

MapXtreme 2004 リリース 6.1 開発者ガイドサプリメント

2005年1月

内容 :

- ◆ はじめに..... 1
- ◆ [MapXtreme 2004 リリース 6.1 へのアップグレード](#)..... 2
- ◆ [MapInfo のセッション状態管理](#) 3
- ◆ [MapXtreme 2004 リリース 6.1 のバグフィックス](#)..... 17
- ◆ [既知の問題](#)..... 29

はじめに

このドキュメントの対象読者

このドキュメントは、『MapXtreme 2004 開発者ガイド』のサプリメントです。このサプリメントでは、アップグレード方法のみならず、新しい機能や、既存の『MapXtreme 2004 開発者ガイド』では取り上げていない重要な手法についても解説しています。したがって、MapXtreme 2004 のすべての管理者が必ずお読みください。

また、このドキュメントには、MapXtreme 2004 のすべてのリリースのリリースノートも取り込まれています。

- バージョン 6.0、2004 年 4 月 30 日 (ビルド 75)
- 限定リリース、2004 年 6 月 30 日 (ビルド 78)
- バージョン 6.1、2004 年 12 月 3 日 (ビルド 104)

リリース 6.1 に関する最新情報については、[17 ページの「MapXtreme 2004 リリース 6.1 のバグフィックス」](#)を参照してください。現在調査中または解決策を検討中の問題点については、[29 ページの「既知の問題」](#)を参照してください。

MapXtreme 2004 リリース 6.1 へのアップグレード

MapXtreme の試用版またはベータ版がインストールされている場合は、アンインストールしてください。MapXtreme 2004 リリース 6.1 にアップグレードするには、Setup.exe を実行し、画面の指示に従って操作します。リリース 6.1 は、リリース 6.0 のビルド 75 と共存してインストールできます。インストール済みのビルド番号が不明な場合は、[スタート]メニューの[ファイル名を指定して実行]をクリックし、「assembly」と入力します。

注意： 限定リリースであるビルド 78 をインストールしている場合は、リリース 6.1 をインストールする前にビルド 78 を必ずアンインストールしてください。

ビルド 75 のデスクトップアプリケーション

リリース 6.1 にアップグレードする場合、ビルド 75 で作成したデスクトップアプリケーションは再コンパイルできます。場合によっては、コントロールをフォームに追加し直すことが必要です。

また、再コンパイルを行わず、新しい MapInfo アセンブリを使用するようにアプリケーションをリダイレクトすることもできます。この手法では、参照先のアセンブリの新しいバージョンを検索するよう、アプリケーションに対して指定します。このアプリケーション構成手順の詳細については、9 ページの「アセンブリバージョンのリダイレクト」を参照してください。

ビルド 75 の Web アプリケーション

ビルド 75 で作成した Web アプリケーションについては、リリース 6.1 で開発を行うためには、プロジェクトのバージョン番号を、古い番号から 1.1.0.104 に変更する必要があります。プロジェクトのバージョン番号の変更方法については、8 ページの「Web アプリケーションの作成または再コンパイル」を参照してください。

ライセンスファイルの再配置

現行の MapXtreme 2004 (リリース 6.0) のユーザは、ライセンスファイルを、リリース 6.1 の当該フォルダ (C:\Program Files\MapInfo\MapXtreme\6.1) にコピーする必要があります。

Web サンプルアプリケーション

Web サンプルアプリケーションについては、適切な仮想ディレクトリを作成する必要があります。C# のサンプルアプリケーションについては、仮想ディレクトリはデフォルトで作成されません。C# アプリケーション用に作成された仮想ディレクトリを VB 用として使用するために、その参照先を VB アプリケーションの場所に変更することができます。

次に例を示します。デフォルトのインストールでは、C# のサンプルアプリケーション ThematicsWeb (場所は C:\Program Files\MapInfo\MapXtreme\6.1\Samples\Features\ThematicsWeb\cs) には、仮想ディレクトリがデフォルトでインストールされます。この仮想ディレクトリを Visual Basic アプリケーションの ThematicsWeb で使用するためには、仮想ディレクトリを変更する必要があります。

1. コントロールパネルから、インターネットインフォメーションサービスを開きます。デフォルトの Web サイトで ThematicsWeb フォルダを探します。
2. このフォルダを右クリックし、ローカルパスを C:\Program Files\MapInfo\MapXtreme\6.1\Samples\Features\ThematicsWeb\cs から C:\Program Files\MapInfo\MapXtreme\6.1\Samples\Features\ThematicsWeb\vb に変更します。

3. IIS を再起動します。
4. タスクマネージャを使用して、aspnet_wp.exe を停止および再起動します。Windows 2003 オペレーティングシステムの場合、aspnet プロセスは w3wp.exe です。

他の Visual Basic サンプルアプリケーションについても、基本的な手順は同じです。

サンプルデータのインストーラ

MapXtreme 2004 リリース 6.0 のサンプルデータをインストールしている場合は、リリース 6.1 の無償のサンプルデータインストーラで新しいサンプルデータをインストールする前に、古いサンプルデータをアンインストールしてください。アンインストールでは、ユーザが変更を加えたデータも削除されます。

注意： データのアンインストールでは、たとえユーザが変更を加えていたとしても、データは削除されます。サンプルデータに加えた変更を保持しておきたいユーザは、データを別の場所に待避してからアンインストールを行う必要があります。

リリース 6.1 のサンプルデータインストーラでは、古いサンプルデータが検出された場合はインストールは失敗します。

Windows XP Service Pack 2

Microsoft Windows XP Service Pack 2 (SP2) を導入している場合には、MapXtreme 2004 でいくつかの問題が生じることが判明しています。その多くは、SP2 で追加されたセキュリティ機能に関連する問題です。Windows XP SP2 のデフォルトでは、これまで他のコンピュータとの通信に使用されてきた数多くのポートが閉じられます。また、Windows ファイアウォールが有効になっている場合には、要求なしで一方的に受けた接続はブロックされます。これが原因で、MapXtreme 2004 で問題が生じます。

Microsoft のサポート技術情報には、Windows XP SP2 のインストール前と同様にプログラムが動作するように指定する方法についての解説があります。MapXtreme 2004 を「例外の一覧」に追加すれば、Windows XP SP2 で MapXtreme 2004 を使用する場合に生じ得る問題を回避できます。

次の Microsoft サポート技術情報を参照してください。

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

Windows XP Service Pack 2 (SP2) が SQL Server および MSDE に及ぼす影響に関してよくある質問と回答については、次を参照してください。

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

MapInfo のセッション状態管理

ISession インターフェイス

MapXtreme 2004 の MapInfo.Session 名前空間には ISession インターフェイスが含まれています。アプリケーションの中核となる情報の大半 (たとえばマップやテーブルなど) に対するアクセスを定義するインターフェイスです。また、ISession は、MapXtreme アプリケーションが必要とするリソースの初期化も管理します。

ASP.NET アプリケーションでは、`ISession` の状態は ASP.NET セッションごとにそれぞれ別個です。1 つのセッションは 1 つのクライアントに対応し、各クライアントはそれぞれ独自の状態を持つからです。クライアントがリクエストを行うたびに、セッションの最後の状態が取得され、実行が行われた後で、新しい状態が再びストレージに格納されます。

`ISession` インスタンスを自動的に準備するのは `WebSessionActivator` の役目です。これにより、静的プロパティ `Session.Current` を通して `ISession` にアクセスでき、処理中の ASP.NET セッションに対応する適切な状態が保持されます。`WebSessionActivator` は `IHttpModule` で、リクエストの有効期間内の処理を定める ASP.NET アプリケーション固有のイベントにアクセスできます。`WebSessionActivator` がこれらのイベントの間に行う処理を以下に示します。

- **BeginRequest**

`WebSessionActivator` は新しい `ISession` インスタンスを作成し (または、`web.config` ファイルで `MapInfo.Engine.Session.Pooled` が `true` に設定されている場合にはプールからインスタンスを取得し)、リクエスト固有のストレージに格納します。この結果、`Session.Current` プロパティを通して `ISession` にアクセスできるようになります。

`web.config` ファイルで、`sessionState` モード設定が `InProc`、`MapInfo.Engine.Session.State` 設定が `HttpSessionState` に指定されている場合には、このステップは `AcquireRequestState` まで先延ばしされます。`ISession` インスタンスはメモリに格納されており、メモリストレージから取得する必要があるからです。

- ...

- **AcquireRequestState**

`MapInfo.Engine.Session.State` 設定が `HttpSessionState`、`sessionState` モード設定が `StateServer` または `SQLServer` の場合には、状態のデシリアライゼーションは、リクエスト固有のストレージの現在の `ISession` インスタンスに対して行われます。

`sessionState` モード設定が `InProc` の場合には、`WebSessionActivator` は既存の `ISession` インスタンスをメモリ (ASP.NET セッション状態) から取得し、リクエスト固有のストレージに格納します。この結果、`Session.Current` プロパティを通してアクセス可能となります。

- ...

- **IHttpHandler の実行**

`Web` ページまたは `Web` サービス固有のコードが実行されるステップです。リクエスト固有のストレージの `ISession` インスタンスには、`Session.Current` プロパティを通してアクセスできます。

- ...

- **PostRequestHandlerExecute**

`MapInfo.Engine.Session.State` 設定が `HttpSessionState` の場合には、ここで `WebSessionActivator` が `ISession` インスタンスを ASP.NET セッション状態に格納します (`sessionState` モード設定に関係なく行われます)。

- ...

- **ReleaseSessionState**

ASP.NET エンジン、`sessionState` モード設定に応じて、ASP.NET セッション状態をメモリに保存するか、またはサーバストレージにシリアライズします。

- ...

