative numbers, the bar chart does not stay constrained by the bounding box of the chart.
- 30175 The Legend border is not updated correctly after adding the range back to the legend in the Modify Ranged Theme Legend tab of the Workspace Manager.
- 30977 Individual Value (Label) and Ranged (Label) compact legends do not have a symbol, just text.
- 31479 The Ranges Natural Break method does not generate the same results in MapXtreme 2004 as in MapInfo Professional.

- 31463 Bar and Pie chart themes change Z-ordering randomly when rendered.

## Web Applications

- 31177 Themes still appear in the MapControl if the table used to generate the theme is set to not visible in the LayerControl, LegendControl, or MapControl.
- 31502 Group Layers and Label Group Layers do not have the option to display within zoom range or scale range when set from the LayerControl.
- 31510 Line direction, show nodes, and centroids are not displayed in a web application for a Layer when set at design time using the LayerControl.
- 31263 Copying and pasting a MapControl in a windows form leaves some properties blank. This causes the MapAlias and WorkspaceFile property values to remain the same, forcing changes on one map to inadvertently show up on another.
- 31354 Images that represent layers using the web LayerControl may not appear the same as the desktop LayerControl.
- 31427 The web LayerControl does not preserve its state.
- 31700 The Workspace is persisted incorrectly after adding a table without a map.
- 31893 The Style Override of a Raster layer is not accessible via the web LayerControl.
- 31725 You cannot load a named connection file at design time for web applications.
- 31743 The default VB ASP.NET template throws an error when no maps are on the page. The error is "Microsoft JScript runtime error: 'undefined' is null or not an object."
- 31854 When you set an invalid MapID to a web tool, it incorrectly throws a "NullReferenceException"; it should throw an "ArgumentException".
- 31845 Under certain conditions, an ASP.NET application renders a blank map and appears to have problems tracking its own state.
- 31933 Web Sample Applications have a hard coded path in their MWS files, which do not work if MapXtreme 2004 is not installed to the default directory. To work around this limitation, clear the Map from the MapControl and then reload the tables and/or workspaces that are associated with the project to the same location as your sample data. The following sample applications use the following data files: ThematicsWeb - World.tab; Tools - World.mws; FindWebSample - World.mws.
- 31496 Removing a group layer in a LayerControl (Web or Desktop) provides no warning or error message at the time it is removed.
- 31654 The web RadiusSelect Tool is not drawing correctly. The visible transparent circle that should be visible when drawing with this tool is chopped off when selected.
- 31645 Turning off the visibility of a group label layer parent node does not automatically turn off the visibility of the child nodes. These properties are set in the LayerControl.
- 31925 Marquee Zoom does not appear if the display is set to 16 colors (the default for Windows 2000).
- 31929, 31881 If the ID for the MapControl is changed with the map loaded in the control, the old MWS file is left on the hard drive. Also, when a MapControl is deleted from a webform that has tables loaded it also does not remove the MWS file. This old MWS file can be deleted manually.

## Workspace Manager

- 30802 There is no way to create a transparency for a Raster Layer using the Workspace Manager. You can create transparency programmatically or save this information to the tab file.
- 30164 When using the Workspace Manager, the use of Shift and Ctrl keys are reversed. When selecting multiple objects hold down the Ctrl key. When cancelling the selecting of individual objects, hold down the Shift key. This behavior is opposite from that in MapInfo Professional.
- 31183 When printing from the Workspace Manager, style overrides for raster layers do not display in the output. For example, a raster with a translucency style override does not print with any transparency.
- 31673 Workspace Manager does not display an error when opening a remote OCI or IUS table on machines without the OCI or IUS clients installed. When this action is performed, nothing happens at all. In a web application, opening a layer from a remote OCI or IUS source from a source without those clients installed at design time results in an empty map at runtime with no error indicated.

## Installation

- After the removal of MapXtreme 2004, there are MapInfo assemblies left in the GAC (Global Assembly Cache). This can happen for two reasons:  
The aspnet\_wp.exe process still has an open reference to the assemblies which the un-install does not detect. To fix this:

- a. Stop the web services using Control Panel->Administrative Tools->Internet Information Services. Select your web sever and stop the service.
- b. Go to the "Removing the assemblies" section below.

