Autodesk MapGuide Viewer API

Introduction 4
About This Help System 4
About the Autodesk MapGuide Viewer ActiveX Control 5
About the Autodesk MapGuide Viewer Plug-In 5
About the Autodesk MapGuide Viewer, Java Edition 6
API Summary for Autodesk MapGuide Viewers 6
Additional Information 9
Viewer API Object Traversal Model 10
What's New in Release 6.0 11
New and Enhanced Features 11
Autodesk DWG Data Source Support 11
Enhanced Support for Map Draw Operations 14
Symbol Bitmap Support 14
Map Mode Retrieval Support 14
Replaced Methods and Workarounds 16
Issues and Workarounds 18
Examples 19
Basic Examples 19
Advanced Examples 19
Plug-In/Java Edition Downloadable Files 21
Objects 22
MGBufferSetup Object 23
MGCollection Object 31
MGDatabaseSetup Object 35
MGDwgDataSources Object 63
MGEdgeAttr Object 88
MGError Object 94
MGExtent Object 97
MGExtentEx Object 104
MGFillAttr Object 109
MGLineAttr Object 116
MGMap Object 120

MGMapApplet Object....231
MGMapComponent Object....232
MGMapContext Object....233
MGMapLayer Object....235
MGMapLayerGroup Object....266
MGMapLayerSetup Object....278
MGMapLayerStyle Object....301
MGMapObject Object....304
MGPageElement Object....321
MGPageSetup Object....327
MGPoint Object....337
MGPrimitive Object....344
MGPrintInfo Object....351
MGPrintLayout Object....354
MGRedlineSetup Object....355
MGSelection Object....360
MGSymbolAttr Object....368
MGTextAttr Object....375
Events....391
Observers and Event Methods....410
MGAddMapLayerObserver....411
Event Methods....411
MGBusyStateChangedObserver....412
Event Methods....412
MGDigitizeCircleObserver....413
Event Methods....414
MGDigitizePointObserver....415
Event Methods....415
MGDigitizePolygonObserver....416
Event Methods....416
MGDigitizePolygonObserverEx....417
Event Methods....417
MGDigitizePolylineObserver....418
Event Methods....418

MGDigitizePolylineObserverEx 419
Event Methods 419
MGDigitizeRectangleObserver 420
Event Methods 421
MGDoubleClickObjectObserver 422
Event Methods 422
MGMapLoadedObserver 423
Event Methods 423
MGPrintingObserver 424
Event Methods 424
MGSelectionChangedObserver 426
Event Methods 427
MGViewChangedObserver 428
Event Methods 428
MGViewChangingObserver 429
Event Methods 429
MGViewDistanceExObserver 430
Event Methods 431
MGViewDistanceObserver 432
Event Methods 432
Index 434

Introduction

The Autodesk MapGuide® Viewer API provides developers with programmatic access to and control of Autodesk MapGuide Viewer. Using the API, you can create applications that extend the functionality of the Autodesk MapGuide Viewer. These applications can be *Web-based* or *stand-alone*. A Web-based application typically consists of HTML pages that host a map and access it programmatically with JavaScript, JScript, or VBScript code. A stand-alone application hosts the Autodesk MapGuide Viewer without the aid of a browser and is written in a programming language such as Java, C++, or Visual Basic.

Autodesk MapGuide Viewer comes in three different versions: Autodesk MapGuide Viewer ActiveX Control, Autodesk MapGuide Viewer Plug-In, and Autodesk MapGuide Viewer, Java Edition. For additional information about these versions of the Autodesk MapGuide Viewer, see "Available Versions" in the *Autodesk MapGuide Viewer Help*. Each Autodesk MapGuide Viewer exposes an API that contains all of the Autodesk MapGuide Viewer functionality. For a summary of APIs available in the three Autodesk MapGuide Viewers, see API Summary for Autodesk MapGuide Viewers.

About This Help System

Autodesk MapGuide Viewer API Help is a complete reference to the Autodesk MapGuide Viewer API. It provides the following information:

- This introduction and an overview of the object traversal model.
- A reference that documents the objects, methods, properties, events, and observers in the Autodesk MapGuide Viewer API. If differences exist between the Autodesk MapGuide Viewer versions, those differences are also documented.
- A set of basic and advanced examples for the Autodesk MapGuide® Viewer Plug-In, Autodesk MapGuide® Viewer ActiveX Control, and Autodesk MapGuide® Viewer, Java™ Edition.

Note: This Help system does not cover conceptual and procedural information. You can find this kind of information, including common programming tasks and examples, in the *Autodesk MapGuide Developer's Guide*. You can also find useful background information about Autodesk MapGuide® in the *Autodesk MapGuide User's Guide*. Both of these documents are available in print and in PDF formats.

In addition to these documents, there are many other resources available to help you learn about Autodesk MapGuide, HTML, ActiveX Controls, plug-ins, and Java. Click here for additional information.

Navigating

There are various ways to navigate through the pages in this Help system. You can use the:

- Contents in the left navigation frame to go to the introductory pages for objects, events, and observers. From there, you can locate a particular item and click its link to go to its page.
- Index to look up a specific object, method, property, event, or observer by name, and click its link to go to its page.
- Search tool to find topics containing a specific word, set of words, or phrase.
- Quick Reference drop-down menu to go to objects, events, and observers. You can find Quick Reference under Contents in the left navigation frame.

Within a topic page, you can:

- Use your browser's Back button to return to a previously viewed page.
- Click the button to go to the object traversal model.

- Click the button to go to the parent of the current topic. For example, clicking this button when viewing a method will take you to the parent object for the method.
- Click the and buttons to browse through other pages within a section.

Printing

If you need to print all or part of this Help system, you can use the PDF version, which is available in the file *MGViewerAPIHelp.pdf*, located in the top-level directory of the Autodesk MapGuide CD. You can also view the PDF file from the Autodesk MapGuide Documentation page.

Copyright Information

Autodesk and Autodesk MapGuide are trademarks of Autodesk, Inc., in the USA and/or other countries.

Apple and Macintosh are trademarks of Apple Computer, Inc., registered in the U.S. and other countries.

Netscape and Netscape Navigator are registered trademarks of Netscape Communications Corporation in the United States and other countries.

Microsoft and ActiveX Control are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Java, all Java-based trademarks and logos, Sun, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

All other brand names, product names, or trademarks belong to their respective holders.

About the Autodesk MapGuide Viewer ActiveX Control

The Autodesk MapGuide Viewer ActiveX Control works only with Microsoft® Internet Explorer on Windows systems. It exposes an API that is accessible from VBScript, JScript (Microsoft's implementation of JavaScript), and Java. The Autodesk MapGuide Viewer ActiveX Control API is exposed through ActiveX Control/COM technology. If you develop an application only for the Autodesk MapGuide Viewer ActiveX Control, users can access that application only with Internet Explorer. If someone tries to view your application with Netscape, it will fail. You can also write stand-alone applications for the Autodesk MapGuide Viewer ActiveX Control with Visual Basic or C++. In this case, no browser is required, as your stand-alone application takes the place of a browser.

Click here for a summary of APIs available in the three Autodesk MapGuide Viewers.

About the Autodesk MapGuide Viewer Plug-In

The Autodesk MapGuide Viewer Plug-In works within Netscape® Navigator for Windows. It exposes an API that is accessible from JavaScript and Java. The Autodesk MapGuide Viewer Plug-In API objects are exposed via the Netscape LiveConnect API. The API classes are defined in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder. If you develop an application only for the Autodesk MapGuide Viewer Plug-In, users can access that application only with Netscape. Internet Explorer users will be unable to view your application.

Depending on your application needs, Autodesk MapGuide Viewer Plug-In may need certain Java files provided by Autodesk MapGuide. For more information about what files your application may need, go to Plug-In/Java Edition Downloadable Files.

Click here for a summary of APIs available in the three Autodesk MapGuide Viewers.

About the Autodesk MapGuide Viewer, Java Edition

The Autodesk MapGuide Viewer, Java Edition works with Internet Explorer and Netscape Navigator, on Windows, MacOS®, and Solaris® systems. It exposes an API that is accessible from JScript, JavaScript, and Java. The API classes are defined in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). All configurations expose the Autodesk MapGuide Viewer, Java Edition API to other Java code if it has been compiled to directly instantiate MGMapApplet or MGMapComponent. See "Accessing Maps" in the *Autodesk MapGuide Developer's Guide* for more information.

Depending on your application needs, Autodesk MapGuide Viewer, Java Edition applications may need certain Java files provided by Autodesk MapGuide. For more information about what files your application may need, go to Plug-In/Java Edition Downloadable Files.

Click here for a summary of APIs available in the three Autodesk MapGuide Viewers.

The Autodesk MapGuide Viewer, Java Edition is almost identical to the Autodesk MapGuide Viewer Plug-In API with the following exceptions:

- The Autodesk MapGuide Viewer Plug-In API resides in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder. The Autodesk MapGuide Viewer, Java Edition API resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).
- Classes in the Autodesk MapGuide Viewer Plug-In API are Java interfaces in the Autodesk MapGuide Viewer, Java Edition API.
- The Autodesk MapGuide Viewer, Java Edition has some cosmetic and functional differences from the other Autodesk MapGuide Viewers. For details about these differences, see "About the Java Edition" in the *Autodesk MapGuide Viewer Help*.

Some objects and methods are implemented exclusively for the Autodesk MapGuide Viewer, Java Edition. Specifically, the following objects and methods are available only for this Autodesk MapGuide Viewer:

MGMapApplet	An applet that implements the MGMap interface.
MGMapComponent	Instantiates Autodesk MapGuide Viewer as a user interface object.
MGMapContext	A Java interface whose methods provide browser services to MGMapComponent.
MGMap.stopEx	Aborts any downloading or redrawing that is currently taking place.

The following methods are *not* available in the Autodesk MapGuide Viewer, Java Edition:

MGMap.stop	Aborts any downloading or redrawing that is currently taking place. Use MGMap.stopEx instead.
MGMap.saveMWF	Saves the map to a local MWF.

API Summary for Autodesk MapGuide Viewers

Depending on which operating system and browser your users have, their Viewer will be either the Autodesk MapGuide Viewer ActiveX Control, Autodesk MapGuide Viewer Plug-In, or Autodesk MapGuide Viewer, Java Edition. If all of your users are running the same operating system and browser (for example, on a corporate intranet), you can develop your application only for the one Autodesk MapGuide Viewer that works with their operating system and browser. However, if your users have various combinations of operating systems and browsers, you may need to develop your application to work with all three Autodesk

MapGuide Viewers. For a discussion of which Autodesk MapGuide Viewers you should support, see "Choosing a Viewer/Browser Environment" in the *Autodesk MapGuide User's Guide*.

In general, most objects and their methods in the Autodesk MapGuide Viewer API are available to all three Autodesk MapGuide Viewers. However, object properties are only available to the Autodesk MapGuide Viewer ActiveX Control. Events are available to all three Autodesk MapGuide Viewers as well, with ActiveX Control events for the Autodesk MapGuide Viewer ActiveX Control, and observer event methods for the Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition. Event handling for the Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition is supported through observer interfaces. Finally, some objects and methods are implemented exclusively for the Autodesk MapGuide Viewer, Java Edition.

The following table summarizes the APIs available for the three viewer versions:

Item	ActiveX Control	Plug-In	Java Edition	Notes
Objects	Yes	Yes	Yes	All objects are available to all Viewer versions, with the exception of the MGMapApplet, MGMapComponent, and MGMapContext objects, and their methods, which are exclusive to the Autodesk MapGuide Viewer, Java Edition.
Methods	Yes	Yes	Yes	All methods are available to all Viewer versions, with exceptions listed in the table below.
Properties	Yes	**No**	**No**	Properties are only available in the Autodesk MapGuide Viewer ActiveX Control.
Events	Yes	Yes	Yes	All events are available to all Autodesk MapGuide Viewers: ActiveX Control events for Autodesk MapGuide Viewer ActiveX Control and observer event methods for the Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition.
Observers	**No**	Yes	Yes	Observers for event handling are available in the Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition only.

There are some exceptions to the general rules outlined in the table above. The following table summarizes these exceptions:

MGMap Methods	ActiveX Control	Plug-In	Java Edition
digitizePolygonEx	**No**	Yes	Yes
digitizePolylineEx	**No**	Yes	Yes
getBusyStateChangedObserver	**No**	Yes	Yes

getDoubleClickObserver	**No**	Yes	Yes
getMapLoadedObserver	**No**	Yes	Yes
getPrintingObserver	**No**	Yes	Yes
getSelectionChangedObserver	**No**	Yes	Yes
getViewChangedObserver	**No**	Yes	Yes
getViewChangingObserver	**No**	Yes	Yes
getViewDistanceExObserver	**No**	Yes	Yes
getViewDistanceObserver	**No**	Yes	Yes
saveMWF	Yes	Yes	**No**
setBusyStateChangedObserver	**No**	Yes	Yes
setDoubleClickObserver	**No**	Yes	Yes
setMapLoadedObserver	**No**	Yes	Yes
setPrintingObserver	**No**	Yes	Yes
setSelectionChangedObserver	**No**	Yes	Yes
setViewChangedObserver	**No**	Yes	Yes
setViewChangingObserver	**No**	Yes	Yes
setViewDistanceExObserver	**No**	Yes	Yes
setViewDistanceObserver	**No**	Yes	Yes
stop	Yes	Yes	**No**
stopEx	**No**	**No**	Yes

Additional Information

Developing applications with the Autodesk MapGuide Viewer API requires knowledge and experience with the Autodesk MapGuide product architecture, HTML, frames, Java, JavaScript, JScript, and more. This page provides links to more information on these topics. Please note that although these links were available when this Help file was released, some of them are not Autodesk pages and may become unavailable.

Working with Autodesk MapGuide

Autodesk MapGuide Web Site: http://www.autodesk.com/mapguide

On this Web site, you can find additional information about Autodesk MapGuide development. The site provides many examples of applications developed with Autodesk MapGuide, both demo applications and real customer sites. You will also find links to resources, such as API examples, general product documentation and Help, and discussion groups.

Autodesk MapGuide Product Documentation: http://www.autodesk.com/mapguidedocs

Working with Netscape and Java

Netscape JavaScript Guide:
http://developer.netscape.com/docs/manuals/communicator/jsguide4/index.htm

Netscape JavaScript Reference: http://developer.netscape.com/docs/manuals/communicator/jsref/index.htm

Sun Microsystems Java Resources: http://java.sun.com/nav/read/products.html

Java Technology Home Page: http://java.sun.com/

Netscape LiveConnect/Plug-in Developers Guide:
http://developer.netscape.com/docs/manuals/communicator/plugin/index.htm

Working with HTML and Scripts

Resources for Creating Web Sites: http://home.netscape.com/home/how-to-create-web-services.html

Web Programming and Scripting: http://home.cnet.com/webbuilding/0-3882.html

Working with Microsoft Scripts

Microsoft Windows Script Technologies: http://msdn.microsoft.com/scripting/

Viewer API Object Traversal Model

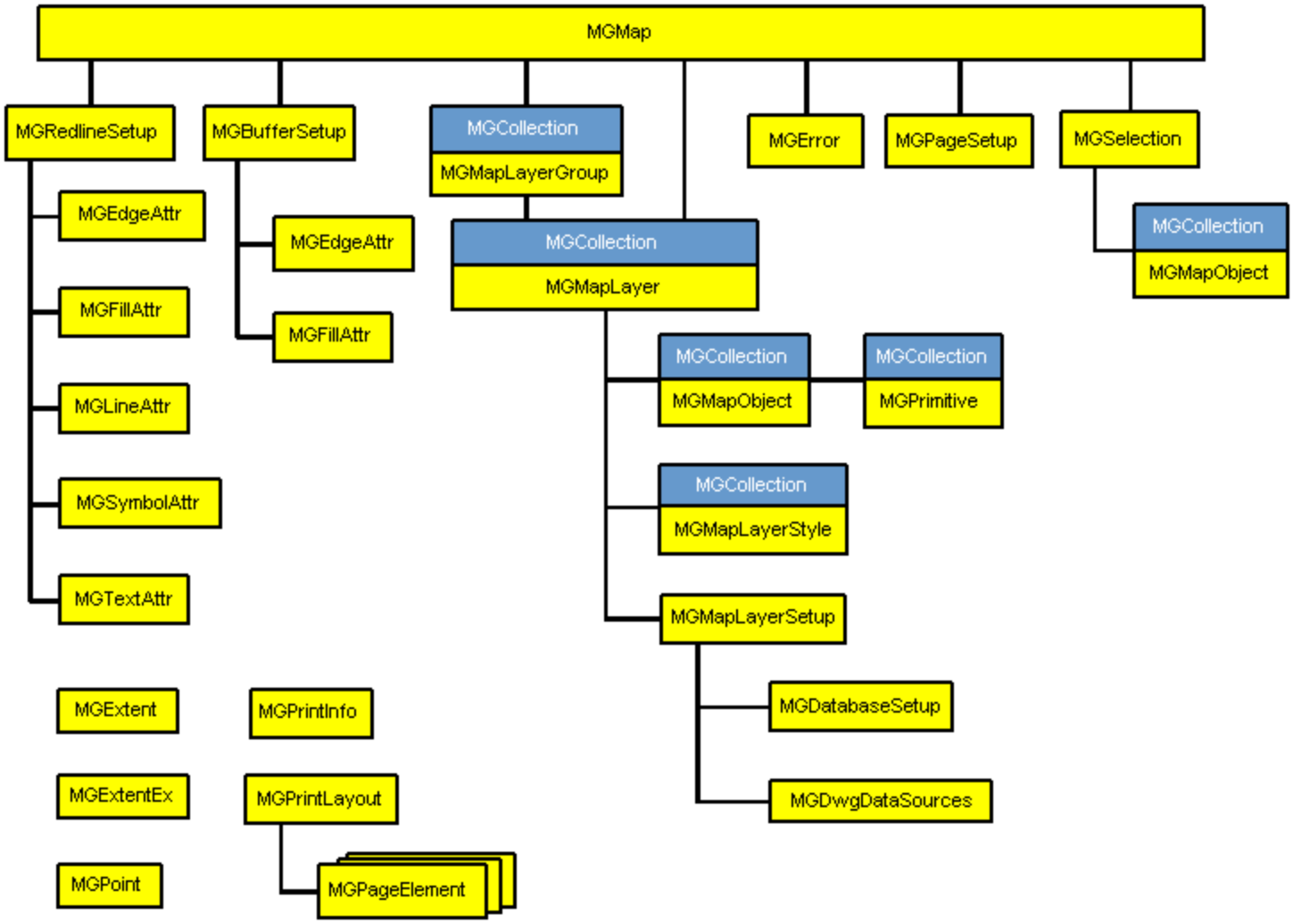
The Autodesk MapGuide Viewer API exposes its functionality as a hierarchy of objects within Autodesk MapGuide Viewer. This hierarchy is represented in the following object traversal model.

Note that this is not an object-oriented inheritance model. Nothing is inherited from an object-oriented programming perspective.

You can use this object traversal model as a guide for traversing objects to get to a specific object. For example, from MGMap, you can access one of the MGMapLayerGroup objects by obtaining an MGCollection object containing all the MGMapLayerGroup objects using the MGMap.getMapLayerGroups method. Alternatively, you can access an MGMapLayerGroup object directly if you have its name by using the MGMap.getMapLayerGroup method.

To use this model, click an object to get detailed reference information, or refer to the Objects introduction for an alphabetical list of object descriptions.

Note: If you are developing an application with JavaScript or VBScript, these objects have no constructors (with the exception of MGPoint, MGCollection, and MGExtentEx objects). Only Autodesk MapGuide Viewer can instantiate Autodesk MapGuide objects: they cannot be constructed directly from JavaScript or VBScript.



What's New in Release 6.0

This release of the Autodesk MapGuide Viewer API includes one new object and several new and changed methods and properties. This topic lists these changes and additions and describes what features they support. When using the API reference and index, you can also identify which objects, methods, and properties are new by looking for the NEW symbol. To identify methods and properties that have been changed, look for the CHANGED symbol. For a list of methods and properties that have been replaced in this and earlier releases, see Replaced Methods and Properties.

Note: The version number of this release is 6.0. You can also determine the version programatically using the getApiVersion method.

New and Enhanced Features

Release 6.0 provides the following new and enhanced features. Each of these are described in more detail in the following sections:

- Autodesk DWG Data Source Support
- Enhanced Support for Map Redraw Operations
- Symbol Bitmap Support
- Map Mode Retrieval Support

Autodesk DWG Data Source Support

Several changes and additions were made to Autodesk MapGuide and the Autodesk MapGuide Viewer API Help to support Autodesk drawing (DWG) data sources. The most significant addition is a new MGDwgDataSources object. Methods and properties were also added or changed in MGMapLayerSetup to support DWG data sources.

MGDwgDataSources Methods and Properties

getDataSource	NEW Gets the DWG data source name that establishes a connection between Autodesk MapGuide Server and the data provider.
getDwg	NEW Gets the Autodesk DWG file name or project file for the Autodesk DWG layer.
getKeyColumn	NEW Gets the name of the column that contains the primary key for each map feature on the DWG layer.
getKeyColumnType	NEW Gets the DWG key column data type.
getKeyTable	NEW Gets the name of the DWG key table to use as primary keys for either map feature properties, external database links, or object data in the Autodesk DWG.
getLayerFilter	NEW Gets the layer filter, which specifies which layers to extract from the specified Autodesk DWG or project file.

getNameColumn	NEW Gets the name of the column that contains the name for each map feature.
getNameTable	NEW Gets the name of the DWG table for MapTips about map feature properties, external database links, or object data in the Autodesk DWG.
getUrlColumn	NEW Gets the name of the column that contains the URL associated with each map feature.
getUrlTable	NEW Gets the name of the DWG URL table containing URLs to which Autodesk DWG map features are linked.
setDataSource	NEW Sets the DWG data source name that establishes a connection between Autodesk MapGuide Server and the data provider.
setDwg	NEW Sets the Autodesk DWG file name or project file for the Autodesk DWG layer.
setKeyColumn	NEW Sets the name of the column that contains the primary key for each map feature on the DWG layer.
setKeyColumnType	NEW Sets the DWG key column data type.
setKeyTable	NEW Sets the name of the DWG key table to use as primary keys for either map feature properties, external database links, or object data in the Autodesk DWG.
setLayerFilter	NEW Sets the layer filter, which specifies which layers to extract from the specified Autodesk DWG or project file.
setNameColumn	NEW Sets the name of the column that contains the name for each map feature.
setNameTable	NEW Sets the name of the DWG table for MapTips about map feature properties, external database links, or object data in the Autodesk DWG.
setUrlColumn	NEW Sets the name of the column that contains the URL associated with each map feature.
setUrlTable	NEW Sets the name of the DWG URL table containing URLs to which Autodesk DWG map features are linked.
DataSource	NEW Gets or sets the DWG data source name that establishes a connection between Autodesk MapGuide Server and the data provider.
Dwg	NEW Gets or sets the Autodesk DWG file name or project file for the Autodesk DWG layer.
KeyColumn	NEW Gets or sets the name of the column that contains the primary key for each map feature on the DWG layer.
KeyColumnType	NEW Gets or sets the DWG key column data type.