- EndRequest

WebSessionActivator は ISession インスタンスを解放します (または、MapInfo.Engine.Session.Pooled 設定が true の場合には、プールに戻します)。

MapInfo.Engine.Session.State 設定が HttpSessionState、sessionState モード設定が InProc の場合には、ISession インスタンスはリクエスト固有のストレージから非アクティブ化されるのみで、解放はされません。同じセッションインスタンスの次のリクエストで再利用されるからです。

Session.Dispose(HttpSessionState) メソッドの使用

ここでは、Session_End から Session.Dispose (HttpSessionState) メソッドをいつどのように呼び出すかについて説明します。

MapXtreme 2004 の Web コントロールを使用している Web アプリケーションでは、web.config のデフォルトの設定は必ず次のようになります (web.config ファイルに下記の設定がない場合でも同様です)。

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

1 行目の設定は、MapInfo.Engine.ISession インスタンスの状態を保存および復元するためのメカニズムを制御する、アプリケーション固有の設定です。このインスタンスには、静的プロパティ MapInfo.Engine.Session.Current を通してアクセスできます。HttpSessionState という設定は、ISession インスタンスは ASP.NET セッション状態を通して保存および復元されるという意味です。このセッション状態は現在の HttpContext を通して公開され、HttpSessionState 型です。

2 行目の設定は、HttpSessionState がどのように保存および復元されるかを制御する、ASP.NET の設定です。デフォルトの InProc という設定は、ASP.NET セッション状態の内容はメモリ内に格納され、ASP.NET セッションインスタンスごとに固有であるという意味です。

これらの設定を使用した場合、各 ASP.NET セッションに対してそれぞれ固有の ISession インスタンスがあり、ASP.NET セッションの有効期間全体を通してこのインスタンスが HttpSessionState に格納されています。セッションがタイムアウトまたは終了したときに ISession インスタンスがきちんと解放されるようにするためには、Global.asax の Session_End メソッドに次のステートメントを追加する必要があります。

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

上記は C# の場合です。

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

上記は VB.NET の場合です。

この呼び出しを行うと、ISession インスタンスがきちんと解放され、メモリも確実に解放されるので、アプリケーションがメモリ不足にならずに済みます。

他の設定を使用している場合には、この呼び出しは必要ありません。他の設定では、ISession インスタンスの格納先はメモリではないからです。

MapInfo.Session.Engine の使用

ここでは、web.config の MapInfo.Session.Engine.UseCallContext 設定をいつどのように使用するかについて説明します。

Response.Redirect および Server.Transfer の使用時の Session.Current の問題

Web アプリケーションで Response.Redirect または Server.Transfer を使用する場合には、同一の ASP.NET セッションインスタンスに対する複数のリクエストを Web サーバが同時に受け取る可能性があります。これにより負荷が高まると、リクエスト実行の一部が別のスレッドで行われることがあります。たとえば、BeginRequest と IHttpHandler が別のスレッドで実行されることがあり、その場合には未処理例外が発生します。複数のフレームが導入されている場合には、このリスクはさらに高まります。

WebSessionActivator は ISession インスタンスをリクエスト固有のストレージに格納します。デフォルトでは、このストレージは、スレッド固有のストレージを使用して内部的に実装されています。Response.Redirect や Server.Transfer を使用する場合には、前述のような動作が原因で、以下の問題が生じる可能性があります。

- IHttpHandler が別スレッドで実行される場合、リクエスト固有の ISession インスタンスにアクセスできない可能性があります。その場合、次のアプリケーション例外がスローされます。

WebSessionActivator が登録されていないか IHttpHandler のコンストラクタにアクセスしようとしているため、セッションがアクティブではありません。WebSessionActivator を登録するには、次のエントリ : <httpModules><add type="MapInfo.Engine.WebSessionActivator, MapInfo.CoreEngine, Version=<version>, Culture=neutral, PublicKeyToken=93e298a0f6b95eb1" name="WebSessionActivator" /></httpModules> を web.config ファイルに追加します。<version> は現在のバージョンの MapInfo.CoreEngine.dll に置き換えます。

- 別のリクエストが実行されていたスレッドで WebSessionActivator が実行されていることが原因で、WebSessionActivator が ISession インスタンスをリクエスト固有のストレージに格納できません。

アクティブなセッションが既に存在します。

この問題を解決するには、ISession のストレージの種類をスレッド固有から呼び出しコンテキスト固有に切り替える方法があります。呼び出しコンテキスト固有のストレージを使用すれば、ASP.NET リクエストの各部分がどのスレッドで実行されるかに関係なく、リクエストの有効期間全体を通して同じ ISession インスタンスが使用されます。呼び出しコンテキスト固有のストレージの機能を有効にするには、アプリケーション固有の次の設定を web.config ファイルに追加します。

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

この設定を使用すると、前述の問題は発生しなくなります。

UseCallContext 設定はパフォーマンスに影響し、アプリケーションの実行が遅くなる場合があります。

Web サービスクライアントからの非同期呼び出しでの Session.Current の問題

Visual Studio .NET の Web サービスクライアントでは、Web サービスメソッドの非同期呼び出しがサポートされています。SoapHttpClientProtocol.BeginInvoke メソッドの使用方法に関する情報は、MSDN Web サイト (<http://www.microsoft.com/japan/msdn/default.asp>) で検索することによって入手できます。

サーバが ASP.NET Web サービス実装 (asmx) の場合に、クライアントから同一のセッションインスタンスに対して複数のメソッドを非同期で呼び出すと、次の問題が生じる可能性があります。

1. サーバサイドで ASP.NET セッション状態を特に有効にしている Web サービスメソッドの場合 (クライアントサイドできちんと動作させるためには、クッキーコンテナの設定が必要です)。
 - a. 同一のセッション状態を使用してリクエストが同期的に処理されるのではなく、一部のリクエストが別のセッション状態の下で処理されます。つまり、セッション状態が正しくなくなります。この問題は MapXtreme 2004 とは無関係に発生します。推奨される解決法はありません。
 - b. リクエストが同一のセッション状態を使用する場合であっても、リクエストの呼び出しコンテキスト外でデシリアライゼーションが行われることがあります。UseCallContext 設定が true の場合、これにより例外が発生します。セッションにアクセスできない (コンテキストが利用できない) からです。UseCallContext が false の場合、デシリアライゼーションは別の ISession インスタンスで行われ、他のリクエスト実行からはこのインスタンスにアクセスできません。これにより、状態が失われます。これら 2 つの問題には、推奨される回避法や解決法はなく、クライアントからの非同期呼び出しを避ける以外に方法はありません。
2. ASP.NET セッション状態を有効にしているかどうかにかかわらず、すべての Web サービスメソッドで起こり得るのは、リクエストの有効期間内の一部の処理が別スレッドで実行されるという状況です。この場合、前述の Response.Redirect/Server.Transfer と同様の問題が生じます。この場合は、UseCallContext を true に設定して問題を解決することをお勧めします。

Web 構成の変更

WMS/WFS サーバの更新

アップグレードの際には、新しい WMS サーバおよび WFS サーバを使用できるように Web 構成ファイルを更新する必要があります。

1. web.config ファイルを編集し、WMS サーバおよび WFS サーバに MapXtreme 2004 ビルド 104 を指定します。たとえば、現状がビルド 75 の場合には、web.config ファイルには「Version=1.0.0.75」と指定されています。これを、バージョン 1.1.0.104 を使用するように変更する必要があります。更新後の web.config ファイルは次のようになります。

```
<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. IIS を停止および再起動します。

8. タスク マネージャを使用して、aspnet_wp.exe を停止および再起動します。Windows 2003 オペレーティングシステムの場合、aspnet プロセスは w3wp.exe です。

アセンブリバージョンのリダイレクト

ここでは、再コンパイルを行わずに新しいバージョンの MapInfo アセンブリを使用するようにアプリケーションを構成する方法について説明します。アセンブリバージョンをリダイレクトするとは、参照先のアセンブリの新しいバージョンを検索するようにアプリケーションに指定するという意味です。Microsoft では、これを実現するためのメカニズムを何種類か用意しています。

詳細については、<http://www.microsoft.com/japan/msdn/default.asp> を開き、「アセンブリバージョンのリダイレクト」を検索して、適切なトピックを参照してください。

既存のアプリケーションの変更を伴わないアセンブリバージョンのリダイレクト

アセンブリバージョンのリダイレクトにはいくつかの方法があります。

アプリケーション構成ファイル

アセンブリバージョンのリダイレクトには、アプリケーション構成ファイルの使用をお勧めします。

アプリケーションのリダイレクトを構成するには、コントロールパネルの管理ツールの下にある Microsoft .NET Framework 1.1 Configuration ツールを使用できます。このツールを使用すると、アプリケーション用のアプリケーション構成ファイルの作成または編集を行うことができます。

アプリケーション構成ファイルの設定は、発行者のポリシーファイルの設定よりも優先されます。

MapXtreme 2004 リリース 6.1 をインストールした後、以下の手順に従って操作して、アセンブリバージョンをリダイレクトします。

1. アプリケーションをリダイレクトさせ、MapXtreme 6.1 の新しいアセンブリを読み込ませるように、アプリケーション構成ファイルを作成します。

注意： 構成ファイルは、アプリケーションと同じディレクトリに配置し、アプリケーション名に基づいた名前を付ける必要があります。たとえば、myApp.exe というアプリケーションの構成ファイル名は myApp.exe.config とする必要があります。Microsoft の構成ツールを使用すれば、このような名前が付けられます。

アプリケーションの .config ファイルの例を次に示します。新しいバージョンはビルド 104 です。

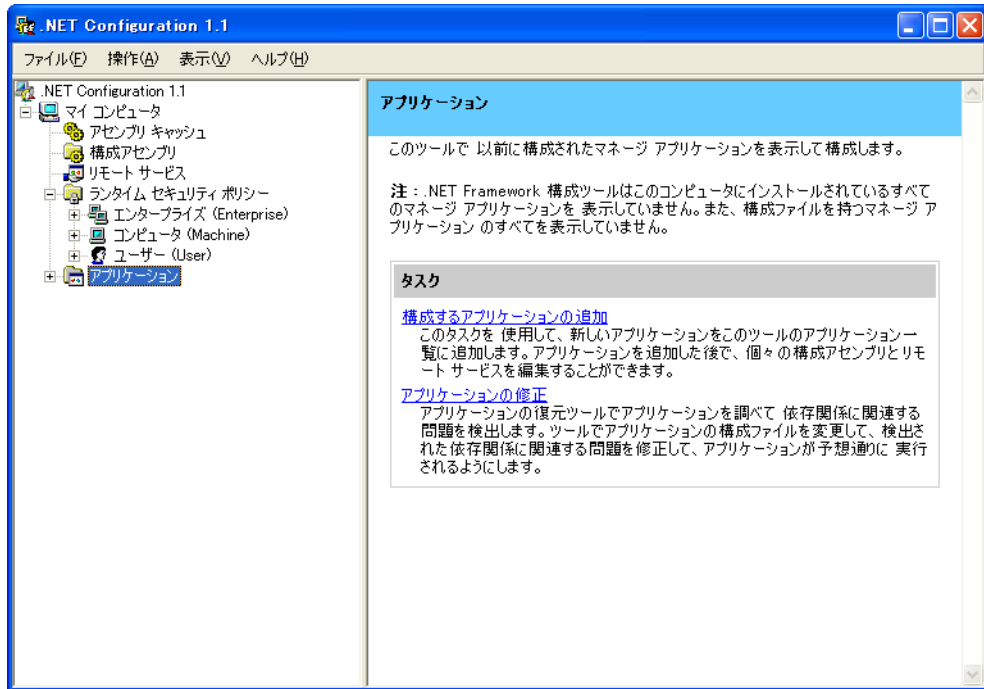
```
<?xml version="1.0"?>
<configuration>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="MapInfo.CoreEngine"
publicKeyToken="93e298a0f6b95eb1" />
        <bindingRedirect oldVersion="1.0.0.75" newVersion="1.1.0.104"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

```

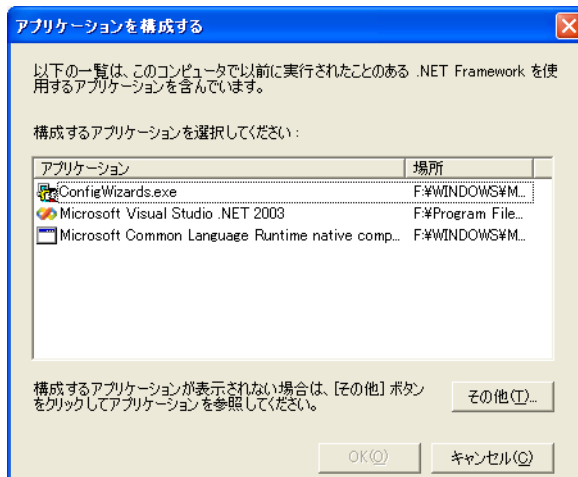
    </dependentAssembly>
  </assemblyBinding>
</runtime>
</configuration>

