

Autodesk MapGuide 6.5 から新しい MapGuide テクノ ロジーへの移行

このホワイト ペーパーでは、Autodesk MapGuide® 6.5 以前のリリースから新しい MapGuide テクノロジーへのいくつかの変更点の概要を説明します。さらに、Autodesk MapGuide 6.5 の既存のアプリケーションから新しい MapGuide テクノロジーへの移行方法についての指針も説明しています。

簡潔に示すために、本ホワイト ペーパーでは、新しい MapGuide テクノロジーのオープン ソース版と商用版は、どちらも「Autodesk MapGuide® Enterprise」として記載しています。Autodesk MapGuide 6.5 以前のバージョンは「Autodesk MapGuide」として記載しています。ここで説明する概念はすべて、新しい MapGuide テクノロジーのオープン ソース版と商用版の両方にあてはまります。Autodesk MapGuide® Studio は、別途ご購入いただくことができます。

目次

新しいテクノロジーが登場した背景	3
Autodesk MapGuide Enterprise — 新しいアーキテクチャ	3
Autodesk MapGuide 6.5 と Autodesk MapGuide Enterprise のコンポーネント の比較	6
.....	8
MWX/MWF の必要性	9
移行計画	10
準備段階 — 計画と準備	11
目標	11
機能のリスト	11
Web インタフェース	11
開発プラットフォームの選択	12
サーバの構成	12
第 1 段階 — データの設定とロード	12
第 2 段階 — マップ画層の作成/マップの作成	13
第 3 段階 — アプリケーションの移行	16
第 4 段階 — テストとフィードバック	17
第 5 段階 — 最終的な変更	17
第 6 段階 — 配置	17
Web 層 API の解説	17
MapGuide の API	17
PHP 5 を使用して開発する	18
Java を使用して開発する	20
ASP .NET を使用して開発する	20
コード例	21
Web ページにビューアを組み込む	21
ユーザ情報とセッション情報を作成する: サイト サービス API	22
クラスの MgMap ライブラリを使用してマップ情報にアクセスする	23
画層の表示を列挙、更新、変更する	24
デジタイズする	26
赤入れする	29
Autodesk MapGuide 6.5 ベースの既存のレポートを呼び出す	29
製品関連情報	31
付録: Invoke Report ソース コード(PHP サンプル)	32
付録: 赤入れサンプル コード(PHP サンプル)	40

新しいテクノロジーが登場した背景

Autodesk MapGuide® Enterprise は、さまざまな顧客や市場の牽引者に応じて開発されました。Web 開発のツールとテクノロジーは、この 10 年で大幅に進歩しています。MapGuide テクノロジーも順調に発展を遂げてきたものの、これらすべてを反映して変化してきたわけではありませんでした。最近の Web アプリケーションは、ブラウザを使用して表示する一方で、「ビジネス ロジック」は Web 層または中間層で実行しています。MapGuide の複雑なアプリケーションには、クライアント層に重要なロジックが必要になります。さらに、Web 開発のために選択できるテクノロジーとして PHP、ASP.NET、JSP がありますが、核となる MapGuide プラットフォームはこれらのテクノロジーと統合されていません。そして、バッファリングのようなデータに関する処理は、変換されたりクリップされたグラフィックスに対してではなく、ジオメトリ データに対して行う必要があります。

Autodesk MapGuide Enterprise もまた、より大きな市場に対応します。現在、Windows Server® オペレーティング システムの普及率がおよそ 48 % のみにとどまっているのに対し、Linux® オペレーティング システムの採用が急速に拡大しています。ある環境では、Linux® のサポートが提供されていないことが障壁となっている場合があります。Apache は圧倒的によく使われている Web サーバであるため、Apache に対するサポートも重要になります。

広範なインターネット ユーザに対応するためには、Web マッピング ソリューションにラスタベースの表示技術がますます必要になってきます。ユーザは、パフォーマンスに優れ、拡張性があり、かつ核となるプラットフォームに適合するラスタベースの表示技術を必要としています。このため、MapGuide のプラグイン バージョンと同等のラスタベースのプラットフォームの機能を提供することが必要になります。

Autodesk MapGuide Enterprise もまた、DWF™ テクノロジーを使用し、さらにこれを強化しています。特に、DWF の強力な印刷および出力機能を利用して、AutoCAD® 図面を見事に正確に表示することができます。また、多くの企業は単独のビューア テクノロジー ベースに移行することもできます。

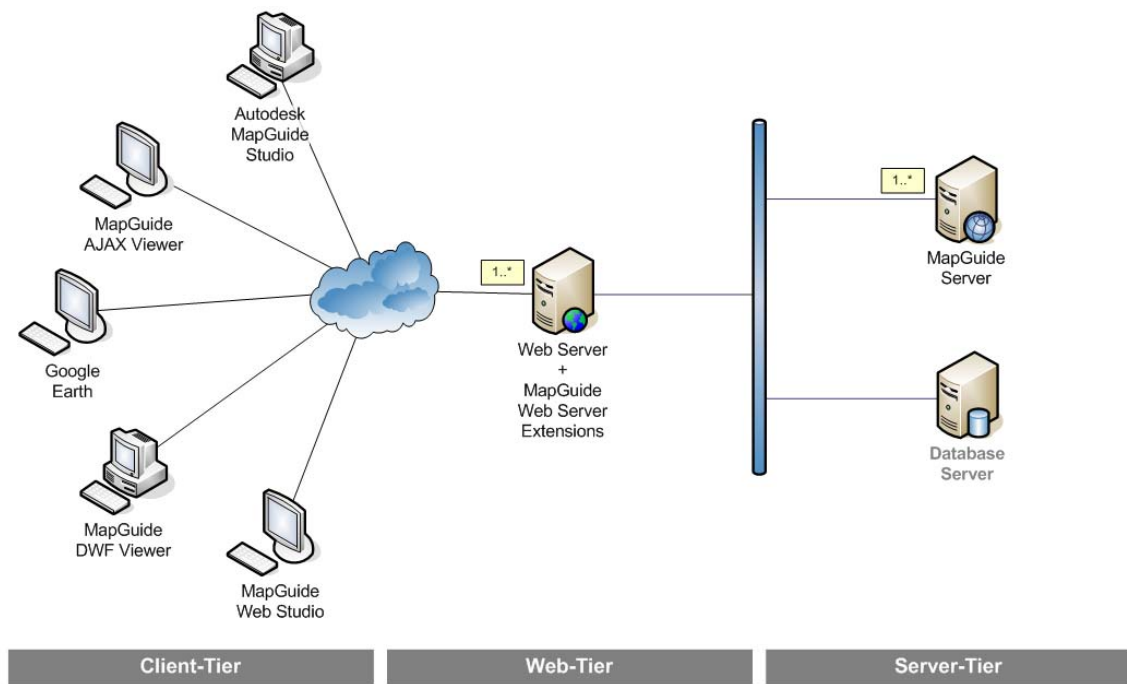
MapGuide のアーキテクチャは 10 年以上の歴史があります。2.5 次元データおよび 3 次元表示によるデータ分析の将来的なサポート、WFS(Web Feature Service)のような OGC(Open Geospatial Consortium)によるさらなるイニシアチブ、そして豊富なメタデータに対する Web サービスとサポートを提供するためには、プラットフォームを近代化することが求められます。

Autodesk MapGuide Enterprise — 新しいアーキテクチャ

新しい Web マッピング プラットフォームである Autodesk MapGuide Enterprise は Autodesk MapGuide 6.5 をはるかに超えており、新しいアーキテクチャ、プログラミング言語のサポート、データ アクセス方法、表示オプション、およびオーサリング環境を提供します。

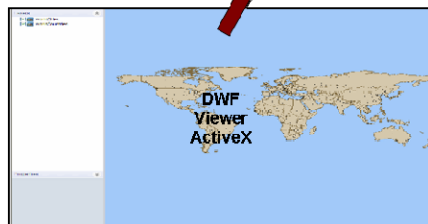
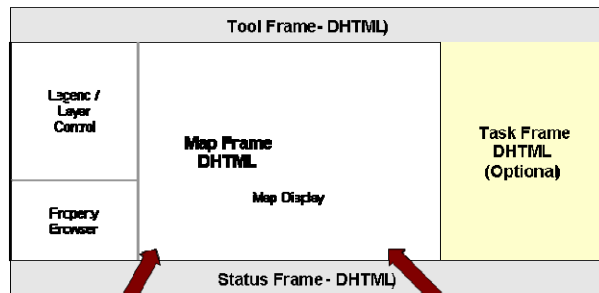
新しいプラットフォームは、オープン ソース ソフトウェア(MapGuide Open Source)として、または商用版(Autodesk MapGuide Enterprise)として入手することができます。新しい MapGuide テクノロジーは、Linux オペレーティング システムを含む複数のプラットフォーム上でインストール、カスタマイズ、および開発することができます。

Autodesk MapGuide のアーキテクチャ



Autodesk MapGuide Enterprise は、Autodesk MapGuide 6.5 と異なるアーキテクチャを持ち、サーバサイドの機能に移したことにより、マッピング機能と地理空間機能の大半がサーバ上で実行されます。API(アプリケーション プログラミング インタフェース)は、主に Web Server Extensions により公開され、.NET、Java®、PHP 開発環境で利用することができます。アーキテクチャのこの移行の結果、ビューア テクノロジー(DWF テクノロジーを使用した ActiveX® ベースのビューアおよび AJAX テクノロジーに基づいた非プラグインのビューア)を選択できるようになりました。DWF ビューアと AJAX ビューアで使用できるアプリケーションの機能に違いはありません。おそらく最も重要なことは、Autodesk MapGuide Enterprise が、サーバサイド アプリケーション機能へ移行したことによって、空間クエリの処理が可能な Web ベースの空間エンジンとして機能できるようになったことです。空間クエリは、SHP、SDF、Oracle®、ArcSDE®™ といった各種データ ソースに対して実行することができ、なによりも、単一の FDO(Feature Data Objects) API を使用して実行することができます。この機能により、

Common Viewer Elements



DWF Viewer MapFrame Content

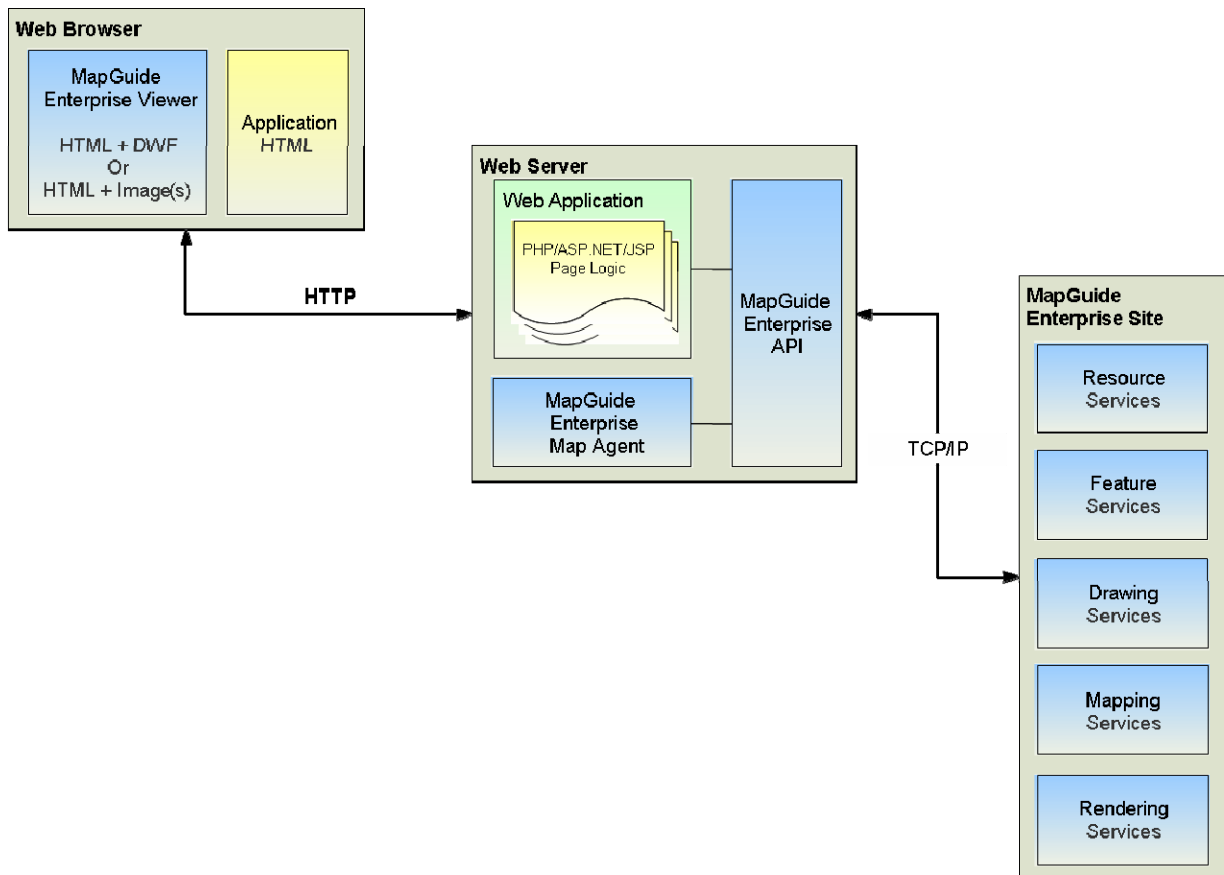


AJAX HTML Viewer MapFrame Content

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

開発者は異なるデータセットに対して1つのAPIだけを使用できるようになるため、データ形式よりもむしろアプリケーションに集中することができます。

アーキテクチャのこの移行により、Autodesk MapGuide 6.5 のさまざまなコンポーネントを使用して構築されたアプリケーションを移行するための対策をとる必要があります。新しい MapGuide テクノロジーによって、サーバ環境のセットアップ、データソースの設定、データのロード、アプリケーションのプログラミングといった面倒なプロセスは大幅に削減されています。移行作業には余分な労力が必要になるように思うかもしれませんが、Autodesk MapGuide Enterprise では、その強力な API によって基本的な機能と高度な機能のどちらについても開発時間が短縮されるため、移行による負担よりも新しいテクノロジーの利点の方が上回っています。これに対して Autodesk MapGuide では、Autodesk MapGuide 6.5 のクライアントサイド API の制限があるため、カスタム機能を開発するには、大きなクライアントサイド スクリプトが必要になります。