KeyTable	NEW Gets or sets the name of the DWG key table to use as primary keys for either map feature properties, external database links, or object data in the Autodesk DWG.
LayerFilter	NEW Gets or sets the layer filter, which specifies which layers to extract from the specified Autodesk DWG or project file.
NameColumn	NEW Gets or sets the name of the column that contains the name for each map feature.
NameTable	NEW Gets or sets the name of the DWG table for MapTips about map feature properties, external database links, or object data in the Autodesk DWG.
UrlColumn	NEW Gets or sets the name of the column that contains the URL associated with each map feature.
UrlTable	NEW Gets or sets the name of the DWG URL table containing URLs to which Autodesk DWG map features are linked.

MGMapLayerSetup DWG Methods and Properties

getDataFile	CHANGED Returns the name of the data file from which Autodesk MapGuide Server retrieves the map data for this map layer. Support for Autodesk DWG layers was added. This methods returns the Autodesk DWG file. This use of this method is retained in this release for backward compatibility and will be deprecated in the next release. Instead, you should use the MGDwgDataSources getDwg method.
getDwgDataSources	NEW Returns the MGDwgDataSources object for this layer. Support for DWG layers was added. This will set the Autodesk DWG file. This use of this method in this release is retained for backward compatibility and will be deprecated in the next release. Instead, you should use the MGDwgDataSources setDwg method.
setDataFile	CHANGED Sets the name of the data file from which Autodesk MapGuide Server retrieves data for this map layer.
DataFile	CHANGED Gets or sets the name of the data file from which Autodesk MapGuide Server retrieves map data for this map layer. Support for DWG layers was added. This property will get or set the Autodesk DWG file. This use of this property is retained in this release for backward compatibility and will be deprecated in the next release. Instead, you should use the MGDwgDataSources Dwg property.
MGDwgDataSources	NEW Returns the MGDwgDataSources object for this layer.

Enhanced Support for Map Draw Operations

New methods and a property were added to MGMap to allow more control over how frequently, and under what conditions, Autodesk MapGuide Viewer draws a map. Maps are now updated every 1.5 seconds by

default, which allows the end user to view changes to the map as it is being drawn instead of waiting until the entire map is rendered.

MGMap Intermediate Updates Methods and Property

getIntermediateUpdatesEnabled	NEW Gets the current setting of the intermediate updates flag.
setIntermediateUpdatesEnabled	NEW Specifies whether or not the map window should update every 1.5 seconds during a drawing operation or only at the end of the operation when all data has been loaded and rendered.
IntermediateUpdatesEnabled	NEW Specifies whether or not the map window should update every 1.5 seconds during a drawing operation or only at the end of the operation when all data has been loaded and rendered.

Symbol Bitmap Support

Changes were made to MGSymbolAttr methods and properties to support bitmaps in symbols.

MGSymbolAttr Bitmap Support Methods and Properties

setRotation	CHANGED Sets the rotation of the point. This method was changed to return an error if the symbol contains a bitmap since such symbols cannot be rotated.
setSymbol	CHANGED Sets the symbol to be used as the symbol attribute. This method was changed so that if the specified symbol contains bitmaps, and the current rotation angle is non-zero, the symbol will be set, and the rotation angle will be set to zero.
Rotation	CHANGED The rotation of the point, in degrees.This property was changed to return an error if the symbol contains a bitmap since such symbols cannot be rotated.
Symbol	CHANGED The symbol to be used as the point attribute. This property was changed so that if the specified symbol contains bitmaps, and the current rotation angle is non-zero, the symbol will be set, and the rotation angle will be set to zero.

Map Mode Retrieval Support

A new method and property was added to MGMap to allow retrieval of a map's mode (zoom, pan, and so on).

MGMap Mode Retrieval Method and Property

getMode	NEW Gets the current mode of the Autodesk MapGuide Viewer.
Mode	NEW Gets the current viewer mode of the map, such as ZoomIn, Digitize, Pan, and so on.

Replaced Methods and Workarounds

The following table lists the methods and properties that have been or will be replaced. We strongly encourage you to use the newer methods and properties since the older ones may be removed from the API in a future release. Methods and properties replaced with this release are indicated by the NEW symbol. All the old methods and properties will still work in this release.

Old Method or Property	*Instead use...*
MGMap.digitizePolygon	MGMap.digitizePolygonEx
MGMap.digitizePolyline	MGMap.digitizePolylineEx
MGMap.getMapLayers	MGMap.getMapLayersEx
MGMap.getUrlList (no longer documented)	MGMap.getUrlListState
MGMap.selectPolygon	MGMap.selectPolygonEx
MGMap.setUrlList (no longer documented)	MGMap.setUrlListState
MGMapLayer.getMapObjects	MGMapLayer.getMapObjectsEx
MGMapLayerSetup.getDataFile	For Text, Point, Polyline, and Polygon map layers from Spatial Data Provider (SDP) Data Sources, use MGMapLayerSetup.getSdpFeatureTable instead. NEW For DWG layers, this method will be deprecated in the next release. Instead, you should use the MGDwgDataSources.getDwg method. This method is still valid for other data source types.
MGMapLayerSetup.setDataFile	For Text, Point, Polyline, and Polygon map layers from Spatial Data Provider (SDP) Data Sources use MGMapLayerSetup.setSdpFeatureTable instead. NEW For DWG layers, this method will be deprecated in the next release. Instead, you should use the MGDwgDataSources.setDwg method. This method is still valid for other data source types.

MGMapLayerSetup.DataFile	For Text, Point, Polyline, and Polygon map layers from Spatial Data Provider (SDP) data sources, use the SdpFeatureTable property instead. NEW For DWG layers, this property will be deprecated in the next release. Instead, you should use the MGDwgDataSources.Dwg property. This property is still valid for other data source types.
MGSelection.addObjects	MGSelection.addObjectsEx
MGSelection.getMapObjects	MGSelection.getMapObjectsEx
MGSelection.removeObjects	MGSelection.removeObjectsEx

Issues and Workarounds

This topic describes known issues and workarounds for the Autodesk MapGuide Viewer API. To learn about known issues and workarounds for Autodesk MapGuide Viewers in general, refer to the Known Issues topic in the *Autodesk MapGuide Viewer Help*. To learn about known issues and workarounds for other Autodesk MapGuide products, refer to the Autodesk MapGuide *MGReadme* file located in the top-level folder of each Autodesk MapGuide product you installed. For example, this file is located in *C:\Program Files\Autodesk\MapGuideViewerActiveX6\MGReadme.htm* for a default Windows installation of Autodesk MapGuide Viewer ActiveX Control.

getLength method Displays Incorrect Lengths for Dashed Lines

When you use either the MGMapObject.getLength or MGPrimitive.getLength methods to measure the length of a dashed line, it returns an incorrect value. To workaround this problem, use solid lines instead of dashed lines for lines to be measured.

Examples

Basic Examples

These examples show you how to use the Autodesk MapGuide Viewer API. They are not meant to be complete applications, but instead serve to illustrate how the API works. Please feel free to reuse and modify any of the sample code. For advanced examples, click here.

Note: Most of these examples reference data on the Autodesk MapGuide Web site, so you will need Internet access to run them and, of course, you will need to download and install Autodesk MapGuide Viewer on your system.

Simple HTML	This example demonstrates how to embed the Autodesk MapGuide Viewer in a simple Web page. Users can use the buttons below the map to load different maps into the Autodesk MapGuide Viewer and display the *About Autodesk MapGuide* dialog box.
Simple Frameset	This example demonstrates the use of the Autodesk MapGuide Viewer in a multi-frame HTML page. The Autodesk MapGuide Viewer is in its own frame, and is specified to grow or shrink as the window is resized. Users can use buttons in another frame to load different maps into the Autodesk MapGuide Viewer and display the *Preferences* and *About Autodesk MapGuide* dialog boxes.
Simple Table	This example shows another approach to interface design. It provides the same API functionality as the Simple Frameset example, but uses JavaScript HREFs and an HTML table to create a different look. Note that the map legend and status bar have been suppressed.
Navigation Techniques	This example demonstrates Viewer navigation techniques. Users can use buttons in the lower frame to pan and zoom and controls in the side frame to scroll the map and zoom to predefined locations.
Selection Techniques	This example demonstrates selection techniques with the Autodesk MapGuide Viewer. A list box to the right of the map shows the United States. When users select states from the list box, they are selected on the map. Users can use additional controls to display the names and keys of selected states, clear the display, and select features by radius.
Basic JavaScript and HTML Examples	This file contains short JavaScript and HTML examples that you use in your own applications.

Advanced Examples

These examples show you how to use the Autodesk MapGuide Viewer API. They are not meant to be complete applications, but instead serve to illustrate how the API works. Please feel free to reuse and modify any of the sample code. For basic examples, click here.

Note: Most of these examples reference data on the Autodesk MapGuide Web site, so you will need Internet access to run them, and, of course, you will need to download and install Autodesk MapGuide Viewer on your system.

Event Handling This example demonstrates how to automatically refresh a map layer every few seconds, and provides event-handling techniques, including the use of Java observers with the Autodesk MapGuide Viewer Plug-In. Users can use controls to the right of the map to enable or disable selection and double-click events, and buttons below the map to digitize features on the map. For additional information about setting up event handler interfaces, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Filter DWG Layers This example demonstrates filtering DWG layers and selecting map features. A list to the right of the map shows the the DWG filters that can be applied to an architectural map of a floor plan. When a user selects one or more filters from the list, the corresponding DWG layer(s) is displayed on the map. The user can also select one or more features on the map and display properties, including names and keys, and DWG layer properties for the current map layer.

Java Edition Examples These examples demonstrate techniques for implementing the Autodesk MapGuide Viewer, Java Edition. Also refer to the list of Plug-In/Java Edition Downloadable Files required for building Java Edition applications.

Plug-In/Java Edition Downloadable Files

Following is a list of downloadable MapGuide java class and source files for developing Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition applications. If you configured an editor for *.java* files, or have default Java file editor installed on your machine, clicking on these files will open them in that editor. If you do not have a default editor for *.java* files, clicking these files will display them in your browser. To save the file to another location, right-click it and choose *Save Target As* or *Save Link As* from the pop-up menu.

MapGuideObserver6.class	The Autodesk MapGuide Viewer Plug-In Observer class
MapGuideObserver6.java	Source for the Autodesk MapGuide Viewer Plug-In Observer class
MapGuideObserver6J.class	The Autodesk MapGuide Viewer, Java Edition Observer class
MapGuideObserver6J.java	Source for the Autodesk MapGuide Viewer, Java Edition Observer class.
mgjava.cab	Archive file containing all the Java Viewer class files; for use with the Autodesk MapGuide Viewer, Java Edition on Windows systems
mgjava.jar	Archive file containing all the Java Viewer class files; for use with the Autodesk MapGuide Viewer, Java Edition on Windows, Macintosh, or Solaris systems
MGDetectClass.class	Detects whether Autodesk MapGuide Viewer Plug-In or Autodesk MapGuide Viewer, Java Edition is installed on a client machine, and installs the corresponding observer applet. For an example that uses this class, see Installing Observers for Plug-In and Java Edition.
MGDetectClass.java	Source for the MGDetectClass class.
MGDetectClass.zip	Archive file containing the MGDetectClass.class file.

Objects

The Autodesk MapGuide Viewer API provides the following objects for Autodesk MapGuide Viewer development. Most objects apply to all three Autodesk MapGuide Viewer versions (ActiveX Control, Plug-In, and Java Edition). Three objects (MGMapApplet, MGMapComponent, and MGMapContext) apply only to the Java Edition.

All properties apply only to ActiveX Control development. Most, but not all methods apply to all three Autodesk MapGuide Viewer APIs. When there are differences, an **Applies To** field in the method description indicates which Autodesk MapGuide Viewers apply. For a summary of available APIs for the three Autodesk MapGuide Viewers, see API Summary for Autodesk MapGuide Viewers.

MGBufferSetup	Organizes the buffer setup properties, which control how Autodesk MapGuide Viewer creates buffers with the viewBuffer and viewBufferDlg methods and the popup menu View Buffer command.
MGCollection	Represents collections of objects, such as MGPoint, MGMapObject, Double, Integer, or any other type of object.
MGDatabaseSetup	Provides access to the database setup for map layers that are linked to database tables.
MGDwgDataSources	NEW! Implements methods and properties to support the use of DWG data sources.
MGEdgeAttr	Contains the polygon edge attributes used to draw buffer or redline polygons.
MGError	Contains error diagnostic information about the most recently executed API method.
MGExtent	Represents an extent of a map in lat/lon (*X*/*Y*) coordinates.
MGExtentEx	Represents spatial extents or a bounding box.
MGFillAttr	Contains the polygon fill attributes used to draw buffer or redline polygons.
MGLineAttr	Contains the attributes used to draw lines and polylines on redline layers.
MGMap	The top-level object of the API for the Autodesk MapGuide ActiveX Control, Plug-In, and Java Edition.
MGMapApplet	Applet that implements the MGMap interface in the Autodesk MapGuide Viewer, Java Edition API.
MGMapComponent	Instantiates Autodesk MapGuide Viewer as a user interface object in the Autodesk MapGuide Viewer, Java Edition API. MGMapComponent extends the Component class and implements the MGMap interface.
MGMapContext	Java interface whose methods provide browser services to MGMapComponent in theAutodesk MapGuide Viewer, Java Edition API.
MGMapLayer	Represents a layer on the map.
MGMapLayerGroup	Represents a map layer group.
MGMapLayerSetup	Provides run-time access to the map layer setup.
MGMapLayerStyle	Provides access to the display scale range of a map layer style.
MGMapObject	Represents a feature on the map, such as a point, polyline, polygon, or text feature.

MGPageElement	Represents a page element on the printout.
MGPageSetup	Organizes the Page Setup properties, which control how Autodesk MapGuide prints maps.
MGPoint	Represents a point on the map expressed in MCS or lat/lon coordinates.
MGPrimitive	Provides access to the individual primitives that make up a multi-primitive feature.
MGPrintInfo	Provides information about the current print operation.
MGPrintLayout	Controls the layout of the page elements on the printout.
MGRedlineSetup	Provides access to the style information used to create features on a redline layer.
MGSelection	Represents the list of selected map features.
MGSymbolAttr	Contains the attributes used to draw symbols on redline layers.
MGTextAttr	Contains the attributes used to draw text on redline layers.

MGBufferSetup Object

The MGBufferSetup object organizes the buffer setup properties, which control how Autodesk MapGuide Viewer creates buffers with the viewBuffer and viewBufferDlg methods and the popup menu View Buffer command.

To get an instance of the MGBufferSetup object, use the MGMap.getBufferSetup method.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGBufferSetup extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGBufferSetup is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getDistance	Gets the buffer offset distance.
getEdgeAttr	Gets an object that groups the polygon edge attributes used to draw the buffer polygon(s).
getFillAttr	Gets an object that groups the polygon fill attributes used to draw the buffer polygon(s).
getLayerName	Gets the name of the map layer that will contain the buffer objects generated from the viewBuffer method of MGMap.
getOneObject	Returns a value indicating whether one large merged buffer object should be created for all buffered features.
getUnit	Gets the units used to specify the size of the buffer.
setDistance	Specifies the buffer offset distance.

setLayerName Sets the name of the map layer that will contain the buffer objects generated from the viewBuffer method of MGMap.

setOneObject Specifies whether one large merged buffer object should be created for all buffered features.

setUnit Sets the units used to specify the size of the buffer.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

Distance The buffer offset distance.

EdgeAttr The object that groups the polygon edge attributes used to draw the buffer polygon(s).

FillAttr The object that groups the polygon fill attributes used to draw the buffer polygon(s).

LayerName The name of the buffer layer to use for the buffer objects.

OneObject Specifies whether or not one large merged buffer object should be created for all buffered features.

Unit The distance unit, which is used with setDistance and getDistance to determine the distance that buffers will extend from the selected features.

MGBufferSetup.getDistance

Syntax

```
double getDistance()
```

Description

Gets the buffer offset distance.

The distance is expressed in the current distance unit, which you can specify using setUnit.

Returns

double - The buffer offset distance.

Error Codes

none

See Also

setDistance, Distance, getUnit

MGBufferSetup.getEdgeAttr

Syntax

```
MGEdgeAttr getEdgeAttr()
```

Description

Gets an object that groups the polygon edge attributes used to draw the buffer polygon(s).

Returns

MGEdgeAttr - The polygon edge attributes object.

Error Codes

none

See Also

EdgeAttr

MGBufferSetup.getFillAttr

Syntax

```
MGFillAttr getFillAttr()
```

Description

Gets an object that groups the polygon fill attributes used to draw the buffer polygon(s).

Returns

MGFillAttr - The polygon fill attributes object.

Error Codes

none

See Also

FillAttr

MGBufferSetup.getLayerName

Syntax

```
string getLayerName()
```

Description

Gets the name of the map layer that will contain the buffer objects generated from the viewBuffer method of MGMap.

The name may reference an existing buffer map layer or a new map layer that will be created.

Returns

String - The name of the buffer layer.

Error Codes

none

See Also

setLayerName, LayerName

MGBufferSetup.getOneObject

Syntax

```
boolean getOneObject()
```

Description

Returns a value indicating whether one large merged buffer object should be created for all buffered features.

Returns

boolean - Returns True to create one object; or False to create individual buffer objects for each map feature that is buffered.

Error Codes

none

See Also

setOneObject, OneObject

MGBufferSetup.getUnit

Syntax

```
String getUnit()
```

Description

Gets the units used to specify the size of the buffer.

Returns

String - The units used to specify the size of the buffer, either MI (miles), KM (kilometers), FT (feet), or M (meters).

Error Codes

none

See Also

setUnit, Unit, getDistance, setDistance

MGBufferSetup.setDistance

Syntax

```
void setDistance(double distance)
```

Description

Specifies the buffer offset distance.

Parameters

distance - The buffer offset distance expressed in the current unit, which you can specify using setUnit.

Error Codes

-3 (illegal argument)

See Also

getDistance, Distance, getUnit

MGBufferSetup.setLayerName

Syntax

```
boolean setLayerName(String name)
```

Description

Sets the name of the map layer that will contain the buffer objects generated from the viewBuffer method of MGMap.

Parameters

name - The name of the map layer that will contain the buffer objects. You can enter an existing buffer map layer name to add buffers to that layer, or enter a new name to create a new layer for the buffers. Do not leave the name blank or use the name of an existing non-buffer map layer.

Returns

boolean - Returns True if successful; otherwise, returns False.

False may be returned if an invalid layer name is specified.

Error Codes

-3 (illegal argument)

See Also

getLayerName, LayerName

MGBufferSetup.setOneObject

Syntax

```
void setOneObject(boolean flag)
```

Description

Specifies whether one large merged buffer object should be created for all buffered features.

Parameters

flag - True to create one object; False to create individual buffer objects for each map feature.

Error Codes

none

See Also

getOneObject, OneObject

MGBufferSetup.setUnit

Syntax

```
void setUnit(String unit)
```

Description

Sets the units used to specify the size of the buffer.

Parameters

unit - The units used to specify the size of the buffer, either MI (miles), KM (kilometers), FT (feet), or M (meters).

Error Codes

-3 (illegal argument)

See Also

getUnit, Unit, getDistance, setDistance

MGBufferSetup.Distance

Description

Gets or sets the buffer offset distance.

The distance is expressed in the current units, which you can set with the Unit property.

Type

double (read/write)

Error Codes

-3 (illegal argument)

See Also

getDistance, setDistance

MGBufferSetup.EdgeAttr

Description

Gets the object that groups the polygon edge attributes used to draw the buffer polygon(s).

Type

MGEdgeAttr (read-only)

Error Codes

none

See Also

getEdgeAttr

MGBufferSetup.FillAttr

Description

Gets the object that groups the polygon fill attributes used to draw the buffer polygon(s).

Type

MGFillAttr (read-only)

Error Codes

none

See Also

getFillAttr

MGBufferSetup.LayerName

Description

Gets or sets the name of the buffer layer to use for the buffer objects.

Type

boolean (read/write)

Error Codes

-3 (illegal argument)

See Also

getBufferSetup, setLayerName

MGBufferSetup.OneObject

Description

Specifies whether or not one large merged buffer object should be created for all buffered features.

Type

boolean (read/write)

Error Codes

none

See Also

getOneObject, setOneObject

MGBufferSetup.Unit

Description

Gets or sets the distance unit, which is used with setDistance and getDistance to determine the distance that buffers will extend from the selected features.

Type

String (read/write)

Error Codes

-3 (illegal argument)

See Also

getDistanceUnit, setDistanceUnit

MGCollection Object

The MGCollection object represents collections of objects, such as MGPoint, MGMapObject, Double, Integer, or any other type of object.

The MGCollection object has similar capabilities to the Java Vector class, but it can be used with the APIs of all three versions of Autodesk MapGuide Viewer (the Autodesk MapGuide Viewer Plug-In, the Autodesk MapGuide Viewer ActiveX Control, and the Autodesk MapGuide Viewer, Java Edition).

You can create instances of the MGCollection object using createObject from MGMap.

Java Development for the Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGCollection extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for the Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGCollection is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

add	Adds the specified object to the collection.
isEmpty	Returns a value indicating whether the collection is empty or contains any objects.
item	Gets the object at the specified index in the collection.
remove	Removes the object at the specified index from the collection.
removeAll	Removes all of the objects from the collection.
size	Gets the number of map features in the collection that have been displayed in the view since the last reload or rebuild of the map layer.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

Count	Gets the number of map features in the collection that have been displayed in the view since the last reload or rebuild of the map layer.

MGCollection.add

Syntax

```
void add(object obj)
```

Description

Adds the specified object to the collection.

Parameters

obj - The object to be added.