```

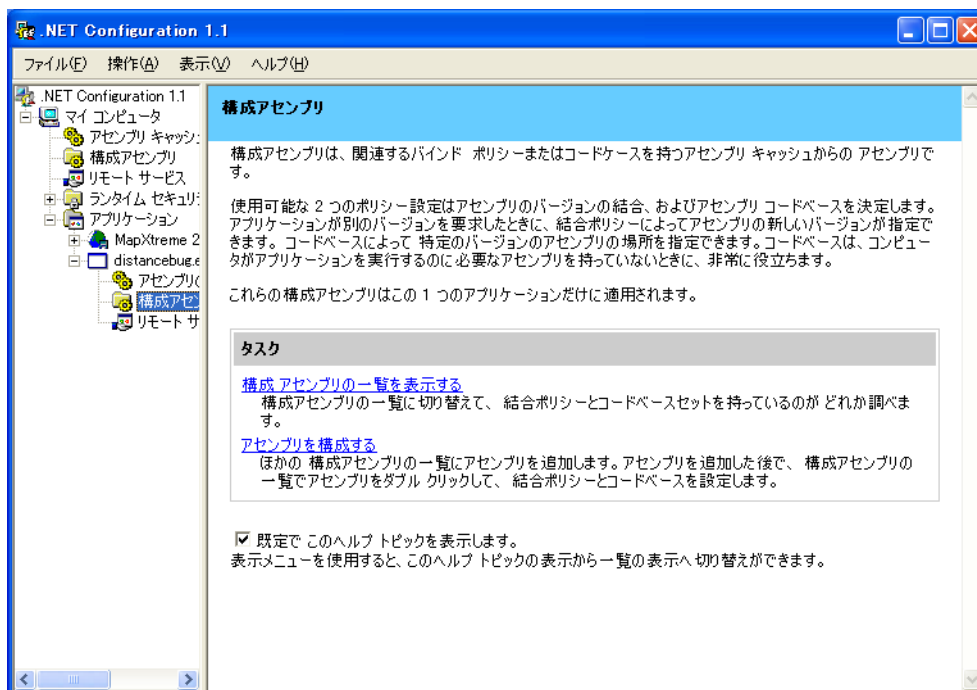
2. コントロールパネルの [管理ツール] をクリックします。
3. [Microsoft .NET Framework 1.1 Configuration] をクリックします。ウィンドウの左側で、[アプリケーション] をクリックします。



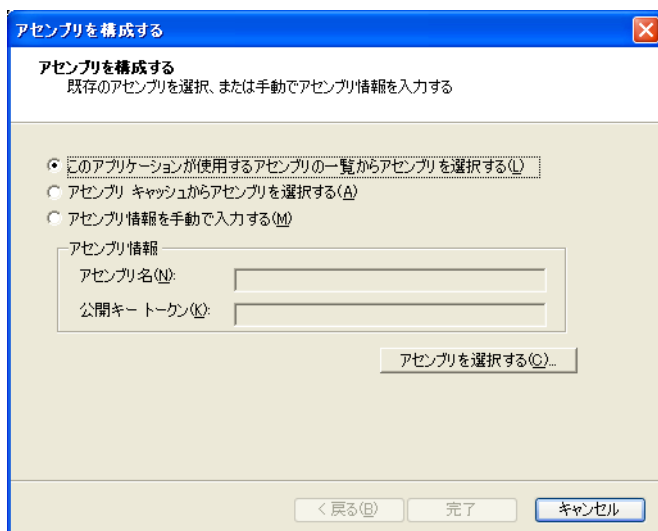
4. [構成するアプリケーションの追加] をクリックします。



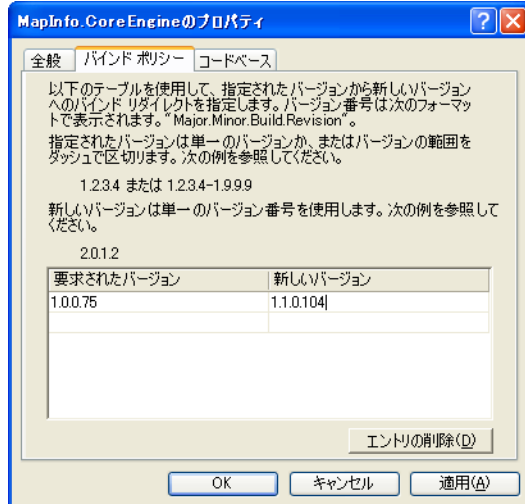
5. [その他] をクリックし、アプリケーションの実行可能ファイル名を指定します。指定したアプリケーションが、ウィンドウの左側の [アプリケーション] の下に表示されます。次の例では、アプリケーション名は `distancebug.exe` です。[+] 記号をクリックしてツリー表示を展開します。ウィンドウの左側で、[構成アセンブリ] をクリックします。次のようなダイアログが表示されます。



6. タスクの一覧で、[アセンブリを構成する] をクリックします。または、[構成アセンブリ] を右クリックし、[追加] をクリックします。



7. アセンブリの一覧からアセンブリを選択するよう、先頭のラジオボタンをオンにして、[アセンブリを選択する] をクリックします。アセンブリ (この例では MapInfo.CoreEngine) を選択し、[完了] をクリックします。すると、次のようなダイアログボックスが表示されます。[バインドポリシー] タブを選択します。



8. [MapInfo.CoreEngine のプロパティ] ダイアログボックスで、要求されたバージョンおよび新しいバージョンを指定します。要求されたバージョンは、アプリケーションの .exe が実際に参照しているバージョンです (たとえば 1.0.0.75)。新しいバージョンは、アプリケーションのリダイレクト先のバージョンです。アプリケーションを MapXtreme 2004 リリース 6.1 にリダイレクトするには、新しいバージョンに 1.1.0.104 を指定します。[OK] をクリックします。
9. MapInfo.CoreTypes など、新しいバージョンにリダイレクトする他のアセンブリを同様に構成します。つまり、手順 7 を繰り返します。ただし、MapInfo.CoreEngine ではなく、MapInfo.CoreTypes (または他の適切なアセンブリ) を選択します。
- ここで、アプリケーションの配置場所を参照すると、myApp.exe.config または myApp.dll.config (myApp はアプリケーションの名前) というファイルがあります。

発行者ポリシーファイル

リダイレクトの設定を含む発行者ポリシーファイルのアセンブリと共に GAC にインストールすることができます。しかしこれは、MapInfo 構成としてサポートされていません。アセンブリバージョンをリダイレクトするための推奨される方法については、9 ページの「アプリケーション構成ファイル」を参照してください。

マシン構成ファイル

アセンブリのリダイレクトを構成するには、コントロールパネルの管理ツールの下にある Microsoft .NET Framework 1.1 Configuration ツールを使用できます。このツールを使用すると、WINDIR¥Microsoft.NET¥Framework¥v1.1.4322¥Config に配置されているマシン構成ファイルを編集できます。

ここでリダイレクトを設定すると、指定したアセンブリを参照しているすべてのアプリケーションが新しいバージョンを使用するようになります。この方法によるリダイレクトの使用には注意が必要です。マシン構成ファイルの設定は、アプリケーション構成ファイルと発行者ポリシーファイルのどちらの設定よりも優先されるからです。

MapXtreme 2004 の新機能

MapXtreme 2004 のリリース 6.1 には、以下の拡張が加えられています。

- 共存可能なインストーラ
- Web アプリケーションの配布
- カスタムラスタハンドラの構成
- カスタムシンボルディレクトリのカスタマイズ
- コンテキストメニューのカスタマイズ
- キーボードショートカットのプログラム機能

注意：各説明文の冒頭にある番号は、MapInfo の問題追跡用番号です。

共存可能なインストーラ

共存可能なインストーラにより、既存のバージョンに影響を与えることなく、新しいバージョンの MapXtreme SDK をインストールできます。既存のバージョンと、新しくインストールした SDK は、それぞれ独立して機能します。これを実現するために、インストーラは、仮想ディレクトリ、個別のディレクトリ、およびレジストリキーを別個に作成します。

Web アプリケーションの配布

『MapXtreme 2004 開発者ガイド』では、独自のインストーラを使用してアプリケーションを配布するための手順について解説しています。『MapXtreme 2004 開発者ガイド』の第 2 章で、「カスタムインストーラを使った配布」というトピック、および具体的な手順を解説したサブトピックを参照してください。その最後に、Web アプリケーション用の仮想ディレクトリを作成する手順があります。この手順は不要となりました。仮想ディレクトリは自動的に作成されるようになったためです。

MapInfo では、ホワイトペーパーも提供しています。タイトルは『How to Create, Package, and Deploy a MapXtreme 2004 Application』です。このホワイトペーパーは、MapInfo の次の Web サイトからダウンロードできます。

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

[Publications] の一覧から、[MapInfo MapXtreme for Windows] を選択し、[MapXtreme 2004 Deployment] というホワイトペーパーを選択します。

カスタムラスタハンドラの構成

MapXtreme 2004 アプリケーションの構成により、MapXtreme 2004 にデフォルトで含まれている以外のラスタハンドラを使用させたり、まったく新しいラスタタイプに対応させることができます。使用するラスタハンドラの優先順位を変更することもできます。

MapXtreme 2004 のデフォルトのインストールでは、ラスタハンドラはすべて <program files>%Common Files%MapInfo%MapXtreme%6.1%RasterGridHandlers に配置されます。また、mirasteru.dll ファイルの配置場所もここです。アプリケーションで他のラスタハンドラを使用する場合も、ここにインストールすることをお勧めします。このデフォルトの配置場所を使用すれば、他の構成手順は不要です。

カスタムラスタハンドラをデフォルト以外の場所に配置したい場合は、その場所をアプリケーション構成ファイル(デスクトップアプリケーションの場合)または web.config ファイル(Webアプリケーションの場合)で指定する必要があります。指定するには、<ApplicationDataPaths> 以下で <Path> 要素または <SpecialPath> 要素を定義し、選択したラスタハンドラをそのフォルダにコピーします。