The installer services corrupts the registry. To determine if this is the case:

- a. Open the Registry Editor. (Start->Run and enter "regedit".)
- b. Navigate to HKEY\_CLASSES\_ROOT\Installer\Assemblies\Global.
- c. Check the (Default) key's type and see if it is REG\_MULTI\_SZ.
- d. If the value is REG\_MULTI\_SZ, select the (Default) key, and then choose Edit->Delete.

The key should re-appear with the type REG\_SZ.

- e. Go to the "Removing the assemblies" section below.

Removing the assemblies from the GAC:

- a. From the **Start->Run** menu, type **assembly**.
- b. Select any MapInfo.\* assemblies and delete them.
- c. If you receive an error, check the AddRemove programs control panel for any MapXtreme 2004 references.

- When the installation process is running a command window appears which is used to register the MapInfo.CoreEngine with the COM+ services. There are three errors displayed (the hex number returned will be different on your system):  
Unhandled Exception: System.DllNotFoundException: Exception from HRESULT: 0x80131524. at \_\_DllMainCRTStartup@12(Void\* , UInt32 , Void\* )  
Unhandled Exception: System.DllNotFoundException: Exception from HRESULT: 0x80131524. at \_\_DllMainCRTStartup@12(Void\* , UInt32 , Void\* )  
Unhandled Exception: System.DllNotFoundException: Exception from HRESULT: 0x80131524. at \_\_DllMainCRTStartup@12(Void\* , UInt32 , Void\* )  
These errors can be safely ignored.
- In the Sample Data provided with MapXtreme 2004, the following tables' coordsys entries in the table metadata is incorrect:
  - US\_CNTY: Coordsys metadata refers to Long/Lat (no datum), but map file is Long/Lat (NAD 27 for Continental US).
  - Mexico: Coordsys metadata refers to Conformal Projection (North America), but map file is Long/Lat (NAD 27 for Central America).
  - USA: Coordsys metadata refers to Conformal Projection (North America), but map file is Long/Lat (WGS 84).
- 29781 When running the .NET Services Installation tool (regsvcs.exe), the Installer throws a "system dll not found" exception error while registering the MapInfo Engine in COM+. This is caused by the unloading and loading of mixed dll's. A fix for this issue is being pursued with Microsoft.
- 31883 The serial number in the registry is deleted after an installation repair is performed. To avoid this, replace the registry entry HKEY\_LOCAL\_MACHINE\Software\MapInfo\MapXtreme\6.0\SerialNumber key with your serial number.
- 31701 When installing as one user and then logging in as another user, the MapXtreme Toolbox is not available in Visual Studio.
- 31832 If you modify your MapXtreme 2004 installation by removing the tools, removing the program subsequent to the modification will throw an error.

## Tools

- 31855 Coordinates returned by tool events are offset a significant distance depending on the zoom level. Tool events currently return approximate values for mouse click points; even if Snap mode is on, the returned coordinate may be a significant distance from the selected point. The problem is that the event handlers are automatically "snapping" all coordinates to their integer pixel equivalents. This means that errors are larger at higher zoom levels -- and also at lower map widths, even at the same zoom level. Stock Select tools and Add tools do work correctly with Snap -- but their ToolUsed events report that the user is clicking on locations up to five miles away from the actual location. To work around this, derive customized tools from stock Select and Add tools rather than from Custom tools, and work with the actual objects that are selected or created rather than using the coordinates returned by ToolUsed.
- 24604 Stock tools AddPolyline and AddPolygon do not draw consistently. There is no way to predict the Z-order of drawn objects when using these tools. A workaround to this problem is to add each feature as its own layer. This ensures the ordering as desired.
- 30205 When a Label is selected (by holding down the Select Tool), the rotate handle is not functional.
- 30813 The InfoTipsEnabled property of the MapTool class does not respect the per-tool override. Setting this property on a particular object does not limit the setting to just that object; it takes effect on a global basis.