Error Codes

none

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It uses the add method to populate an MGCollection called mapObjects. Note the use of getMapObject to obtain the objects by key before adding them to the collection:

```
function selectSomeStates()
{
    // Get map object; then get "States" layer
    map = getMap();
    var mapLayer = map.getMapLayer("States");
    // Create an MGCollection object called mapFeatures
    var mapFeatures = map.createObject("MGCollection");
    // Get features by key, then add to mapFeatures collection
    mapFeatures.add(mapLayer.getMapObject("AZ")); // key is "AZ"
    mapFeatures.add(mapLayer.getMapObject("CA")); // key is "CA"
    mapFeatures.add(mapLayer.getMapObject("NV")); // key is "NV"
    // Select and display contents of mapFeatures
    map.getSelection().addObjectsEx(mapFeatures, true);
}
```

MGCollection.isEmpty

Syntax

```
boolean isEmpty()
```

Description

Returns a value indicating whether the collection is empty or contains any objects.

Returns

boolean - Returns True if the collection is empty; False if the collection contains at least one object.

Error Codes

none

MGCollection.item

Syntax

```
object item(int index)
```

Description

Gets the object at the specified index in the collection.

Parameters

index - The zero-based index of the desired object.

Returns

object - The object at the specified index.

Error Codes

-7 (no such element)

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays a dialog box listing all the layers in the map. First, MGMap.getMapLayersEx is used to obtain the map layer collection and to assign it to a variable named layers. Then, size and item are used in a for loop to operate on each item in the collection:

```
function listLayers()
{
    var map = getMap();
    var msg = "";

    // Assign collection to layers variable
    var layers = map.getMapLayersEx();
    // Use size() to step through collection
    for (var i = 0; i < layers.size(); i++)
    {
        // Get each item in collection...
        var layer = layers.item(i);
        // ...then get item name and add it to msg
        msg = msg + layer.getName() + "\n";
    }
    // Display the layer names
    alert(msg);
}
```

MGCollection.remove

Syntax

`void remove(int index)`

Description

Removes the object at the specified index from the collection.

Parameters

index - The zero-based index of the object to be removed.

Error Codes

-7 (no such element)

See Also

removeAll

MGCollection.removeAll

Syntax

```
void removeAll()
```

Description

Removes all of the objects from the collection.

Error Codes

none

See Also

remove

MGCollection.size

Syntax

```
int size()
```

Description

Gets the number of map features in the collection that have been displayed in the view since the last reload or rebuild of the map layer.

Returns

int - The size of the collection.

Error Codes

none

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays a dialog box listing all the layers in the map. First, MGMap.getMapLayersEx is used to obtain the map layer collection and to assign it to a variable named layers. Then, size and item are used in a for loop to operate on each item in the collection:

```
function listLayers()
{
    var map = getMap();
```

```
    var msg = "";

    // Assign collection to layers variable
    var layers = map.getMapLayersEx();
    // Use size() to step through collection
    for (var i = 0; i < layers.size(); i++)
    {
        // Get each item in collection...
        var layer = layers.item(i);
        // ...then get item name and add it to msg
        msg = msg + layer.getName() + "\n";
    }
    // Display the layer names
    alert(msg);
}
```

MGCollection.Count

Description

Gets the number of map features in the collection that have been displayed in the view since the last reload or rebuild of the map layer.

Type

int (read-only)

Error Codes

none

See Also

size

MGDatabaseSetup Object

The MGDatabaseSetup object provides access to the database setup for map layers that are linked to database tables.

Database setup options that you can access include the OLE DB data source name, the database table name, various column names, and the SQL Where clause. These are the same database settings the map author specifies on the Data Sources tab of the Map Layer Properties dialog box in Autodesk MapGuide Author. If you attempt to query or modify a setting that does not apply to the layer of the map feature you are working on, the method call fails and sets an error code in the MGError object.

Java Development for the Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGDatabaseSetup extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for the Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGDatabaseSetup is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getDataSource	Gets the name of the OLE DB data source from which Autodesk MapGuide Server retrieves data.
getKeyColumn	Gets the database column name that contains the keys of the map features on this layer.
getKeyColumnType	Gets the key column type.
getLatColumn	Gets the name of the database column that contains the latitude (or *Y* coordinate) values of the map features on this layer.
getLonColumn	Gets the name of the database column that contains the longitude (or *X* coordinate) values of the map features on this layer.
getNameColumn	Gets the name of the database column that contains the names of the map features on this layer.
getSymbolAngleColumn	Gets the name of the column containing the rotation angle of symbols for point features.
getSymbolHeightColumn	Gets the name of the column containing the height of symbols for point features.
getSymbolWidthColumn	Gets the name of the column containing the width of symbols for point features.
getTable	Gets the database table name, within the OLE DB data source, from which Autodesk MapGuide Server retrieves data for this layer.
getTextAngleColumn	Gets the name of the database column that contains the rotation angle for text map features on this layer.
getTextHeightColumn	Gets the name of the database column that contains the height for text map features on this layer.
getTextHorizAlignColumn	Gets the name of the column containing horizontal alignment values for text.
getTextVertAlignColumn	Gets the name of the column containing vertical alignment values for text.
getUrlLinkColumn	Gets the name of the database column that contains URLs that are linked to map features on this layer.
getWhereClause	Gets the SQL Where clause that Autodesk MapGuide Server uses to query the OLE DB data source that provides data for this layer.
setDataSource	Sets the name of the OLE DB data source from which the Autodesk MapGuide Server retrieves its data.
setKeyColumn	Sets the database column you specify as the key column for this layer.
setKeyColumnType	Sets the data type of the key column to the data type you specify.
setLatColumn	Sets the name of the database column that contains the latitude (or *Y* coordinate) value of map features on this layer.

setLonColumn	Sets the name of the database column that contains the longitude (or *X* coordinate) values for the map features on this layer.
setNameColumn	Sets the name of the database column containing the names of the map features on this layer.
setSymbolAngleColumn	Sets the name of the column containing the rotation angle of symbols for point features.
setSymbolHeightColumn	Sets the name of the column containing the height of symbols for point features.
setSymbolWidthColumn	Sets the name of the column containing the width of symbols for point features.
setTable	Sets the name of the database table, within the OLE DB data source, from which Autodesk MapGuide Server retrieves data for this layer.
setTextAngleColumn	Sets the name of the database column that contains the rotation angle for the text map features on this layer.
setTextHeightColumn	Sets the name of the database column that contains the height of the text map features on this layer.
setTextHorizAlignColumn	Sets the name of the column containing horizontal alignment values for text features.
setTextVertAlignColumn	Sets the name of the column containing vertical alignment values for text features.
setUrlLinkColumn	Sets the name of the database column that contains the URLs that are linked to map features on this layer.
setWhereClause	Sets the SQL Where clause that Autodesk MapGuide uses to query the OLE DB data source that provides data for this layer.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

Properties that set layer setup attributes set the layer's rebuild flag, but not update the display. To update the display, call the refresh method of MGMap.

DataSource	Gets or sets the name of the OLE DB data source from which Autodesk MapGuide Server retrieves data for this layer.
KeyColumn	Gets or sets the name of the database column that contains the keys of the map features on this layer.
KeyColumnType	Gets or sets the data type of the database column containing the keys of map features.

LatColumn	Gets or sets the name of the database column that contains the latitude (or *Y* coordinate) values of the map features on this layer.
LonColumn	Gets or sets the name of the database column that contains the longitude (or *X* coordinate) values of the map features on this layer.
NameColumn	Gets or sets the name of the database column containing the names of map features on this layer.
SymbolAngleColumn	Gets or sets the name of the column containing the rotation angle of symbols for point features.
SymbolHeightColumn	Gets or sets the name of the column containing the height of symbols for point features.
SymbolWidthColumn	Gets or sets the name of the column containing the width of symbols for point features.
Table	Gets or sets the name of the database table, within the OLE DB data source, from which Autodesk MapGuide Server retrieves data for this layer.
TextAngleColumn	Gets or sets the name of the database column that contains the rotation angle of text map features on this layer.
TextHeightColumn	Gets or sets the name of the database column that contains the height of text map features on this layer.
TextHorizAlignColumn	Gets or sets the name of the column containing horizontal alignment values for text.
TextVertAlignColumn	Gets or sets the name of the column containing vertical alignment values for text.
UrlLinkColumn	Gets or sets the name of the database column that contains the URLs that are linked to map features on this layer.
WhereClause	Gets or sets the SQL Where clause that Autodesk MapGuide Server uses to query the OLE DB data source that provides data for this layer.

MGDatabaseSetup.getDataSource

Syntax

```
String getDataSource()
```

Description

Gets the name of the OLE DB data source from which Autodesk MapGuide Server retrieves data.

Returns

String - The name of the OLE DB data source, or empty string if this method does not apply.

Error Codes

none

See Also

setDataSource, DataSource

MGDatabaseSetup.getKeyColumn

Syntax

```
String getKeyColumn()
```

Description

Gets the database column name that contains the keys of the map features on this layer.

Returns

String - The name of the key column, or empty string if this method does not apply.

Error Codes

none

See Also

setKeyColumn, KeyColumn

MGDatabaseSetup.getKeyColumnType

Syntax

```
String getKeyColumnType()
```

Description

Gets the key column type.

Returns

String - The key column data type. Values can be: Boolean, Byte, Date, Decimal, Double, Float, Integer, Numeric, SmallInt, String, Time, TimeStamp, UnsignedByte, UnsignedInteger, or UnsignedSmallInt.

In previous releases, this method also returned Char, Real, VarChar, or LogVarChar types. Autodesk MapGuide Viewers reading version 4.*x* MWFs will now return these types as String, Float, String, and String, respectively.

We recommend that you do not use Double, Float, or Real types because of the inherent inaccuracies in comparing them.

Note that the Autodesk MapGuide Viewer API uses string values while the OLEDBFeatureDataSource KeyColumnType element in Map Window XML (MWX) files uses enumerated constants. For more information about MWX files, refer to the *Autodesk MapGuide Dynamic Authoring Toolkit Developer's Guide*.

Error Codes

-15 (does not apply)

See Also

setKeyColumnType, KeyColumnType

MGDatabaseSetup.getLatColumn

Syntax

```
String getLatColumn()
```

Description

Gets the name of the database column that contains the latitude (or *Y* coordinate) values of the map features on this layer.

Returns

String - The name of the latitude (or *Y* coordinate) column, or an empty string if this method does not apply.

Error Codes

-15 (does not apply)

See Also

setLatColumn, LatColumn, getLonColumn, setLonColumn, LonColumn

MGDatabaseSetup.getLonColumn

Syntax

```
String getLonColumn()
```

Description

Gets the name of the database column that contains the longitude (or *X* coordinate) values of the map features on this layer.

Returns

String - The name of the longitude (or *X* coordinate) column, or an empty string if this method does not apply.

Error Codes

-15 (does not apply)

See Also

setLatColumn, getLatColumn, LatColumn, setLonColumn, LonColumn

MGDatabaseSetup.getNameColumn

Syntax

```
String getNameColumn()
```

Description

Gets the name of the database column that contains the names of the map features on this layer.

Returns

String - The column name, or an empty string if this method does not apply.

Error Codes

-15 (does not apply)

See Also

setNameColumn, NameColumn

MGDatabaseSetup.getSymbolAngleColumn

Syntax

```
String getSymbolAngleColumn()
```

Description

Gets the name of the column containing the rotation angle of symbols for point features.

The rotation angle is specified in decimal degrees measured counterclockwise.

Returns

String - The column name, or an empty string if this method does not apply.

Error Codes

-15 (does not apply)

See Also

setSymbolHeightColumn, SymbolHeightColumn

MGDatabaseSetup.getSymbolHeightColumn

Syntax

```
String getSymbolHeightColumn()
```

Description

Gets the name of the column containing the height of symbols for point features.

The symbol height is specified in Mapping Coordinate System (MCS) units.

Returns

String - The column name, or an empty string if this method does not apply.

Error Codes

-15 (does not apply)

See Also

setSymbolAngleColumn, SymbolAngleColumn

MGDatabaseSetup.getSymbolWidthColumn

Syntax

```
String getSymbolWidthColumn()
```

Description

Gets the name of the column containing the width of symbols for point features.

The symbol width is specified in Mapping Coordinate System (MCS) units.

Returns

String - The column name, or an empty string if this method does not apply.

Error Codes

-15 (does not apply)

See Also

setSymbolWidthColumn, SymbolWidthColumn

MGDatabaseSetup.getTable

Syntax

```
String getTable()
```

Description

Gets the database table name, within the OLE DB data source, from which Autodesk MapGuide Server retrieves data for this layer.

Returns

String - The name of the database table.

Error Codes

-15 (does not apply)

See Also

setTable, Table

MGDatabaseSetup.getTextAngleColumn

Syntax

```
String getTextAngleColumn()
```

Description

Gets the name of the database column that contains the rotation angle for text map features on this layer.

Returns

String - The name of the rotation angle column, or an empty string if this method does not apply.

Error Codes

-15 (does not apply)

See Also

setTextAngleColumn, TextAngleColumn

MGDatabaseSetup.getTextHeightColumn

Syntax

```
String getTextHeightColumn()
```

Description

Gets the name of the database column that contains the height for text map features on this layer.

Returns

String - The name of the text height column, or an empty string if this method does not apply.

Error Codes

-15 (does not apply)

See Also

setTextHeightColumn, TextHeightColumn

MGDatabaseSetup.getTextHorizAlignColumn

Syntax

```
String getTextHorizAlignColumn()
```

Description

Gets the name of the column containing horizontal alignment values for text.

Valid alignment values found within the column are the integers *0* (left), *1* (center), and *2* (right).

Returns

String - The column name, or an empty string if this method does not apply.

Error Codes

-15 (does not apply)

See Also

setTextHorizAlignColumn, TextHorizAlignColumn

MGDatabaseSetup.getTextVertAlignColumn

Syntax

```
String getTextVertAlignColumn()
```

Description

Gets the name of the column containing vertical alignment values for text.

Valid alignment values found within the column are the integers *0* (bottom), *1* (baseline), *2* (halfline), *3* (capline), and *4* (top).

Returns

String - The column name, or an empty string if this method does not apply.

Error Codes

-15 (does not apply)

See Also

setTextVertAlignColumn, TextVertAlignColumn

MGDatabaseSetup.getUrlLinkColumn

Syntax

```
String getUrlLinkColumn()
```

Description

Gets the name of the database column that contains URLs that are linked to map features on this layer.

Returns

String - The name of the URL column, or an empty string if this method does not apply.

Error Codes

none

See Also

setUrlLinkColumn, UrlLinkColumn

MGDatabaseSetup.getWhereClause

Syntax

```
String getWhereClause()
```

Description

Gets the SQL Where clause that Autodesk MapGuide Server uses to query the OLE DB data source that provides data for this layer.

This method is functionally equivalent to the MGMapLayer getSQLWhere method.

Returns

String - The SQL Where clause, or an empty string if this method does not apply.

Error Codes

none

See Also

getSQLWhere, setWhereClause, WhereClause

MGDatabaseSetup.setDataSource

Syntax

boolean setDataSource(String *dataSource*)

Description

Sets the name of the OLE DB data source from which Autodesk MapGuide Server retrieves its data.

Parameters

dataSource - The name of the OLE DB data source.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

See Also

getDataSource, DataSource

MGDatabaseSetup.setKeyColumn

Syntax

`boolean setKeyColumn(String keyColumn)`

Description

Sets the database column you specify as the key column for this layer.

This column provides the keys for map features on this layer.

Parameters

keyColumn - The name of the key column.

Error Codes

-1 (busy)

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

See Also

getKeyColumn, KeyColumn

MGDatabaseSetup.setKeyColumnType

Syntax

`boolean setKeyColumnType(String keyColumnType)`

Description

Sets the data type of the key column to the data type you specify.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

keyColumnType - The data type you want the key column set to.

Values can be: Boolean, Byte, Date, Decimal, Double, Float, Integer, Numeric, SmallInt, String, Time, TimeStamp, UnsignedByte, UnsignedInteger, or UnsignedSmallInt.

For compatibility with previous releases, setKeyColumnType also accepts the values Char, LongVarChar, Real, or VarChar.

We recommend that you do not use Float, Double, and Real types because of the inherent inaccuracies in comparing them.

Returns

boolean - Returns True and sets the layer's rebuild flag to True if the method is successful.

Error Codes

-1 (busy)

-3 (illegal argument)

-15 (does not apply)

See Also

getKeyColumnType, KeyColumnType

MGDatabaseSetup.setLatColumn

Syntax

`boolean setLatColumn(String latColumn)`

Description

Sets the name of the database column that contains the latitude (or *Y* coordinate) value of map features on this layer.

Parameters

latColumn - The name of the latitude (or Y) column.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getLatColumn, getLonColumn, setLonColumn, LatColumn, LonColumn

MGDatabaseSetup.setLonColumn

Syntax

```
boolean setLonColumn(String lonColumn)
```

Description

Sets the name of the database column that contains the longitude (or *X* coordinate) values for the map features on this layer.

Parameters

lonColumn - The name of the longitude (or X) column.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

setLatColumn, getLatColumn, getLonColumn, LonColumn, LatColumn

MGDatabaseSetup.setNameColumn

Syntax

```
boolean setNameColumn(String column)
```

Description

Sets the name of the database column containing the names of the map features on this layer.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

column - The name you want the column set to.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getNameColumn, NameColumn

MGDatabaseSetup.setSymbolAngleColumn

Syntax

`boolean setSymbolAngleColumn(String column)`

Description

Sets the name of the column containing the rotation angle of symbols for point features.

Rotation angles are specified in decimal degrees measured counterclockwise and can range from -360.0º to 360.0º.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

column - The column name.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getSymbolAngleColumn, SymbolAngleColumn

MGDatabaseSetup.setSymbolHeightColumn

Syntax

`boolean setSymbolHeightColumn(String column)`

Description

Sets the name of the column containing the height of symbols for point features.

Symbol height is specified in Mapping Coordinate System (MCS) units.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

column - The column name.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getSymbolHeightColumn, SymbolHeightColumn

MGDatabaseSetup.setSymbolWidthColumn

Syntax

```
boolean setSymbolWidthColumn(String column)
```

Description

Sets the name of the column containing the width of symbols for point features.

Symbol width is specified in Mapping Coordinate System (MCS) units.

Note that this method does not work during a busy state. For additional information about the busy state, see "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

column - The column name.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getSymbolWidthColumn, SymbolWidthColumn

MGDatabaseSetup.setTable

Syntax

```
boolean setTable(String table)
```

Description

Sets the name of the database table, within the OLE DB data source, from which Autodesk MapGuide Server retrieves data for this layer.

Parameters

table - The name of the database table.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

See Also

getTable, Table

MGDatabaseSetup.setTextAngleColumn

Syntax

```
boolean setTextAngleColumn(String angleColumn)
```

Description

Sets the name of the database column that contains the rotation angle for the text map features on this layer.

Parameters

angleColumn - The name of the rotation angle column.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getTextAngleColumn, TextAngleColumn

MGDatabaseSetup.setTextHeightColumn

Syntax

```
boolean setTextHeightColumn(String heightColumn)
```

Description

Sets the name of the database column that contains the height of the text map features on this layer.

Parameters

heightColumn - The name of the text height column.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getTextHeightColumn, TextHeightColumn

MGDatabaseSetup.setTextHorizAlignColumn

Syntax

```
boolean setTextHorizAlignColumn(String column)
```

Description

Sets the name of the column containing horizontal alignment values for text features.

Valid alignment values found within the column are the integers *0* (left), *1* (center), and *2* (right).

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

column - The column name.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getTextHorizAlignColumn, TextHorizAlignColumn

MGDatabaseSetup.setTextVertAlignColumn

Syntax

`boolean setTextVertAlignColumn(String column)`

Description

Sets the name of the column containing vertical alignment values for text features.

Valid alignment values found within the column are the integers *0* (bottom), *1* (baseline), *2* (halfline), *3* (capline), and *4* (top).

Note that this method does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

column - The column name.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getTextVertAlignColumn, TextVertAlignColumn

MGDatabaseSetup.setUrlLinkColumn

Syntax

`boolean setUrlLinkColumn(String linkColumn)`

Description

Sets the name of the database column that contains the URLs that are linked to map features on this layer.

Parameters

linkColumn - The name of the URL column.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

See Also

getUrlLinkColumn, UrlLinkColumn

MGDatabaseSetup.setWhereClause

Syntax

`boolean setWhereClause(String whereClause)`

Description

Sets the SQL Where clause that Autodesk MapGuide uses to query the OLE DB data source that provides data for this layer.

This method is functionally equivalent to the MGMapLayer setSQLWhere method.

Note

If you use the comparison operators =, <, >, <>, or <=, the value is assumed to be a string. Thus, numerical comparisons can return unexpected results. For example, if the SQL Where clause includes FEATURE_ID > 55, a row in the database table that has a feature ID of 6 would be returned, because the string "6" is greater than the string "55".

Parameters

whereClause - The SQL Where clause to use to run queries on the data source.

Returns

boolean - Returns True and sets the layer's rebuild flag to True if successful. Returns False if the method does not apply.

Error Codes

-1 (busy)

See Also

getWhereClause, WhereClause

MGDatabaseSetup.DataSource

Description

Gets or sets the name of the OLE DB data source from which Autodesk MapGuide Server retrieves data for this layer.

Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getDataSource, setDataSource

MGDatabaseSetup.KeyColumn

Description

Gets or sets the name of the database column that contains the keys of the map features on this layer.

Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getKeyColumn, setKeyColumn

MGDatabaseSetup.KeyColumnType

Description

Gets or sets the data type of the database column containing the keys of map features.

Values can be: Boolean, Byte, Date, Decimal, Double, Float, Integer, Numeric, SmallInt, String, Time, TimeStamp, UnsignedByte, UnsignedInteger, or UnsignedSmallInt.

For compatibility with previous releases, this property also accepts the values Char, LongVarChar, Real, or VarChar.

We recommend that you do not use Float, Double, or Real types because of the inherent inaccuracies in comparing them.

This property returns an empty string if it does not apply. If modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getKeyColumnType, setKeyColumnType

MGDatabaseSetup.LatColumn

Description

Gets or sets the name of the database column that contains the latitude (or *Y* coordinate) values of the map features on this layer.

Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getLatColumn, setLatColumn

MGDatabaseSetup.LonColumn

Description

Gets or sets the name of the database column that contains the longitude (or *X* coordinate) values of the map features on this layer.

Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getLonColumn, setLonColumn

MGDatabaseSetup.NameColumn

Description

Gets or sets the name of the database column containing the names of map features on this layer.

Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getNameColumn, setNameColumn

MGDatabaseSetup.SymbolAngleColumn

Description

Gets or sets the name of the column containing the rotation angle of symbols for point features.

The rotation angle is specified in decimal degrees measured counterclockwise. Rotation angles are specified in decimal degrees measured counterclockwise and can range from -360.0º to 360.0º.

This property returns an empty string if it does not apply. If modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getSymbolAngleColumn, setSymbolAngleColumn

MGDatabaseSetup.SymbolHeightColumn

Description

Gets or sets the name of the column containing the height of symbols for point features.

Symbol height is specified in Mapping Coordinate System (MCS) units.This property returns an empty string if it does not apply. If modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getSymbolHeightColumn, setSymbolHeightColumn

MGDatabaseSetup.SymbolWidthColumn

Description

Gets or sets the name of the column containing the width of symbols for point features.

Symbol width is specified in Mapping Coordinate System (MCS) units.This property returns an empty string if it does not apply. If modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getSymbolWidthColumn, setSymbolWidthColumn

MGDatabaseSetup.Table

Description

Gets or sets the name of the database table, within the OLE DB data source, from which Autodesk MapGuide Server retrieves data for this layer.

Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the layer's rebuild flag to true.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getTable, setTable

MGDatabaseSetup.TextAngleColumn

Description

Gets or sets the name of the database column that contains the rotation angle of text map features on this layer.

Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getTextAngleColumn, setTextAngleColumn

MGDatabaseSetup.TextHeightColumn

Description

Gets or sets the name of the database column that contains the height of text map features on this layer.

Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getTextHeightColumn, setTextHeightColumn

MGDatabaseSetup.TextHorizAlignColumn

Description

Gets or sets the name of the column containing horizontal alignment values for text.

Valid alignment values found within the column are the integers *0* (left), *1* (center), and *2* (right).

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getTextHorizAlignColumn, setTextHorizAlignColumn

MGDatabaseSetup.TextVertAlignColumn

Description

Gets or sets the name of the column containing vertical alignment values for text.

Valid alignment values found within the column are the integers *0* (bottom), *1* (baseline), *2* (halfline), *3* (capline), and *4* (top).

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getTextVertAlignColumn, setTextVertAlignColumn

MGDatabaseSetup.UrlLinkColumn

Description

Gets or sets the name of the database column that contains the URLs that are linked to map features on this layer.

Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getUrlLinkColumn, setUrlLinkColumn

MGDatabaseSetup.WhereClause

Description

Gets or sets the SQL Where clause that Autodesk MapGuide Server uses to query the OLE DB data source that provides data for this layer.

Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

This property is equivalent to the MGMapLayer SQLWhere property.

Note

If you use the comparison operators =, <, >, <>, or <=, the value is assumed to be a string. Thus, numerical comparisons can return unexpected results. For example, if the SQL Where clause includes FEATURE_ID > 55, a row in the database table that has a feature ID of 6 would be returned, because the string "6" is greater than the string "55".

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getWhereClause, setWhereClause

MGDwgDataSources Object

This object implements methods and properties to support Autodesk drawing (DWG) data sources. Autodesk DWG is a worldwide standard drawing file format across vertical industries, such as architectural design, and facilities planning and maintenance.

For an example that demonstrates filtering DWG data sources, refer to the Filter DWG Layers example.

Understanding Layers in Autodesk MapGuide

When working with Autodesk DWG data especially, it is important to understand the difference between *Autodesk DWG layers*, *Autodesk MapGuide map layers*, and *Autodesk DWG map layers*. These are not the same and are defined as follows:

Autodesk DWG layer

A single layer within an Autodesk drawing file (Autodesk DWG).

Autodesk MapGuide map layer

A single map layer defined within a MapGuide map window file (MWF). There can be several types of MapGuide map layers, including ones sourced from SDF, SHP, DWG, and so on.

Autodesk DWG map layer

A single Autodesk MapGuide map layer that is sourced from an Autodesk DWG or Autodesk Map project drawing.

Changing Map Layer Data Source Properties

When using Viewer API methods that set Autodesk MapGuide map layer data source properties, including those for Autodesk DWG data, keep in mind that the changes you make can affect the validity of other data source properties for that layer. For example, if you changed the DWG data source using the setDataSource method (or the DataSource property), this change could make existing values for other properties (such as the Autodesk DWG file) invalid since they might not exist in the new data source.

Methods

getDataSource	NEW Gets the DWG data source name that establishes a connection between Autodesk MapGuide Server and the data provider.
getDwg	NEW Gets the Autodesk DWG file name or Autodesk Map project drawing for the Autodesk DWG layer.
getKeyColumn	NEW Gets the name of the column that contains the primary key for each map feature on the DWG layer.
getKeyColumnType	NEW Gets the DWG key column data type.
getKeyTable	NEW Gets the name of the DWG key table to use as primary keys for either map feature properties, external database links, or object data in the Autodesk DWG.

getLayerFilter — NEW Gets the layer filter, which specifies which layers to extract from the specified Autodesk DWG file or Autodesk Map project drawing.

getNameColumn — NEW Gets the name of the column that contains the name for each map feature.

getNameTable — NEW Gets the name of the DWG table for MapTips about map feature properties, external database links, or object data in the Autodesk DWG.

getUrlColumn — NEW Gets the name of the column that contains the URL associated with each map feature.

getUrlTable — NEW Gets the name of the DWG URL table containing URLs to which Autodesk DWG map features are linked.

setDataSource — NEW Sets the DWG data source name that establishes a connection between Autodesk MapGuide Server and the data provider.

setDwg — NEW Sets the Autodesk DWG file name or Autodesk Map project drawing for the Autodesk DWG layer.

setKeyColumn — NEW Sets the name of the column that contains the primary key for each map feature on the DWG layer.

setKeyColumnType — NEW Sets the DWG key column data type.

setKeyTable — NEW Sets the name of the DWG key table to use as primary keys for either feature properties, external database links, or object data in the Autodesk DWG.

setLayerFilter — NEW Sets the layer filter, which specifies which layers to extract from the specified Autodesk DWG or project file.

setNameColumn — NEW Sets the name of the column that contains the name for each map feature.

setNameTable — NEW Sets the name of the DWG table for MapTips about feature properties, external database links, or object data in the Autodesk DWG.

setUrlColumn — NEW Sets the name of the column that contains the URL associated with each map feature.

setUrlTable — NEW Sets the name of the DWG URL table containing URLs to which Autodesk DWG map features are linked.

Properties

Note that direct access to properties is only available in the Autodesk MapGuide ActiveX Control API. In the Autodesk MapGuide Plug-In and Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition.

DataSource — NEW Gets or sets the DWG data source name that establishes a connection between Autodesk MapGuide Server and the data provider.

Dwg — NEW Gets or sets the Autodesk DWG file name or project file for the Autodesk DWG layer.

KeyColumn	NEW! Gets or sets the name of the column that contains the primary key for each map feature on the DWG layer.
KeyColumnType	NEW! Gets or sets the DWG key column data type.
KeyTable	NEW! Gets or sets the name of the DWG key table to use as primary keys for either feature properties, external database links, or object data in the Autodesk DWG.
LayerFilter	NEW! Gets or sets the layer filter, which specifies which layers to extract from the specified Autodesk DWG or project file.
NameColumn	NEW! Gets or sets the name of the column that contains the name for each map feature.
NameTable	NEW! Gets or sets the name of the DWG table for MapTips about feature properties, external database links, or object data in the Autodesk DWG.
UrlColumn	NEW! Gets or sets the name of the column that contains the URL associated with each map feature.
UrlTable	NEW! Gets or sets the name of the DWG URL table containing URLs to which Autodesk DWG map features are linked.

MGDwgDataSources.getDataSource

Syntax

```
String getDataSource()
```

Description

Gets the DWG data source name that establishes a connection between Autodesk MapGuide Server and the data provider.

This method is not affected by the busy state.

Parameters

none

Returns

String - The name of the Data Source.

Error Codes

none

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function getProperties()
{
   // Get the map and selected features
```

```
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   var dataSource = dwgMapLayerDataSource.getDataSource();
}
```

See Also

setDataSource, DataSource

MGDwgDataSources.getDwg

Syntax

```
String getDwg()
```

Description

Gets the Autodesk DWG file name or Autodesk Map project drawing for the Autodesk DWG layer.

If the DWG contains no path, then the first one found in the DWG search path (ADL) will be used.

This method is not affected by the busy state.

Parameters

none

Returns

String - The name of the Autodesk DWG file or Autodesk Map project drawing.

Error Codes

none

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function getProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   var dwgFile = dwgMapLayerDataSource.getDwg();
}
```

See Also

setDwg, Dwg

MGDwgDataSources.getKeyColumn

Syntax

```
string getKeyColumn()
```

Description

Gets the name of the column that contains the primary key for each map feature on the DWG layer.

This method is not affected by the busy state.

Parameters

none

Returns

String - The name of the key column.

Error Codes

none

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function getProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   var keyColumn = dwgMapLayerDataSource.getKeyColumn();
}
```

See Also

setKeyColumnType, KeyColumn

MGDwgDataSources.getKeyColumnType

Syntax

```
string getKeyColumnType()
```

Description

Gets the DWG key column data type.

This method is not affected by the busy state.

Parameters

None

Returns

String - The key column data type. Values can be: Boolean, Byte, Date, Decimal, Double, Float, Integer, Numeric, SmallInt, String, Time, TimeStamp, UnsignedByte, UnsignedInteger, and UnsignedSmallInt.

We recommend that you do not use Float or Double types because of the inherent inaccuracies in comparing them.

Note that the Autodesk MapGuide Viewer API uses string values while the DWGFeatureDataSource KeyColumnType element in Map Window XML (MWX) files uses enumerated constants. For more information about MWX, refer to the *Autodesk MapGuide Dynamic Authoring Toolkit Developer's Guide*.

Error Codes

none

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function getProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   var keyColumnType = dwgMapLayerDataSource.getKeyColumnType();
}
```

See Also

setKeyColumnType, KeyColumnType

MGDwgDataSources.getKeyTable

Syntax

```
string getKeyTable()
```

Description

Gets the name of the DWG key table to use as primary keys for either map feature properties, external database links, or object data in the Autodesk DWG.

This method is not affected by the busy state.

Parameters

none

Returns

String - The name of the key table.

Error Codes

none

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function getProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   var keyTable = dwgMapLayerDataSource.getKeyTable();
}
```

See Also

setKeyTable, KeyTable

MGDwgDataSources.getLayerFilter

Syntax

```
String getLayerFilter()
```

Description

Gets the layer filter, which specifies which layers to extract from the specified Autodesk DWG file or Autodesk Map project drawing.

The layer filter is a comma-delimited string.

This method is not affected by the busy state.

Parameters

none

Returns

String - The layer filter.

Error Codes

none

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function getProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   var layerFilter = dwgMapLayerDataSource.getLayerFilter();
}
```

See Also

setLayerFilter, LayerFilter

MGDwgDataSources.getNameColumn

Syntax

```
string getNameColumn()
```

Description

Gets the name of the column that contains the name for each map feature.

The feature's name appears in a MapTip and in the status bar when the user pauses the mouse pointer over the feature.

This method is not affected by the busy state.

Parameters

none

Returns

String - The name of the Name column.

Error Codes

none

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function getProperties()
{
```

```
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   var nameColumn = dwgMapLayerDataSource.getNameColumn();
}
```

See Also

setNameColumn, NameColumn

MGDwgDataSources.getNameTable

Syntax

```
string getNameTable()
```

Description

Gets the name of the DWG table for MapTips about map feature properties, external database links, or object data in the Autodesk DWG.

This method is not affected by the busy state.

Parameters

none

Returns

String - The name of the name table.

Error Codes

none

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function getProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   var nameTable = dwgMapLayerDataSource.getNameTable();
}
```

See Also

setNameTable, NameTable

MGDwgDataSources.getUrlColumn

Syntax

```
string getUrlColumn()
```

Description

Gets the name of the column that contains the URL associated with each map feature.

This enables the user to go to a Web page that is related to a map feature simply by double-clicking that feature. Typically, you store HTTP URLs in this column, but you could also specify commands with other protocols, such as FTP URLs, or even JavaScript commands. Use caution, however, with non-standard protocols such as Javascript, as not all browsers support them.

This method is not affected by the busy state.

Parameters

none

Returns

String - The Name of the URL column.

Error Codes

none

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function getProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   var urlColumn = dwgMapLayerDataSource.getUrlColumn();
}
```

See Also

setUrlColumn, UrlColumn

MGDwgDataSources.getUrlTable

Syntax

```
string getUrlTable()
```

Description

Gets the name of the DWG URL table containing URLs to which Autodesk DWG map features are linked.

URLs can link to external databases, Web sites, and files via HTTP, FTP, or file protocols.

This method is not affected by the busy state.

Parameters

none

Returns

String - The name of the URL table.

Error Codes

none

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function getProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   var urlTable = dwgMapLayerDataSource.getUrlTable();
}
```

See Also

setUrlTable, UrlTable

MGDwgDataSources.setDataSource

Syntax

```
boolean setDataSource(string dataSource)
```

Description

Sets the DWG data source name that establishes a connection between Autodesk MapGuide Server and the data provider.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

When changing the DWG data source, several other properties may also have to be updated. These properties include Dwg, KeyTable, KeyColumn, KeyColumnType, NameTable, NameColumn, UrlTable, UrlColumn, and LayerFilter.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

dataSource - The name of the data source.

Returns

boolean - Returns True if successful.

Error Codes

-1 (busy)

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function setProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   dwgMapLayerDataSource.setDataSource("newDataSource");
}
```

See Also

getDataSource, DataSource

MGDwgDataSources.setDwg

Syntax

```
boolean setDwg(string DWG)
```

Description

Sets the Autodesk DWG file or Autodesk Map project drawing for the Autodesk DWG layer.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information. If the DWG contains no path, then the first one found in the DWG search path will be used.

When changing the Autodesk DWG file, several other properties may also have to be updated, including KeyTable, KeyColumn, KeyColumnType, NameTable, NameType, NameColumn, UrlTable, UrlColumn, and LayerFilter.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

DWG - The name of the Autodesk DWG file.

Returns

boolean - Returns True if successful.

Refresh

NA

Error Codes

-1 (busy)

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function setProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   dwgMapLayerDataSource.setDwg("newDwgFile");
}
```

See Also

getDwg, Dwg

MGDwgDataSources.setKeyColumn

Syntax

```
boolean setKeyColumn(string column)
```

Description

Sets the name of the column that contains the primary key for each map feature on the DWG layer.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

column - The name that you want the key column to be set to.

Returns

boolean - Returns True if successful and rebuild occurs when the property changes.

Error Codes

-1 (busy)

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function setProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   dwgMapLayerDataSource.setKeyColumn("newKeyColumn");
}
```

See Also

getKeyColumn, KeyColumn

MGDwgDataSources.setKeyColumnType

Syntax

```
boolean setKeyColumnType(string keyColumnType)
```

Description

Sets the DWG key column data type.

This method is not case-sensitive.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

keyColumnType - The data type that you want the key column to be set to. Values can be: Boolean, Byte, Date, Decimal, Double, Float, Integer, Numeric, SmallInt, String, Time, TimeStamp, Unsigned Byte, UnsignedInteger, or UnsignedSmallInt.

Returns

boolean - Returns True if successful; otherwise returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function setProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   dwgMapLayerDataSource.setKeyColumnType("Integer");
}
```

See Also

getKeyColumnType, KeyColumnType

MGDwgDataSources.setKeyTable

Syntax

```
boolean setKeyTable(string keyTable)
```

Description

Sets the name of the DWG key table to use as primary keys for either feature properties, external database links, or object data in the Autodesk DWG.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

When changing the key table, several other properties may also have to be updated, including KeyColumn and KeyColumnType.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

keyTable - The name of the key table.

Returns

boolean - Returns True if successful and rebuild occurs when the property changes.

Error Codes

-1 (busy)

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function setProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   dwgMapLayerDataSource.setKeyTable("newKeyTable");
}
```

See Also

getKeyTable, KeyTable

MGDwgDataSources.setLayerFilter

Syntax

`boolean setLayerFilter(string layerFilter)`

Description

Sets the DWG layer filter to identify the DWG layers to extract from the Autodesk DWG file or Autodesk Map project drawing.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

layerFilter - The layer filter. The layer filter is a comma-delimited string containing the filters for each DWG layer to be applied. An empty layer filter string ("") applies all filter layers.

Returns

boolean - Returns True if successful; otherwise returns False.

Refresh

N/A

Error Codes

-1 (busy)

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function setProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   // Set the layer filter to show Interior Walls only
   dwgMapLayerDataSource.setLayerFilter("Interior Walls");
   map.refresh();
   // Set the layer filter to show Interior Walls and Equipment only
   dwgMapLayerDataSource.setLayerFilter("Interior Walls, Equipment");
   map.refresh();
   // Set the layer filter to show all DWG layers
   dwgMapLayerDataSource.setLayerFilter("");
   map.refresh();
}
```

See Also

getLayerFilter, LayerFilter

MGDwgDataSources.setNameColumn

Syntax

```
boolean setNameColumn(string column)
```

Description

Sets the name of the column that contains the name for each map feature.

The feature's name appears in a MapTip and in the status bar when the user pauses the mouse pointer over the feature.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

column - The name that you want the name column to be set to.

Returns

boolean - Returns True if successful; otherwise, returns False. If successful, sets the layer's rebuild flag to True.

Error Codes

-1 (busy)

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function setProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   dwgMapLayerDataSource.setNameColumn("newNameColumn");
}
```

See Also

getNameColumn, NameColumn

MGDwgDataSources.setNameTable

Syntax

```
boolean setNameTable(string NameTable)
```

Description

Sets the name of the DWG table for MapTips about feature properties, external database links, or object data in the Autodesk DWG.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

When changing the name table, several other properties may also have to be updated, including NameColumn.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

NameTable - The name of the name table.

Returns

boolean - Returns True if successful and rebuild occurs when the property changes.

Refresh

NA

Error Codes

-1 (busy)

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function setProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   dwgMapLayerDataSource.setNameTable("newNameTable");
}
```

See Also

getNameTable, NameTable

MGDwgDataSources.setUrlColumn

Syntax

```
boolean setUrlColumn(string column)
```

Description

Sets the name of the column that contains the URL associated with each map feature.

This enables the user to go to a Web page that is related to a map feature simply by double-clicking that feature. Typically, you store HTTP URLs in this column, but you could also specify commands with other protocols, such as FTP URLs, or even JavaScript commands. Use caution, however, with non-standard protocols such as Javascript:, as not all browsers support them.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

column - The name that you want the URL column to be set to.

Returns

boolean - Returns True if successful; otherwise returns False.

Error Codes

-1 (busy)

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function setProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   dwgMapLayerDataSource.setUrlColumn("newUrlColumn");
}
```

See Also

getUrlColumn, UrlColumn

MGDwgDataSources.setUrlTable

Syntax

`boolean setUrlTable(string urlTable)`

Description

Sets the name of the DWG URL table containing URLs to which Autodesk DWG map features are linked.

URLs can link to external databases, Web sites, and files via HTTP, FTP, or file protocols.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

When changing the URL table, several other properties may also have to be updated, including URLColumn.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

urlTable - The name of the URL table.

Returns

boolean - Returns True if successful and rebuild occurs when the property changes.

Error Codes

-1 (busy): For write operations only.

Example

This code is designed to work with the *1stfloorplanEx6.mwf* file used by the *Filter DWG Layers* example:

```
function setProperties()
{
   // Get the map and selected features
   var map = getMap();
   var mapSel = map.getSelection();
   // Use the single map layer Floor Plan
   var mapLayer = map.getMapLayer("Floor Plan");
   // Get DWG data source object for the layer
   var mapLayerSetup = mapLayer.getLayerSetup();
   var dwgMapLayerDataSource = mapLayerSetup.getDwgDataSources();
   dwgMapLayerDataSource.setUrlTable("newUrlTable");
}
```

See Also

getUrlTable, UrlTable

MGDwgDataSources.DataSource

Description

Gets or sets the DWG data source name that establishes a connection between Autodesk MapGuide Server and the data provider.

If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

When changing the DWG Data Source, several other properties may also have to be updated. These properties include Dwg, KeyTable, KeyColumn, KeyColumnType, NameTable, NameColumn, UrlTable, UrlColumn, and LayerFilter.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getDataSource, setDataSource

MGDwgDataSources.Dwg

Description

Gets or sets the Autodesk DWG file name or project file for the Autodesk DWG layer.

If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that if the DWG contains no path, then the first one found in the DWG search path will be used.

When changing the Autodesk DWG file, several other properties may also have to be updated.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getDwg, setDwg

MGDwgDataSources.KeyColumn

Description

Gets or sets the name of the column that contains the primary key for each map feature on the DWG layer.

If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getKeyColumn, setKeyColumn

MGDwgDataSources.KeyColumnType

Description

Gets or sets the DWG key column data type.

If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Values can be: Boolean, Byte, Date, Decimal, Double, Float, Integer, Numeric, SmallInt, String, Time, TimeStamp, UnsignedByte, UnsignedInteger, and UnsignedSmallInt.

We recommend that you do not use Floatand Double types because of the inherent inaccuracies in comparing them.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write). This property is not case-sensitive.

Error Codes

-1 (busy): For write operations only.

-3 (illegal argument)

See Also

getKeyColumnType, setKeyColumnType

MGDwgDataSources.KeyTable

Description

Gets or sets the name of the DWG key table to use as primary keys for either feature properties, external database links, or object data in the Autodesk DWG.

If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

When changing the key table, several other properties may also have to be updated.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getKeyTable, setKeyTable

MGDwgDataSources.LayerFilter

Description

Gets or sets the layer filter, which specifies which layers to extract from the specified Autodesk DWG or project file.

If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that the layer filter is a comma-delimited string.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getLayerFilter, setLayerFilter

MGDwgDataSources.NameColumn

Description

Gets or sets the name of the column that contains the name for each map feature.

The feature’s name appears in a MapTip and in the status bar when the user pauses the mouse pointer over the feature.

Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getNameColumn, setNameColumn

MGDwgDataSources.NameTable

Description

Gets or sets the name of the DWG table for MapTips about feature properties, external database links, or object data in the Autodesk DWG.

If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

When changing the Name table, several other properties may also have to be updated. Properties include NameColumn.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getNameTable, setNameTable

MGDwgDataSources.UrlColumn

Description

Gets or sets the name of the column that contains the URL associated with each map feature.

This enables the user to go to a Web page that is related to a map feature simply by double-clicking that feature. Typically, you store HTTP URLs in this column, but you could also specify commands with other protocols, such as FTP URLs, or even JavaScript commands. Use caution, however, with non-standard protocols such as Javascript:, as not all browsers support them.

Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getUrlColumn, setUrlColumn

MGDwgDataSources.UrlTable

Description

Gets or sets the name of the DWG URL table containing URLs to which Autodesk DWG map features are linked.

URLs can link to external databases, Web sites, and files via HTTP, FTP, or file protocols.

If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

When changing the URL table, several other properties may also have to be updated,including UrlColumn.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getUrlTable, setUrlTable

MGEdgeAttr Object

The MGEdgeAttr object contains the polygon edge attributes used to draw buffer or redline polygons.

Buffer polygons are used when Autodesk MapGuide executes the view buffer command. To get an instance of the MGEdgeAttr object for use as a buffer, first use the MGMap.getBufferSetup method to get an instance of the MGBufferSetup object. Then use the MGBufferSetup.getEdgeAttr method.

Redline polygons are used as primitive features on redline layers; they can't currently be created through the Autodesk MapGuide user interface, but only through the API. To get an instance of the MGEdgeAttr object for use as a redline feature, first use the MGMap.getRedlineSetup method to get an instance of the MGRedlineSetup object. Then use the MGRedlineSetup.getEdgeAttr method.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGEdgeAttr extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGEdgeAttr is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getColor	Gets the edge color index used to draw the polygons.
getStyle	Gets the edge style used to draw the polygons.
getThickness	Gets the edge thickness used to draw the polygons.
getVisible	Returns a value indicating whether the edge should be drawn around the polygons.
setColor	Specifies the edge color index used to draw the polygons.
setStyle	Specifies the edge style used to draw the polygons.
setThickness	Specifies the edge thickness used to draw the polygons.

setVisible Specifies whether the edge should be drawn around the polygons.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

Color Gets or sets the edge color index used to draw the polygons.

Style Gets or sets the edge style used to draw the polygons.

Thickness Gets or sets the edge thickness used to draw the polygons.

Visible Gets or sets whether the edge should be drawn around the polygons.

MGEdgeAttr.getColor

Syntax