たとえば、デスクトップアプリケーションのカスタムラスタハンドラを、デフォルトと異なる MyAppData ディレクトリに配置する場合には、次のような .config ファイルを使用できます。

```
<?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>
```

この例では、<Personal> タグは、ユーザの My Documents で定義されている特別な場所です。これにより、My Documents の MyAppData というフォルダが参照されます。<ApplicationDataPaths> タグ内では、<SpecialPath> タグまたは <Path> タグを使用します。両方のタグを使用することは、ラスタハンドラをいずれかの“MyAppData”フォルダに配置できることを意味します。両方のタグを使用してもエラーにはなりません、意図する操作ではないでしょう。これはデフォルトの構成ではないので、mirasteru.dll も同じディレクトリに配置する必要があります。

アプリケーションデータが .NET Framework の特別なシステムフォルダからの相対パスにある場合は、<SpecialPath> 構文を使用します。たとえば、My Documents ディレクトリの下の MyAppData というディレクトリにアプリケーションデータがある場合、構成ファイルのエントリは次のようになります。

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

ここで“Personal”は、My Documents ディレクトリを表す、.NET Framework の Environment.SpecialFolder 列挙体の値です。

この構成方法は、複数のラスタハンドラによって管理できるファイルの種類について、優先させるラスタハンドラを変更する場合にも使用できます。たとえば、JPEG ファイルは Halo または LEADTOOLS (どちらも MapXtreme 2004 に組み込まれています) で処理できます。通常は Halo が優先されます。Halo のファイル拡張子である *.rhv が LEADTOOLS のファイル拡張子である *.rhx よりもアルファベット順で前に来るからです。ただし、MapXtreme 2004 v6.1 では、構成ファイルで定義されている <ApplicationDataPaths> が最初に検索されるため、デフォルトの [CommonFiles] フォルダ内のハンドラの前に、指定したラスタハンドラが検索されます。したがって、定義したフォルダに LEADTOOLS の *.RHX ハンドラをコピーし、そのハンドラが最初に検索および使用されるように MapXtreme 2004 を構成することもできます。

別の方法として、開発者が [CommonFiles] フォルダ内のファイル拡張子を変更して、目的のラスタハンドラがアルファベット順で先に来るようにできます。ただし、この方法の場合、MapXtreme 2004 で開発したすべてのアプリケーションに影響するため、予想外の副作用が生じる場合があります。また、この方法でファイル拡張子を変更した場合、名前を変更したラスタハンドラは MapXtreme 2004 をアンインストールしても削除されません。このような理由により、ラスタハンドラの場所と優先順位を変更する場合は、構成による方法を使用することをお勧めします。

カスタムシンボルディレクトリのカスタマイズ

カスタムシンボルはアプリケーションと同じフォルダ内にある CustSymb フォルダ、またはそれ以外のフォルダに格納できます。ファイルの格納場所のカスタマイズ方法については、『MapXtreme 2004 開発者ガイド』の付録 B の「アプリケーションデータファイルの場所」を参照してください。

カスタムビットマップシンボルを別の CustSymb フォルダに格納したい場合は、アプリケーションの構成ファイルで <Path> 要素または <SpecialPath> 要素を使用します。この方法は、13 ページの「カスタムラスタハンドラの構成」で説明した方法と非常に似ています。<SpecialPath> タグの例を次に示します。

```
<?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>
    </configuration>
```

この例では、C:¥Documents and Settings¥username¥Application Data¥Your Corporation¥Your Application¥CustSymb などのフォルダでカスタムシンボルが検索されます。<SpecialPath> 要素にはフォルダ名 CustSymb を含めてはいけないことに注意してください。含めた場合、このパスは ...¥Your Application¥CustMapXtreme 2004¥CustSymb¥CustSymb と解釈されます。“CustSymb” が重複しているため、この指定は正しくありません。

アプリケーションデータが .NET Framework の特別なシステムフォルダからの相対パスにある場合は、<SpecialPath> 構文を使用します。たとえば、My Documents ディレクトリの下に MyAppData というディレクトリにアプリケーションデータがある場合、構成ファイルのエントリは次のようになります。

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

ここで“Personal”は、My Documents ディレクトリを表す、.NET Framework の Environment.SpecialFolder 列挙体の値です。

コンテキストメニューのカスタマイズ

MapXtreme 2004 では、ユーザがレイヤツリーで右クリックしたときに表示される、独自の LayerControl コンテキストメニューアイテムを開発者が作成できます。これに対応するために、新しい ContextMenuTargetObject プロパティは、ユーザが右クリックしたオブジェクトを返します。

LayerControlEnhancer クラスの定義方法の例を示すサンプルコードが提供されています。このクラスを使用すると、既存の LayerControl のコンテキストメニューにカスタムアイテムを追加できます。この例は、¥Samples¥Features¥LayerControl ディレクトリ内の LayerControl サンプルアプリケーションにあります。

キーボードショートカットのプログラム機能

LayerControl ツールバー上のすべてのアイテムにアクセスできるようにキーボードショートカットを作成できます。[下] ボタン、[上] ボタン、および [削除] ボタンにプログラムからアクセスできるようにする、PerformDown メソッド、PerformUp メソッド、および PerformRemove メソッドは既に用意されていました。

新しい AddMenuMnemonic プロパティにより、[追加] メニューの表示に関連付けられたキーにもプログラムからアクセスできます。AddMenuMnemonic プロパティの構文は次のとおりです。

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

ツールバーへのカスタムボタンの追加

MapXtreme 2004 では、ツールバーへカスタムツールボタンを追加する手順が非常に簡素化されています。これに対応するために、MapToolBar クラスに ImageList プロパティが追加されています。このプロパティは、設計時のプロパティページに表示されないようにオーバーライドされています。MapToolBarButton の ToolId プロパティは設計時に表示されるようになったため、カスタムツール用の値を設定できます。

ツールバーにカスタムボタンを追加するには、簡素化された以下の手順に従います。

1. フォームに MapToolBar を追加します。
2. Visual Studio の [プロパティ] ウィンドウで、Buttons プロパティを強調表示し、[...] ボタンをクリックして、コレクションエディタダイアログを表示します。
3. [追加] ボタンをクリックし、新しい MapToolBarButton を追加します。
4. ボタンの ButtonType プロパティを CustomTool (ドロップダウンの一覧の末尾のアイテム) に設定します。この時点では、ボタンには何も表示されません。カスタムボタンに関連付けられたイメージがないからです。
5. ボタンの ToolId プロパティを、MapControl の Tools コレクションに追加したカスタムツールの名前に設定します。ToolId の値とコレクション内のツールが一致していない場合、ユーザがボタンをクリックしたときに実行時例外がスローされます。
6. [OK] をクリックして、ボタンのコレクションエディタダイアログを閉じます。
7. フォームのコンストラクタにおいて、InitializeComponent の呼び出しの後で、MapToolBar の ImageList にカスタムビットマップを追加するためのコードを記述します。カスタムビットマップとアプリケーションの関連付けをどのように行うかはプログラマ次第です。1つの方法としては、設計時に ImageList をフォームに追加し、カスタムイメージを格納しておきます。そして、それらのイメージをツールバーの ImageList に転送する処理は実行時に行うようにコードを記述しておきます。そのコードの例を次に示します。

```
// カスタムツールボタンのビットマップをツールバーのイメージリストに追加する
foreach (Image image in this.imageList1.Images) {
    this.mapToolBar1.ImageList.Images.Add(image);
}

// カスタムツールのボタンにビットマップを関連付ける (イメージリストの末尾のイ
メージ)
this.mapToolBarButtonBlueSelect.ImageIndex =
this.mapToolBar1.ImageList.Images.Count-1;
```

MapXtreme 2004 リリース 6.1 のバグフィックス

MapXtreme 2004 リリース 6.1 のバグフィックスでは、次の問題が解決されています。

- [データアクセス](#)
- [ドキュメント化](#)
- [Selection \(選択 \) クラス](#)
- [Legends \(凡例 \)](#)
- [Raster](#)
- [日本語 Legacy Text](#)
- [インストール](#)
- [展開](#)
- [マッピング](#)
- [永続化](#)
- [SDK](#)
- [ユーザインターフェイス](#)
- [Web アプリケーション](#)
- [デスクトップアプリケーション](#)
- [全般](#)

注意： 各説明文の冒頭にある番号は、MapInfo の問題追跡用番号です。

データアクセス

TableInfoServer

31997 TableInfoServer が、Oracle からテーブルを追加する際に where 条件を無視する問題が解決されました。

Table.UpdateFeature()

32002 Table.UpdateFeature が、ステートメントを正しく解析するようになりました。基底の ResultSet テーブルが、フィールドを読み取り専用としてマークしていたことが原因でした。

AddColumns

T11013 `Table.AddColumns` を使って、大量の行を含むフィールドを TAB ファイルに追加する際のパフォーマンスが向上しました。サイズの大きな MS Access、ADO.NET、RDB のテーブルから、ネイティブの TAB ファイルにフィールドを追加するのがより高速になりました。

ビューテーブル内の OrderBy 句

T11023 MapXtreme 2004 は、`TableInfoView()` を使用するクエリ内の `OrderBy` 句を無視します。例外はスローされません。これにより、各 RDBMS データベースの動作の違い (Oracle では許可されるが、SQL Server では許可されないなど) に対応できるようになりました。

Oracle ビューを開くときのエラー

T11188 Oracle ビューをワークスペースマネージャで開くと、エラーが発生することがありました。この問題が解決されました。

Catalog.Search のメモリリーク

T11696 データを読み込んだ後で、`Catalog.Search` がメモリを使い果たしてしまっていました。このメモリリークが修正されました。

MiScrollable リーダー

T11750 現在のプロパティを使おうとすると `MIScrollable` リーダーでエラーが発生する問題が解決されました。

InsertFeature メソッド

T11621 `InsertFeature` メソッドが正しく動作するようになりました。

Table.InsertFeatures メソッド

T11160 `Table.InsertFeatures` メソッドが、テーブルにフィーチャを正しく挿入するようになりました。

Table.UpdateFeature

32002 `Table.UpdateFeature` メソッドが、テーブル内の行を正しく更新するようになりました。

Spatialware ローダのパフォーマンス向上

T11861 SpatialWare 4.81 から MapXtreme へのテーブル読み込みのパフォーマンスが向上しました。

FeatureProcessor クラス

T12279 `SearchALL` を使ってすべてのレコードを返すとき、結果の `IResultFeatureCollection` をバッファしようとする、エラーが発生していました。この問題が解決されました。

メモリ管理

T11696, T12007, T11645メモリ管理が向上し、いくつかのメソッドでメモリリークが発生していた問題が解決されました。

Access と MapInfo テーブルの結合

T11323 特定の状況において、Access テーブルと MapXtreme 2004 USA テーブルを結合すると、正しいマップの主題図が作成されない問題がありました。この問題が解決されました。