- 31656 Selections are being inverted when using the Point Selection Tool. (This occurs because selections cancel each other out in cases where regions overlap, e.g., a selectable ocean layer.)
- 31405 Tools with state (e.g., EditText, AddPolyline, etc.) are not cleaned up when a MapControl loses focus. To fix this situation, write your program so that the map is redrawn periodically.
- 31567 ZoomIn and ZoomOut tools zoom increments are not the same as MapX. Note that this behavior can be overridden and customized for any desired behavior.
- 31569 When using the LayerControl, the zoom tools increment the Map name. For instance if you have a map named Map1, with each zoom, the name of the map changes to Map2, Map3, etc.
- 31799 When adding objects to a map with the Add tools, the new objects do not appear where they should. Instead, they appear near the 0,0 point. Once a view change occurs on the map, this behavior stops. To fix this problem, set `Map.Zoom = Map.Zoom` in the ToolBar's Click event after activating any Add tool.
- 31917 When the Dynamic Selection Tools check box is selected, objects are displayed as selected on the map even after the selection is cancelled.
- 31945 Snap mode does not work when the AddText tool is active.
- 31953 When zooming in to extreme levels, sometimes the zoom level stays at 0 when attempting to zoom out using the ZoomOut tool. To escape this situation, change your view using something such as the View Entire Layer control in the LayerControl.

## General

- 31889 The first Zoom modifier loaded from a geoset always has units of miles, no matter what the units should be.
- MapExport fails in multi-thread mode when using LeadTool's JPEG 2000 and TIFF. JPEG 2000 fails on Windows 2000. TIFF CMYK fails on all platforms.
- The WMS Server exports GIF files that cannot be read by MapInfo Professional. The WMS server in MapXtreme 2004 generates GIF images that are valid but causes a failure in MapInfo Professional v7.5. To work around this, use any of the other listed image formats returned by the server.
- 25344 Writing of shape (\*.shp) files is unsupported in this release.
- 25630 Do not convert a Point in a Non-Earth Coordinate system to MGRS. This conversion should throw an exception or do nothing. In this version that operation produces an unspecified result.
- 25655 When creating a Point type or setting a Point type's Data properties to extreme values in Longitude/Latitude, the map view does not clip the Longitude/Latitude values correctly. For example, if you create a Longitude/Latitude Point or set the Point's Data property to a non-existent value such as (999999999.0, 999999999.0), the values are clipped to 1000, 1000, instead of -360, -90, 360, 90 as designed.
- 30827 Style repositories are not type safe. If you pass an object that is not a style into a style repository, it will crash. This is true for any object that implements IList or ICollection.
- 27788 When setting the Angle property on a Font symbol style to a positive value, the symbol rotates counter-clockwise. Other objects in MapXtreme 2004 rotate in a clockwise direction for positive values of the Angle property.
- 29042 MapXtreme 2004 cannot draw null symbols for bitmap symbols, as it can for lines, fills, font symbols, etc. If you select "N" from Custom Symbols in the Symbol Style Control it will display an undesired symbol. To avoid this, use a different point style and make that one "null."