```
int getColor()
```

Description

Gets the edge color index used to draw the polygons.

Returns

int - Color index taken from the color controls in Autodesk MapGuide Author. Values can range from 1 to 256.

Error Codes

none

See Also

setColor, Color

MGEdgeAttr.getStyle

Syntax

```
String getStyle()
```

Description

Gets the edge style used to draw the polygons.

Returns

String - The name of the line style. See Autodesk MapGuide Line Styles for a list of available line style names.

Error Codes

none

See Also

setStyle, Style

MGEdgeAttr.getThickness

Syntax

```
int getThickness()
```

Description

Gets the edge thickness used to draw the polygons.

Returns

int - The thickness in pixels. Values can range from 1 to 16384.

Error Codes

none

See Also

setThickness, Thickness

MGEdgeAttr.getVisible

Syntax

```
boolean getVisible()
```

Description

Returns a value indicating whether the edge should be drawn around the polygons.

Returns

boolean - Returns True if the edge should be visible; otherwise, false.

Error Codes

none

See Also

setVisible, Visible

MGEdgeAttr.setColor

Syntax

```
void setColor(int color)
```

Description

Specifies the edge color index used to draw the polygons.

Parameters

color - Color index taken from the color controls in Autodesk MapGuide Author. Values can range from 1 to 256.

Error Codes

-3 (illegal argument)

See Also

getColor, Color

MGEdgeAttr.setStyle

Syntax

```
void setStyle(String style)
```

Description

Specifies the edge style used to draw the polygons.

Parameters

style - The name of the line style. See Autodesk MapGuide Line Styles for a list of available line style names.

Error Codes

-3 (illegal argument)

See Also

getStyle, Style

MGEdgeAttr.setThickness

Syntax

```
void setThickness(int thickness)
```

Description

Specifies the edge thickness used to draw the polygons.

Parameters

thickness - The thickness in pixels. Values can range from 1 to 16384.

Error Codes

-3 (illegal argument)

See Also

getThickness, Thickness

MGEdgeAttr.setVisible

Syntax

`void setVisible(boolean visible)`

Description

Specifies whether the edge should be drawn around the polygons.

Parameters

visible - True if the edge should be visible; otherwise, false.

Error Codes

none

See Also

getVisible, Visible

MGEdgeAttr.Color

Description

Gets or sets the edge color index used to draw the polygons.

Type

int (read/write)

Error Codes

-3 (illegal argument)

See Also

getColor, setColor

MGEdgeAttr.Style

Description

Gets or sets the edge style used to draw the polygons.

See Autodesk MapGuide Line Styles for a list of available line style names.

Type

String (read/write)

Error Codes

-3 (illegal argument)

See Also

getStyle, setStyle

MGEdgeAttr.Thickness

Description

Gets or sets the edge thickness used to draw the polygons.

Type

int (read/write)

Error Codes

-3 (illegal argument)

See Also

getThickness, setThickness

MGEdgeAttr.Visible

Description

Gets or sets whether the edge should be drawn around the polygons.

Type

boolean (read/write)

Error Codes

none

See Also

getVisible, setVisible

MGError Object

The MGError object contains error diagnostic information about the most recently executed API method.

Each time the Autodesk MapGuide Viewer application calls an API method, the existing MGError object expires, and a new one is created.

You can get the most recent instance of the MGError object with MGMap.getLastError.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGError extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGError is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getArg Gets the number of an incorrect argument in the event that an illegal argument is passed.

getCode Gets the error code of the MGError object.

isError Returns the error status of the MGError object.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

Arg The number of the incorrect argument in the case of an illegal argument.

Code The error code in the MGError object.

Error The error status in the MGError object.

MGError.getArg

Syntax

```
int getArg()
```

Description

Gets the number of an incorrect argument in the event that an illegal argument is passed.

Returns

int - the argument number, or 0 if there was no illegal argument.

Error Codes

none

Example

```
function checkArgType()
{
   var map = getMap();
   var arg = map.getLastError().getArg();
   alert("ERROR: " + arg);
}
```

See Also

getLastError

MGError.getCode

Syntax

```
int getCode()
```

Description

Gets the error code of the MGError object.

Returns

int - A predefined error code. Current error codes are:

-999	Unknown	**-9**	MapGuide Internal
-99	Null Pointer	**-8**	Construction Failed
-16	Write Permission Denied	**-7**	No Such Element
-15	Does Not Apply	**-6**	Invalid Object
-14	Refresh Disabled	**-5**	Security Violation
-13	Translation Failed	**-3**	Illegal Argument

-12	Invalid Selection	**-2**	Not Ready
-11	Layer's View Disabled	**-1**	Busy
-10	Design Time Only	**0**	None

Error Codes

none

Example

```
function checkError()
{
   var map = getMap();
   var code = map.getLastError().getCode();
   alert("ERROR: " + code);
}
```

See Also

getLastError

MGError.isError

Syntax

```
boolean isError()
```

Description

Gets the error status of the MGError object.

Returns

boolean - Returns True if there is an error; False if there is no error.

Error Codes

none

MGError.Arg

Description

Gets the number of the incorrect argument in the case of an illegal argument.

Type

int (read-only)

Error Codes

none

See Also

getArg

MGError.Code

Description

Gets the error code in the MGError object.

Type

int (read-only)

Error Codes

none

See Also

getCode

MGError.Error

Description

Gets the error status in the MGError object.

Type

boolean (read-only)

Error Codes

none

See Also

getCode

MGExtent Object

The MGExtent object represents an extent in lat/lon coordinates.

Note that instances of the class MGExent cannot be publicly created.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGExtent extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGExtent is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

equals	Compares this MGExtent object to the specified object.
getMaxLat	Gets the maximum latitude (or *Y* coordinate) of the extent in decimal degrees.
getMaxLon	Gets the maximum longitude (or *X* coordinate) of the extent in decimal degrees.
getMinLat	Gets the minimum latitude (or *Y* coordinate) of the extent in decimal degrees.
getMinLon	Gets the minimum longitude (or *X* coordinate) of the extent in decimal degrees.

Properties

Note that direct access to properties is only available in the Autodesk MapGuide ActiveX Control API. In the Autodesk MapGuide Plug-In and Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition.

MaxLat	The maximum latitude (or *Y* coordinate) of the extent in decimal degrees.
MaxLon	The maximum longitude (or *X* coordinate) of the extent in decimal degrees.
MinLat	The minimum latitude (or *Y* coordinate) of the extent.
MinLon	The minimum longitude (or *X* coordinate) of the extent.

MGExtent.equals

Syntax

```
boolean equals(Object obj)
```

Description

Compares this MGExtent object to the specified object.

Parameters

obj - The object with which to compare this MGExtent.

Returns

boolean - Returns True if the two objects are identical; otherwise, returns False.

Error Codes

-3 (illegal argument)

MGExtent.getMaxLat

Syntax

```
double getMaxLat()
```

Description

Gets the maximum latitude (or *Y* coordinate) of the extent in decimal degrees.

Returns

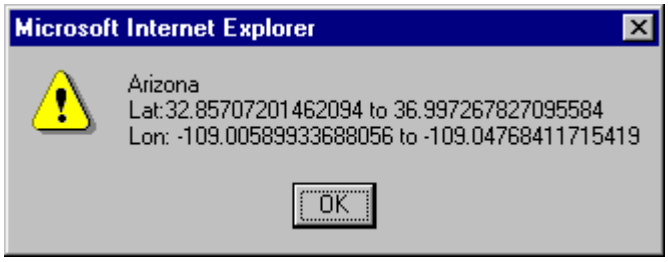
double - The maximum latitude.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the minimum and maximum lat/lon values for each selected object. The values for each object are displayed in an alert similar to the following illustration:




```
function showLatLon()
{
    var map = getMap();
    // Assign collection of selected features to variable
    var selected = map.getSelection().getMapObjectsEx(null);
    // If collection is not empty, step through its members
    if (selected.size() > 0)
    {
        // for each object in collection...
        for (i = 0; i < selected.size(); i++)
        {
            // ...get name and extents
            name = selected.item(i).getName();
            ext = selected.item(i).getExtent();
            // ...and display alert showing lat/lon values for object's extents
            alert (name + "\nLat:" + ext.getMaxLat() + " to " + ext.getMaxLat() +
"\nLon: " + ext.getMaxLon() + " to " + ext.getMaxLon())
        }
    }
    // If collection is empty, display alert
    else
        alert("Please select a feature.");
}
```

See Also

MaxLat

MGExtent.getMaxLon

Syntax

```
double getMaxLon()
```

Description

Gets the maximum longitude (or *X* coordinate) of the extent in decimal degrees.

Returns

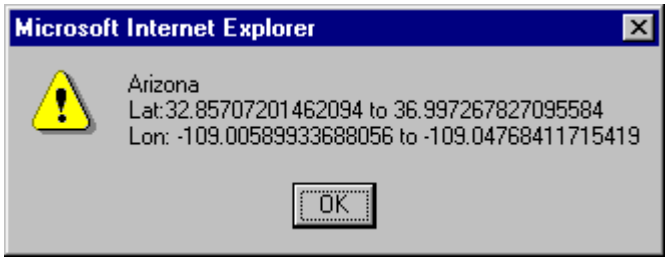
double - The maximum longitude.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the minimum and maximum lat/lon values for each selected object. The values for each object are displayed in an alert similar to the following illustration:




```
function showLatLon()
{
    var map = getMap();
    // Assign collection of selected features to variable
    var selected = map.getSelection().getMapObjectsEx(null);
    // If collection is not empty, step through its members
    if (selected.size() > 0)
    {
        // for each object in collection...
        for (i = 0; i < selected.size(); i++)
        {
            // ...get name and extents
            name = selected.item(i).getName();
            ext = selected.item(i).getExtent();
            // ...and display alert showing lat/lon values for object's extents
            alert (name + "\nLat:" + ext.getMaxLat() + " to " + ext.getMaxLat() +
"\nLon: " + ext.getMaxLon() + " to " + ext.getMaxLon())
        }
    }
    // If collection is empty, display alert
    else
```

```
        alert("Please select a feature.");
}
```

See Also

MaxLon

MGExtent.getMinLat

Syntax

```
double getMinLat()
```

Description

Gets the minimum latitude (or *Y* coordinate) of the extent in decimal degrees.

Returns

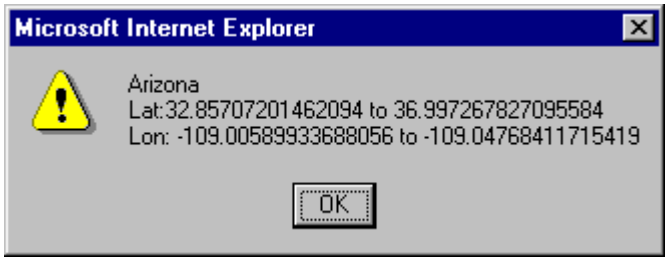
double - The minimum latitude.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the minimum and maximum lat/lon values for each selected object. The values for each object are displayed in an alert similar to the following illustration:




```
function showLatLon()
{
    var map = getMap();
    // Assign collection of selected features to variable
    var selected = map.getSelection().getMapObjectsEx(null);
    // If collection is not empty, step through its members
    if (selected.size() > 0)
    {
        // for each object in collection...
        for (i = 0; i < selected.size(); i++)
        {
            // ...get name and extents
            name = selected.item(i).getName();
            ext = selected.item(i).getExtent();
            // ...and display alert showing lat/lon values for object's extents
            alert (name + "\nLat:" + ext.getMaxLat() + " to " + ext.getMaxLat() +
"\nLon: " + ext.getMaxLon() + " to " + ext.getMaxLon())
        }
```

```
    }
    // If collection is empty, display alert
    else
        alert("Please select a feature.");
}
```

See Also

MinLat

MGExtent.getMinLon

Syntax

```
double getMinLon()
```

Description

Gets the minimum longitude (or *X* coordinate) of the extent in decimal degrees.

Returns

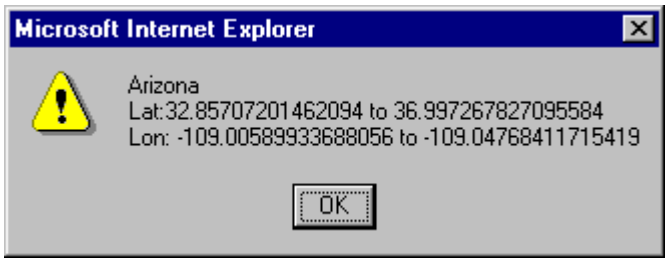
double - The miminum longitude.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the minimum and maximum lat/lon values for each selected object. The values for each object are displayed in an alert similar to the following illustration:




```
function showLatLon()
{
    var map = getMap();
    // Assign collection of selected features to variable
    var selected = map.getSelection().getMapObjectsEx(null);
    // If collection is not empty, step through its members
    if (selected.size() > 0)
    {
        // for each object in collection...
        for (i = 0; i < selected.size(); i++)
        {
            // ...get name and extents
            name = selected.item(i).getName();
            ext = selected.item(i).getExtent();
            // ...and display alert showing lat/lon values for object's extents
```

```
            alert (name + "\nLat:" + ext.getMaxLat() + " to " + ext.getMaxLat() +
"\nLon: " + ext.getMaxLon() + " to " + ext.getMaxLon())
        }
    }
    // If collection is empty, display alert
    else
        alert("Please select a feature.");
}
```

See Also

MinLon

MGExtent.MaxLat

Description

Gets the maximum latitude (or *Y* coordinate) of the extent in decimal degrees.

Type

double (read-only)

Error Codes

none

See Also

getMaxLat

MGExtent.MaxLon

Description

Gets the maximum longitude (or *X* coordinate) of the extent in decimal degrees.

Type

double (read-only)

Error Codes

none

See Also

getMaxLon

MGExtent.MinLat

Description

Gets the minimum latitude (or *Y* coordinate) of the extent.

Type

double (read-only)

Error Codes

none

See Also

getMinLat

MGExtent.MinLon

Description

Gets the minimum longitude (or *X* coordinate) of the extent.

Type

double (read-only)

Error Codes

none

See Also

getMinLon

MGExtentEx Object

The MGExtentEx object represents spatial extents or a bounding box.

This object can be created using the createObject method of MGMap and it is returned from other methods such as MGMap.getMapExtent, MGMapObject.getExtentEx, and MGPrimitive.getExtent.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGExtentEx extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGExtentEx is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

equals Compares this MGExtentEx to the specified object.

getMaxX Gets the maximum *X* value, or longitude, of the extent.

getMaxY Gets the maximum *Y* value, or latitude, of the extent.

getMinX Gets the minimum *X* value, or longitude, of the extent.

getMinY Gets the minimum *Y* value, or latitude, of the extent.

set Sets the extents of the object.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

MaxX Gets the maximum *X* value, or longitude, of the extent.

MaxY Gets the maximum *Y* value, or latitude, of the extent.

MinX Gets the minimum *X* value, or longitude, of the extent.

MinY Gets the minimum *Y* value, or latitude, of the extent.

MGExtentEx.equals

Syntax

```
boolean equals(Object obj)
```

Description

Compares this MGExtentEx to the specified object.

Parameters

obj - The object with which to compare the MGExtentEx.

Returns

boolean - Returns True if the two objects are identical; otherwise, returns False.

Error Codes

-3 (illegal argument)

MGExtentEx.getMaxX

Syntax

```
double getMaxX()
```

Description

Gets the maximum *X* value, or longitude, of the extent.

Returns

double - maximum *X* value of the extent.

Error Codes

none

MGExtentEx.getMaxY

Syntax

```
double getMaxY()
```

Description

Gets the maximum *Y* value, or latitude, of the extent.

Returns

double - maximum *Y* value of the extent.

Error Codes

none

MGExtentEx.getMinX

Syntax

```
double getMinX()
```

Description

Gets the minimum *X* value, or longitude, of the extent.

Returns

double - minimum *X* value of the extent.

Error Codes

none

MGExtentEx.getMinY

Syntax

```
double getMinY()
```

Description

Gets the minimum *Y* value, or latitude, of the extent.

Returns

double - minimum *Y* value of the extent.

Error Codes

none

MGExtentEx.set

Syntax

`boolean set(double minX, double minY, double maxX, double maxY)`

Description

Sets the extents of the object.

Parameters

minX - the minimum *X* value (longitude) of the extent.

minY - the minimum *Y* value (latitude) of the extent.

maxX - the maximum *X* value (longitude) of the extent.

maxY - the maximum *Y* value (latitude) of the extent.

Returns

boolean - Returns True if successful. Returns False if either minimum value is greater than or equal to its corresponding maximum value, unless all four values have been set to zero.

Error Codes

-3 (illegal argument)

See Also

getMinX, getMinY, getMaxX, getMaxY

MGExtentEx.MaxX

Description

Gets the maximum *X* value, or longitude, of the extent.

Type

double (read-only)

Error Codes

none

See Also

getMaxX

MGExtentEx.MaxY

Description

Gets the maximum *Y* value, or latitude, of the extent.

Type

double (read-only)

Error Codes

none

See Also

getMaxY

MGExtentEx.MinX

Description

Gets the minimum *X* value, or longitude, of the extent.

Type

double (read-only)

Error Codes

none

See Also

getMinX

MGExtentEx.MinY

Description

Gets the minimum *Y* value, or latitude, of the extent.

Type

double (read-only)

Error Codes

none

See Also

getMinY

MGFillAttr Object

The MGFillAttr object contains the polygon fill attributes used to draw buffer or redline polygons.

Buffer polygons are used when Autodesk MapGuide executes the view buffer command. To get an instance of the MGFillAttr object for use as a buffer, first use the MGMap.getBufferSetup method to get an instance of the MGBufferSetup object. Then use the MGBufferSetup.getFillAttr method.

Redline polygons are used as primitive features on redline layers; they can't currently be created through the Autodesk MapGuide user interface, only through the API. To get an instance of the MGFillAttr object for use as a redline feature, first use the MGMap.getRedlineSetup method to get an instance of the MGRedlineSetup object. Then use the MGRedlineSetup.getFillAttr method.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGFillAttr extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGFillAttr is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getBackColor	Gets the background color index used to draw the polygons.
getBackMode	Gets the background mode used to draw the polygons.
getColor	Gets the fill color index used to draw the polygons.
getHatchPattern	Gets the fill hatch pattern used to draw the polygons.
getStyle	Gets the fill style used to draw the polygons.
setBackColor	Specifies the background color index used to draw the polygons.
setBackMode	Specifies the background mode used to draw the polygons.
setColor	Specifies the fill color index used to draw the polygons.
setHatchPattern	Specifies the fill hatch pattern used to draw the polygons.
setStyle	Specifies the fill style used to draw the polygons.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

BackColor	Gets or sets the background color index used to draw the polygons.
BackMode	Gets or sets the background mode used to draw the polygons.
Color	Gets or sets the fill color index used to draw the polygons.
HatchPattern	Gets or sets the fill hatch pattern used to draw the polygons.
Style	Gets or sets the fill style used to draw the polygons.

MGFillAttr.getBackColor

Syntax

```
int getBackColor()
```

Description

Gets the background color index used to draw the polygons.

Returns

int - Color index taken from the color controls in Autodesk MapGuide Author. Values can range from 1 to 256.

Error Codes

none

See Also

setBackColor, BackColor

MGFillAttr.getBackMode

Syntax

```
String getBackMode()
```

Description

Gets the background mode used to draw the polygons.

Returns

String - The background mode, either Transparent or Opaque.

Error Codes

none

See Also

setBackMode, BackMode

MGFillAttr.getColor

Syntax

```
int getColor()
```

Description

Gets the fill color index used to draw the polygons.

Returns

int - Color index taken from the color controls in Autodesk MapGuide Author. Values can range from 1 to 256.

Error Codes

none

See Also

setColor, Color

MGFillAttr.getHatchPattern

Syntax

```
String getHatchPattern()
```

Description

Gets the fill hatch pattern used to draw the polygons.

Returns

String - The hatch pattern. See Autodesk MapGuide Fill Styles for a list of available hatch patterns.

Error Codes

none

See Also

setHatchPattern, HatchPattern

MGFillAttr.getStyle

Syntax

```
String getStyle()
```

Description

Gets the fill style used to draw the polygons.

Returns

String - The fill style. Value can be None, Solid, Hatch, or the name of one of the Autodesk MapGuide Fill Styles. If the value is Hatch, you can use getHatchPattern to get the fill style.

Error Codes

none

See Also

setStyle, Style

MGFillAttr.setBackColor

Syntax