MapGuide Enterprise のアプリケーション/API のアーキテクチャ

Autodesk MapGuide 6.5 と Autodesk MapGuide Enterprise の コンポーネントの比較

次の表に、Autodesk MapGuide 6.5 のコンポーネントと、対応する Autodesk MapGuide Enterprise のコンポーネントを示します。

MapGuide コンポーネントの比較

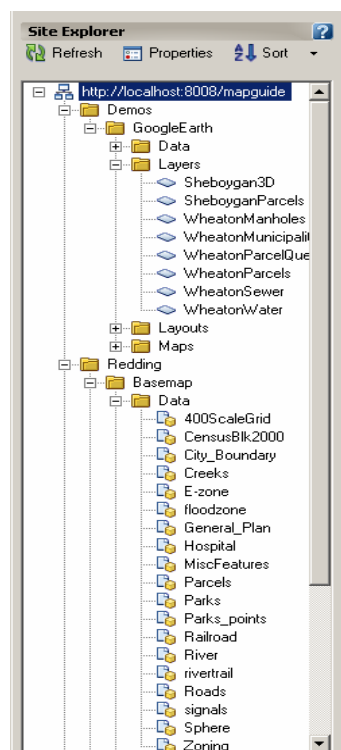
Autodesk MapGuide 6.5 のコンポーネント	Autodesk MapGuide Enterprise のコンポーネント
Server	Server/Web Server Extensions
Author	Autodesk MapGuide Studio
ActiveX Plug-in	Autodesk® DWF™ Viewer
Dynamic Authoring Toolkit	Web Server Extensions API
SDF COM Toolkit	Web Server Extensions API
MapGuide Raster Workshop	Autodesk MapGuide Studio
Autodesk MapGuide API	Web Server Extensions API、クラ イアントサイドのラッパー API
SDF Loader	Autodesk MapGuide Studio
LiteView	AJAX ビューア
Server Administrator	Web ベース サーバの Site Administrator

Autodesk MapGuide 6.5 の MapGuide Server には、空間ソースとデータベース ソースに対するデータ ソースの作成、データ ソースにアクセスするグループとユーザの作成、MapGuide エージェントの起動と停止、および使用状況とエラー ログの表示を行う機能が搭載されていました。これらの機能は、Web ベースの Site Administrator と Autodesk MapGuide Studio で使用できます。

Autodesk MapGuide 6.5 の MapGuide Author では、アプリケーションに組み込まれる MWF(マップ ウィンドウ ファイル)を作成することができました。Autodesk MapGuide Enterprise では、これらの MWF ファイルはデータ ソース、画層定義、マップ定義からなる「リソース」に取って代わり、これらはすべてサーバに常駐します。

ActiveX または Java で使用できる Autodesk MapGuide 6.5 プラグインは、一般的に使用されているオートデスクの DWF Viewer に置き換えられています。DWF Viewer には、機械や建築関連のファイルを含め、DWF ファイルを生成するすべてのオートデスク製品の標準のビューア テクノロジーであるという利点があります。DWF Viewer は、すべてのデジタル デザイン コンテンツ向けの単一の統合ビューアです。

Autodesk MapGuide 6.5 の Dynamic Authoring



サイトのリソース

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

Toolkit(DAT)を使用すると、MWF ファイルを操作することができ、MWF ファイルを XML 形式に変換し、XML データを修正してから、XML を MWF ファイルに変換し直すことができます。MWF ファイルは、その後、ブラウザに配信されることになります。DAT は、さらに強力な Web 層 API に置き換えられ、マップのすべての要素にアクセスできます。ユーザは、マップの任意の部分を瞬時に追加、編集、削除することができます。

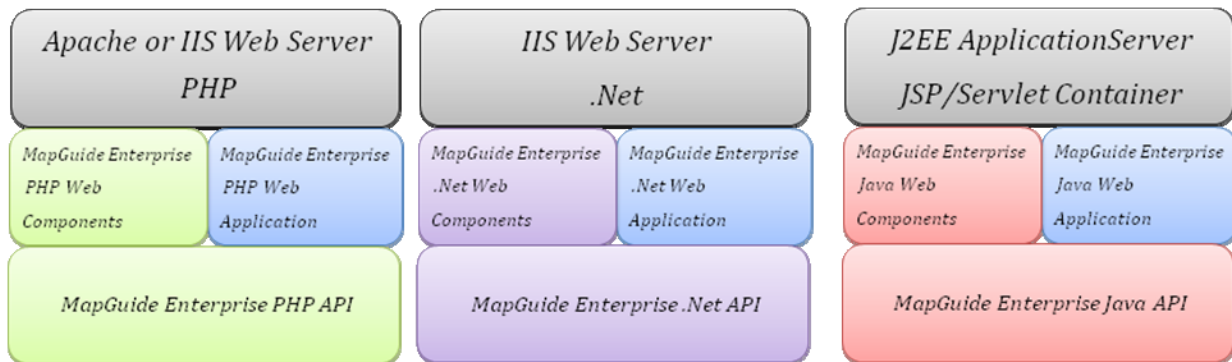
Autodesk MapGuide 6.5 の SDF COM Toolkit はサーバサイド COM オブジェクトであり、Autodesk MapGuide 6.5 では SDF データソースのクエリと操作を行うことができます。Web Extensions のフィーチャ サービス API は、FDO の読み取り/書き込み可能なプロバイダの機能に基づいて構築されたもので、これを使用して SDF や他のサポートされている形式を含め、地理空間データのクエリと操作を行うことができます。必要に応じて実行できるデータストアのダイナミックな作成と操作は、Web Extensions のフィーチャ サービス API により可能です。

Autodesk MapGuide 6.5 の Raster Workshop を使用すると、ラスター イメージ カタログを設定し、ラスター ファイルを操作して、後でラスター イメージの縮尺に応じた画層の作成に使用できるサブサンプル バージョンを作成することができます。Autodesk MapGuide Studio には、これと同じタスクを実行できる機能が標準で搭載されています。

Autodesk MapGuide 6.5 API は、MWF ファイルを操作できるクライアントサイド API です。この API はブラウザに依存するため、Autodesk MapGuide 6.5 のアプリケーションをブラウザ間で互換性を持たせるように作成することは困難でした。また、この API は機能が制限されているため、特別な機能を作成するためには高度なプログラミングが必要になります。



Autodesk MapGuide Enterprise の Web 層 API モデル



Autodesk MapGuide Enterprise の開発オプション

Autodesk MapGuide

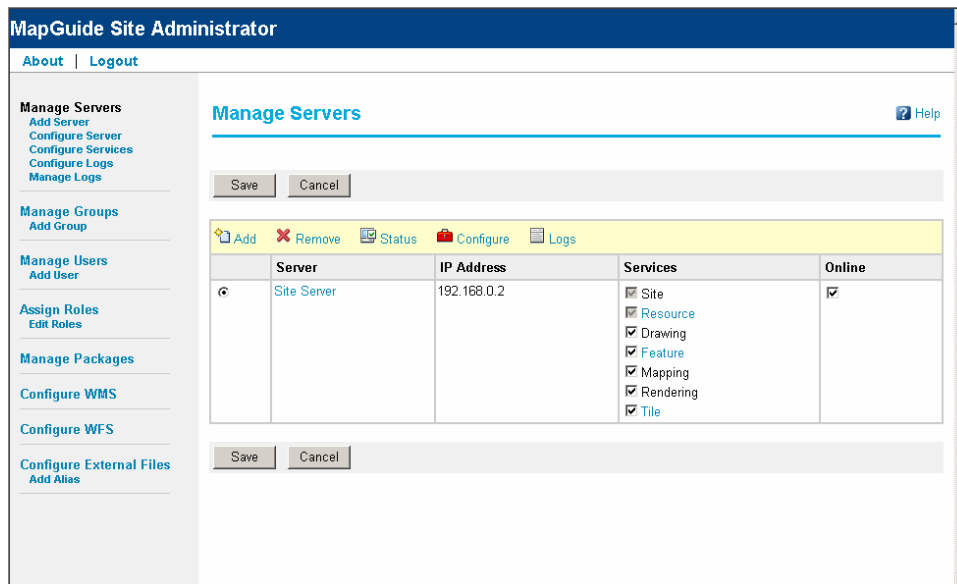
MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

Enterprise の Web 層 API は、サーバサイド API として再設計されました。この結果、ユーザはより一層強力で複雑なアプリケーションを開発できます。API はサーバサイドであるため、ブラウザに依存しません。したがって、ユーザはサポートされている 3 つの開発環境(.NET、Java、PHP)のいずれを使用しても、特定のブラウザに依存しない互換性のあるアプリケーションを記述することができます。ユーザは言語間で簡単に移植でき、言語に関係なくすべての機能を確実に使用できます。Autodesk MapGuide Enterprise には、フレーム間でユーザがインタラクティブに操作できるようにするために、クライアントサイド API が引き続き搭載されていますが、ほとんどのマッピング処理は Web サーバ上のビジネス ロジック層で実行されます。

Autodesk MapGuide 6.5 の SDF Loader は、地理空間データを作成し、これを SDF 形式に変換できるコマンドライン ユーティリティです。また、バッチ ファイルを作成して、定期的にデータの一括変換を実行することもできます。Autodesk MapGuide Enterprise では、SDF Loader の代わりに Studio を使用して、スケジューリングが可能なロード プロシージャを作成することができます。

Autodesk MapGuide 6.5 の LiteView は、プラグインを必要としないイメージ ベースのアプリケーションを構築できるサプレット ベースのソフトウェアです。Autodesk MapGuide Enterprise では、LiteView は AJAX ビューアに置き換えられています。開発者は DWF プラグインまたは AJAX ビューアのどちらかを使用して表示可能な 1 つのアプリケーションを構築できます。非プラグイン ベースのアプリケーションをパブリッシュするために、別々のコードを記述する必要はありません。

Autodesk MapGuide 6.5 の Server Administrator ツールを使用すると、地理空間ソースとデータベース ソースに対するデータ ソースの作成、データ ソースにアクセスするグループとユーザの作成、MapGuide サーバの起動と停止、使用状況とエラー ログの表示を行うことができます。これらの機能は、Autodesk MapGuide Enterprise の Site Administrator と Autodesk MapGuide Enterprise Studio で使用できます。