- 29092 Buffering complex MultiPolygon/Region objects with many line segments throws an error. Avoid this by thinning the data being buffered.
- 29250 Do not try to modify a MapControl at design time while the application is in a run state. A "System.InvalidOperationException" exception is thrown from the MapControl when this is attempted.
- 29391 MapXtreme applications cannot load persistence workspace files that contain a table having a SpatialSchemaXY assigned on the fly. To avoid this, save the spatial metadata into the tab file and then re-create the TableInfo object.
- 29824 The "Apply color" check box in the Symbol Style dialog box should not be available when a new extended custom symbol set is selected. This option should be unavailable, but is not.
- 30528 SelfIntersects does not work correctly for some cases with linear objects (MultiCurve, Curve, LineString) such as the figure 8 and the figure B.
- 30413 Labels are still displayed on a layer if the layer is not visible. To avoid this, turn off the Label Modifier visibility.
- 30423 In some cases you cannot create an ASCII file from a tab file, when running on a computer configured with a Japanese locale.
- 30456 PNG Transparency at 96dpi, 501px X 300px, 8bpp does not work.
- 30812 The Snap-to-node crosshair feature has redraw issues when toggling snap on or off. To avoid this, force a map redraw by panning or refreshing your map.
- 31199 The WMS Client allows for users to "get" a Map without specifying any bounds. You must specify the bounds as part of the GetMap request.
- 31792 Do not try to reload an already-loaded MWS file into a MapControl. This produces an error such as "Error loading File: xxx.mws" where xxx is the file name of the file being loaded.
- 31223 When you modify a table being used in a LabelSource object, these changes are not reflected in the LabelSource object.
- 31267 A LabelSource layer alias is not unique to a Map; it is unique to a Label layer. To work around this limitation, use both the Layer alias and the LabelSource alias to ensure uniqueness.
- 31372 The vertical scrollbars for the Symbol Style custom listbox are not working.
- 31342 A border appears on the top and left of an image while panning on the Web MapControl.
- 31471 Unable to read WBMP files.
- 31689 The LayerControl can be inconsistent on a cached remote table with mixed objects. Remote data sources do not always return features in a predictable order. To avoid this, call LayerControl.SetTableImage to specify an image explicitly.
- 31480 Color of style override is not applied when first setting the style override.
- 31481 GRC and GRD raster handlers are not thread safe.
- 31488 The WFS client does not understand GML2. The WFS client only understands MapInfo-GML3. Because of this, the client only works with MapInfo servers.
- 31811 WriteTabFile writes out invalid TAB file characters for column names when the current codepage is not equal to the data codepage.
- 31803 CoordSysFactory.LoadDefaultProjectionFile() throws a NullReferenceException if the default coordsys is not found.
- 31809 TableInfo.WriteTabFile() can write unusable TAB files. Under certain circumstances this method can overwrite an existing TAB file, making it invalid. Only use this method to create new TAB files rather than update or overwrite pre-existing ones.

- 31617 Edit handles for LegacyText objects do not function correctly.
- 31668 ScaleBarAdornment and ScaleBarAdornmentControl classes can report different sizes of the scale bar.
- 31899 There is a redraw problem when the InsertionLayerFilter is changed programatically after toggling a Dynamic Selection. The problem is that the edit handles of a selected region are not visible until the map is refreshed. To avoid this, do not enable Dynamic Selection if you are not using it in your application. Also you can limit LayerFilter changes to what end users can access through LayerControlDlg.
- 31671 If selected Labels are off screen when a map click occurs, the selection is not cancelled.
- 31726 GML publishing does not properly read or write FeatureCollections with multiple geometry columns.
- 31702 Opening multiple geosets through LoadMapWizard stacks new layers on the bottom, rather than on the top. To fix this problem set `Map.Zoom = Map.Zoom` after activating any Add tool.
- 31815 Map may incorrectly remember hidden selections after a zoom layering change.
- 31791 Mapping.SearchInfoFactory methods do not validate parameters.
- 31850 Closing a temporary table causes its labels to disappear from the display when the animation layer is active. It is important to note that the LabelsAreEditable property set to True is not compatible with animation layers. To fix this behavior call `Mapinfo.Mapping.Map.Invalidate()` often in your application.
- 31920 Cannot modify the font in, or resize, a title adornment after it has been displayed.
- 31836 An error occurs when loading a workspace if it contains an edited label for a raster layer.
- 31846 CursorImageURL URL type is not saved at design time.
- 31890 The anchoring and minimum size for dialog boxes and sample application forms are often wrong or nonexistent. This results in dialog boxes being resized to no apparent effect.
- 31898 Do not use unreasonably large values in scientific notation in dialog boxes. This can result in an error such as "Object reference not set to an instance of an object."
- 31936 Some FillStyle property settings in the FillStyle dialog box disable all the dialog box's pickers.
- 31958 The WFS Sample at runtime generates a System.URLFormatException. To fix this, change the URL constant in the SimpleSample Class in the simplesample.cs file to point to a valid WFS server.