```
void setBackColor(int color)
```

Description

Specifies the background color index used to draw the polygons.

Parameters

color - Color index taken from the color controls in Autodesk MapGuide Author. Values can range from 1 to 256.

Error Codes

-3 (illegal argument)

See Also

getBackColor, BackColor

MGFillAttr.setBackMode

Syntax

```
void getBackMode(String mode)
```

Description

Specifies the background mode used to draw the polygons.

Parameters

mode - The background mode, either Transparent or Opaque.

Error Codes

-3 (illegal argument)

See Also

getBackMode, BackMode

MGFillAttr.setColor

Syntax

```
void setColor(int color)
```

Description

Specifies the fill color index used to draw the polygons.

Parameters

color - Color index taken from the color controls in Autodesk MapGuide Author. Values can range from 1 to 256.

Error Codes

-3 (illegal argument)

See Also

getColor, Color

MGFillAttr.setHatchPattern

Syntax

```
void setHatchPattern(String style)
```

Description

Specifies the fill hatch pattern used to draw the polygons.

In Release 4.0, you had to use two methods to specify the hatch: you would set setStyle to Hatch, and then use setHatchPattern to specify the pattern. As of Release 5, you can specify the hatch pattern directly with setStyle. However, setHatchPattern is still available for backward compatibility.

Parameters

style - The hatch pattern. See Autodesk MapGuide Fill Styles for a list of available hatch patterns.

Error Codes

-3 (illegal argument)

See Also

getHatchPattern, HatchPattern

MGFillAttr.setStyle

Syntax

```
void setStyle(String style)
```

Description

Specifies the fill style used to draw the polygons.

Parameters

style - The fill style. Value can be None, Solid, or the name of one of the Autodesk MapGuide Fill Styles. For backward compatiblity with Release 4.0, you can also specify Hatch and then specify the hatch pattern using setHatchPattern.

Error Codes

-3 (illegal argument)

See Also

getStyle, Style

MGFillAttr.BackColor

Description

Gets or sets the background color index used to draw the polygons.

Type

int (read/write)

Error Codes

-3 (illegal argument)

See Also

getBackColor, setBackColor

MGFillAttr.BackMode

Description

Gets or sets the background mode used to draw the polygons.

Type

String (read/write)

Error Codes

-3 (illegal argument)

See Also

getBackMode, setBackMode

MGFillAttr.Color

Description

Gets or sets the fill color index used to draw the polygons.

Type

int (read/write)

Error Codes

-3 (illegal argument)

See Also

getColor, setColor

MGFillAttr.HatchPattern

Description

Gets or sets the fill hatch pattern used to draw the polygons.

See setHatchPattern for more information about setting hatch patterns.

Type

String (read/write)

Error Codes

-3 (illegal argument)

See Also

getHatchPattern, setHatchPattern

MGFillAttr.Style

Description

Gets or sets the fill style used to draw the polygons.

See setStyle for more information about setting fill styles.

Type

String (read/write)

Error Codes

-3 (illegal argument)

See Also

getStyle, setStyle

MGLineAttr Object

The MGLineAttr object contains the attributes used to draw lines and polylines on redline layers.

Redline features can't currently be created through the Autodesk MapGuide user interface, only through the API. To get an instance of the MGLineAttr object for use as a redline feature, first use the MGMap.getRedlineSetup method to get an instance of the MGRedlineSetup object. Then use the MGRedlineSetup.getLineAttr method.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGLineAttr extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGLineAttr is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getColor	Gets the color index used to draw the lines or polylines.
getStyle	Gets the style used to draw the lines or polylines.
getThickness	Gets the thickness used to draw the lines or polylines.
setColor	Specifies the color index used to draw the lines or polylines.
setStyle	Specifies the style used to draw the lines or polylines.
setThickness	Specifies the thickness used to draw the lines or polylines.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

Color	Gets or sets the color index used to draw the lines or polylines.
Style	Gets or sets the style used to draw the lines or polylines.
Thickness	Gets or sets the thickness used to draw the lines or polylines.

MGLineAttr.getColor

Syntax

```
int getColor()
```

Description

Gets the color index used to draw the lines or polylines.

Returns

int - The color index, based on the color controls in Autodesk MapGuide Author; values range from 1 to 256.

Error Codes

none

See Also

setColor, Color

MGLineAttr.getStyle

Syntax

```
String getStyle()
```

Description

Gets the style used to draw the lines or polylines.

Returns

String - The name of the line style. See Autodesk MapGuide Line Styles for a list of available line styles.

Error Codes

none

See Also

setStyle, Style

MGLineAttr.getThickness

Syntax

```
int getThickness()
```

Description

Gets the thickness used to draw the lines or polylines.

Returns

int - The thickness in pixels; values range from 1 to 16384.

Error Codes

none

See Also

setThickness, Thickness

MGLineAttr.setColor

Syntax

```
void setColor(int color)
```

Description

Specifies the color index used to draw the lines or polylines.

Parameters

int - The color index, based on the color controls in Autodesk MapGuide Author; values range from 1 to 256.

Error Codes

-3 (illegal argument)

See Also

getColor, Color

MGLineAttr.setStyle

Syntax

```
void setStyle(String style)
```

Description

Specifies the style used to draw the lines or polylines.

Parameters

style - The name of the line style. See Autodesk MapGuide Line Styles for a list of available line styles.

Error Codes

-3 (illegal argument)

See Also

getStyle, Style

MGLineAttr.setThickness

Syntax

`void setThickness(int thickness)`

Description

Specifies the thickness used to draw the lines or polylines.

Parameters

thickness - The thickness in pixels; values range from 1 to 16384.

Error Codes

-3 (illegal argument)

See Also

getThickness, Thickness

MGLineAttr.Color

Description

Gets or sets the color index used to draw the lines or polylines.

Type

int (read/write)

Error Codes

-3 (illegal argument)

See Also

getColor, setColor

MGLineAttr.Style

Description

Gets or sets the style used to draw the lines or polylines.

See Autodesk MapGuide Line Styles for a list of available line styles.

Type

String (read/write)

Error Codes

-3 (illegal argument)

See Also

getStyle, setStyle

MGLineAttr.Thickness

Description

Gets or sets the thickness used to draw the lines or polylines.

Type

int (read/write)

Error Codes

-3 (illegal argument)

See Also

getThickness, setThickness

MGMap Object

MGMap is the top-level object of the API for the Autodesk MapGuide ActiveX Control, Plug-In, and Java Edition. For the ActiveX Control, MGMap is a COM interface.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, MGMap is a concrete class that resides in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGMap is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the

mgjava.cab file (for Windows systems only) and the *mgjava.jar* file (for all platforms). MGMap is implemented by the concrete classes MGMapApplet and MGMapComponent.

Example

The following is an example of accessing an MGMap object. If your embedded map has the name myMap, the code to access it in a simple, frameless HTML page will look something like this:

```
// name is "getMap()", but it can be anything you like
function getMap()
{
    if (navigator.appName() == "Netscape")
        return document.myMap;   // access map with Navigator
    else
        return window.myMap;     // access map with IE
}
```

If your application has multiple frames, the code to access the map in a frame called "Left" will look something like this:

```
{
    if (navigator.appName() == "Netscape")
        return parent.Left.document.myMap;
    else
        return parent.Left.myMap;
}
```

Methods

aboutDlg	Displays the Autodesk MapGuide About dialog box, which shows information about Autodesk MapGuide Viewer.
addMapLayer	Downloads and adds the specified Map Layer File (*.mlf*) to the map.
addMapLayers	Downloads and adds the specified Map Layer Files to the map.
copyMap	Copies the current view of the map to the clipboard as an Enhanced Windows Metafile.
copyUrl	Copies the URL of the displayed map to the clipboard.
createLayer	Adds a layer of the specified type to the map.
createMapLayerGroup	Creates a new map layer group and adds it to the map.
createObject	Creates and returns an object of the specified type.
digitizeCircle	Enables the user to define a circle on the map by clicking and dragging the mouse.
digitizeLine	Enables the user to draw a line on the map by clicking and dragging the mouse.
digitizePoint	Enables the user to define a point on the map by clicking the mouse.
digitizePolygon	Enables the user to use the mouse to define a polygon on the map, then fires the event onDigitizedPolygon upon completion.

digitizePolygonEx	Enables the user to use the mouse to define a polygon on the map, then fires the event onDigitizedPolygon upon completion.
digitizePolyline	Enables a user to use the mouse to define a polyline on the map, then fires the event onDigitizedPolyline upon completion.
digitizePolylineEx	Enables a user to use the mouse to define a polyline on the map, then fires the event onDigitizedPolyline upon completion.
digitizeRectangle	Enables the user to define a rectangle on the map by clicking and dragging the mouse, then fires the onDigitizedRectangle event upon completion.
disableApiExceptions	Disables Autodesk MapGuide Viewer from throwing exceptions.
disablePrintingEvents	Disables the Autodesk MapGuide Viewer from sending printing events to the associated event handler during a print process.
disableSelectionChangedEvent	Disables the firing of the onSelectionChanged event during calls to the MGMapLayer methods removeObject, removeObjects, and removeAllObjects.
enableApiExceptions	Enables Autodesk MapGuide Viewer exception throwing.
enablePrintingEvents	Enables the Autodesk MapGuide Viewer to send printing events to the associated event handler during a print process.
enableSelectionChangedEvent	Enables the firing of the onSelectionChanged event during calls to the MGMapLayer methods removeObject, removeObjects, and removeAllObjects.
getApiVersion	Gets the Autodesk MapGuide Viewer API version.
getAutoLinkDelay	Gets the number of seconds before the next HTML page associated with a map feature from the AutoLinkLayer will be displayed in the AutoLinkTarget window.
getAutoLinkLayers	Gets the name of the layer that uses the AutoLink feature.
getAutoLinkTarget	Gets the name of the target window where Autodesk MapGuide will display the HTML pages associated with map features on the AutoLinkLayer.
getAutoRefresh	Returns a boolean value that indicates whether or not the automatic refresh feature is enabled.
getBufferSetup	Gets an object that contains buffering properties, such as the distance from the selected features to draw the buffers.
getBusyStateChangedObserver	Returns a reference to the Java object that observes busy state change events, or null if no observer has been established.
getDefaultTarget	Gets the name of the default target window where Autodesk MapGuide will display HTML documents, including autolinks, errors, feature links, reports, and URL lists.
getDisplayUnits	Gets the current type of display units being used to display the map width, as seen in the status bar.

getDoubleClickObserver	Returns a reference to the Java object being used to observe double-click events, or null if no observer has been established.
getErrorTarget	Gets the name of the target window where Autodesk MapGuide will display error messages.
getHeight	Gets the height of the map.
getIntermediateUpdatesEnabled	NEW Get the current setting of the intermediate updates flag.
getLastError	Returns a reference to the MGError object of the most recent Autodesk MapGuide Viewer API method call or property access.
getLat	Gets the latitude (*Y*) value in decimal degrees of the center of the map area currently displayed.
getLon	Gets the longitude (*X*) value in decimal degrees of the center of the map area currently displayed.
getMapExtent	Gets the extents for the map or current map view.
getMapLayer	Gets the specified map layer.
getMapLayerGroup	Gets the specified map layer group.
getMapLayerGroups	Gets an MGCollection object containing all the map layer groups defined in the map.
getMapLayers	Returns a Vector (or MGCollection) object containing all the map layers defined in the map.
getMapLayersEx	Gets an MGCollection object containing all the map layers defined in the map.
getMapLoadedObserver	Returns a reference to an MGMapLoadedObserver object or null if no observer has been established.
getMcsScaleFactor	Gets the conversion factor to convert from the current Mapping Coordinate System (MCS) to meters.
getMode	NEW Gets the current mode of the Autodesk MapGuide Viewer.
getName	Gets the name of the map, as set in the Preferences dialog box in Autodesk MapGuide Author, or with setName.
getObjectLinkTarget	Gets the name of the target window where Autodesk MapGuide will display the URL associated with a map feature.
getPageSetup	Gets an object containing properties used by Autodesk MapGuide to print maps.
getPrintingObserver	Returns a reference to the Java object being used to observe printing events.
getRedlineSetup	Gets an object containing all properties for drawing redline features.

getReportTarget	Gets the name of the target window where Autodesk MapGuide will display reports from View Reports command.
getScale	Gets the current map display scale in terms of 1:scale.
getSelection	Gets the currently selected map features.
getSelectionChangedObserver	Returns a reference to the Java object being used to observe selection change events, or null if no observer has been established.
getSelectionMode	Gets the selection mode, which controls how map features are selected when a feature is not fully contained in the selection area.
getSymbols	Gets a collection of strings representing the symbols available for use as symbol attributes.
getTrackingID	Gets the customer tracking ID associated with a map.
getUnits	Gets the current units that will be used when getting or setting the MapWidth property, or when using getWidth or setWidth without specifying the *Units* parameter.
getUrl	Gets the URL of the map that is currently displayed by Autodesk MapGuide Viewer.
getUrlListState	Returns True if Autodesk MapGuide will generate an HTML document that lists the names and URLs associated with a map feature as the user pauses the mouse pointer over that feature.
getUrlListTarget	Gets the name of the target window where Autodesk MapGuide will display the names and URLs associated with a map feature as the user pauses the mouse pointer over that feature.
getVersion	Gets the version string of the Autodesk MapGuide Viewer.
getViewChangedObserver	Returns a reference to the Java object being used to observe view change events, or null if no observer has been established.
getViewChangingObserver	Returns a reference to the Java object that observes when a map is about to be refreshed, or null if no such observer has been established.
getViewDistanceExObserver	Returns a reference to the Java object that observes view distance events, or null if no such observer has been established.
getViewDistanceObserver	Returns a reference to the Java object that observes view distance events, or null if no such observer has been established.
getViewerType	Returns a string indicating whether the current Autodesk MapGuide Viewer is the ActiveX Control, Java Edition, or the Autodesk MapGuide Viewer Plug-In.
getWidth	Gets the current width of the map in the specified unit of measure.
isBusy	Returns True if Autodesk MapGuide Viewer is busy loading or drawing the map.

lonLatToMcs	Converts the specified lat/lon coordinate pair to a point using the map coordinate system (MCS) defined for the MWF by the map author.
mcsToLonLat	Converts the specified Mapping Coordinate System (MCS) coordinate pair to lat/lon coordinates.
pageSetupDlg	Displays the Page Setup dialog box, where the user can specify preferences for printing the map.
panMode	Switches to Pan mode.
preferencesDlg	Displays the Preferences dialog box, where the user can specify the map units, selection mode, and options for the status bar.
printDlg	Uses the page setup properties, then displays the Print dialog box.
redrawExtent	Redraws the specified portion of the map.
redrawLegend	Redraws the map legend.
refresh	Refreshes (redraws) the map.
removeMapLayer	Removes the specified map layer from the map.
removeMapLayerGroup	Removes the specified map layer group from the map.
saveMWF	Saves the map to a local MWF.
selectMapObjectsDlg	Displays the Select Map Features dialog box, where the user can select map features by name from a specific map layer.
selectMode	Switches to Select mode.
selectPolygon	Selects all map features that fall within the specified polygon.
selectPolygonEx	Selects all map features that fall within the specified polygon.
selectPolygonMode	Enters the select polygon mode.
selectRadius	Selects all map features that fall within the specified radius.
selectRadiusMode	Selects Select Radius mode.
selectWithin	Selects features from the specified map layers that fall within the currently selected polygon map features.
selectWithinDlg	Displays the Select Within dialog box, where the user chooses the map layer containing the features that will be selected if they fall within the currently selected features.
setAutoLinkDelay	Sets the number of seconds before the next HTML page associated with a map feature from the AutoLinkLayer will be displayed in the AutoLinkTarget window.
setAutoLinkLayers	Sets the name of the layer that will use the AutoLink feature.
setAutoLinkTarget	Sets the name of the target window where Autodesk MapGuide will display the HTML pages associated with map features on the AutoLinkLayer.

setAutoRefresh	Enables or disables the automatic refresh flag.
setBusyStateChangedObserver	Activates the Java object that observes the Autodesk MapGuide Viewer busy state.
setDefaultTarget	Sets the name of the default target window where Autodesk MapGuide will display HTML documents.
setDoubleClickObserver	Sets the Java object for observing double-click events.
setErrorTarget	Sets the name of the target window where Autodesk MapGuide will display error messages.
setIntermediateUpdatesEnabled	NEW Specifies whether or not the map window should update every 1.5 seconds during a drawing operation or only at the end of the operation when all data has been loaded and rendered.
setMapLoadedObserver	Activates the Java object that observes when Autodesk MapGuide loads MWFs, enabling the onMapLoaded event.
setName	Sets the name of the current map.
setObjectLinkTarget	Sets the name of the target window where Autodesk MapGuide will display the URL associated with a map feature.
setPrintingObserver	Sets the Java object that observes printing events.
setReportTarget	Sets the name of the target window where Autodesk MapGuide will display reports from View Reports command.
setScale	Sets the current map display scale in terms of 1:scale.
setSelectionChangedObserver	Sets the Java object for observing selection change events.
setSelectionMode	Sets the selection mode, which controls how map features are selected when a feature is not fully contained in the selection area.
setTrackingID	Sets the customer tracking ID associated with a map.
setUrl	Sets the URL of the map to display.
setUrlListState	Enables or disables automatic generation of an HTML document listing the names and URLs of map features associated with URLs.
setUrlListTarget	Sets the name of the target window where Autodesk MapGuide will display the names and URLs associated with a map feature as the user pauses the mouse pointer over that feature.
setViewChangedObserver	Sets the Java object for observing view change events.
setViewChangingObserver	Sets the Java object for observing view change events.
setViewDistanceExObserver	Sets the Java object for observing view distance events.
setViewDistanceObserver	Sets the Java object for view distance events.
setWidth	Sets the current width of the map in the specified unit of measure.

stop	Aborts any downloading or redrawing that is currently taking place. Used for ActiveX Control and Plug-In Viewers only.
stopEx	Aborts any downloading or redrawing that is currently taking place. Used for Java Edition only.
viewBuffer	Creates buffers around the selected map features.
viewBufferDlg	Displays the View Buffer dialog box, where the user specifies the settings for creating the buffers.
viewDistance	Enters View Distance mode and fires the onViewedDistance event when complete.
viewDistanceEx	Enters View Distance mode and fires the onViewedDistanceEx event when complete.
viewReport	Invokes the specified report as authored in the map window.
viewReportsDlg	Displays the View Reports dialog box, where the user selects the report to display.
zoomGotoDlg	Displays the Zoom Goto dialog box, where the user specifies the zoom width as well as either a place (location), an address, or a ZIP code to zoom to.
zoomGotoLocation	Zooms to the specified category, location, and scale.
zoomInMode	Enters ZoomIn mode.
zoomOut	Zooms to the full extent of the map window as defined in the map window properties.
zoomOutMode	Enters the ZoomOut mode.
zoomPrevious	Sets the current view of the map back to the previous view of the map.
zoomScale	Zooms to the specified map center (lat/lon) and scale.
zoomScaleDlg	Displays the Zoom Scale dialog box, where the user enters the scale to zoom to.
zoomSelected	Zooms to an extent covering the selected map features.
zoomWidth	Zooms to the specified map center and width.
zoomWidthDlg	Displays the Zoom Width dialog box, where the user enters the width of the map window to zoom to.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

ApiVersion	Gets the API version for Autodesk MapGuide ActiveX Control.
AutoLinkDelay	Gets or sets the number of seconds before the next HTML page associated with a map feature from an AutoLinkLayer will be displayed in the AutoLinkTarget window.
AutoLinkLayers	Gets or sets the name of the layer that you want to use as the AutoLink layer.
AutoLinkTarget	Gets or sets the name of the target window where Autodesk MapGuide will display the HTML pages associated with map features on AutoLinkLayers as the user pauses the mouse pointer over those map features.
AutoRefresh	Gets or sets the automatic refresh flag.
BufferSetup	An object that contains buffering properties, such as the distance from the selected features to the edge of the buffers.
Busy	True if Autodesk MapGuide Viewer is busy loading or drawing a map.
DefaultTarget	Gets or sets the name of the default target window where Autodesk MapGuide will display HTML documents.
DisplayUnits	Gets the current display units, either M (meters), KM (kilometers), FT (feet), or MI (miles), as seen in the map width on the status bar.
ErrorTarget	Gets or sets the name of the target window where Autodesk MapGuide will display error messages.
FullUrl	Gets the full URL of the map that is currently displayed by Autodesk MapGuide, including parameters specifying the current location and any non-default options.
IntermediateUpdatesEnabled	NEW Specifies whether or not the map window should update every 1.5 seconds during a drawing operation or only at the end when all data has been loaded and rendered.
LastError	The MGError object that contains the error value and message for the most recent API error.
Lat	Gets or sets the latitude (*Y*) value in decimal degrees of the center of the map area currently displayed.
LayersViewWidth	Gets or sets the width of the legend. In order to change this property at runtime, it must be non-zero when the MGMap object is created.
Lon	Gets or sets the longitude (*X*) value in decimal degrees of the center of the map area currently displayed.
MapHeight	The height of the map. This value is expressed in the units set with the MapUnits property.
MapLayerGroups	A collection containing all the map layer groups defined in the map.
MapLayers	A collection containing all the map layers defined in the map.
MapName	Gets or sets the name of the map.

MapScale	Gets or sets the current map display scale in terms of 1:scale.
MapUnits	Gets or sets the current units that will be used when getting or setting the MapWidth property, either M (meters), KM (kilometers), FT (feet), or MI (miles).
MapWidth	Gets or sets the current map width.
McsScaleFactor	The conversion factor used to convert from the current Mapping Coordinate System (MCS) to meters.
Mode	NEW Gets the current viewer mode of the map, such as ZoomIn, Digitize, Pan, and so on.
ObjectLinkTarget	Gets or sets the name of the target window where Autodesk MapGuide will display the URL associated with a map feature.
PageSetup	An object that contains properties used for printing maps.
RedlineSetup	Gets an object containing all properties for drawing redline features.
ReportTarget	Gets or sets the name of the target window where Autodesk MapGuide will display reports from View Reports command.
Selection	The currently selected map features.
SelectionMode	Gets or sets the selection mode, either Centroid or Intersection.
StatusBar	Gets or sets the state of the status bar, either Off or On.
Symbols	Gets a collection of strings representing the symbols available for use as point attributes.
ToolBar	Gets or sets the state of the toolbar, either Off, On, or Big.
TrackingID	The customer tracking ID associated with a map.
Url	Gets or sets the URL of the map that is currently displayed by Autodesk MapGuide.
UrlListState	Specifies whether Autodesk MapGuide will generate an HTML document that lists the names and URLs (if any) that are associated with the selected map feature(s).
UrlListTarget	Gets or sets the name of the target window where Autodesk MapGuide will display the names and URLs of map features associated with URLs.
Version	The version string of Autodesk MapGuide Viewer.
ViewerType	A string indicating whether the Autodesk MapGuide Viewer is the Autodesk MapGuide Viewer ActiveX Control, the Autodesk MapGuide Viewer Plug-In, or the Autodesk MapGuide Viewer, Java Edition.

MGMap.aboutDlg

Syntax