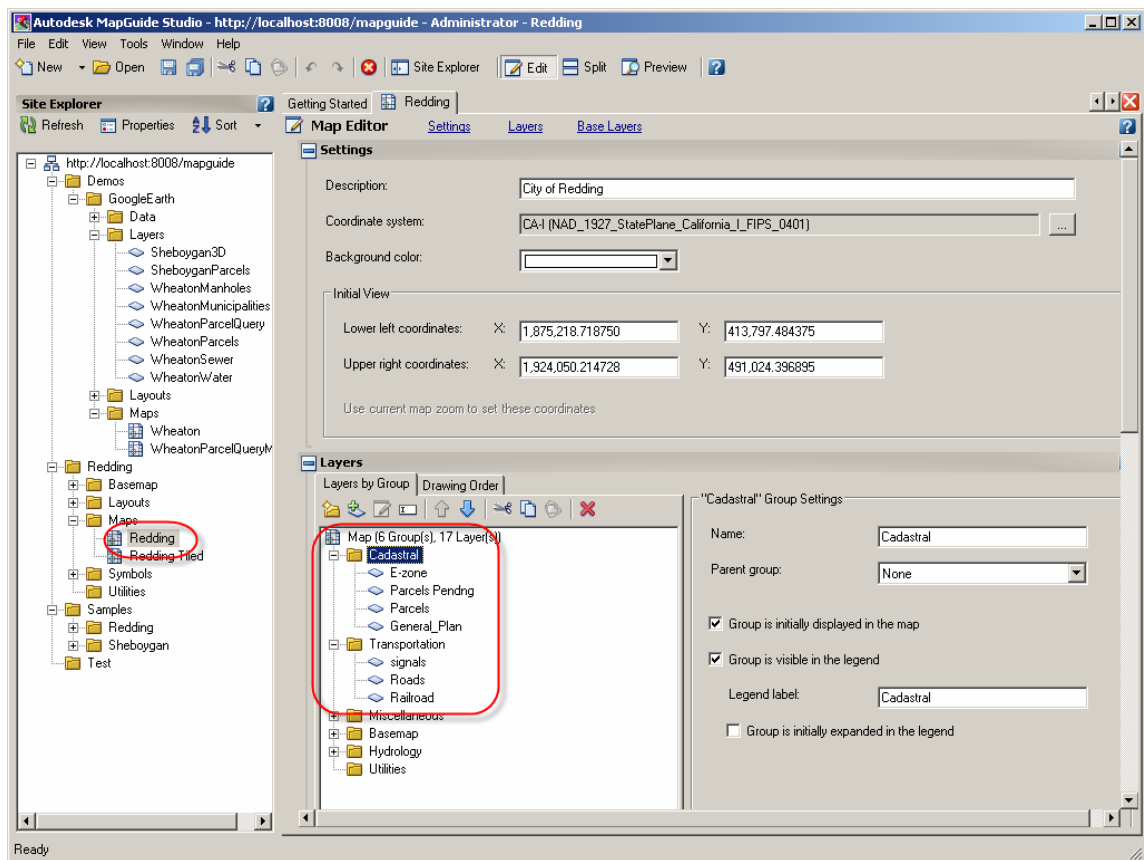


Autodesk MapGuide Enterprise の Web ベースのアドミニストレータ

MWX/MWF の必要性

Autodesk MapGuide Enterprise ではサーバ上でスタイル設定が実行されるため、MWF ファイルや MWX ファイルはもう必要ありません。MWF ファイルおよび MWX ファイルに格納されている定義と規則は、リソースとしてサーバ上に格納されます。マップにはもはや画層は含まれません。代わりに、マップ定義が画層定義を参照するため、一度画層を定義すれば、多くのマップでその画層を使用することができます。

リソースは、マップ、画層、データ接続などを定義します。各リソースは、階層構造のリソース リポジトリに格納される XML ドキュメントによって定義されます。各リポジトリは、リソース データベース内で管理される仮想ファイル システムと考えることができます。リソース データベースは Autodesk MapGuide Enterprise Server 上に常駐し、リソース リポジトリを集約化した一元保管場所となります。

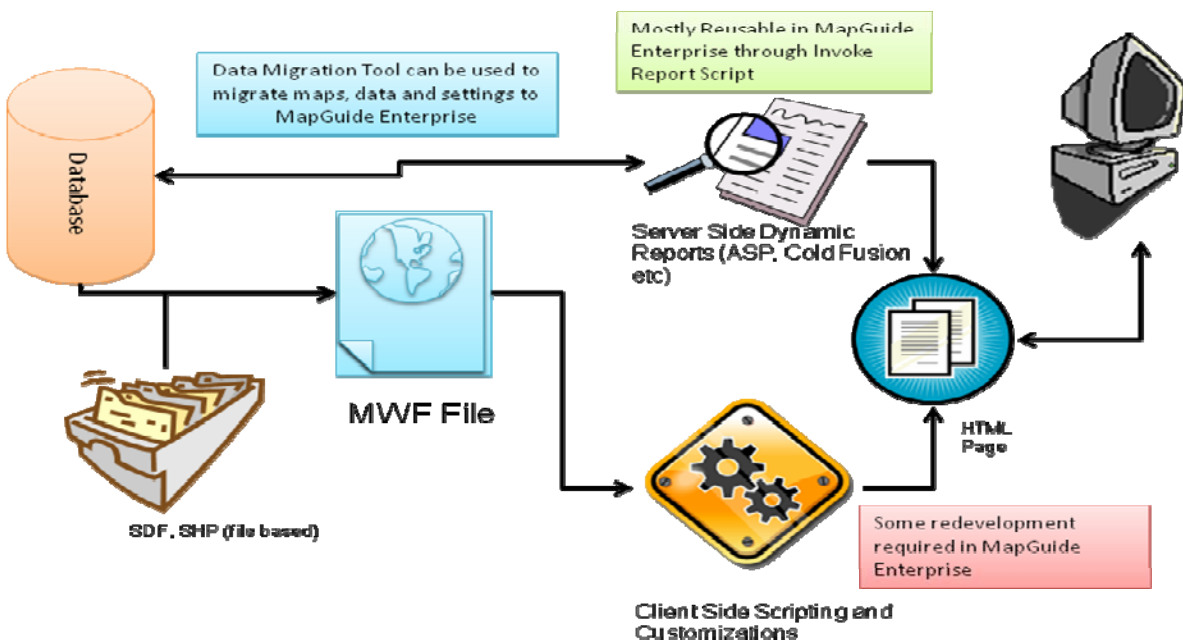


Autodesk MapGuide Studio を使用して作成されたマップ定義

移行計画

次の図は、一般的な Autodesk MapGuide 6.5 のアプリケーションを示しています。青で示されている構成要素は、データマイグレーションツールを使用して Autodesk MapGuide Enterprise に移行することができます。緑のセクションは、レポート要求を Autodesk MapGuide Enterprise から Autodesk MapGuide 6.5 の既存のレポートに転送するスクリプト(ソースコードは、「付録」を参照)によって、ほとんどを再利用することができます。赤のセクションは、Autodesk MapGuide Enterprise ではサーバサイド API に移行されたため、再開発する必要があります。

MapGuide 6.5 の一般的なアプリケーション –
マイグレーション ツール



Autodesk MapGuide 6.5 から Autodesk MapGuide Enterprise に移行する場合は、Autodesk MapGuide 6.5 で行われたカスタマイズに関する計画および再開発が多少なりとも必要になってきます。Autodesk MapGuide 6.5 以前の既存のアプリケーションを Autodesk MapGuide Enterprise に移行する際は、以降のセクションを段階的な指針として利用してください。

準備段階 — 計画と準備

プロセスの最初の段階では、アプリケーションの移行計画を立てる必要があります。この段階では、アプリケーションの目標を定める明確なプランを立て、作成する必要のあるカスタム機能をリストアップし、Web インタフェース設計を策定し、開発プラットフォームを選択し、サーバのハードウェア構成またはアーキテクチャを特定して、これらを記載した移行計画書を作成する必要があります。プロジェクト管理のために、役割分担、スケジュール、中間目標を含めることもできます。移行計画書を作成することで、開発者は大きな問題に遭遇することなく、既存のアプリケーションを成功裏に移行できます。

目標

想定される移行目標として、次のことが挙げられます。

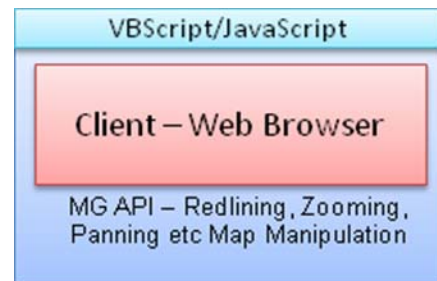
- 機能を拡張する
- プラグイン ビューアを除去する
- 空間データの更新を簡素化する
- Autodesk MapGuide Enterprise の組み込み機能を使用することにより、カスタム プログラミングを減らす
- サーバおよびアプリケーション ソフトウェアをアップグレードする
- アプリケーション開発プラットフォームを変更する

機能のリスト

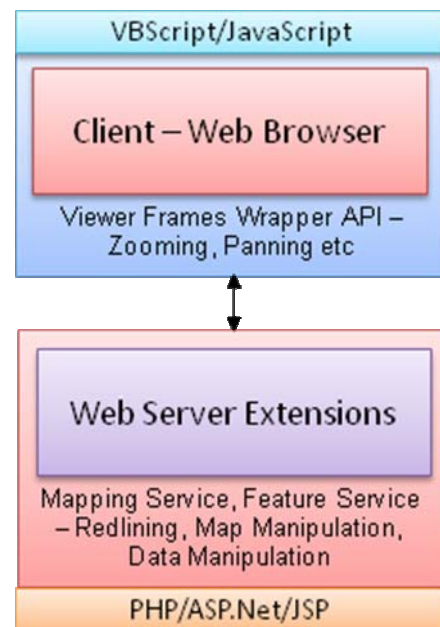
Autodesk MapGuide 6.5 サイトの重要な要素の 1 つは、高度なアプリケーションを開発するために Autodesk MapGuide 6.5 Viewer API を使用して実行されるカスタム プログラミングです。移行計画では、元のアプリケーションに存在していたすべてのカスタム プログラミング機能と、組み込む必要のある新しい機能について考慮する必要があります。既存機能のリストを作成し、それらの機能を Autodesk MapGuide Enterprise でどのようにして再現するのかを確認することは、アプリケーションの移行にどの程度の労力が必要になるのかを明確にする上で有用です。さらに、API を使用してプログラミングする必要のある機能の中には、Autodesk MapGuide Enterprise の標準機能として搭載されているものもあります。たとえば、多くの Autodesk MapGuide のアプリケーションには、[表示位置指定ズーム]というカスタム機能がありますが、この機能は、Studio を使用して作成できるズーム/検索機能に置き換えることができます。

Web インタフェース

Autodesk MapGuide アプリケーションの移行は、既存のインタフェースを再設計する機会を提供することにもなります。Autodesk MapGuide Enterprise には追加の機能が搭載されているため、Autodesk MapGuide サイト用に作成されたカスタム インタフェースのプログラミングと設計の多くは不要になります。たとえば、Autodesk MapGuide Enterprise で



Autodesk MapGuide 6.5 のカスタム機能



Autodesk MapGuide Enterprise のカスタム機能

は、[Properties Pane]という標準で搭載されている機能を使用することにより、ユーザはマップオブジェクトをクリックして属性情報を取得することができます。この機能は Autodesk MapGuide 6.5 には標準で搭載されていないため、多くの開発者はこれらの操作を実行するためにカスタム機能を作成していました。また開発者は、マップをインタラクティブに操作できるようにするために、カスタム ツールバーを作成していました。Autodesk MapGuide Enterprise では、これらのカスタム機能はタスク ペインに組み込むことができるため、カスタム ツールバーを作成する必要はありません。

場合によっては、開発者は、インタフェースをそのまま残した状態で、Autodesk MapGuide Enterprise を既存の Web 設計に組み込むこともできます。これは、ユーザが既存の設計に慣れている場合や、インタフェースを変更するとアプリケーションの使用法に悪影響を及ぼす場合に行うことがあります。そのような場合は、インタフェースを保持し、標準搭載の機能で置き換えられる可能性があるプログラミングは行わない方が賢明でしょう。

開発プラットフォームの選択

Autodesk MapGuide Enterprise は、Web 開発の実行に使用される最も一般的なプログラミング言語 .NET、PHP、JSP をサポートしています。開発者はこれらの言語を使用して、Autodesk MapGuide Enterprise に搭載されている強力な API にアクセスすることができます。クライアント サイドの API がブラウザおよび言語に依存する Autodesk MapGuide 6.5 とは異なり、Autodesk MapGuide Enterprise は、3 つのすべての Web API Extensions 間の完全互換性を提供します。つまり、アプリケーション開発者は任意の開発プラットフォームを使用でき、またあるアプリケーションから別のアプリケーションに移行しても、そのアプリケーションが同じように実行されるという確信を持つことができます。