TableInfoView

T12076 一部の SQL クエリが、TableInfoView を通じて発行された場合に誤った結果を返すことがありました。この問題が解決されました。

サンプルアプリケーションの Find エラー

30578 インデックス化されていないテーブルで Find サンプルアプリケーションが使われた場合、例外がスローされていました。適切なエラーメッセージが表示されるようになりました。

Info Tool コントロール

T12451 AddColumns でフィールドを追加した後で、Info Tool が正常に動作するようになりました。

ドキュメント化

GridRead コンストラクタのサンプルコード

T13262 オンラインヘルプの GridRead Class の例が修正され、Session.Current 呼び出しが含まれています。

LegendControl サンプルコード

T13242 オンラインヘルプの LegendControl サンプルコードが修正されました。

TableServerInfo

31857 TableInfoServer.ConnectString プロパティの説明に、ODBC の接続文字列で“DLG=SQL_DRIVER_NOPROMPT”を指定する必要があるという内容の注記が追加されました。

TableInfoServer による CacheParameter クラスの作成

30173 TableInfoServer クラスは自動的に CacheParameter クラスを作成し、タイプを On に設定します。以前はこの点が明確にドキュメント化されていませんでした。

SpatialSchemaPointRef サンプルコード

T13242 オンラインヘルプの SpatialSchemaPointRef サンプルコードが修正されました。

TableInfoNative クラスサンプルコード

T11830 オンラインヘルプの TableInfoNative クラスサンプルコードが修正され、.TAB ファイルを作成するようになりました。

クエリ内の Order By 句

T11023 TableInfoView クエリでは、Order By 句が無視され、エラーメッセージが表示されません。これは正しい動作であり、今回明確にドキュメント化されました。

TableInfoView

T10899 TableInfoView を使い、機能のサブセットに基づいて新しいレイヤを作成するとき、新しいレイヤはベースレイヤと同じ MBR を持ちます。これは正しい動作であり、今回明確にドキュメント化されました。

Selection クラス サンプルコード

T12120 オンラインヘルプの Selection クラスのサンプルコードが修正されました。

AllowOverlap と AllowDuplicates

T12473 処理対象のラベルで許可されていれば、ラベル間の重複やオーバーラップが許可されます。これは、MapXtreme の以前の動作から変更された点です。この新しい動作は、MapXtreme と MapInfo Professional の両方に共通しています。

TableInfoAscii の最大文字数

30955 ASCII テーブルでは、表示できる文字数は最大で 253 文字です。それを超える部分は切り捨てられます。これは正しい動作であり、今回 TableInfoAscii 向けにドキュメント化されました。

MapExport クラス

T12229 オンラインヘルプの MapExport サンプルコードが修正され、MapControl を使ってマップをエクスポートするためには、そのマップを複製する必要があることが示されました。

FeatureProcessor クラス

T11923 FeatureProcessor クラスのサンプルコードが更新され、Combine メソッドの使用法が分かりやすくなりました。

Selection (選択) クラス

IFeatureCollections を Selection に追加できない

Selection は MultiResultSetFeatureCollection であり、追加できるのは IResultSetFeatureCollections のみです。その他の種類の IFeatureCollections を追加することはできません。

その結果、IFeatureCollection を Selection に追加しようとする则表示されるエラーメッセージが、“この値は、MapInfo.Data.IResultSetFeatureCollection 型である必要があります。”に修正されました。

Selection.Add メソッド

31995 Selection.Add(IFeatureCollection) メソッドが修正され、IResultSetFeatureCollection を IFeatureCollection (ベースインターフェイス) として Selection に追加できるようになりました。次の例では、Selection.Add メソッドを使って IResultSetFeatureCollection 内のフィーチャを選択します。

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

```
' テーブルを使ってマップを読み込み、テーブルを取得する
MapControll.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")

' 最初のフィーチャを取得する
pRdr = pTable.ExecuteReader()
pRdr.Read()
pFtr = pRdr.Current
pRdr.Close()

' 最終的なフィーチャのコレクションを作成する
pFeatureCollection =
MapInfo.Data.FeatureCollectionFactory.CreateResultSetFeatureCollection
(pTable, pTable.TableInfo.Columns)

' このフィーチャをコレクションに追加する
pFeatureCollection.Add(pFtr)

' フィーチャのコレクションを Selection に追加してフィーチャを選択する
MapInfo.Engine.Session.Current.Selections.DefaultSelection.Add
(pFeatureCollection)
```

Legends (凡例)

凡例の枠

T11038 凡例を縦にしても横にしても、周りに正しく枠が描画されるようになりました。

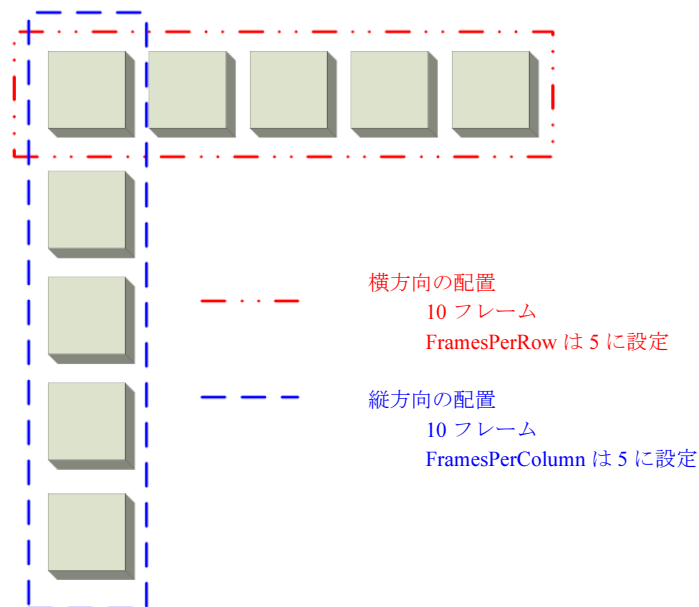
凡例フレーム配置のプロパティの変更

T11036 LegendFormat.FramesPerRow と LegendFormat.FramesPerColumn プロパティのデフォルト値が、1 から 0 に変更され、FramesAlignment プロパティの動作と同じになりました。FramesPerRow および FramesPerColumn は、フレームを縦または横に配置することのできる FrameAlignment と組み合わせて使用されます。

たとえば、10 個のフレームを含む判例を横に配置する場合を考えます。横方向の FrameAlignment とは、FramesPerRow の値を利用するということです。FramesPerRow の値が 5 の場合、結果は、それぞれ 5 つのフレームを含む 2 つの行になります。

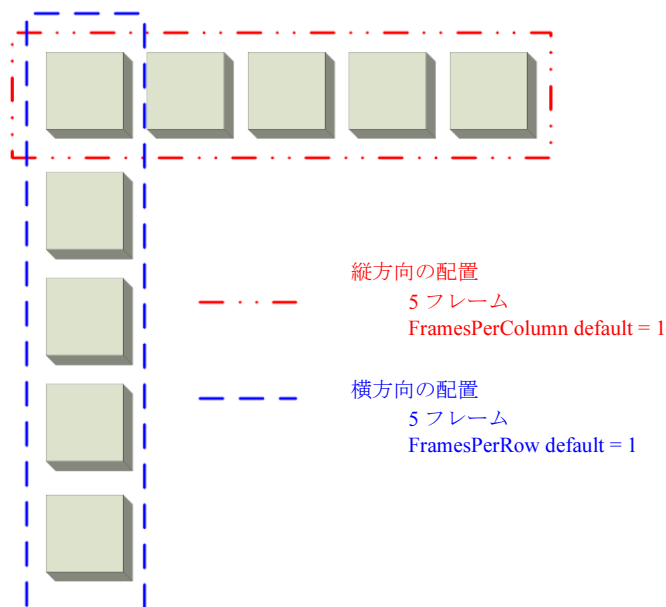
同様に、凡例中で 10 個のフレームを縦に配置する場合に、FramesPerColumn を 5 に設定すると、それぞれ 2 つのフレームを含む 5 つの行になります (各フィールドに 5 つのフレーム)。

10 個のフレームを縦および横に配置した様子を次の図に示します。



デフォルト値として 1 を使用していたビルド 75 では、正常に動作はするものの、配置が意図したものと正反対になってしまっていました。FramesPerRow のデフォルト値が 1 だと、凡例内に複数のフレームがある場合、縦配置のように表示されてしまいます (各行に 1 フレームずつ表示)。次の図を参照してください。

同様に、`FramesPerColumn` が 1 だと、凡例内に複数のフレームがある場合、横配置のように表示されてしまいます (各フィールドに 1 フレームずつ表示)。今回、デフォルト値が 0 になったことで、凡例内のフレームの数にかかわらず、`FrameAlignment` の値のみが考慮されるようになりました。



Raster

T10973 Halo JPEG ラスタハンドラがスレッドセーフになりました。

日本語 Legacy Text

T10922 日本語の `LegacyText` オブジェクトで、デフォルト以外のフォントが正しく表示されるようになりました。

T11144 `LegacyText` オブジェクトの作成時に、日本語のフォント名が表示されるようになりました。

インストール

T12914 WMS/WFS 用 XML スキーマが MapXtreme 2004 CD に収録され、正常にインストールされるようになりました。

展開

ビルド 75 ランタイム

MapXtreme 2004 リリース 6.1 にアップグレードする場合、このアップグレードはリリース 6.0、ビルド 75 と共存できます。

カスタムインストーラを使ったパッケージング

リリース 6.1 は、MapXtreme 2004 でカスタムのアプリケーションインストーラを使用する際のいくつかの問題を次のとおり修正します。

T10905 カスタムインストーラでアプリケーションをパッケージングするとき、そのアプリケーションが COM+ プールを使用している場合、これに MapInfo.Web.MSM と MapInfo.CoreEngine.MSM を含める必要があります。これにより、CoreEngine が COM+ オブジェクトとして登録されます。

T10903 マージモジュール (MSM) を使って作成したアプリケーションインストーラが、Microsoft C ランタイム DLL を正しい場所 (System32 フォルダ) にインストールするようになりました。

T10907 MSM を使って作成したアプリケーションインストーラが、仮想ディレクトリ (VDir) MapXtremeWebResources6_1 を正しく作成するようになりました。

LayerControl を使った展開

T12671 Layer Control を使った MapXtreme 2004 アプリケーションの展開が正常に動作するようになりました。以前は、未処理例外が発生していました。

マッピング

Geoset インポート

T12443 Geoset インポートが正常に動作するようになりました。これは、問題番号 T12473 の AllowOverlap および AllowDuplicates の動作と関連があります。

ワークスペースマネージャ内の修飾

T11702 特定の状況において、.TAB ファイルを開いたときには修飾が表示されるにもかかわらず、ワークスペースを開いたときに修飾が表示されない場合があります。この問題が解決され、ワークスペースを開いたときに意図したとおりに修飾が表示されるようになりました。