```
void aboutDlg()
```

Description

Displays the Autodesk MapGuide About dialog box, which shows information about Autodesk MapGuide Viewer.

This method is the equivalent of the Help > Help About menu command in the user interface.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Error Codes

-1 (busy)

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the Autodesk MapGuide About dialog box:

```
function displayAbout()
{
    getMap().aboutDlg();
}
```

MGMap.addMapLayer

Syntax

ActiveX Control: `void addMapLayer(String URL)`

Plug-In/Java Edition: `void addMapLayer(String URL, MGAddMapLayerObserver observer)`

Description

Downloads and adds the specified Map Layer File (*.mlf*) to the map.

It then fires the event onAddMapLayer either when the file is added, when the stop button is clicked, or when a download error occurs.

For Autodesk MapGuide Viewer Plug-In, the *observer* parameter specifies the object that either handles the event or forwards it back to JavaScript.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

URL - The URL of the Map Layer File.

observer - A Java object that implements MGAddMapLayerObserver (Plug-In and Java Edition only).

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It adds a new layer, named "USStCap.mlf", to a map:

```
function addOneLayer()
{
    var map = getMap();
    var mapUrl = "http://yourserver/maps/USStCap.mlf";
    // check Viewer version (type)...
    if (map.getViewerType() == "ActiveX Control")
        // ...if IE, addMapLayer() takes URL as only argument
        map.addMapLayer(mapUrl);
    else
        // ...if Netscape or Java, second argument also needed
        map.addMapLayer(mapUrl, document.obs);
}
```

For Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition, addMapLayer() requires the observer applet as its second argument. If you are not writing code to handle the onAddMapLayer event, the second argument can be "null", instead of "document.obs":

```
map.addMapLayer(mapUrl, null);
```

See Also

addMapLayers, removeMapLayer, viewBuffer, onAddMapLayer

MGMap.addMapLayers

Syntax

ActiveX Control: `boolean addMapLayers(MGCollection URLs)`

Plug-In/Java Edition: `boolean addMapLayers(MGCollection URLs, MGAddMapLayerObserver observer)`

Description

Downloads and adds the specified Map Layer Files to the map.

It then fires the onAddMapLayer event for each file added. Once downloads are finished, the map is updated.

For Autodesk MapGuide Viewer Plug-In, the *observer* parameter specifies the object that either handles the event or forwards it back to JavaScript.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

URLs - An MGCollection of strings. Each string is a URL that references a map layer file.

observer - A Java object that implements MGAddMapLayerObserver (Plug-In and Java Edition only).

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It adds two layers, named "USStCap.mlf" and "USInterstates.mlf", to a map:

```
function addTwoLayers()
{
    var map = getMap();
    var urlCollection = map.createObject("MGCollection");
    urlCollection.add("http://yourserver/maps/USStCap.mlf")
    urlCollection.add("http://yourserver/maps/USInterstates.mlf")
    /// check Viewer version (type)...
    if (map.getViewerType() == "ActiveX Control")
        // ...if IE, addMapLayers() takes URL as only argument
        map.addMapLayers(urlCollection);
    else
        // ...if Netscape or Java, second argument also needed
        map.addMapLayers(urlCollection, document.obs);
}
```

Like addMapLayer(), addMapLayers() requires a second argument when you use it with Autodesk MapGuide Viewer Plug-In or the Autodesk MapGuide Viewer, Java Edition.

See Also

addMapLayer, removeMapLayer, viewBuffer, onAddMapLayer

MGMap.copyMap

Syntax

```
void copyMap()
```

Description

Copies the current view of the map to the clipboard as an Enhanced Windows Metafile.

This method is the equivalent of the Copy > Copy menu command in the user interface.

Error Codes

-1 (busy)

Example

This simple function is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It copies the current view to the clipboard, where it can then be pasted into a file by the user or by a program you write:

```
function copyMapToClipboard()
{
    getMap().copyMap();
    alert ("Current view has been copied to Clipboard!");
}
```

MGMap.copyUrl

Syntax

```
void copyUrl()
```

Description

Copies the URL of the displayed map to the clipboard.

This method is the equivalent of the Copy > Copy As URL menu command in the user interface.

Note that this method *only* copies the URL; it does not return it as a value.

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It copies the URL of the map to the clipboard, where it can then be pasted into a file by the user or by a program you write:

```
function copyUrlToClipboard()
{
    var url = getMap().copyUrl();
    alert ("Map URL has been copied to the Clipboard!");
}
```

See Also

getUrl, setUrl

MGMap.createLayer

Syntax

```
MGMapLayer createLayer(String type, String name)
```

Description

Creates and adds a layer of the specified type to the map.

At this time, only redline layers can be added to the map. For more information on redline layers, refer to "Applications" in the *Autodesk MapGuide Developer's Guide*.

Parameters

type - The layer type; currently redline is the only supported type.

name - The layer name.

Returns

MGMapLayer - An MGMapLayer object if successful, null otherwise.

Error Codes

-1 (busy)

-3 (illegal argument)

-5 (security violation)

MGMap.createMapLayerGroup

Syntax

`MGMapLayerGroup createMapLayerGroup(String name)`

Description

Creates a new map layer group and adds it to the map.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

name - The name of the map layer group you wish to create. To avoid conflicts, you should give each layer group a unique name.

Returns

MGMapLayerGroup - The new MGMapLayerGroup object.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

removeMapLayerGroup, redrawLegend

MGMap.createObject

Syntax

`object createObject(String type)`

Description

Creates and returns an object of the specified type.

Parameters

type - The type of object you wish to create, either MGCollection, MGExtentEx, or MGPoint. Note that this parameter is not case-sensitive.

Returns

object - The newly created MGCollection, MGExtentEx, or MGPoint object.

Error Codes

-3 (illegal argument)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It shows the use of createObject() with an MGCollection:

```
function selectSomeStates()
{
    // Get map object; then get "States" layer
    map = getMap();
    var mapLayer = map.getMapLayer("States");
    // Create an MGCollection object called mapFeatures
    var mapFeatures = map.createObject("MGCollection");
    // Get features by key, then add to mapFeatures collection
    mapFeatures.add(mapLayer.getMapObject("AZ")); // key is "AZ"
    mapFeatures.add(mapLayer.getMapObject("CA")); // key is "CA"
    mapFeatures.add(mapLayer.getMapObject("NV")); // key is "NV"
    // Select and display contents of mapFeatures
    map.getSelection().addObjectsEx(mapFeatures, true);
}
```

To see an interesting use of this method, refer to the Selection Techniques example.

MGMap.digitizeCircle

Syntax

ActiveX Control: `boolean digitizeCircle(String units)`

Plug-In/Java Edition: `boolean digitizeCircle(String units, MGDigitizeCircleObserver observer)`

Description

Enables the user to define a circle on the map by clicking and dragging the mouse.

It fires the onDigitizedCircle event upon completion. The digitizeCircle method does not perform any functions other than sending the circle coordinates to onDigitizedCircle.

For Autodesk MapGuide Viewer Plug-In, the *observer* parameter specifies the object that either handles the event or forwards it back to JavaScript.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

units - Units to specify radius in, one of the following codes: M (meters), KM (kilometers), FT (feet), MI (miles), "", or null. "" and null apply to the Autodesk MapGuide Viewer Plug-In and Java Edition only. They use the current default units.

observer - A Java object that implements MGDigitizeCircleObserver (Plug-In and Java Edition only).

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

digitizePoint, digitizePolygonEx, digitizePolylineEx

MGMap.digitizeLine

Syntax

ActiveX Control: `boolean digitizeLine()`

Plug-In/Java Edition: `boolean digitizeLine(MGDigitizePolylineObserverEx observer)`

Description

Enables the user to draw a line on the map by clicking and dragging the mouse.

It then fires the onDigitizedPolyline event upon completion. This method does not perform any functions other than sending the line coordinates to onDigitizedPolyline.

For Autodesk MapGuide Viewer Plug-In and Java Edition, the *observer* parameter specifies the object that either handles the event or forwards it back to JavaScript.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

observer - A Java object that implements MGDigitizePolylineObserverEx (Plug-In and Java Edition only).

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

See Also

onDigitizedPolyline

MGMap.digitizePoint

Syntax

ActiveX Control: `boolean digitizePoint()`

Plug-In/Java Edition: `boolean digitizePoint(MGDigitizePointObserver observer)`

Description

Enables the user to define a point on the map by clicking the mouse.

After the point has been specified, digitizePoint fires the onDigitizedPoint event, passing it the point coordinates, the map object, and for Netscape and the Autodesk MapGuide Viewer, Java Edition, the observer applet.

This method is meant to be used with an onDigitizedPoint event handler; see the example for details.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

observer - A Java object that implements MGDigitizePointObserver (Plug-In and Java Edition only).

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It shows how to display the coordinates of a a user-specified point. First, a function named getThePoint invokes the appropriate digitize method:

```
function getThePoint()
{
    if (navigator.appName == "Netscape")
        getMap().digitizePoint(document.obs);
    else
        getMap().digitizePoint();
}
```

Then, if you've set up your event handler properly, digitizePoint() fires the onDigitizedPoint event, passing it the map object and the user-specified point. If the browser is Netscape, the observer applet is passed as

well. See "Handling Events" in the *Autodesk MapGuide Developer's Guide* for information about working with events and observers, including setting up event handlers.

Finally, the onDigitizedPoint event looks for a JavaScript function of the same name and, if that function exists, executes it:

```
function onDigitizedPoint(map, point)
{
    alert("The coordinates are:\n\nLAT = " + point.getY() + ", LON = " +
point.getX());
}
```

See Also

digitizeCircle, digitizePolygonEx, digitizePolylineEx

MGMap.digitizePolygon

Syntax

ActiveX Control: `boolean digitizePolygon()`

Plug-In/Java Edition: `boolean digitizePolygon(MGDigitizePolygonObserver observer)`

Description

Enables the user to use the mouse to define a polygon on the map, then fires the event onDigitizedPolygon upon completion.

digitizePolygon does not perform any functions other than sending the polygon coordinates to onDigitizedPolygon. For Autodesk MapGuide Viewer Plug-In, the *observer* parameter specifies the object that either handles the event or forwards it back to JavaScript. As of Release 1.3 of the API, there is a newer version of this method for use with the Autodesk MapGuide Viewer Plug-In and Java Edition: digitizePolygonEx.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

observer - A Java object that implements MGDigitizePolygonObserver (Plug-In and Java Edition only).

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It shows how to display the coordinates of a a user-specified polygon.

Note: The example works with the Autodesk MapGuide Viewer Plug-In and Java Edition under Netscape, and with the ActiveX Control under Internet Explorer. It does not work with the Autodesk MapGuide Viewer, Java Edition under Internet Explorer. (All methods using the observer applet have this problem,

because Internet Explorer doesn't allow for the passing of non-primitives between JavaScript and Java.) You can get around this limitation by calling the methods from a wrapper applet, as shown in Java Edition: Example 6.

First, a function called getPolygon invokes the appropriate digitize method:

```
function getPolygon()
{
    // If browser is Netscape...
    if (navigator.appName == "Netscape")
        // ...use newer Ex version, and pass observer applet.
        getMap().digitizePolygonEx(document.obs);
    // If browser is IE...
    else
        // ...use older non-Ex version, with no argument.
        getMap().digitizePolygon();
}
```

Then, if you've set up your event handler properly, digitizePolygon() or digitizePolygonEx() fires the onDigitizedPolygon event, passing it the map object, the number of polygon vertices, and the coordinates of those vertices. If the browser is Netscape, the observer applet is passed as well. See "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide* for information about working with events and observers, including setting up event handlers.

Finally, the onDigitizedPolygon event looks for a JavaScript function of the same name and, if that function exists, executes it:

```
function onDigitizedPolygon(map, numPoints, points)
{
    alert("You just digitized a polygon with: " + numPoints + " Points");
    for (i = 0;  i <  numPoints; i++)
        alert("Point[" + i + "] = " + points.item(i).getY() + ", " +
points.item(i).getX());
}
```

To see how these functions work in an application, refer to the Event Handling example.

See Also

onDigitizedPolygon, digitizePolygonEx

MGMap.digitizePolygonEx

Syntax

`boolean digitizePolygonEx(MGDigitizePolygonObserverEx` *`observer`*`)`

Description

Enables the user to use the mouse to define a polygon on the map, then fires the event onDigitizedPolygon on completion.

digitizePolygonEx does not perform any function other than sending the polygon coordinates to onDigitizedPolygon. The *observer* parameter specifies the object that either handles the event or forwards it back to JavaScript.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Applies To

Plug-In and Java Edition only. This method is an updated version of digitizePolygon. digitizePolygon still works in Release 6.

Parameters

observer - A Java object that implements MGDigitizePolygonObserverEx.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It shows how to display the coordinates of a a user-specified polygon.

Note: The example works with the Autodesk MapGuide Viewer Plug-In and Java Edition under Netscape, and with the ActiveX Control under Internet Explorer. It does not work with the Autodesk MapGuide Viewer, Java Edition under Internet Explorer. (All methods using the observer applet have this problem, because Internet Explorer doesn't allow for the passing of non-primitives between JavaScript and Java.) You can get around this limitation by calling the methods from a wrapper applet, as shown in Java Edition: Example 6.

First, a function called getPolygon invokes the appropriate digitize method:

```
function getPolygon()
{
    // If browser is Netscape...
    if (navigator.appName == "Netscape")
        // ...use newer Ex version, and pass observer applet.
        getMap().digitizePolygonEx(document.obs);
    // If browser is IE...
    else
        // ...use older non-Ex version, with no argument.
        getMap().digitizePolygon();
}
```

Then, if you've set up your event handler properly, digitizePolygon() or digitizePolygonEx() fires the onDigitizedPolygon event, passing it the map object, the number of polygon vertices, and the coordinates of those vertices. If the browser is Netscape, the observer applet is passed as well. See "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide* for information about working with events and observers, including setting up event handlers.

Finally, the onDigitizedPolygon event looks for a JavaScript function of the same name and, if that function exists, executes it:

```
function onDigitizedPolygon(map, numPoints, points)
{
    alert("You just digitized a polygon with: " + numPoints + " Points");
    for (i = 0;  i <  numPoints; i++)
        alert("Point[" + i + "] = " + points.item(i).getY() + ", " +
points.item(i).getX());
}
```

To see how these functions work in an application, refer to the Event Handling example.

See Also

digitizeCircle, digitizePoint, digitizePolylineEx, selectPolygonEx

MGMap.digitizePolyline

Syntax

ActiveX Control: `boolean digitizePolyline()`

Plug-In/Java Edition: `boolean digitizePolyline(MGDigitizePolylineObserver observer)`

Description

Enables a user to use the mouse to define a polyline on the map, then fires the event onDigitizedPolyline upon completion.

digitizePolyline does not perform any functions other than sending the polyline coordinates to onDigitizedPolyline. For and Java Edition, the *observer* parameter specifies the object that either handles the event or forwards it back to JavaScript. As of Release 1.3 of the API, there is a newer version of this method for use with the Autodesk MapGuide Viewer Plug-In and Java Edition: digitizePolylineEx.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

observer - A Java object that implements MGDigitizePolylineObserver (Plug-In and Java Edition only).

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It shows how to display the coordinates of a user-specified polyline.

Note: The example works with the Autodesk MapGuide Viewer Plug-In and Java Edition under Netscape, and with the ActiveX Control under Internet Explorer. It does not work with the Autodesk MapGuide Viewer, Java Edition under Internet Explorer. (All methods using the observer applet have this problem, because Internet Explorer doesn't allow for the passing of non-primitives between JavaScript and Java.) You can get around this limitation by calling the methods from a wrapper applet, as shown in Java Edition Example 96.

First, a function called getPolyline invokes the appropriate digitize method:

```
function getPolyline()
{
    // If browser is Netscape...
```

```
    if (navigator.appName == "Netscape")
        // ...use newer Ex version, and pass observer applet.
        getMap().digitizePolylineEx(document.obs);
    // If browser is IE...
    else
        // ...use older non-Ex version, with no argument.
        getMap().digitizePolyline();
}
```

Then, if you've set up your event handler properly, digitizePolyline() or digitizePolylineEx() fires the onDigitizedPolyline event, passing it the map object, the number of polyline vertices, and the coordinates of those vertices. If the browser is Netscape, the observer applet is passed as well. See "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide* for information about working with events and observers, including setting up event handlers.

Finally, the onDigitizedPolyline event looks for a JavaScript function of the same name and, if that function exists, executes it:

```
function onDigitizedPolyline(map, numPoints, points)
{
    alert("You just digitized a polyline with: " + numPoints + " Points");
    for (i = 0; i < numPoints; i++)
        alert("Point[" + i + "] = " + points.item(i).getY() + ", " +
points.item(i).getX());
}
```

To see how these functions work in an application, refer to the Event Handling example.

See Also

onDigitizedPolygon, digitizePolylineEx

MGMap.digitizePolylineEx

Syntax

`boolean digitizePolylineEx(MGDigitizePolylineObserverEx` *`observer`*`)`

Description

Enables a user to use the mouse to define a polyline on the map, then fires the event onDigitizedPolyline upon completion.

digitizePolylineEx does not perform any functions other than sending the polyline coordinates to onDigitizedPolyline. The *observer* parameter specifies the object that either handles the event or forwards it back to JavaScript.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Applies To

Plug-In and Java Edition only. This method is an updated version of digitizePolyline. The older method still works in Release 6.

Parameters

observer - A Java object that implements MGDigitizePolylineObserverEx.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It shows how to display the coordinates of a user-specified polyline.

Note: The example works with the Autodesk MapGuide Viewer Plug-In and Java Edition under Netscape, and with the ActiveX Control under Internet Explorer. It does not work with the Autodesk MapGuide Viewer, Java Edition under Internet Explorer. (All methods using the observer applet have this problem, because Internet Explorer doesn't allow for the passing of non-primitives between JavaScript and Java.) You can get around this limitation by calling the methods from a wrapper applet, as shown in Java Edition: Example 6

First, a function called getPolyline invokes the appropriate digitize method:

```
function getPolyline()
{
    // If browser is Netscape...
    if (navigator.appName == "Netscape")
        // ...use newer Ex version, and pass observer applet.
        getMap().digitizePolylineEx(document.obs);
    // If browser is IE...
    else
        // ...use older non-Ex version, with no argument.
        getMap().digitizePolyline();
}
```

Then, if you've set up your event handler properly, digitizePolyline() or digitizePolylineEx() fires the onDigitizedPolyline event, passing it the map object, the number of polyline vertices, and the coordinates of those vertices. If the browser is Netscape, the observer applet is passed as well. See "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide* for information about working with events and observers, including setting up event handlers.

Finally, the onDigitizedPolyline event looks for a JavaScript function of the same name and, if that function exists, executes it:

```
function onDigitizedPolyline(map, numPoints, points)
{
    alert("You just digitized a polyline with: " + numPoints + " Points");
    for (i = 0; i < numPoints; i++)
        alert("Point[" + i + "] = " + points.item(i).getY() + ", " +
points.item(i).getX());
}
```

To see how these functions work in an application, refer to the Event Handling example.

See Also

digitizeCircle, digitizePoint, digitizePolygonEx

MGMap.digitizeRectangle

Syntax

ActiveX Control: `boolean digitizeRectangle()`

Plug-In/Java Edition: `boolean digitizeRectangle(MGDigitizeRectangleObserver observer)`

Description

Enables the user to define a rectangle on the map by clicking and dragging the mouse, then fires the onDigitizedRectangle event upon completion.

digitizeRectangle does not perform any functions other than sending the rectangle coordinates to onDigitizedRectangle. Users can cancel digitizeRectangle by pressing the ESC key or by right-clicking the mouse. For Autodesk MapGuide Viewer Plug-In, the *observer* parameter specifies the object that either handles the event or forwards it back to JavaScript.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

observer - A Java object that implements MGDigitizeRectangleObserver (Plug-In and Java Edition only).

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

See Also

digitizePolygon

MGMap.disableApiExceptions

Syntax

`void disableApiExceptions()`

Description

Disables Autodesk MapGuide Viewer from throwing exceptions.

Error Codes

none

See Also

enableApiExceptions

MGMap.disablePrintingEvents

Syntax

```
void disablePrintingEvents()
```

Description

Disables the Autodesk MapGuide Viewer from sending printing events to the associated event handler during a print process.

Error Codes

none

See Also

enablePrintingEvents, onBeginLayout, onEndLayout

MGMap.disableSelectionChangedEvent

Syntax

```
void disableSelectionChangedEvent()
```

Description

Disables the firing of the onSelectionChanged event during calls to the MGMapLayer methods removeObject, removeObjects, and removeAllObjects.

Error Codes

none

See Also

enableSelectionChangedEvent, onSelectionChanged

MGMap.enableApiExceptions

Syntax

```
void enableApiExceptions()
```

Description

Enables Autodesk MapGuide Viewer exception throwing.

Error Codes

none

See Also

disableApiExceptions

MGMap.enablePrintingEvents

Syntax

```
void enablePrintingEvents()
```

Description

Enables the Autodesk MapGuide Viewer to send printing events to the associated event handler during a print process.

Once enablePrintingEvents is called, all methods that initiate a printing process -- as well as the print command on the Autodesk MapGuide Viewer popup menu -- use the new printing model until disablePrintingEvents is called.

Error Codes

none

See Also

disablePrintingEvents, onBeginLayout, onEndLayout

MGMap.enableSelectionChangedEvent

Syntax

```
void enableSelectionChangedEvent()
```

Description

Enables the firing of the onSelectionChanged event during calls to the MGMapLayer methods removeObject, removeObjects, and removeAllObjects.

Error Codes

none

See Also

disableSelectionChangedEvent, onSelectionChanged

MGMap.getApiVersion

Syntax