どの開発プラットフォームを選択するかは、プログラマの好み、経験、スキルによって異なります。各プラットフォームには、それぞれ独自の長所と短所があります。

サーバの構成

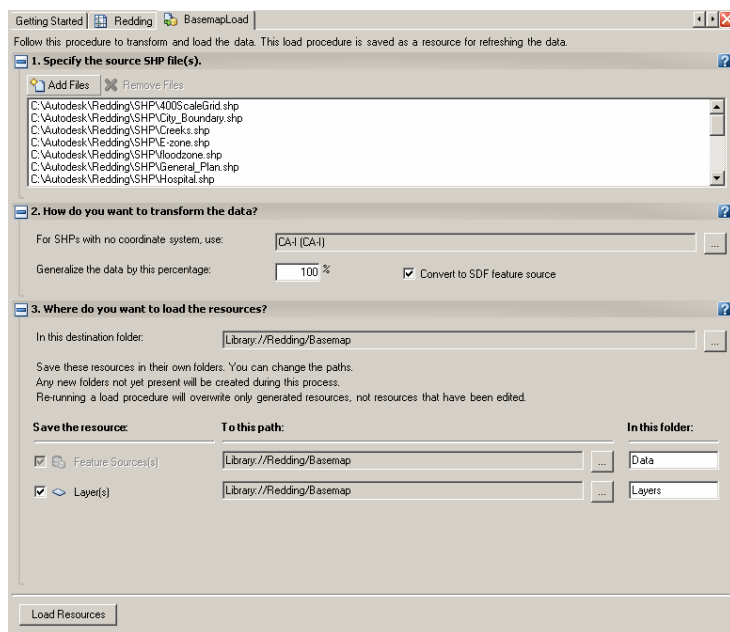
Autodesk MapGuide 6.5 のアプリケーションを移行するには、開発サーバを用意する必要があります。開発者は、通常、テスト用に少なくとも 1 つのステージング サーバまたは開発サーバを所有しています。Autodesk MapGuide Enterprise では、2 つのサーバを構成して、このソフトウェアの専用サーバとして使用することをお勧めします。一方のサーバはステージング用として使用し、もう一方は開発サーバとして使用します。Autodesk MapGuide Enterprise を Autodesk MapGuide と同じサーバ上にインストールすることもできますが、お勧めできません。

Autodesk MapGuide Enterprise の実行に必要なハードウェアは、組織のネットワーク インフラ、開発プラットフォームの規格、開発者の能力などさまざまな要因によって異なります。

Autodesk MapGuide Enterprise は、Microsoft® Windows® オペレーティング システムだけでなく Linux 上でも実行できるようになったため、組織の IT スタッフの役割が意思決定プロセスにおいてより重要になります。

第 1 段階 — データの設定とロード

移行の第 1 段階は、空間データと表形式データを設定して Autodesk MapGuide Enterprise にロードし、外部データベースに接続することです。Autodesk MapGuide Enterprise では、データソースの設定は Studio を使用して行います。Autodesk MapGuide 6.5 と同様、編成されたデータを維持するためには、編成されたデータ構造を作成することが非常に重要です。Autodesk MapGuide Enterprise では、サーバのディスクドライブにファイル フォルダを作成するのではなく、Studio



を使用してフォルダを作成します。ファイル フォルダは、既存の Autodesk MapGuide 構造と同じ構造にして、一貫性を維持することをお勧めします。しかし、既存の構造が十分に編成されていない場合は、この移行プロセスが、ファイル構造とデータを適切に編成する機会になります。

フォルダを作成したら、Studio を使用して、空間データと表形式データを Autodesk MapGuide Enterprise に読み込むことができます。Autodesk MapGuide 6.5 とは異なり、空間データ ソースをアドミニストレータで作成する必要はありません。データがロードされると、Studio がデータ ソースへの接続を確立します。

Autodesk MapGuide 6.5 と Autodesk MapGuide Enterprise のもう 1 つの大きな違いは、Autodesk MapGuide Enterprise では、データをロードするときに複雑なデータ更新ルーチンを作成する必要がないという点です。Autodesk MapGuide サイトの多くの管理者は、空間データを更新するプロセスを定義するために SDF Loader を使用してバッチ ファイルを作成します。このプロセスは、自動または手動で実行されます。ほとんどの場合、管理者は更新ルーチンを作成せずに、プロセス全体を手動で実行します。Autodesk MapGuide Enterprise では、データがロードされると、ロード プロシージャが作成されます。データの更新が必要ときに、ロード プロシージャをすばやく簡単に実行できるように、ロード プロシージャに複数のデータ セットを含めることができます。

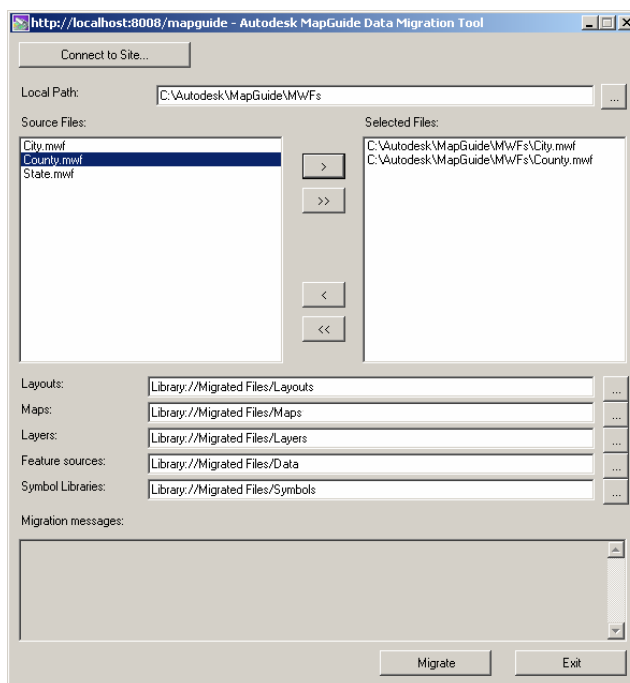
Autodesk MapGuide Studio でのロード プロシージャ

第 2 段階 — マップ画層の作成/マップの作成

移行の次の段階は、Autodesk MapGuide Enterprise のマップで使用できる画層を作成することです。このプロセスは、Autodesk MapGuide および Autodesk MapGuide Author で使用されていたこれまでの手順とは微妙に異なります。Autodesk MapGuide では、マップに画層を追加する必要がある場合は、Autodesk MapGuide Author を起動して、MWF ファイル内に画層を作成することになります。アプリケーションに複数の MWF ファイルがある場合は、各 MWF ファイルに画層を作成するか、コピーする必要があります。また、各 MWF ファイル内の各画層それぞれに書式を設定しなければなりません。このプロセスでは、2 つの別の画層が同じ目的に使用される場合でも、異なる書式が設定されてしまうという可能性が生じます。Autodesk MapGuide Enterprise では、画層を作成し、主題やツールチップなど画層の外観を一度だけ定義すれば、その後はさまざまなマップ定義内の画層定義を使用することになります。Autodesk MapGuide Enterprise のこの新しい機能により、画層を何度も作成しなければならないような重複する作業を排除することができます。

もちろん、既存のアプリケーションを Autodesk MapGuide Enterprise に移行する以上は、ユーザは Autodesk MapGuide のアプリケーションに存在する画層のスタイルを複製したいと思うでしょう。しかし、Autodesk MapGuide Enterprise には非常に優れたレンダリング機能が搭載されているため、ユーザは以前のアプリケーションの画層の書式設定を保持するかどうか決断することが求められます。

一度マップ画層を定義したら、マップを作成することができます。Studio では、マップの画層は既に書式が設定されています。マップは、基本的には、定義されているデータと画層のビューとして機能します。このアプローチの利点は、画



MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

層の書式設定や表示を変更しなければならない場合に、画層を一度だけ変更すれば、その画層を使用しているあらゆる場所に反映できることです。これに対して Autodesk MapGuide 6.5 では、画層が複数の MWF ファイル上に表示されている場合、各画層を各 MWF ファイル内で個別に変更しなければなりません。

Autodesk MapGuide の多くの既存アプリケーションには、情報に関する画層と MWF ファイルが多数含まれているため、画層を再作成するプロセスに時間がかかる可能性があります。既存の MWF ファイルの情報を Autodesk MapGuide Enterprise に移行するには、データマイグレーション ツール(DMT)を Studio とともに使用します。DMT を使用すると、MWF ファイルにアクセスし、その情報を Autodesk MapGuide Enterprise Server に取り込むことができます。

次の表に、Autodesk MapGuide 6.5 のファイルとその要素が、Autodesk MapGuide Enterprise のリソースにどのようにして移行されるのかを示します。

注: すべての Autodesk MapGuide Studio ユーザは、このデータマイグレーション ツールを <http://www.autodesk.com/mapguideenterprise> から無償でダウンロードすることができます。

MWF/MWX	
Web レイアウト	既定値の Web レイアウトは、MWF ポップアップ メニューが右クリックメニューに変換される場所に作成されます。
マップ定義	マップ ウィンドウのプロパティは、マップ定義のプロパティに変換されます。
画層定義	以下の「MLF」の項目を参照してください。 注: リモート MapGuide サーバを参照している画層は移行されません。
シンボル ライブラリ	MWF API シンボルおよびポイント画層シンボルから作成されます。
フィーチャソース	以下の「MLF」および「UDL」の項目を参照してください。
図面ソース	以下の「MLF」の項目を参照してください。

MLF	
ダイナミック テキスト画層、ポリライン画層、ポリゴン画層	<p>フィーチャ画層の定義</p> <ul style="list-style-type: none"> テキスト画層は、ポイントのスタイル設定を作成します。
フィーチャ画層の定義	<p>フィーチャソース</p> <ul style="list-style-type: none"> SHP SDP 用の SHP プロバイダ フィーチャソース SDF 2.x SDP 用の SDF3 プロバイダ フィーチャソース DWG™ 用の SDF3 プロバイダ フィーチャソース Oracle SDP 用の Oracle プロバイダ フィーチャソース Microsoft SQL SDP 用の Microsoft SQL Server プロバイダ フィーチャソース <p>属性ソース</p> <p>2 次テーブルごとに、属性ソースが作成され、フィーチャソースにジョイントが追加されます。</p>

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

MLF	
	<ul style="list-style-type: none"> Microsoft Access OLE DB 用の ODBC プロバイダ フィーチャ ソース SHP SDP 用の SHP プロバイダ フィーチャ ソース SDF 2.x SDP 用の SDF3 プロバイダ フィーチャ ソース DWG 用の SDF3 プロバイダ フィーチャ ソース Oracle SDP 用の Oracle プロバイダ フィーチャ ソース Microsoft SQL OLE DB 用の Microsoft SQL Server プロバイダ フィーチャ ソース
ダイナミック ポイント 画層	<p>フィーチャ画層の定義</p> <ul style="list-style-type: none"> 上記に加えて、次が追加されます。 Microsoft Access SDP 用の ODBC プロバイダ フィーチャ ソース <p>2 次テーブル用の属性ソース</p> <ul style="list-style-type: none"> 上記と同様 <p>シンボル ライブラリ</p>
ダイナミック ラスター 画層	<p>ラスター画層の定義</p> <p>ラスター プロバイダ フィーチャ ソース</p> <ul style="list-style-type: none"> TIFF ベースのフィーチャ ソースに変換された JPEG(.jpg) TIFF ベースのフィーチャ ソースに変換された Microsoft Windows ビットマップ(.bmp) TIFF ベースのフィーチャ ソースに変換された TrueVision Targa 2.0 形式(.tga) TIFF ベースのフィーチャ ソースに変換された ポータブル ネットワーク グラフィックス(.png) TIFF ベースのフィーチャ ソースに変換された タグ付きイメージ ファイル形式(.tif) TIFF ベースのフィーチャ ソースに変換された CALS MIL-R-28002A タイプ 1(.cal) ECW ベースのフィーチャ ソースに変換された Enhanced Compression Wavelet(.ecw) MrSID ベースのフィーチャ ソースに変換された MrSID®(.sid)

MLF	
ダイナミック DWF 画層	<ul style="list-style-type: none"> 図面画層の定義 図面ソース
ダイナミック DWG 画層	<ul style="list-style-type: none"> 図面画層の定義 図面ソース
スタティック画層	<p>次のデータ ソース例外を使用して、上記と同様に移行されます。</p> <ul style="list-style-type: none"> SDF 3 プロバイダに変換されたポイント SDF 3 プロバイダに変換されたテキスト SDF 3 プロバイダに変換されたポリライン

MLF	
	<ul style="list-style-type: none"> • SDF 3 プロバイダに変換されたポリゴン • TIFF ベースのラスター プロバイダに変換されたラスター • 図面ソースに変換された DWG • 図面ソースに変換された DWF
バッファ画層	<ul style="list-style-type: none"> • 図面画層の定義 • 図面ソース
赤入れ画層	<ul style="list-style-type: none"> • 図面画層の定義 • 図面ソース