ScaleBarAdornment

T11703 ワークスペースマネージャ内でマップのズームが変更されたときに、ScaleBarAdornment の値が正しく変更されるようになりました。

変更されたラベルの動作

T11819 表示範囲内で Autolabel が有効になっているとき、変更 (Select ツールでドラッグ) されたラベルが正しく表示されるようになりました。

WMF エクスポート

T12122 WMF としてエクスポートされたイメージが、正しいファイル形式でエクスポートされるようになりました。

[レイヤ管理] から Geoset を開く

31692 特定の状況において、[レイヤ管理] から Geoset を開くとクラッシュすることがありました。この問題が解決されました。

SearchWithinScreenRadius および SearchWithinScreenRect

T13037 ResultSetCombineMode.Replace が使われたときに、SearchWithinScreenRadius と SearchWithinScreenRect が正しく更新されるようになりました。

ラベルの配置

T11590 ラインのデフォルトの配置が Alignment.TopCenter になりました。ポイントのデフォルトの配置が Alignment.CenterRight になりました。これは、MapInfo Professional の動作と同様です。

ラベルの Callout Line

T11933 ラベルの Callout Lines が、非表示でなければならない場合にも表示されていました。これは修正され、CalloutLine.Use = False のときに Callout Line が表示されることはなくなりました。

Geoset のズーム速度

T10985 ワークスペースマネージャで開いた Geoset の速度が大幅に向上しました。

永続化

ワークスペースのマップ単位

T12097, T12043 MapXtreme 2004 では、マップ単位が “survey ft” および “sft” に設定されているワークスペースファイルを読み込めるようになりました。

ワークスペースに保存されたビューテーブル

T11833 ビューテーブルをワークスペース内で保存するとき、ネイティブの .tab ファイルとして正常に保存されるようになりました。

ラベルの位置

T12654 Above または Below オフセットの使用時に、ワークスペースマネージャでラベルの位置を正しく保存できるようになりました。

LabelLayer の永続化

T11350 LabelLayer の可視性が、ワークスペースマネージャで正しく永続化されるようになりました。

ラベルフォントの永続化

T12086 編集されたフォントが、ワークスペースマネージャで正しく永続化されるようになりました。

Legend.Format.FramesPerRow

T11274 FrameAlignment が Horizontal に設定されているとき、Legend.Format.FramesPerRow が正しく保存されるようになりました。

ラベルの Callout Line

31907 ラベル上のカスタムの Callout Line がワークスペースマネージャで永続化されるようになりました。

GML の属性

T12891 GML の属性が、W3C 標準に従って書き出されるようになりました。

SDK

デスクトップライセンスとプーリング Web アプリケーション

T12909 以前は、プーリング Web アプリケーションがデスクトップライセンスで実行でき、実行時のウォーターマークもありませんでした。この問題が解決されました。

Web ライセンスの問題

T12911 Web アプリケーションでプールが有効になっており、Web ランタイムライセンスしかない場合、Web アプリケーションは実行できず、例外がスローされていました。この問題が解決されました。

ワークスペースマネージャのクラッシュ

T13132 特定の状況において、すべてのテーブルを閉じると、ワークスペースマネージャが `ArgumentNullException` をスローしてクラッシュすることがありました。この問題が解決されました。

棒グラフ主題図

T12719 棒グラフ主題図を作成し、それを積上げグラフにすると、クラッシュすることがありました。この問題が解決されました。

投影法の追加

T11546 WFS では、MapInfo Professional 7.8 で新しく追加された投影法を含むテーブルを読み込もうとすると、サーバ例外が発生することがありました。この問題が解決されました。

カスタムメニューアイテム

T13021 カスタムメニューアイテムを追加することにより、アイテムがレイヤツリーに追加されると、`LayerControl` のコンテキストメニューで一時的に問題が起きることがありました。この問題が解決されました。

ユーザインターフェイス

Info Tool アイコン

31693 Info Tool アイコンを選択したとき、アクティブであることが分かりやすくなりました。

Layer Control から主題図のスタイルを変更する

31969 特定の状況において、Layer Control から主題図のスタイルを変更すると未処理例外が発生していました。この問題が解決されました。

FeatureStyles サンプルアプリケーション

31891 FeatureStyles サンプルアプリケーションで、メインフォームを縮小して MapControl が表示されないようにすると、System.ArgumentException が起こっていました。この問題が解決されました。

ModifyRangedThemeDlg

T11279 主題図の追加ウィザードを使って新しい RangedLabelTheme を作成するとき、レンジの数値を調整できるようになりました。

ワークスペースマネージャ

T11704 修飾を含むワークスペースを開いた後で、[ファイル] メニューから [新規作成] を選択し、[変更をワークスペースに保存しますか ?] の問いに「いいえ」を選択すると、例外が発生していました。この問題が解決されました。

主題図の作成ウィザード

31935 [マップ] メニューから主題図の作成ウィザードを起動した後で、レンジラベル主題図を追加すると、未処理例外が発生することがありました。この問題が解決されました。

Expression ダイアログ

T11609, 31864 主題図の作成ウィザードの Expression ダイアログが正常に閉じるようになりました。

ラベルレイヤの削除

31926 主題図の作成ウィザードでラベル主題図を作成した後で、マップからラベルレイヤを削除すると MapException エラーが発生していました。この問題が解決されました。

ラベルの式

29997 正しいラベルの式の一部が無効として解釈されてしまうことがありました。この問題が解決されました。

AddAdornments

31643 以前は、既存の修飾を含むマップに修飾を追加すると、新しい修飾が上部ではなく下部に追加されていました。これにより、スケールバーがタイトルの後ろに隠れてしまうなどの問題が起きていました。この問題が解決されました。

修飾のドラッグおよび移動

31641 マップ上の凡例、タイトル、スケールバーの修飾をドラッグして移動することができるようになりました。

Web アプリケーション

SelectionChanged ハンドラ

T11648 PolygonSelectionTool、RadiusSelectionTool、および RectangleSelectionTool の SelectionChanged ハンドラが、Web アプリケーションで正常に動作するようになりました。

State サンプルアプリケーション

31587 Web からアクセスする State サンプルアプリケーションが正常に動作するようになりました。

ASP.NET アプリケーション

31273 VB ASP.NET アプリケーションがサーバへのポストバック操作で正常に動作するようになりました。

Response.Redirect と Server.Transfer

T13463 特定の状況において、Response.Redirect と Server.Transfer が未処理例外をスローすることがありました。この問題が解決されました。この解決策では、新しい Session.Dispose メソッドを使って ISession インスタンスを適切に破棄し、新しい web.config エントリを使って ISession インスタンスの状況固有のストレージを制限しています。MapInfo.Engine.ISession インスタンスを管理する方法の詳細については、[3 ページの「MapInfo のセッション状態管理」](#)を参照してください。

プールされたアプリケーションでのメモリリーク

T11483以前は、凡例が破棄された後でもメモリが解放されませんでした。この問題が解決されました。

デスクトップアプリケーション

追加 (Add) ツール

31799 ツールを追加するときに、適切な場所と適切なズームレベルで追加されるようになりました。

全般

ライセンスのエラー

A10015 デンマーク語バージョンの MapXtreme 2004 をインストールするときのライセンスのエラーが解決されました。

インストーラのエラー

T10905 マージ モジュール (MSM) を使って MapXtreme 2004 アセンブリを配布すると、COM+ を使って MapXtreme にアクセスするアプリケーションが失敗します。

AddWebForm ウィザード

T13025 MapXtreme AddWebForm ウィザードのパスが修正されました。これで、ウィザードが正常に起動するようになりました。

Geoset の [選択可能] プロパティ

T13249 MapXtreme Geoset マネージャで設定される [選択可能] および [編集可能] プロパティへの変更が強化されました。

ワークスペースマネージャの TitleAdornment

T11713 ファイルがワークスペースマネージャに読み込まれたとき、TitleAdornment が正常に表示されるようになりました。

Geometry オブジェクト上の Distance メソッド

T12588 2つのオブジェクトのジオメトリがそれぞれ違う座標系にある場合でも、Distance メソッドが正常に動作するようになりました。

LoadMapWizard

31778 LoadMapWizard の戻り値が正しくなりました。現在は、すべてのテーブルが問題なく開かれたかどうかに基づいて決まるようになっています。

日本語ランタイム

T11536 ワークスペースマネージャが、日本語ランタイムで正常に動作するようになりました。

ズームレベル

T11076 最大および最小のズームレベルが、Geoset マネージャおよびワークスペースマネージャと統一されました。

検索結果

T13344 検索結果が複数ある場合、これらの結果が正しくリストボックスに表示されるようになりました。

既知の問題

次に挙げる問題は、リリース 6.1 のリリースノートで既知の問題として指摘されているものです。これらは現在調査中、または解決に向けて作業中です。

- [データアクセス](#)
- [デスクトップアプリケーション](#)
- [主題図](#)
- [Web アプリケーション](#)
- [ワークスペースマネージャ](#)
- [ワークスペースマネージャ](#)
- [ツール](#)
- [全般](#)

データアクセス

- 24759 非常に小さな座標境界に Oracle レイヤを追加するとエラーが発生します。
- 30165 新しい MIPparameter に、無効な dbtype を設定した場合、NotSupportedException がスローされるべきですが、実際にはスローされません。

- 30375 TAB ファイルにパスワードが含まれていない Oracle データベースからダウンロードしたテーブルを開くときに、ユーザはパスワードを要求されません。これを防ぐには、TAB ファイルにパスワードを追加します。
- 30503 TableInfoServer 経由で Access テーブルを追加することができません。TableInfo にフィールド情報が含まれていない *.mdb ファイルを直接開くことはサポートされていません。これを防ぐには、TAB ファイルで Access テーブルを開くか、TableInfo オブジェクトにフィールド情報を含めます。
- 30839 Ä や Ü などの特殊文字を使った GROUP BY で、正しい結果が返されません。
- 31278 TAB ファイルにパスワード情報が含まれていない、リンクされた ODBC テーブルを MapControl に読み込むと、Web ページがタイムアウトになります。これを防ぐには、TAB ファイルにパスワードを追加します。
- 31529 MapTableLoader が DBMS テーブルの読み込みに失敗すると、メッセージ部分が不完全な "TableException" がスローされます。つまり、メッセージにテーブル名が含まれません。
- Web コントロールがフレームをサポートしません。