UDL	
	<ul style="list-style-type: none"> • SHP SDP 用の SHP プロバイダ フィーチャ ソース • SDF SDP 用の SDF プロバイダ フィーチャ ソース • Microsoft Access OLE DB 用の ODBC プロバイダ フィーチャ ソース

DMT を使用すると情報の基本的な変換を実行できるため、開発者はマップの作成を最初から始めなくてもよいことに注目してください。しかし、情報が変換された後も、書式を設定しなければならない場合があります。実際には、ほとんどの開発者は、Autodesk MapGuide Enterprise の新機能を使用してマップの書式を強化することを望むでしょう。

第 3 段階 — アプリケーションの移行

マップを作成した後は、開発プラットフォームおよびサーバ アーキテクチャの選択とセットアップが完了していれば、開発者はアプリケーションの移行プロセスを開始することができます。このプロセスが、移行する上で最も時間のかかる部分です。移行計画で確認した機能のリストは、開発プロセスにとって重要です。Autodesk MapGuide Enterprise に標準で搭載されている機能を使用することで除去できる機能をこのリストに反映するために、リストを絞り込む必要があります。

移行するべき最も重要な機能は、オブジェクトの選択、サーバとのインタラクティブ操作、[表示位置指定ズーム]機能、レポート機能です。これらの機能は、Autodesk MapGuide のアプリケーションで最もよく使用される機能です。

開発者は 次の 2 つのアプローチのいずれかを採用して、サイトの実際のコードを記述することができます。最初から始めるか全体のコード ベースを記述して、新しい設計または従来の設計をアプリケーションに組み込むことができます。あるいは、既存のアプリケーションを複製し、古いコードを削除したり新しいコードを追加したりすることによって修正することもできます。どちらのアプローチが効果的なのかは開発者次第ですが、API が完全に異なっていることと、従来の API 呼び出しはいずれも再利用できないことから、最初から始めた方がより効率的かもしれません。Autodesk MapGuide で使用するために作成されたレポートは、Invoke Report スクリプト(ソース コードは、「付録」を参照)によって Autodesk MapGuide Enterprise で再利用することができます。

Web 層 API およびサンプル コードの詳細については、本ドキュメントのコード例のセクションに記載しています。

第 4 段階 — テストとフィードバック

新しいサイトが完成した後は、開発者が開発時に機能のテストを行ってれば、ベータ テスト用にアプリケーションを配布することができます。ユーザ グループは従来のアプリケーションの内部動作に慣れているので、新しいサイトの使用方法のトレーニングを実施したり、アプリケーションの新しい機能や相違点に関する情報を知らせたりする必要があります。また、問題点が開発者に明確に伝わるように、ベータ ユーザにフィードバック フォームを提供する必要もあります。

第 5 段階 — 最終的な変更

次の段階は、ベータ テスト ユーザからのフィードバックを整理して理解することです。このとき、フィードバックが明確に伝わるように、ベータ テスト ユーザとミーティングを持たなければならない場合もあります。フィードバック項目のリストがまとまったら、ベータ テスト ユーザはアプリケーションの使用を中止して、開発者がアプリケーションをアップデートできるようにする必要があります。再度、開発者はテストを行い、問題点を修正することになります。

第 6 段階 — 配置

最終的な変更を加えたら、サイトをライブ サーバ上に配置します。新しいサイトに加えて従来の MapGuide サイトも維持するのか、あるいは従来のサイトを置き換えるのかは、個々の組織によって異なります。新しいサイトを立ち上げる前に、アプリケーションの新しい機能についてユーザにトレーニングを実施しなければならない場合もあります。ユーザが新しいサイトでの作業を開始したら、フィードバックを収集するための計画を立てる必要があります。ステージング サーバ上でさらに開発を進め、必要に応じてライブ サーバに配置することができます。従来のサイトの運用を中止する期限をすべてのユーザに知らせる必要もあります。

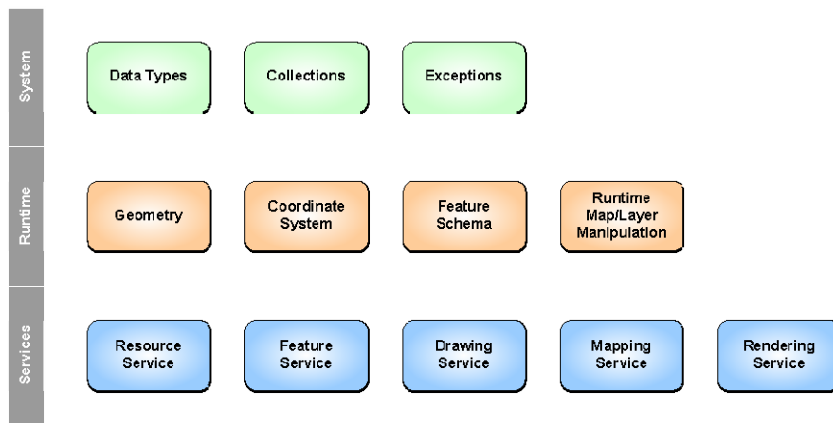
Web 層 API の解説

MapGuide を Microsoft Windows Server で実行するのか Red Hat® Linux Server で実行するのかを選択できる柔軟性により、組織は最も適切なオペレーティング システムと開発ツールを選択することができます。Web 層 API は、両方のオペレーティング システム間で完全に整合性があります。

MapGuide の API

Autodesk MapGuide Enterprise を使用する開発者は、複数の API オブジェクトを利用することができます。

Autodesk MapGuide Studio には .NET API が付属しており、ユーザはこれを使用して、Studio のコンポーネントを使用したカスタム アプリケーションを構築することができます。開発者がサーバ上のフィーチャ ジオメトリのあらゆる側面にアクセスしたい場合は、FDO API を使用して、フィーチャを変更、操作、再作成、分析します。



Web Server Extension API モデル

使用できる API は次のとおりです。

サービス	機能
リソース サービス	リポジトリとリソースを操作することができます。また、データの操作とロードに使用することもできます。

フィーチャ サービス	FDO プロバイダにアクセスすることができます。
マッピング サービス	マップおよびマップ内の画層にアクセスすることができます。
図面サービス	図面ソースへの下位アクセスを行うことができます。DWF ファイルを操作することもできます。
レンダリング サービス	マップをビットマップ イメージにレンダリングします。このイメージは、通常、HTML ビューアで表示するために使用されます。
座標系	座標系の変換をサポートします。
ジオメトリ	ジオメトリ オブジェクトを操作します。
サイト サービス	ユーザ、グループ、ユーザ セッションを設定します。

リソースの定義、設定、画層表示の設定など、Autodesk MapGuide と Autodesk MapGuide Studio の大部分は XML で動作するため、開発の多くは、API のインスタンス化とコントロールが可能な XML の DOM(Document Object Model)を使用できるアプリケーションで行う必要があります。

次の 3 つの Web 開発環境は 3 つのマップエージェントに相当し、既に Autodesk MapGuide 用に開発され、開発例が製品に同梱されています。

- PHP 5
- JAVA(JSP)
- ASP.NET

PHP 5 を使用して開発する

PHP は、次のような多くのプラットフォーム上で実行されるサーバ サイドの Web ベースのスクリプト言語です。

- Linux(Red Hat、Fedora™ など)
- Microsoft Windows
- Unix®(Solaris® など)
- Mac OS® X

PHP はオープン ソースであるため、組織は購入する必要がありません。PHP は絶えず改良とデバッグが行われているため、堅固さにおいては信頼性があります。PHP は、Apache、Microsoft Internet Information Server(IIS)、Netscape®、O'Reilly といったほとんどの Web サーバ上で動作します。Autodesk MapGuide Enterprise の Site Administrator は PHP で記述されています。この管理ツールは、Web Extensions を使用してインストールされます。

PHP マップエージェントをインストールした場合は、これを使用することにより、PHP を用いて Autodesk MapGuide Enterprise API のさまざまな機能を呼び出すことができます。

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

PHP 内で Autodesk MapGuide Enterprise からマップ セッションにアクセスする例を次に示します。

```
<?php

include 'AppConstants.php';

MgInitializeWebTier ($configFilePath); //path to the webconfig.ini file

$userInfo = new MgUserInformation("Administrator", "admin");

$site = new MgSite();

$site->Open($userInfo);

$HTTP_SESSION_VARS['MgSessionId'] = $site->CreateSession();

$mapDefinition = "Library://DUBLIN CA/5. MAPS/Dublin
Map.MapDefinition";

$webLayout = "Library://DUBLIN CA/6. WEB LAYOUTS/Dublin Web
Layout.WebLayout";

?>

<frameset rows="110,*" frameborder="NO" border="0"
framespacing="0">

<frame src="Title.php?AppName=Sample Application"
name="TitleFrame" scrolling="NO" noresize>

<frame src="/mapviewerphp/dwfviewer.php?SESSION=<?=
$HTTP_SESSION_VARS['MgSessionId'] ?>&WEBLAYOUT=<?=
$webLayout ?>"
name="ViewerFrame">

</frameset>
```

注: このコード例の *Library://DUBLIN CA/6. WEB LAYOUTS/Dublin Web Layout.WebLayout* が指定されている箇所では、サーバ上にあるライブラリを使用しています。これは、Autodesk MapGuide Studio で作成された Web レイアウトです。

PHP によるマップ インスタンスの作成例:

```
$map = new MgMap();

$map->Open($resourceService, 'Dublin Map');
```

Autodesk MapGuide Enterprise の API は強力であるため、PHP を使用してデータのアップロード、FDO へのリンク、新しい情報を含む画層の作成を行い、その後マップを作成して、Web レイアウトにそのマップを追加することができます。Autodesk MapGuide Studio で見られるプロセス全体は、API を使用して自動化することもできます。

Java を使用して開発する

MapGuide Web Server Extensions を使用してインストールされるマップエージェントが、*mapagent.jsp* と呼ばれる Java Server Page(JSP)です。Sun Microsystems 社によって開発された JSP は、サーバに要求を送信する Web サーバ スクリプト エージェントです。JSP は、アプレット(クライアント上で実行される java アプリケーション)としてではなく、サーブレットとしてサーバサイドでコンパイルされます。JSP は、サーバ上にある既存のサーブレットや Java サーブレットと通信することができます。JSP ページを実行するには、Java を実行できる Web サーバが必要になります。最も一般的なのは Apache Tomcat サーバです。Tomcat もオープン ソース アプリケーションで、<http://tomcat.apache.org> から無償でダウンロードでき、Linux または Windows オペレーティング システムのどちらにもインストールすることができます。Windows を使用している場合や、別の Web サーバと同時に Tomcat を実行している場合は、Tomcat をインストールするときに別のポートを選択しなければならない場合があります。たとえば、Autodesk MapGuide Enterprise の Web Extensions をインストールし、同時に Java マップエージェントを使用することもできます。Java マップエージェントは、<http://hogwarts:8080/JavaMapAgent/MapAgent.jsp> から入手することができます。

JSP によるマップ インスタンスの作成例:

```
<%@ page import="org.osgeo.mapguide.*" %>

MgMap map = new MgMap();

map.Open(resourceService, "Dublin Map");
```

Autodesk MapGuide でのすべての API 呼び出しは、Java Web Server Extension でも利用することができます。

ASP .NET を使用して開発する

Active Server Pages(ASP) .NET は、Microsoft .NET Framework に基づいています。このプログラミング フレームワークは、Microsoft Windows オペレーティング システム上にインストールされます。PHP や Java とは異なり、.NET は Microsoft IIS でのみ動作します。.NET が提供する Microsoft Web サービスを使用すると、Microsoft 環境で多くのサーバサイド カスタム アプリケーションを作成することができます。ASP.NET は、次のような多くのスクリプト言語でプログラミングすることができます。

- C#(C シャープ)
- Visual Basic (VB)
- JScript

ASP.NET を使用する開発者は、通常、C# または VB.NET を選択します。Microsoft® Visual Studio® .NET または Microsoft® Visual Basic® .NET のような開発ツールを使用すると、統合開発環境(IDE)の WYSIWYG(What You See Is What You Get)機能を使用して、複雑なアプリケーションを簡単に統合することができます。

MapGuide Web Server Extensions には、同様に ASP.NET マップエージェントが付属しています。Web Server Extensions にバンドルされているサンプル ファイルは C# .NET で開発されていますが、ASP.NET と互換性のある .NET スクリプト言語であればいずれもアプリケーションの開発に使用することができます。

ASP.NET によるマップ インスタンスの作成例:

```
<%@ Import Namespace="OSGEO.MapGuide" %>

MgMap map = new MgMap();
```

```
map.Open(resourceService, "Dublin Map");
```

コード例

注: Autodesk MapGuide Enterprise の開発者用ガイドは、コード サンプルの優れたリソースです。製品をインストールすることにより、開発手法を利用することができます。

多くのアプリケーションでカスタム インタフェースが必要になり、また Studio で利用できる機能をさらに拡張する必要があります。Autodesk MapGuide Enterprise では、これらは API (ASP.NET、PHP、JSP) の Web 層で作成されます。これらの API には、MapGuide API の関数とメソッドすべてが含まれ、またさらに拡張されています。実際のところ、そのほとんどにすぐに慣れることでしょう。MapGuide API と 空間アプリケーション サーバ API の使用上の主な違いは、次のとおりです。

- 空間アプリケーション サーバ API は、サーバサイド サービスに分類されます(本ドキュメント前述の「Web 層 API の解説」を参照)。マップに対する操作を行うたびに、アプリケーションの状態を保存するためには、空間アプリケーション サーバの Web 層に通知する必要があります。これは、状態を格納する ResourceService オブジェクトを作成することにより行うことができます。
- ビューアは変更通知を受ける必要があります(これは、変更が実際にはクライアント上でなくサーバ上で行われるためです)。これを行うための最も一般的な方法は、操作が実行されるときにページ(たとえば、組み込まれたビューア)を更新する DHTML OnLoad() イベントを使用することです。
- Autodesk MapGuide では、API はブラウザ クライアントに組み込まれた MWF オブジェクトを参照します。空間アプリケーション サーバでは、API はサーバ上のサイトリソースを参照します。このように、API クラス ライブラリにアクセスする方法がまったく異なります。

次の例は、サイト サービスとマッピング サービスに的を絞っています。これらのサービスの中でも、MgSite、MgSiteConnection、MgWebActions、MgLayer、MgMap の API クラスに関する例を示しています。

- Web ページにビューアを組み込む
- ユーザ情報とセッション情報を作成する: サイト サービス API
- クラスの MgMap ライブラリを使用してマップ情報にアクセスする
- 画層の表示を列挙、更新、変更する
- デジタイズ、および赤入れする
- Autodesk MapGuide 6.5 ベースの既存のレポートを呼び出す

ソフトウェアには、その他のコード サンプルも付属しています。

Web ページにビューアを組み込む

Web ページにビューアを組み込む方法はいくつかありますが、最も一般的な方法は、Web 層の仮想ディレクトリにインストールされる *dwfviewer.aspx* を呼び出し、これに Web レイアウトの名前を渡すことです。このアプローチを使用すると、表示をカスタマイズしたり、カスタム コマンドを追加したりすることができます。また、クライアントを介してユーザによるダイナミックなインタラクティブ操作を行うこともできます。ユーザが表示を変更すると、DWF Viewer は更新されたメタデータを空間アプリケーション サーバから要求し、また、表示するために更新されたグラフィックスを要求します。

次のコードは、2 つの ASP.NET ページを呼び出しています。一方のページにはタイトル グラフィックスが含まれ、もう一方には Viewer Definition .aspx ページへのポインタが含まれています。このページには、どの Web レイアウトを参照するのかを宣言するパラメータが渡されず、参考までに、この機能を実行する従来の方法も同様に記載します。

ASP.NET (サーバ)	<pre><frameset rows="120, *"> <frame src="Title.aspx?AppName=Sample Application"> <frame src="http://localhost/mapguide/mapviewernet/dwfvviewer.aspx?WEBLAYOUT=Library:// <YOUR DIRECTORY>/Layouts/<YOUR LAYOUT>.WebLayout" name="ViewerFrame"> </frameset></pre>
MapGuide 6.5 (クライアント)	<pre><OBJECT ID="map" WIDTH="100%" HEIGHT="100%" CLASSID="CLSID:62789780-B744-11D0-986B-00609731A21D"> <PARAM NAME="URL" VALUE="http://localhost/example.mwf"> </OBJECT</pre>

ユーザ情報とセッション情報を作成する: サイト サービス API

前出の単純なコードを使用すると、開発者は既定のツールを使用することにより、空間アプリケーション サーバの Web レイアウトを表示し、マップをインタラクティブに操作することができます。しかし、API を使用してマップをインタラクティブに操作するためには、ユーザ情報とセッション変数を設定する必要があります。空間アプリケーション サーバはサーバ上の Web 層 API を使用するため、どのアプリケーションであっても、アプリケーションの状態を反映するセッションデータを格納して、サーバとビューア間で受け渡しを行う必要があります。これは、サイト サービス API を使用してアクセスできる空間アプリケーション サーバ セッションとセッション変数を使用して実行されます。

次のコード行は、サーバ サイトを確立し、後で参照するためのセッション ID を作成しています。

ASP.NET (サーバ)	<pre><%@ Page language="c#" %> <%@ Import Namespace="OSGEO.MapGuide" %> <% String mapDefinition="Library://<Path to your map resource>/<your map>.MapDefinition"; String webLayout=" Library://<Path to your Layout resource>/<your Web Layout>.WebLayout"; String SessionId=""; MapGuideApi.MgInitializeWebTier(configPath); //configPath is the path to the webconfig.ini file MgUserInformation userInfo = new MgUserInformation("UserName","Password");</pre>
---------------	--

	<pre>MgSite = new MgSite(); Site.Open(userInfo); sessionId = site.CreateSession() %></pre>
MapGuide 6.5(クライアント)	同様のメソッドはありません。

このコードについては、多少の説明が必要です。

- シンボル `<% ..%>` は、コード ブロックの開始と終了を示しています。
- コードの最初の 2 行では、使用する言語(この場合は、C#)と、Import メソッドを使用してアクセスされるクラスのライブラリを定義しています。このコードで、すべての API ライブラリ、関連クラス、メソッドがアプリケーションに読み込まれます('Mg' ライブラリ)。
- 次に、使用されるマップおよび Web レイアウトのリソースの位置に関する情報を格納する変数と、サイトを識別する変数の 3 つの変数を作成しています。
- 次に、Web 層を初期化しています。
- 次に、与えられたユーザ情報を使用してユーザを定義し、サイトを作成して開いています。
- 最後に、ユーザの一意のセッション識別子を作成しています。これで、サーバは、この変数を使用してセッションを追跡できるようになります。

前出のビューアと Web レイアウトを開くためのコードは、既定のツールを使用することによりマップをインタラクティブに操作できますが、API を使用した場合は操作できないことを示していました。今では、このコードで定義したサイトとセッションの情報を使用することにより、次のようにセッション ID 変数を追加するだけで、どのアプリケーションにもビューアを含めることができます。

```
<frame src="http://localhost/mapguide/mapviewernet/ajaxiewer.aspx?SESSION=<%= SessionId %>&WEBLAYOUT=<%= webLayout %>"
```

クラスの MgMap ライブラリを使用してマップ情報にアクセスする

次の例は、マップの名前とその範囲を取得して表示するために、Autodesk MapGuide Enterprise の Web 層 API にアクセスして、マップにアクセスし、マップに関する単純な情報を取得する方法を示しています。このコードは、呼び出し関数または呼び出しスクリプトからこのスクリプトにセッション ID 変数が渡されることを前提としています。

ASP.NET(サーバ)	<pre><%@ Page language="c#" %> <%@ Import Namespace="OSGEO.MapGuide" %> <% String MgSessionId = Request.QueryString["SESSION"]; MapGuideApi.MgI initializeWebTier(configPath) //configPath is the path to the webconfig.ini file</pre>
--------------	---

	<pre> MgUserInfo userInfo = new awUserInfo(MgSessionId); MgSiteConnection siteConnection = new MgSiteConnection(); siteConnection.open(userInfo); MgResourceService resourceService = siteConnection.CreateService(MgServiceType.ResourceService); MgMap map = new MgMap(); map.Open(resourceService, "<Map Name>"); MgEnvelope envelope = map.GetMapExtent(); Double lowerX = envelope.GetLowerLeftCoordinates().GetX(); Double lowerY = envelope.GetLowerLeftCoordinates().GetY(); String mapName = map.GetName(); %> </pre>
MapGuide 6.5(クライアント)	<pre> <SCRIPT LANGUAGE="JAVASCRIPT"> function getMap() { if (navigator.appName == "Netscape") return parent.mapframe.document.map; else return parent.mapframe.map; } function getCoordinates(){ var map = getMap(); var ext = map.getMapExtent(false,true); var lowerX = ext.getMinX(); var lowerY = ext.getMinY(); var mapName = map.getName(); } </SCRIPT> </pre>

画層の表示を列挙、更新、変更する

マップ内のすべての画層名を取得するには、MgLayerCollection クラスにアクセスし、次に画層ごとに MgLayer クラスにアクセスします。

ASP.NET(サーバ)	<pre> <% MgLayerCollection LayerCol = map.GetLayers(); MgLayer layer = null; String layername = ""; for (int i = 0; i < LayerCol.GetCount(); i++) { layer = LayerCol.GetItem(i); layername = layer.GetName(); } %> </pre>
MapGuide 6.5(クライアント)	<pre> <SCRIPT LANGUAGE="JAVASCRIPT"> function getMap() { if (navigator.appName == "Netscape") return parent.mapframe.document.map; else return parent.mapframe.map; } function getLayerNames(){ var map = getMap(); var layers = map.getMapLayersEx(); for (var i = 0; i < layers.size(); i++) { var layer = layers.item(i); var layername = layer.getName() ; } } </SCRIPT> </pre>

次の例を続けると、画層の表示/非表示の切り替えができます。

ASP.NET(サーバ)	<pre> <% MgLayerCollection LayerCol = map.GetLayers(); MgLayer layer = LayerCol.GetItem("Layer Name"); if (layer.IsVisible()){ layer.SetVisible(false); } %> </pre>
--------------	---

	<pre> } else{ layer.SetVisible(true); } } %> </pre>
MapGuide 6.5 (クライアント)	<pre> <SCRIPT LANGUAGE="JAVASCRIPT"> function getMap() { if (navigator.appName == "Netscape") return parent.mapframe.document.map; else return parent.mapframe.map; } function layerToggle(name) { var map = getMap(); var layer = map.getMapLayer(name); layer.setVisibility(!layer.getVisibility()); map.refresh(); } </SCRIPT> </pre>

デジタイズする

Autodesk MapGuide Enterprise のビューアには、ユーザがアプリケーションをインタラクティブに操作できるようにするラッパー API が搭載されています。デジタイズと赤入れは、通常はユーザ入力を伴う機能であり、このために提供されている API を利用することができます。

Autodesk MapGuide Enterprise Viewer API には、ユーザ入力をデジタイズするための関数がいくつか含まれています。次の例は、ユーザがボタンをクリックして、ポイントをデジタイズする場合の例です。

```
<input type="button" value=" Point " onclick="DigitizePoint();">
```

the script calls the JavaScript function:

```
function DigitizePoint() {
parent.parent.mapFrame.DigitizePoint(OnPointDigitized);
}
```

このコードは、マップ フレーム内で Viewer API の DigitizePoint() メソッドを順番に呼び出しています。また、現在のスクリプト内で定義されているコールバック関数の名前 OnPointDigitized も渡しています。DigitizePoint() は、ポイントがデジタイズされた後にこの関数を呼び出し、ポイントのデジタイズされた座標を渡しています。このコールバック関数を使用して、デジタイズされ

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

た座標を希望どおりに処理することができます。次の例のスクリプトは、タスク ペインに座標を単純に表示します。

```
function OnPointDigitized(point) {
  ShowResults("X: " + point.X + ", Y: " + point.Y);
}
```

<p>MapGuide Enterprise Viewer API(クライアント)</p>	<pre><Script Language="Javascript"> function digitizeMyLine(myLineDigitizeHandler) { parent.MapFrame.digitizeLineString(); } Function myLineDigitizeHandler(oLinestring) { // LineString Object Var sPoints; for (i=0;i<oLineString.count();i++) { //generate point list string sPoints = oLineString.point(i).X + "," + oLineString.point(i).Y + ","; } //submit points to server side script to generate redline layer and feature source var params = new Array("points",sPoints); parent.FormFrame.submit("../CreateRedline.aspx", params, "FormFrame"); } </Script></pre>
<p>MapGuide 6.5(クライアント)</p>	<pre><SCRIPT LANGUAGE="JAVASCRIPT"> function digitizeMyLine() {</pre>

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

	<pre>getMap().digitizePolyline(); } </SCRIPT> <SCRIPT LANGUAGE="VBSCRIPT"> Sub onDigitizePolyline(map, oPoints) //create Redline Layer on client Map End sub </SCRIPT></pre>