デスクトップアプリケーション

- 31268 デスクトップの MapControl をコピーして Windows フォームに貼り付けると、一部のプロパティが空白のまま残ります。空白になるのは、LeftButtonTool、MiddleButtonTool、RightButtonTool の各プロパティです。これを修正するには、[プロパティ] ウィンドウで、コントロールのプロパティを手動でリセットします。
- 30800 [スタイル] ダイアログボックスを最初に開くとき、2 度目以降の呼び出しよりも時間がかかります。
- 30365 修飾キーまたはマウスホイールが使用された場合、ラバーバンドアウトラインを持つツールを閉じた後に、マップ上にラバーバンドイメージの残骸が残ります。これを防ぐには、ツールの使用時にマウスホイールサポートをオフにします。
- 29444 [ポリゴン選択] ツールまたはカスタムのポリゴン / 折れ線ツールを使用するとき、描画が完了しても、ラバーバンドイメージが完全に消えない場合があります。再描画のスピードが問題にならない場合、Map.Invalidate() メソッドは ToolUsed イベントの後で呼び出してください。
- 30310 Pan などのツールを使っているときにマウスホイールで移動すると、マップの描画が崩れます。アプリケーションでマウスホイールを無効にすると、この問題を回避できます。マップが定期的に再描画されるようなプログラムを作成してもよいでしょう。
- 31955 GridInfoForm サンプルアプリケーションは、LayerControl の一番上のレイヤとしてグリッドレイヤが使われている場合にしか動作しません。このアプリケーションが動作できるようにするには、グリッドレイヤを LayerControl の一番上のレイヤに移動します。

主題図

- 31915 CreateThemeWizard では、ラベルリソースが存在しないときに、ユーザがラベル主題図を選択することができます。これにより、“Could not bind to the new display member. Parameter name: newDisplayMember.” という不可解なエラーが発生します。
- 31555 ハッチパターンが使用されているとき、ズーム時に円グラフ主題図内の各要素の色が変わってしまいます。

- 31234 円グラフ主題図では、フィーチャのない行に対する円グラフが表示されます。円グラフ主題図はテーブル内のすべての行(フィーチャのない行も含む)に対して表示されます。
- 30036 マイナスの数値を含む棒グラフを作成する場合、この棒グラフがグラフの境界ボックス内に収まりません。
- 30175 ワークスペースマネージャのレンジ主題図変更ダイアログの凡例タブで凡例に範囲を追加した場合、凡例の枠が正常に更新されません。
- 30977 個別値(ラベル)およびレンジ(ラベル)コンパクトサイズ凡例には、テキストのみで、シンボルがありません。
- 31479 MapXtreme 2004 と MapInfo Professional では、範囲の自然分布メソッドがそれぞれ違う結果を生成します。
- 31463 棒グラフまたは円グラフの主題図で、レンダリングされたときに Z-ordering がランダムに変わります。

Web アプリケーション

- 31177 主題図の生成に使われたテーブルが LayerControl、LegendControl、または MapControl で非表示に設定されているとき、主題図が MapControl に表示されます。
- 31502 LayerControl から設定された場合、グループレイヤとラベルグループレイヤに、ズーム範囲または縮尺範囲内に表示するオプションがありません。
- 31510 設計時に LayerControl を使って設定した場合、レイヤの Web アプリケーションで、ラインの方向、表示ノード、中心点が表示されません。
- 31263 MapControl をコピーして Windows フォームに貼り付けると、フォームの一部のフィールドが空白になります。これにより、MapAlias と WorkspaceFile プロパティの値が同じになり、1つのマップで行った変更が別のマップに誤って表示されてしまいます。
- 31354 Web LayerControl を使用しているレイヤを表すイメージが、デスクトップの LayerControl と同じように表示されません。
- 31427 Web LayerControl が状態を保存しません。
- 31700 マップを持たないテーブルを追加した後、ワークスペースが正しく永続化されません。
- 31893 ラスタレイヤのスタイルオーバーライドに、Web LayerControl 経由でアクセスできません。
- 31725 Web アプリケーションの設計時に名前付き接続ファイルを読み込むことができません。
- 31743 ページ上にマップがない場合、デフォルトの VB ASP.NET テンプレートでエラーが発生します。発生するエラーは、“Microsoft JScript ランタイムエラー: ‘undefined’ が null であるか、オブジェクトではありません”です。
- 31854 Web ツールに無効な MapID を設定すると、本来スローされるはずの “ArgumentException” ではなく “NullReferenceException” がスローされます。
- 31845 特定の条件において、ASP.NET アプリケーションは空白のマップをレンダリングし、自らの状態を追跡するのが困難になります。

- 31933 Web サンプルアプリケーションの MWS ファイルにハードコードされているパスは、MapXtreme 2004 がデフォルトの場所にインストールされていないと機能しません。この問題を回避するには、MapControl からマップをクリアし、そのプロジェクトに関連付けられているテーブルおよびワークスペースを、サンプルデータと同じ場所に再読み込みします。ThematicsWeb サンプルアプリケーションでは World.tab ファイル、Tools サンプルアプリケーションでは World.mws ファイル、FindWebSample サンプルアプリケーションでは World.mws ファイルを使用します。
- 31496 LayerControl (Web またはデスクトップ) 内のグループレイヤを削除しても、警告やエラーメッセージが表示されません。
- 31654 Web RadiusSelect ツールで正しく描画することができません。このツールを使った描画中に表示されるべき透過円が、選択時に分断されます。
- 31645 グループラベルレイヤの親ノードの可視性をオフにしても、子ノードの可視性が自動的にオフになりません。これらのプロパティは LayerControl で設定されます。
- 31925 ディスプレイの設定が 16 色 (Windows 2000 のデフォルト) のとき、長方形ズームが表示されません。
- 31929, 31881 コントロールにマップが読み込まれて MapControl の ID が変更された場合に、古い MWS ファイルがハードディスク上に残ります。また、テーブルが読み込まれた webform から MapControl が削除されても、MWS ファイルは削除されません。この古い MWS ファイルは手動で削除できます。

ワークスペースマネージャ

- 30802 ワークスペースマネージャを使って、ラスタレイヤに透過を作成する方法がありません。透過をプログラムによって作成するか、この情報を TAB ファイルに保存することができます。
- 30164 ワークスペースマネージャの使用時に、Shift キーと Ctrl キーの役割が逆転します。複数のオブジェクトを選択する場合には Ctrl キーを押下します。各オブジェクトの選択をキャンセルするには、Shift キーを押下します。MapInfo Professional の場合と役割が逆になっています。
- 31183 ワークスペースマネージャから印刷を行う場合、ラスタレイヤのスタイルのオーバーライドが出力に表示されません。たとえば、半透明のスタイルオーバーライドを持つラスタでは、透過が印刷結果に反映されません。
- 31673 OCI または IUS クライアントがインストールされていないマシンでリモート OCI または IUS テーブルを開くときに、ワークスペースマネージャでエラーが表示されません。この操作を行っても、何も起こりません。Web アプリケーションでは、クライアントがインストールされていない場合に設計時にリモート OCI または IUS ソースからレイヤを開こうとすると、実行時に空白のマップが表示され、エラーは発生しません。

インストール

- MapXtreme 2004 を削除した後、GAC (Global Assembly Cache) に MapInfo アセンブリが残ります。これは、次の 2 つの原因で起こります。

aspnet_wp.exe プロセスが、アンインストールにおいて検出されないアセンブリに対してオープンな参照を保持しています。これを解決するには、次の作業を行います。

- a. コントロールパネルの管理ツールにあるインターネットインフォメーションサービスを使って Web サービスを停止します。Web サーバを選択してサービスを停止します。
- b. この後の「アセンブリの削除」セクションの手順に従います。

インストーラサービスがレジストリを破損しています。これが起きているかどうかを確認するには、次の作業を行います。

- a. レジストリエディタを開きます ([スタート] メニューの [ファイル名を指定して実行] を開き、「regedit」と入力します)。
 - b. HKEY_CLASSES_ROOT¥Installer¥Assemblies¥Global に移動します。
 - c. (デフォルト) キーの種類が REG_MULTI_SZ かどうかを確認します。
 - d. 値が REG_MULTI_SZ の場合、(デフォルト) キーを選択し、[編集] メニューの [削除] をクリックします。
キーの種類が REG_SZ になり、再表示されます。
 - e. この後の「アセンブリの削除」セクションの手順に従います。
- GAC からアセンブリを削除するには、次の操作を行います。
- a. [スタート] メニューの [ファイル名を指定して実行] を開き、「assembly」と入力します。
 - b. MapInfo.* アセンブリをすべて選択し、削除します。
 - c. エラーが表示されたら、[プログラムの追加と削除] コントロールパネルで MapXtreme 2004 のすべてのリファレンスを確認します。
- インストールプロセスの実行中、MapInfo.CoreEngine を COM+ サービスに登録するためのコマンド画面が表示されます。3 つのエラーが表示されます (返される 16 進数はシステムによって異なります)。


```
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* )
```

 これらのエラーは無視してかまいません。
 - MapXtreme 2004 で提供されているサンプルデータでは、テーブルメタデータ内の次のテーブルの coordsys エントリが誤っています。
 - US_CNTY: coordsys メタデータは緯度 / 経度 (測地系なし) を参照しますが、マップファイルは緯度 / 経度 (NAD 27 for Continental US) です。
 - Mexico: coordsys メタデータは等角投影法 (北米) を参照しますが、マップファイルは緯度 / 経度 (NAD 27 for Central America) です。
 - USA: coordsys メタデータは等角投影法 (北米) を参照しますが、マップファイルは緯度 / 経度 (WGS 84) です。
 - 29781 .NET Services Installation ツール (regsvcs.exe) の実行中、MapInfo Engine を COM+ へ登録するときにインストーラが “システム dll が見つかりません” という例外エラーをスローします。これは、混合 dll のアンロードと読み込みによって起こります。この問題の解決策は、Microsoft と共に調査中です。
 - 31883 インストールの修復の実行後、レジストリ内のシリアルナンバーが削除されます。これを防ぐには、レジストリのエントリ HKEY_LOCAL_MACHINE¥Software¥MapInfo¥MapXtreme¥6.0¥SerialNumber キーをシリアルナンバーで置き換えます。
 - 31701 インストールした後、別のユーザとしてログインすると、Visual Studio で MapXtreme ツールボックスが利用できません。

- 31832 ツールを削除して MapXtreme 2004 インストールを変更すると、この変更後にプログラムを削除するとエラーが発生します。

ツール

- 31855 ツールイベントで返された座標は、ズームレベルによって相当の距離をオフセットされます。ツールイベントは現在、マウスでクリックした地点のおおよその値を返します。スナップモードがオンでも、返された座標は選択された地点よりもかなり遠くなる可能性があります。問題は、イベントハンドラがすべての座標を整数のピクセル値に自動的にスナップすることです。つまり、ズームレベルが高い場合、また、同じズームレベルでもマップの幅が狭い場合に、より大きなエラーになります。Stock Select ツールと Add ツールはスナップで正常に動作しますが、ToolUsed イベントは、ユーザがクリックした地点から最高で 5 マイル離れた場所を報告します。この問題を回避するには、Custom ツールからではなく stock Select および Add ツールからカスタムツールを派生させ、ToolUsed で返された座標を使うのではなく、選択または作成された実際のオブジェクトで作業を行います。
- 24604 Stock ツールの AddPolyline と AddPolygon の描画に一貫性がありません。これらのツールを使う場合、描画されたオブジェクトの Z-order を予測する方法はありません。この問題を回避するには、各フィーチャをそれぞれレイヤとして追加します。これで、意図どおりの順序が保たれます。
- 30205 ラベルが選択されているとき (Select ツールを押下)、回転ハンドルが機能しません。
- 30813 MapTool クラスの InfoTipsEnabled プロパティでは、オーバーライドをツール単位で反映させることができません。特定のオブジェクトでは、このプロパティを設定すると、設定がそのオブジェクトだけに限定されず、グローバルで適用されます。
- 31656 Point Selection ツールを使うと、選択が反転されます (これは、選択可能な海レイヤなど、リージョンがオーバーラップする場合に選択がお互いをキャンセルするためです)。
- 31405 MapControl がフォーカスを失ったとき、状態 (EditText、AddPolyline など) を持つツールがクリーンアップされません。これを修正するには、マップが定期的に再描画されるようなプログラムを作成します。
- 31567 ZoomIn と ZoomOut ツールのズーム間隔が MapX とは異なります。この動作はオーバーライド可能で、希望する動作でカスタマイズすることもできます。
- 31569 LayerControl の使用時、ズームツールがマップ名をインクリメントします。たとえば、Map1 という名前のマップがある場合、ズームするごとにその名前が Map2、Map3、というように変わります。
- 31799 Add ツールでマップにオブジェクトを追加するとき、新しいオブジェクトが表示されるべき場所に表示されません。0,0 地点の近くに表示されてしまいます。マップ上でビューに変更があると、この動作は治まります。この問題を修正するには、いずれかの Add ツールを有効にした後で、ツールバーの Click イベントで Map.Zoom = Map.Zoom を設定します。
- 31917 [動的選択ツール] チェックボックスがオンの場合、オブジェクトの選択がキャンセルされても、マップ上では選択されているように表示されます。
- 31945 AddText ツールがアクティブの場合、スナップモードが機能しません。
- 31953 極端なレベルに拡大すると、その後 ZoomOut ツールで縮小しようとしたときにズームレベルが 0 のままになることがあります。これを回避するには、LayerControl の [レイヤ全体を表示] などのコントロールを使用してビューを変更します。

全般

- 31889 geoset から読み込まれた最初のズーム修飾子の単位が常にマイルになってしまいます。
- LeadTool の JPEG 2000 および TIFF を使用しているとき、マルチスレッドモードで MapExport が失敗します。JPEG 2000 は Windows 2000 で失敗します。TIFF CMYK はすべてのプラットフォームで失敗します。
- WMS サーバは、MapInfo Professional で読み込むことができない GIF ファイルをエクスポートします。MapXtreme 2004 の WMS サーバで生成される GIF イメージは有効ですが、MapInfo Professional v7.5 ではエラーになります。これを回避するには、サーバから返された他のイメージフォーマットを使用します。
- 25344 このリリースでは、Shape (*.shp) ファイルの書き込みがサポートされていません。
- 25630 一般図の座標系にある Point を MGRS に変換しないでください。例外がスローされるか、何も起こりません。このバージョンでは、予期しない結果になる可能性があります。
- 25655 Point タイプを作成したり、Point タイプの Data プロパティの緯度/経度を極端な値に設定したりすると、マップビューは緯度/経度の値を正しくクリップしません。たとえば、緯度/経度の Point を作成したり、Point の Data プロパティを (999999999.0, 999999999.0) などの実在しない値に設定したりすると、この値が本来は -360, -90, 360, 90 にクリップされるはずが、1000, 1000 にクリップされてしまいます。
- 30827 スタイルリポジトリがタイプセーフではありません。スタイルでないオブジェクトをスタイルリポジトリに渡すと、クラッシュします。これは、IList または ICollection を実装するすべてのオブジェクトに言えることです。
- 27788 フォントシンボルスタイルの Angle プロパティを正の値に設定すると、シンボルが反時計回りに回転します。MapXtreme 2004 のその他のオブジェクトは、Angle プロパティに正の値が設定されると時計回りに回転します。
- 29042 MapXtreme 2004 では、null シンボルの描画は、ライン、フィル、フォントシンボルなどでは可能ですが、ビットマップシンボルでは行えません。シンボルスタイルコントロールのカスタムシンボルで“N”を選択すると、意図したものとは別のシンボルが表示されます。これを避けるには、別のポイントスタイルを使用して、それを“null”にします。
- 29092 多くのラインセグメントを持つ複雑な MultiPolygon/Region オブジェクトをバッファするとエラーが発生します。これを防ぐには、バッファされるデータを簡略化します。
- 29250 アプリケーションが実行中の状態で、設計時に MapControl を変更しないでください。MapControl から“SystemInvalidOperation”例外がスローされます。
- 29391 MapXtreme アプリケーションは、オンザフライで SpatialSchemaXY が割り当てられたテーブルが含まれる、永続的なワークスペースファイルを読み込めません。これを防ぐには、空間メタデータを TAB ファイルに保存し、TableInfo オブジェクトを再作成します。
- 29824 [シンボルスタイル] ダイアログボックスの [シンボル色を適用] チェックボックスは、新しい拡張カスタムシンボルが選択されているときには利用不可能になっているべきです。実際は、利用不可能になっていません。
- 30528 figure 8 および figure B のような線形オブジェクト (MultiCurve、Curve、LineString) で、SelfIntersects が正常に機能しないことがあります。
- 30413 レイヤが非表示の場合、ラベルがレイヤに表示されたままになります。これを防ぐには、ラベル修飾子の可視性をオフにします。

- 30423 日本語ロケールで設定されているコンピュータで実行されているとき、TAB ファイルから ASCII ファイルを作成できない場合があります。
- 30456 96dpi、501px X 300px、8bpp の PNG 透過性が機能しません。
- 30812 スナップモードのクロスヘアフィーチャで、スナップのオンまたはオフを切り替えると、再描画の問題が発生します。これを防ぐには、マップを移動またはリフレッシュして、強制的に再描画します。
- 31199 WMS クライアントでは、ユーザが境界を指定せずにマップを “get” できます。GetMap リクエストの一部として境界を指定しなければなりません。
- 31792 既に MapControl に読み込まれている MWS ファイルを再読み込みしないでください。「ファイル読み込みエラー：xxx.mws」のようなエラーが表示されます。xxx は読み込もうとしたファイルの名前です。
- 31223 LabelSource オブジェクトで使用されているテーブルを変更すると、この変更が LabelSource オブジェクトに反映されません。
- 31267 LabelSource レイヤのエイリアスは、マップ内で一意ではなく、Label レイヤ内で一意である必要があります。この制約を回避するには、Layer エイリアスと LabelSource エイリアスの両方を使って一意性を実現します。
- 31372 Symbol Style のカスタムリストボックスで、縦方向のスクロールバーが機能しません。
- 31342 Web MapControl 上で移動すると、イメージの左上に境界線が表示されます。
- 31471 WBMP ファイルを読み込めません。
- 31689 LayerControl は、さまざまなオブジェクトを含むキャッシュされたリモートテーブルで一貫性がなくなります。リモートデータソースは、常に予期できる順序でフィーチャを返すとは限りません。これを避けるには、LayerControl.SetTableImage を呼び出してイメージを明示的に指定します。
- 31480 最初にスタイルオーバーライドを設定するとき、スタイルオーバーライドの色が適用されません。
- 31481 GRC および GRD ラスタハンドラはスレッドセーフではありません。
- 31488 WFS クライアントが GML2 を解釈できません。WFS クライアントは MapInfo-GML3 しか解釈できません。このため、クライアントは MapInfo サーバでしか機能しません。
- 31811 現在のコードページがデータのコードページと等しくない場合、WriteTabFile は、フィールド名として無効な TAB ファイル文字を書き出します。
- 31803 CoordSysFactory.LoadDefaultProjectionFile() は、デフォルトの coordsys が見つからない場合 NullReferenceException をスローします。
- 31809 TableInfo.WriteTabFile() が、使用不可能な TAB ファイルを書き出すことがあります。特定の状況において、このメソッドは既存の TAB ファイルを上書きして、無効にしてしまうことがあります。このメソッドは、既存の TAB ファイルの更新や上書きではなく、新規 TAB ファイルの作成にのみ使用してください。
- 31617 LegacyText オブジェクトのハンドルの編集が正常に行えません。
- 31668 ScaleBarAdornment および ScaleBarAdornmentControl の 2 つのクラスが、スケールバーのサイズについて異なる値を返すことがあります。

- 31899 [動的選択ツール] チェックボックスのオン/オフを切り替えた後で `InsertionLayerFilter` をプログラムによって変更すると、再描画の問題が発生します。選択領域の編集ハンドルが、マップをリフレッシュするまで表示されません。これを避けるには、アプリケーションで使用していないときには動的選択ツールを無効にしておきます。また、`LayerFilter` の変更を、エンドユーザが `LayerControlDlg` 経由でアクセスできるものだけに限定するという方法もあります。
- 31671 マップがクリックされたときに、選択されているラベルがスクリーンの外にある場合、この選択がキャンセルされません。
- 31726 GML の公開において、複数のジオメトリフィールドを持つ `FeatureCollections` が正常に読み書きされません。
- 31702 `LoadMapWizard` 経由で複数の `geoset` を開くと、新しいレイヤが一番上ではなく一番下に配置されます。この問題を解決するには、いずれかの `Add` ツールを有効にした後で `Map.Zoom = Map.Zoom` を設定します。
- 31815 ズームのレイヤリングが変わると、マップが非表示の選択を誤って記憶することがあります。
- 31791 `Mapping.SearchInfoFactory` メソッドがパラメータを検証しません。
- 31850 アニメーションレイヤが有効なとき、一時テーブルを閉じると、そのラベルがディスプレイから消えます。`LabelsAreEditable` プロパティを `True` に設定するのはアニメーションレイヤには効果がありません。この動作を修正するには、アプリケーションから `Mapinfo.Mapping.Map.Invalidate()` を頻繁に呼び出します。
- 31920 タイトル修飾を表示した後で、フォントやサイズを変更できません。
- 31836 編集されたラベルやラスタレイヤを含むワークスペースを読み込むと、エラーが発生します。
- 31846 `CursorImageURL` URL タイプが、設計時に保存されません。
- 31890 ダイアログボックスおよびサンプルアプリケーションフォームのアンカーと最小サイズが、誤っているか、存在しないことがあります。これにより、ダイアログボックスのサイズ変更が反映されません。
- 31898 ダイアログボックス内の指数表記で極端に大きな値を使用しないでください。“オブジェクトの参照がオブジェクトのインスタンスに対して設定されていません”などのエラーが発生することがあります。
- 31936 `FillStyle` ダイアログボックスの、特定の `FillStyle` プロパティの設定で、ダイアログボックスのすべてのピッカーが無効になります。
- 31958 WFS サンプルが、実行時に `System.URLFormatException` を生成します。これを修正するには、`samplesample.cs` ファイル内の `SimpleSample` クラスの URL 定数を、有効な WFS サーバを指すように変更します。