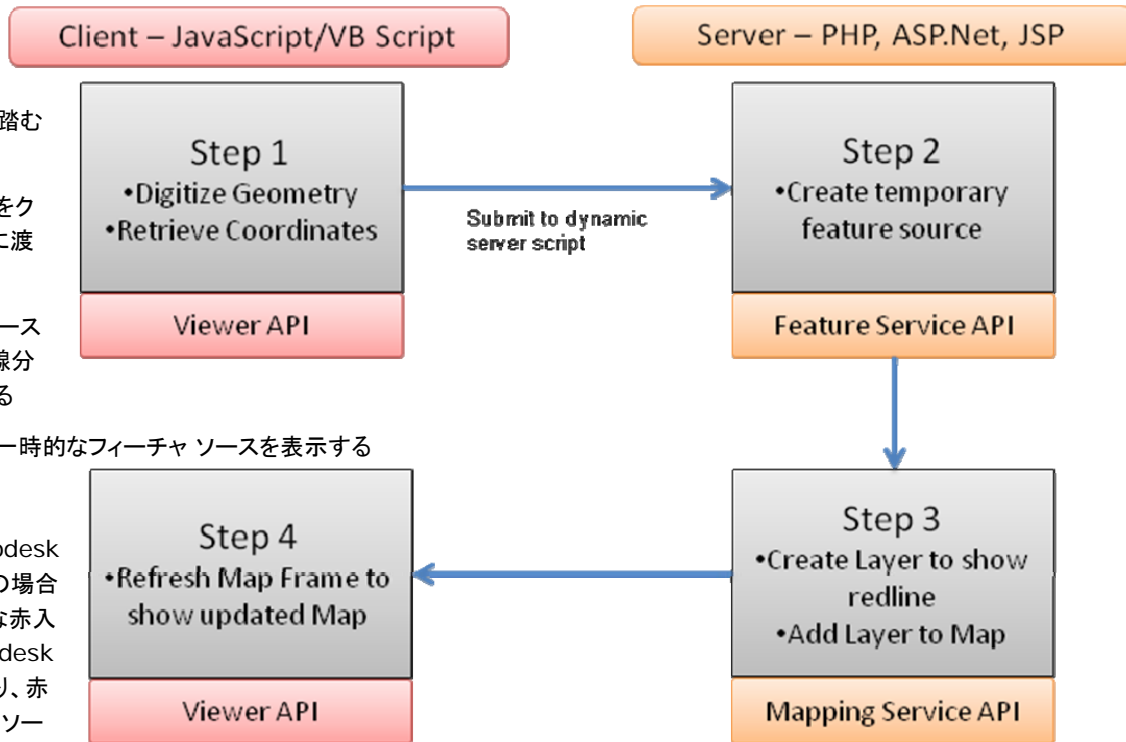
赤入れする

Autodesk MapGuide Enterprise で赤入れするためには、3 つの段階を踏む必要があります。

1. デジタイズされた座標をクライアントからサーバに渡す
2. 一時的なフィーチャ ソースを作成する。これは、線分を描くために使用される
3. 画層を作成して、その一時的なフィーチャ ソースを表示する

赤入れ処理によって、Autodesk MapGuide Enterprise の場合のクライアント上に一時的な赤入れ画層が作成される Autodesk MapGuide 6.5 とは異なり、赤入れは一時的なフィーチャ ソースとしてサーバ上に作成され、マップを更新すると、新しく作成された赤入れフィーチャ ソースが反映されます。

赤入れのサンプル コードは、付録に記載しています。



Autodesk MapGuide Enterprise での赤入れフロー チャート

Autodesk MapGuide 6.5 ベースの既存のレポートを呼び出す

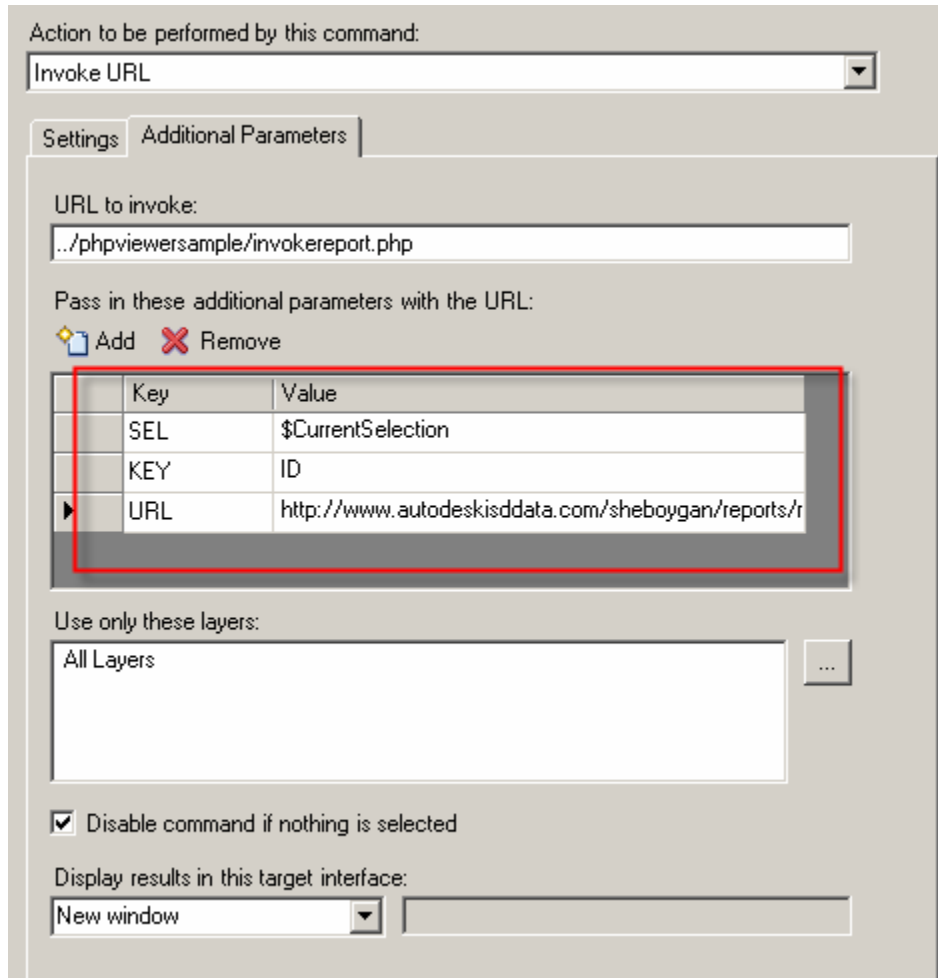
Invoke Report スクリプトを使用すると、Autodesk MapGuide 6.5 の既存のレポートを再利用することができます。このスクリプトを組み込むには、Invoke URL コマンドを使用します (ソース コードは、「付録」を参照)。

SEL - \$CurrentSelection (MapGuide での現在の選択)

KEY—OBJ_KEYS または MG6.5 のレポート内の KEY フィールドのすべての名前

URL—MG6.5 のレポートの URL

METHOD—GET または POST



Invoke Report スクリプトを追加することにより、MapGuide 6.5 のレポートにレポート要求を転送することが可能

METHOD という 4 番目のパラメータを任意に追加できることに注意してください(上図では示されていません)。このパラメータを使用して GET または POST のいずれかを設定することができます。

この例では、ASP.Net www.autodeskisddata.com/sheboygan/reports/rptParcel.aspx を使用して記述されたMG6.5 アプリケーションの既存のレポートを再利用しています。

製品関連情報

- **Autodesk MapGuide プロダクト センター** — オートデスクの公式 Web サイト (www.autodesk.co.jp/mapguide)から、無償でダウンロード可能なデータマイグレーション ツールなど、MapGuide のすべてのバージョンの最新情報を入手することができます。
- **Autodesk MapGuide ディスカッション グループ** — このサイトでは、MapGuide ディスカッション グループが見つかります。MapGuide ディスカッション グループは、ユーザ間で情報を共有、入手するための優れた情報源です。
www.autodesk.com/discussiongroup-mapguide をご覧ください。
- **Autodesk MapGuide デベロッパ ディスカッション グループ** — MapGuide デベロッパ ディスカッション グループは、開発関連の問題を扱うもう 1 つの優れた情報源です。www.autodesk.com/discussiongroup-mapguide-developer をご覧ください。
- **MapGuide Open Source の Web サイト(OSGeo.org)** — このサイトでは、MapGuide Open Source に関する情報を提供しています。
mapguide.osgeo.org から最新リリースをダウンロードしたり、メーリング リストに参加したりすることができます。
- **Autodesk Design Review プロダクト センター** — MapGuide Open Source とともに使用できる Autodesk Design Review Viewer(DWF ビューアを含む)を <http://www.autodesk.co.jp/designreview> からダウンロードすることができます。
- **Autodesk DWF デベロッパ センター** — DWF Viewer API など、MapGuide Open Source とともに使用できる DWF Viewer 用の開発ツールを www.autodesk.com/developdwf から提供しています。

付録: Invoke Report ソース コード (PHP サンプル)

次の PHP スクリプトは、レポート要求を Autodesk MapGuide Enterprise から Autodesk MapGuide 6.5 の既存のレポートに転送します。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"
/>
<title>Invoke Report</title>
</head>
<body onload="OnPageLoad()">

<?php
include '../mapviewerphp/constants.php';

$configFilePath = "C:\Program
Files\MapGuideOpenSource\WebServerExtensions\www\webconfig.ini";

try
{
    // Parse the Parameters
    // - SESSION - MapGuide Session Identifier
    // - MAPNAME - Name of the Map
    // - SEL    - Current Selection
    // - URL    - URL of the MapGuide 6.5 Report
    // - METHOD  - GET or POST (default = POST)

    $session = urldecode(($_SERVER['REQUEST_METHOD'] == "POST")?
$_POST['SESSION']: $_GET['SESSION']);

    $mapName = urldecode(($_SERVER['REQUEST_METHOD'] == "POST")?
$_POST['MAPNAME']: $_GET['MAPNAME']);

    $selText = urldecode(($_SERVER['REQUEST_METHOD'] == "POST")?
$_POST['SEL']: $_GET['SEL']);
```

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

```
$reportUrl = urldecode(($_SERVER['REQUEST_METHOD'] == "POST")?
$_POST['URL']: $_GET['URL']);

$keyName = urldecode(($_SERVER['REQUEST_METHOD'] == "POST")?
$_POST['KEY']: $_GET['KEY']);

$method = "POST";
if ($_SERVER['REQUEST_METHOD'] == "POST")
{
    if (array_key_exists('METHOD', $_POST))
        $method = $_POST['METHOD'];
}
else
{
    if (array_key_exists('METHOD', $_GET))
        $method = $_GET['METHOD'];
}
$objkeys = "";
$errorMsg = null;

// Initialize the Web Extensions and connect to the MapGuide Site using
// session identifier supplied by the caller.

MgInitializeWebTier ($configFilePath);

$userInfo = new MgUserInformation($session);
$siteConnection = new MgSiteConnection();
$siteConnection->Open($userInfo);

// Create an instance of the Resource and Mapping Services and use them
to open
// the map and initialize a selection object.

$resourceSrcv = $siteConnection-
>CreateService(MgServiceType::ResourceService);
```

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

```
$featureSrv = $siteConnection-  
>CreateService(MgServiceType::FeatureService);  
$map = new MgMap();  
$map->Open($resourceSrv, $mapName);
```

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

```
$sel = new MgSelection($map, $selText);
$selLayers = $sel->GetLayers();

// For each layer in the selection, query the selected features and append
the key
// values to the string $objkeys.

for($li = 0; $li < $selLayers->GetCount(); $li++)
{
    $selLayer = $selLayers->GetItem($li);
    $featureSource = new MgResourceIdentifier($selLayer-
>GetFeatureSourceId());

    $featureClassName = $selLayer->GetFeatureClassName();
    $filter = $sel->GenerateFilter($selLayer, $featureClassName);
    $query = new MgFeatureQueryOptions();
    $query->SetFilter($filter);

    $featureReader = $featureSrc->SelectFeatures($featureSource,
$featureClassName, $query);
    while($featureReader->ReadNext())
    {
        if (strlen($objkeys) != 0)
        {
            $objkeys .= ',';
        }

        $propertyType = $featureReader->GetPropertyType($keyName);
        switch($propertyType)
        {
            case MgPropertyType::Boolean :
                $objkeys .= $featureReader->GetBoolean($keyName);
                break;

            case MgPropertyType::Byte :
```

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

```
$objkeys := $featureReader->GetByte($keyName);  
break;  
  
    case MgPropertyType::Single :  
$objkeys := $featureReader->GetSingle($keyName);  
break;  
  
case MgPropertyType::Double :  
$objkeys := $featureReader->GetDouble($keyName);  
break;  
  
case MgPropertyType::Int16 :  
$objkeys := $featureReader->GetInt16($keyName);  
break;  
  
case MgPropertyType::Int32 :  
$objkeys := $featureReader->GetInt32($keyName);  
break;  
  
case MgPropertyType::Int64 :  
$objkeys := $featureReader->GetInt64($keyName);  
break;  
  
case MgPropertyType::String :  
$objkeys := "" . $featureReader->GetString($keyName) . "";  
break;  
  
case MgPropertyType::DateTime :  
case MgPropertyType::Null :  
case MgPropertyType::Blob :  
case MgPropertyType::Clob :  
case MgPropertyType::Feature :  
case MgPropertyType::Geometry :  
case MgPropertyType::Raster :
```

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

```
$errorMsg = 'The data type of the key property is not supported by Invoke  
Report.';
```

```
    break;
```

```
};
```

```
}
```

```
$featureReader->Close();
```

```
}
```

```
}
```

```
catch (MgException $mge)
```

```
{
```

```
    $errorMsg = $mge->GetMessage() . "<br><br>" . $mge->GetDetails();
```

```
}
```

```
catch (Exception $e)
```

```
{
```

```
    $errorMsg = $e->getMessage();
```

```
}
```

```
// If no errors were detected output a form with a hidden OBJ_KEYS  
parameter and JavaScript
```

```
// to submit it. Otherwise output an error message.
```

```
if ($errorMsg == null)
```

```
{
```

```
?>
```

```
<form action="<?= $reportUrl ?>" method="<?= $method ?>"  
enctype="application/x-www-form-urlencoded" name="invokeReport"  
target="_self">
```

```
<input name="OBJ_KEYS" type="hidden" value="<?= $objkeys ?>" />
```

```
</form>
```

```
<script language="JavaScript" type="text/javascript">
```

```
function OnPageLoad()
```

```
{
```

```
    document.invokeReport.submit();
```

```
}
```

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

```
</script>
```

```
<?php
```

```
}
```

```
else
```

```
{
```

```
?>
```

```
<h1>Error</h1>
```

```
<p><?= $errorMsg ?></p>
```

```
<script language="JavaScript" type="text/javascript">
```

```
function OnPageLoad()
```

```
{
```

```
}
```

```
</script>
```

```
<?php
```

```
}
```

```
?>
```

```
</body>
```

```
</html>
```

付録: 赤入れサンプル コード(PHP サンプル)

このコード サンプルは、インストレーション CD に含まれている『MapGuide Enterprise 開発者用ガイド』から引用しています。完全なサンプル アプリケーションについては、開発者用ガイドを参照してください。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<!--
// Copyright (C) 2004-2006 Autodesk, Inc.
//
// This library is free software; you can redistribute it and/or
// modify it under the terms of version 2.1 of the GNU Lesser
// General Public License as published by the Free Software
// Foundation.
//
// This library is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty
// of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
// the GNU
// Lesser General Public License for more details.
//
// You should have received a copy of the GNU Lesser General Public
// License along with this library; if not, write to the Free Software
// Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-
// 1301 USA
-->
<head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-
Type">
  <title>Draw a Line</title>

  <meta http-equiv="content-style-type" content="text/css">
  <link href="../styles/globalStyles.css" rel="stylesheet"
type="text/css">
  <link href="../styles/otherStyles.css" rel="stylesheet"
type="text/css">
  <link href="/mapgudie/viewerfiles/viewer.css" rel="stylesheet"
type="text/css">

  <meta http-equiv="content-script-type" content="text/javascript">
  <script language="javascript" type="text/javascript">
    function OnPageLoad()
    {
      parent.mapFrame.Refresh();
    }
  </script>
```

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

</head>

```
<body onLoad="OnPageLoad()">
  <?php
    require_once('../common.php');
    require_once($webExtensionsDirectory .
'www/mapviewerphp/layerdefinitionfactory.php');
    require_once('../modifying_maps_and_layers/layer_functions.php');

    // Get the parameters passed in from the task pane
    $x0 = ($_SERVER['REQUEST_METHOD'] == "POST")?$_POST['x0']:
$_GET['x0'];
    $y0 = ($_SERVER['REQUEST_METHOD'] == "POST")?$_POST['y0']:
$_GET['y0'];
    $x1 = ($_SERVER['REQUEST_METHOD'] == "POST")?$_POST['x1']:
$_GET['x1'];
    $y1 = ($_SERVER['REQUEST_METHOD'] == "POST")?$_POST['y1']:
$_GET['y1'];
    $sessionId = ($_SERVER['REQUEST_METHOD'] ==
"POST")?$_POST['SESSION']: $_GET['SESSION'];
    $mapName = ($_SERVER['REQUEST_METHOD'] ==
"POST")?$_POST['MAPNAME']: $_GET['MAPNAME'];

    try
    {
      // -----//
      // Basic initialization needs to be done every time.
      MgInitializeWebTier($webconfigFilePath);

      // Get the session information passed from the viewer.
      $sessionId = ($_SERVER['REQUEST_METHOD'] ==
"POST")?$_POST['SESSION']: $_GET['SESSION'];

      // Get the user information using the session id,
      // and set up a connection to the site server.
      $userInfo = new MgUserInformation($sessionId);
      $siteConnection = new MgSiteConnection();
      $siteConnection->Open($userInfo);

      // Get an instance of the required service(s).
      $resourceService = $siteConnection->
CreateService(MgServiceType::ResourceService);
      $featureService = $siteConnection-
>CreateService(MgServiceType::FeatureService);

      //-----//
      // Open the map
      $map = new MgMap();
      $map->Open($resourceService, $mapName);

      $layerName = "Lines";
      $layerLegendLabel = "New Lines";
      $groupName = "Analysis";
      $groupLegendLabel = "Analysis";
```

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

```
//-----//
// Does the temporary feature source already exist?
// If not, create it
$featureSourceName =
"Session:$sessionId//TemporaryLines.FeatureSource";
$resourceIdentifier = new
MgResourceIdentifier($featureSourceName);

$featureSourceExists = DoesResourceExist($resourceIdentifier,
$resourceService);
if (! $featureSourceExists)
{
    // Create a temporary feature source to draw the lines on

    // Create a feature class definition for the new feature
    // source
    $classDefinition = new MgClassDefinition();
    $classDefinition->SetName("Lines");
    $classDefinition->SetDescription("Lines to display.");
    $geometryPropertyName="SHPGEOM";
    $classDefinition->
        SetDefaultGeometryPropertyName( $geometryPropertyName);

    // Create an identify property
    $identityProperty = new MgDataPropertyDefinition("KEY");
    $identityProperty->SetDataType(MgPropertyType::Int32);
    $identityProperty->SetAutoGeneration(true);
    $identityProperty->SetReadOnly(true);
    // Add the identity property to the class definition
    $classDefinition->GetIdentityProperties()->
        Add($identityProperty);
    $classDefinition->GetProperties()->Add($identityProperty);

    // Create a name property
    $nameProperty = new MgDataPropertyDefinition("NAME");
    $nameProperty->SetDataType(MgPropertyType::String);
    // Add the name property to the class definition
    $classDefinition->GetProperties()->Add($nameProperty);

    // Create a geometry property
    $geometryProperty = new
        MgGeometricPropertyDefinition($geometryPropertyName);
    $geometryProperty->
        SetGeometryTypes(MgFeatureGeometricType::Surface);
    // Add the geometry property to the class definition
    $classDefinition->GetProperties()->Add($geometryProperty);

    // Create a feature schema
    $featureSchema = new MgFeatureSchema("SHP_Schema",
        "Line schema");
    // Add the feature schema to the class definition
    $featureSchema->GetClasses()->Add($classDefinition);
}
```

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

```
// Create the feature source
$wkt = $map->GetMapSRS();
$sdfParams = new MgCreateSdfParams("ArbitraryXY",
    $wkt, $featureSchema);
$featureService->CreateFeatureSource($resourceIdentifier,
    $sdfParams);
}

// Add the line to the feature source
$batchPropertyCollection = new MgBatchPropertyCollection();
$propertyCollection = MakeLine("Line A", $x0, $y0, $x1, $y1);
$batchPropertyCollection->Add($propertyCollection);

// Add the batch property collection to the feature source
$cmd = new MgInsertFeatures($layerName,
$batchPropertyCollection);
$featureCommandCollection = new
MgFeatureCommandCollection();
$featureCommandCollection->Add($cmd);

// Execute the "add" commands
$featureService->UpdateFeatures($resourceIdentifier,
$featureCommandCollection, false);

//-----//
$layerExists = DoesLayerExist($layerName, $map);
if (! $layerExists )
{
    // Create a new layer which uses that feature source

    // Create a line rule to stylize the lines
    $ruleLegendLabel = 'Lines Rule';
    $filter = "";
    $color = 'FF0000FF';
    $factory = new LayerDefinitionFactory();
    $lineRule = $factory->CreateLineRule($ruleLegendLabel, $filter,
$color);

    // Create a line type style
    $lineTypeStyle = $factory->CreateLineTypeStyle($lineRule);

    // Create a scale range
    $minScale = '0';
    $maxScale = '1000000000000';
    $lineScaleRange = $factory->CreateScaleRange($minScale,
$maxScale, $lineTypeStyle);

    // Create the layer definition
    $featureName = 'SHP_Schema:Lines';
    $geometry = 'SHPGEOM';
    $layerDefinition = $factory-
>CreateLayerDefinition($featureSourceName, $featureName,
$geometry, $lineScaleRange);
```

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

```
//-----//
// Add the layer to the map
$newLayer = add_layer_definition_to_map($layerDefinition,
$layerName, $layerLegendLabel, $sessionId, $resourceService,
$map);
// Add the layer to a layer group
add_layer_to_group($newLayer,$groupName, $groupLegendLabel,
$map);
}

// -----//
// Turn on the visibility of this layer.
// (If the layer does not already exist in the map, it will be visible
by default when it is added.
// But if the user has already run this script, he or she may have set
the layer to be invisible.)
$layerCollection = $map->GetLayers();
if ($layerCollection->Contains($layerName))
{
$linesLayer = $layerCollection->GetItem($layerName);
$linesLayer->SetVisible(true);
}

$groupCollection = $map->GetLayerGroups();
if ($groupCollection->Contains($groupName))
{
$analysisGroup = $groupCollection->GetItem($groupName);
$analysisGroup->SetVisible(true);
}

//-----//
// Save the map back to the session repository
$sessionIdName = "Session:$sessionId//$mapName.Map";
$sessionResourceID = new MgResourceIdentifier($sessionIdName);
$sessionResourceID->Validate();
$map->Save($resourceService, $sessionResourceID);

//-----//

}
catch (MgException $e)
{
echo "<script language=\"javascript\" type=\"text/javascript\">
\n";
echo "    alert(\" " . $e->GetMessage() . " \"); \n";
echo "</script> \n";
}

////////////////////////////////////
////
function MakeLine($name, $x0, $y0 , $x1, $y1)
{
$propertyCollection = new MgPropertyCollection();
```

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

```
$nameProperty = new MgStringProperty("NAME", $name);
$propertyCollection->Add($nameProperty);

$wktReaderWriter = new MgWktReaderWriter();
$agfReaderWriter = new MgAgfReaderWriter();

$geometry = $wktReaderWriter->Read("LINESTRING XY ($x0 $y0,
$x1 $y1)");
$geometryByteReader = $agfReaderWriter->Write($geometry);
$geometryProperty = new MgGeometryProperty("SHPGEOM",
$geometryByteReader);
$propertyCollection->Add($geometryProperty);

return $propertyCollection;
}

////////////////////////////////////
////
function DoesResourceExist($resourceIdentifier, $resourceService)
// Returns true if the resource already exists, or false otherwise
{
    try
    {
        $resourceService->GetResourceContent($resourceIdentifier);
    }
    catch (MgResourceNotFoundException $e)
    {
        return false;
    }

    return true;
}

////////////////////////////////////
////
function DoesLayerExist($layerName, $map)
// Returns true if the layer already exists, or false otherwise
{
    $layerCollection = $map->GetLayers();
    return( $layerCollection->Contains($layerName) ? true : false ) ;
}

////////////////////////////////////
////
?>
</body>
</html>
```

MIGRATING TO THE NEW MAPGUIDE TECHNOLOGY

Occasionally, Autodesk makes statements regarding planned or future development efforts for our existing or new products and services. These statements are not intended to be a promise or guarantee of future delivery of products, services, or features but merely reflect our current plans, which may change. The Company assumes no obligation to update these forward-looking statements to reflect any change in circumstances, after the statements are made.

Autodesk, AutoCAD, Autodesk MapGuide, DWF, DWG, and DXF are registered trademarks or trademarks of Autodesk, Inc., in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product offerings and specifications at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.

© 2007 Autodesk, Inc. All rights reserved.