```
string getApiVersion()
```

Description

Gets the Autodesk MapGuide Viewer API version.

Returns

String - The version of the Autodesk MapGuide Viewer API, which will be one of the following:

- Autodesk MapGuide Release 2.5 uses API version *1.0*
- Autodesk MapGuide Release 2.5.5 uses API version *1.1*
- Autodesk MapGuide Release 3.0 uses API version *1.2*
- Autodesk MapGuide Release 4.0.3.4 uses API version *1.3*
- Autodesk MapGuide Release 4.0.3.6 uses API version *1.4*
- Autodesk MapGuide Release 4.1 uses API version*1.5*
- Autodesk MapGuide Release 5.0 uses API version *5.0*
- Autodesk MapGuide Release 6.0 uses API version *6.0*

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays a dialog box showing the API version:

```
function whatsTheApi()
{
    var msg = getMap().getApiVersion();
    alert("API version is: " + msg);
}
```

Here's another way to write the same function:

```
function whatsTheApi()
{
    alert("API version is: " + getMap().getApiVersion());
}
```

See Also

ApiVersion

MGMap.getAutoLinkDelay

Syntax

```
int getAutoLinkDelay()
```

Description

Gets the number of seconds before the next HTML page associated with a map feature from the AutoLinkLayer will be displayed in the AutoLinkTarget window.

If the delay value is 0, the automatic timed update feature is disabled, and the AutoLinkTarget window will be updated whenever the mouse passes over a map feature on the AutoLinkLayer. The default value is 20 seconds.

Returns

int - The delay in seconds between showing another link.

Error Codes

none

See Also

setAutoLinkDelay, AutoLinkDelay

MGMap.getAutoLinkLayers

Syntax

```
string getAutoLinkLayers()
```

Syntax

Returns the name of the layer that uses the AutoLink feature.

AutoLink automatically displays the linked HTML pages of a map feature when the user pauses the mouse button over that feature. The HTML pages display in the AutoLinkTarget window.

Returns

String - Name of the layer that will use AutoLinks, or an empty string if the AutoLink feature is disabled.

Error Codes

none

See Also

setAutoLinkLayers, AutoLinkLayers

MGMap.getAutoLinkTarget

Syntax

```
string getAutoLinkTarget()
```

Description

Gets the name of the target window where Autodesk MapGuide will display the HTML pages associated with map features on the AutoLinkLayer.

Returns

String - The name of the window or frame where Autodesk MapGuide will display the HTML pages; returns an empty string if no window or frame is set.

Error Codes

none

See Also

setAutoLinkTarget, AutoLinkTarget

MGMap.getAutoRefresh

Syntax

```
boolean getAutoRefresh()
```

Description

Returns a boolean value that indicates whether or not the automatic refresh feature is enabled.

Returns

boolean - Returns True if automatic refresh is enabled; otherwise, returns False.

Error Codes

none

See Also

setAutoRefresh, refresh, autoRefresh

MGMap.getBufferSetup

Syntax

```
MGBufferSetup getBufferSetup()
```

Description

Gets an object that contains buffering properties, such as the distance from the selected features to draw the buffers.

Autodesk MapGuide uses the properties to create buffer objects.

Returns

MGBufferSetup - An MGBufferSetup object that contains the buffer setup properties.

Error Codes

none

See Also

BufferSetup, setLayerName, viewBufferDlg, viewBuffer

MGMap.getBusyStateChangedObserver

Syntax

```
MGBusyStateChangedObserver getBusyStateChangedObserver()
```

Description

Returns a reference to the Java object that observes busy state change events, or null if no observer has been established.

Applies To

Plug-In and Java Edition only

Returns

MGBusyStateChangedObserver - The observer interface implementation.

Error Codes

none

See Also

setBusyStateChangedObserver

MGMap.getDefaultTarget

Syntax

```
string getDefaultTarget()
```

Description

Gets the name of the default target window where Autodesk MapGuide will display HTML documents, including autolinks, errors, feature links, reports, and URL lists.

Returns

String - The name of the window/frame where Autodesk MapGuide will display HTML documents, or an empty string if no default is set.

Error Codes

none

See Also

setDefaultTarget, DefaultTarget, getAutoLinkTarget, getErrorTarget, getObjectLinkTarget, getReportTarget, getUrlListTarget

MGMap.getDisplayUnits

Syntax

```
string getDisplayUnits()
```

Description

Gets the current type of display units being used to display the map width, as seen in the status bar.

Note that this method returns the current display units, not the default units. To retrieve the default units, call getUnits().

Returns

String - The current display units, either M, KM, FT, or MI.

Error Codes

none

See Also

getUnits, DisplayUnits

MGMap.getDoubleClickObserver

Syntax

```
MGDoubleClickObjectObserver getDoubleClickObserver()
```

Description

Returns a reference to the Java object being used to observe double-click events, or null if no observer has been established.

Applies To

Plug-In and Java Edition only

Returns

MGDoubleClickObjectObserver - The observer interface implementation.

Error Codes

none

See Also

setDoubleClickObserver, onDoubleClickObject

MGMap.getErrorTarget

Syntax

```
string getErrorTarget()
```

Description

Gets the name of the target window where Autodesk MapGuide will display error messages.

Returns

String - The name of the window/frame in which error messages will be displayed, or an empty string if no window/frame is set.

Error Codes

none

See Also

setErrorTarget, ErrorTarget

MGMap.getHeight

Syntax

```
double getHeight(String units)
```

Description

Gets the height of the map.

The height value is returned in the current default units, unless you specify another unit with the *units* parameter.

Parameters

units - The type of height units to use, either M (meters, the default), KM (kilometers), FT (feet), or MI (miles). The Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition also take "" and null. They use the current default units.

Returns

double - The height of the map.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It uses the KM argument to display a dialog box showing the height and width of the map in kilometers:

```
function showSize()
{
    var map = getMap();

    var width = map.getWidth("KM");
    var height = map.getHeight("KM");
    alert("width: " + width + ", height: " + height);
}
```

Here's a more complex use of getHeight() and getWidth(). In the following example, an HTML form lets the user select the unit of measurement from a drop-down list:

```
<FORM NAME="myForm">
    <SELECT NAME="optionList" size="1" onChange="showSize();">
        <OPTION VALUE="MI">miles</OPTION>
        <OPTION VALUE="FT">feet</OPTION>
        <OPTION VALUE="KM">kilometers</OPTION>
        <OPTION VALUE="M">meters</OPTION>
    </SELECT>
    <INPUT TYPE="button" VALUE="Go..." ONCLICK="showSize();">
</FORM>
```

The form calls a JavaScript function, passing it the value and text properties of the selected form option:

```
function showSize()
{
    var map = getMap();
    // For selected form option, get array index, VALUE
    // attribute, and option name as it appears in the browser
    var idx = window.document.myForm.optionList.selectedIndex;
    var val = window.document.myForm.optionList.options[idx].value;
    var type = window.document.myForm.optionList.options[idx].text;
    // Use val and type to call height/width methods and display alert
    alert( "Size is " + map.getWidth(val) + " by " + map.getHeight(val) + " " +
type);
}
```

If the user selected kilometers from the form, showSize would read the selected item's value and text properties ("KM" and "kilometers", respectively). In that case, the line:

```
    alert( "Size is " + map.getWidth(val) + " by " + map.getHeight(val) + " " +
type);
```

Would evaluate as:

```
    alert( "Size is " + map.getWidth("KM") + " by " + map.getHeight("KM") + " " +
"kilometers");
```

See Also

MapHeight

MGMap.getIntermediateUpdatesEnabled

Syntax

```
boolean getIntermediateUpdatesEnabled()
```

Description

Get the current setting of the intermediate updates flag.

The default mode for intermediate updates is True.

Parameters

none

Returns

boolean - True if intermediate updates are enabled; Otherwise, returns False.

Example

The following functions determine the current status of intermediate updates using getIntermediateUpdatesEnabled. They then use setIntermediateUpdatesEnabled to either disable and enable intermediate updates.

```
function disableIntermediateUpdates()
{
   var map = getMap();
   var status = map.getIntermediateUpdatesEnabled();
   if (status == true) map.setIntermediateUpdatesEnabled(false);
}

function enableIntermediateUpdates()
{
   var map = getMap();
   var status = map.getIntermediateUpdatesEnabled();
   if (status == false) map.setIntermediateUpdatesEnabled(true);
}
```

See Also

setIntermediateUpdatesEnabled, IntermediateUpdatesEnabled

MGMap.getLastError

Syntax

```
MGError getLastError()
```

Description

Returns a reference to the MGError object of the most recent Autodesk MapGuide Viewer API method call or property access.

Returns

MGError - The error object.

Error Codes

none

Example

```
function checkError()
{
   var map = getMap();
   var code = map.getLastError().getCode();
   alert("ERROR: " + code);
}
```

See Also

getLastError, LastError

MGMap.getLat

Syntax

```
double getLat()
```

Description

Gets the latitude (*Y*) value in decimal degrees of the center of the map area currently displayed.

Latitudes are positive north of the equator and negative south of the equator.

Returns

double - Latitude, or the *Y* coordinate if an arbitrary coordinate system is being used.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays a dialog box showing the Lat/Lon coordinates for the center of the visible map area:

```
function showCenterCoords()
{
    var map = getMap();
    alert( "Lat: " + map.getLat() + "\nLon: " + map.getLon() );
}
```

See Also

Lat, getLon, zoomWidth

MGMap.getLon

Syntax

```
double getLon()
```

Description

Gets the longitude (*X*) value in decimal degrees of the center of the map area currently displayed.

Longitudes are positive east of the Greenwich meridian and negative west of the meridian.

Returns

double - Longitude, or the*X* coordinate if an arbitrary coordinate system is being used.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays a dialog box showing the Lat/Lon coordinates for the center of the visible map area:

```
function showCenterCoords()
{
    var map = getMap();
    alert( "Lat: " + map.getLat() + "\nLon: " + map.getLon() );
```

See Also

Lon, getLat, zoomWidth

MGMap.getMapExtent

Syntax

`MGExtentEx getMapExtent(boolean forCurrentView, boolean useMcs)`

Description

Gets the extents for the map or current map view.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

forCurrentView - A boolean value representing the type of extents to return. If this value is True, getMapExtent returns the extents for the current view. If it is False, getMapExtent returns the extents for the entire map as it was created in Autodesk MapGuide Author.

useMcs - A boolean value representing the type of coordinates to be returned. If this value is True, coordinates are returned in the Mapping Coordinate System (MCS). If it is False, coordinates are returned as lat/lon values.

Returns

MGExtentEx - An MGExtentEx object containing the requested extents.

Error Codes

-1 (busy)

MGMap.getMapLayer

Syntax

`MGMapLayer getMapLayer(String layername)`

Description

Gets the specified map layer.

Parameters

layername - Name of the map layer to return.

Returns

MGMapLayer - The specified map layer, or null if no layer exists with the given name.

Error Codes

-3 (illegal argument)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It takes a layer name as its argument. If the layer exists, its visibility is toggled. Otherwise, an alert displays:

```
function layerToggle(name)
{
    var map = getMap();
    var layer = map.getMapLayer(name);

    if (layer == null)
        alert("layer not found.");
    else
        layer.setVisibility(!layer.getVisibility());

    map.refresh();
}
```

You might use the following HTML to call the function on a layer called "World Countries":

```
<FORM>
<INPUT TYPE="button" VALUE="Toggle" ONCLICK="layerToggle('World Countries');">
</FORM>
```

See Also

MapLayers, getMapLayersEx, getMapLayers

MGMap.getMapLayerGroup

Syntax

```
MGMapLayerGroup getMapLayerGroup(String
  groupname)
```

Description

Gets the specified map layer group.

Parameters

groupname - The name of the map layer group to return.

Returns

MGMapLayerGroup - The specified map layer group, or null if no layer exists with the given name.

Error Codes

-3 (illegal argument)

See Also

getMapLayerGroups, mapLayerGroups

MGMap.getMapLayerGroups

Syntax

```
MGCollection getMapLayerGroups()
```

Description

Gets an MGCollection object containing all the map layer groups defined in the map.

Returns

MGCollection - An MGCollection of MGMapLayerGroup objects. Returns an empty MGCollection if unsuccessful.

Error Codes

none

See Also

getMapLayerGroup, mapLayerGroups

MGMap.getMapLayers

Syntax

ActiveX Control: `MGCollection getMapLayers()`

Plug-In/Java Edition: `Vector getMapLayers()`

Description

Returns a Vector (or MGCollection) object containing all the map layers defined in the map.

For compatibility between the Autodesk MapGuide Viewer Plug-In, Java Edition, and ActiveX Control, use getMapLayersEx instead of getMapLayers.

Returns

ActiveX Control: MGCollection - A collection of MGMapLayer objects. Returns empty MGCollection if unsuccessful.

Plug-In/Java Edition: Vector - A java.util.Vector of map layer objects. Returns empty Vector if unsuccessful.

Error Codes

none

See Also

MapLayers, getMapLayer, getMapLayersEx

MGMap.getMapLayersEx

Syntax

```
MGCollection getMapLayersEx()
```

Description

Gets an MGCollection object containing all the map layers defined in the map.

This method replaces the getMapLayers method in earlier versions of the Autodesk MapGuide Viewer API and supports the Autodesk MapGuide Viewer Plug-In, Java Edition, and ActiveX Control with the same syntax. The older method will still work in Release 6, but you should use getMapLayersEx in all future development.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Returns

MGCollection - An MGCollection object.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays a dialog box listing all the layers in the map. The getMapLayersEx() method is used to obtain the map layer collection and to assign it to a variable named layers. Note the use of the size() and item() methods, which operate on the collection variable:

```
function listLayers()
{
    var map = getMap();
    var msg = "";

    // Assign collection to layers variable
    var layers = map.getMapLayersEx();
    // Use size() to step through collection
    for (var i = 0; i < layers.size(); i++)
    {
        // Get each item in collection...
        var layer = layers.item(i);
        // ...then get item name and add it to msg
        msg = msg + layer.getName() + "\n";
    }
    // Display the layer names
    alert(msg);
}
```

See Also

MapLayers, getMapLayer, getMapLayers

MGMap.getMapLoadedObserver

Syntax

```
MGMapLoadedObserver getMapLoadedObserver()
```

Description

Returns a reference to an MGMapLoadedObserver object or null if no observer has been established.

Applies To

Plug-In and Java Edition only

Returns

MGMapLoadedObserver - An MGMapLoadedObserver object.

Error Codes

none

See Also

setMapLoadedObserver, onMapLoaded

MGMap.getMcsScaleFactor

Syntax

```
double getMCSScaleFactor()
```

Description

Gets the conversion factor to convert from the current Mapping Coordinate System (MCS) to meters.

The MCS can vary based on factors such as the current projection of the map.

Returns

double - The conversion factor to convert from MCS to Meters.

Error Codes

none

Example

This simple function is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the MCS-to-meters conversion factor:

```
function mcsConversionFactor()
{
    alert("Conversion factor is " + getMap().getMCSScaleFactor());
}
```

See Also

MCSScaleFactor

MGMap.getMode

Syntax

```
string getMode()
```

Description

Gets the current viewer mode of the map, such as zoomIn, digitize, pan, and so on.

Parameters

None

Returns

Busy, DigitizeCircle, DigitizeLine , DigitizePoint, DigitizePolyline, DigitizePolygon, DigitizeRectangle, Pan, Select, SelectPolygon, SelectRadius, ViewDistance, ZoomIn, ZoomOut, and Unknown.

Error Codes

None

See Also

Mode

MGMap.getName

Syntax

```
string getName()
```

Description

Gets the name of the map, as set in the Preferences dialog box in Autodesk MapGuide Author, or with setName.

Returns

String - The name of the map.

Error Codes

none

Example

This simple function is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays an alert showing the name of the map:

```
function showNameInAlert()
{
    alert("The map is named " + getMap().getName() + ".");
}
```

You can also use getName() to display the map name dynamically as an HTML heading. The following example assumes you are working with a two-frame frameset. One frame (called mapFrame) displays a page containing the map. The other frame ('titleFrame') displays a blank HTML document that will be replaced later by the map name.

First, write a function that uses getName() to write the map name to the title frame. This function should be in the page containing the map (or in a *.js* file referenced by the page):

```
function showNameInFrame()
{
    parent.titleFrame.document.open();
    parent.titleFrame.document.write("<H1>" + getMap().getName() + "</H1>");
    parent.titleFrame.document.close();
}
```

Then, call the function from the onLoad event of the page containing the map:

```
<BODY onLoad="showNameInFrame()">
```

When the map page is fully loaded, the onLoad event fires, invoking the showNameInFrame() function. The showNameInFrame() function replaces the contents of the title frame with an H1 element containing the map name.

The map name isn't available until the map is fully loaded. We got around this problem by redirecting the name to a separate frame after the map loaded. Another approach would be to use *dynamic HTML* to display the name in the same page as the map. See a third-party dynamic HTML reference for details.

See Also

setName, MapName

MGMap.getObjectLinkTarget

Syntax

```
string getObjectLinkTarget()
```

Description

Gets the name of the target window where Autodesk MapGuide will display the URL associated with a map feature.

Returns

String - The name of the window/frame that will display map features associated URLs, or an empty string if the window/frame is not set.

Error Codes

none

See Also

setObjectLinkTarget, ObjectLinkTarget

MGMap.getPageSetup

Syntax

```
MGPageSetup getPageSetup()
```

Description

Gets an object containing properties used by Autodesk MapGuide to print maps.

Returns

MGPageSetup - An MGPageSetup object that contains the page setup properties.

Error Codes

none

See Also

PageSetup, pageSetupDlg, printDlg

MGMap.getPrintingObserver

Syntax

```
MGPrintingObserver getPrintingObserver()
```

Description

Returns a reference to the Java object being used to observe printing events.

Applies To

Plug-In and Java Edition only

Returns

MGPrintingObserver - An MGPrintingObserver object, or null if no observer has been setup for the printing events.

Error Codes

none

See Also

setPrintingObserver, onBeginLayout, onEndLayout

MGMap.getRedlineSetup

Syntax

```
MGRedlineSetup getRedlineSetup()
```

Description

Gets an object containing all properties for drawing redline features.

This object provides access to the properties used to draw each of the individual feature types.

Returns

MGRedlineSetup - An MGRedlineSetup object if successful, null otherwise.

Error Codes

none

See Also

RedlineSetup

MGMap.getReportTarget

Syntax

```
string getReportTarget()
```

Description

Gets the name of the target window where Autodesk MapGuide will display reports from View Reports command.

Returns

String - The name of the window/frame that reports will display reports, or an empty string if the window/frame is not set.

Error Codes

none

See Also

setReportTarget, ReportTarget

MGMap.getScale

Syntax

```
double getScale()
```

Description

Gets the current map display scale in terms of 1:scale.

Returns

double - The map display scale.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays a message box showing the current map scale:

```
function showScale()
{
    alert ( "Scale is 1:" + getMap().getScale() + ".");
}
```

See Also

setScale, MapScale

MGMap.getSelection

Syntax

```
MGSelection getSelection()
```

Description

Gets the currently selected map features.

Returns

MGSelection - The current selection.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It zooms to the selected feature or features. If nothing is selected, an alert displays:

```
function zoomToSelected()
{
    var map = getMap();
    // Assign collection of selected features to variable
    var selected = map.getSelection().getMapObjectsEx(null);
    // If collection is not empty, zoom to its members...
    if (selected.size() > 0)
        map.zoomSelected();
    // ...otherwise, display alert
    else
        alert("Please make a selection first.");
}
```

See Also

Selection

MGMap.getSelectionChangedObserver

Syntax

```
MGSelectionChangedObserver getSelectionChangedObserver()
```

Description

Returns a reference to the Java object being used to observe selection change events, or null if no observer has been established.

Applies To

Plug-In and Java Edition only

Returns

MGSelectionChangedObserver - The observer interface implementation.

Error Codes

none

See Also

setSelectionChangedObserver, onSelectionChanged

MGMap.getSelectionMode

Syntax

```
string getSelectionMode()
```

Description

Gets the selection mode, which controls how map features are selected when a feature is not fully contained in the selection area.

Returns

String - The selection mode: either Centroid (selects features whose center points fall within the selection area), or Intersection (selects features that are completely or partially contained within the selection area).

Error Codes

none

See Also

SelectionMode, setSelectionMode

MGMap.getSymbols

Syntax

```
MGCollection getSymbols()
```

Description

Gets a collection of strings representing the symbols available for use as symbol attributes.

Returns

MGCollection - A collection of strings representing all available symbols.

Error Codes

none

See Also

Symbols

MGMap.getTrackingID

Syntax

```
String getTrackingID()
```

Description

Gets the customer tracking ID associated with a map.

This method is available only if the map was created with the *Allow Access to Tracking ID from API* option selected in Autodesk MapGuide Author.

Returns

String - The tracking ID string.

Error Codes

-5 (security violation)

See Also

setTrackingID, TrackingID

MGMap.getUnits

Syntax

```
string getUnits()
```

Description

Gets the current units that will be used when getting or setting the MapWidth property, or when using getWidth or setWidth without specifying the *Units* parameter.

Not to be confused with the coordinate system units, which the user can set to control the distance units in the Autodesk MapGuide Viewer, or default width units, which the map author uses to set the default width of the MWF in the Properties dialog box in Autodesk MapGuide Author.

Returns

String - One of the following: M (meters), KM (kilometers), FT (feet), or MI (miles)

Error Codes

none

See Also

MapUnits, getWidth, setWidth

MGMap.getUrl

Syntax

```
string getUrl()
```

Description

Gets the URL of the map that is currently displayed by Autodesk MapGuide Viewer.

The returned URL includes parameters that describe the current map location and the parameters that control the state of the various Autodesk MapGuide user interface elements.

For the ActiveX Control and Plug-In, note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

String - A string that represents the URL of the map.

Error Codes

-1 (busy): For ActiveX Control and Plug-In only.

"" (empty string)

See Also

setUrl, URL, FullUrl

MGMap.getUrlListState

Syntax

```
boolean getUrlListState()
```

Description

Returns True if Autodesk MapGuide will generate an HTML document that lists the names and URLs associated with a map feature as the user pauses the mouse pointer over that feature.

The HTML document is displayed in the UrlListTarget window.

Returns

boolean - Returns True if the URL list feature is enabled; otherwise, returns False.

Error Codes

none

See Also

setUrlListState, UrlListState, getUrlListTarget

MGMap.getUrlListTarget

Syntax

```
string getUrlListTarget()
```

Description

Gets the name of the target window where Autodesk MapGuide will display the names and URLs associated with a map feature as the user pauses the mouse pointer over that feature.

The URL lists are generated only if this functionality is enabled through setUrlListState. To verify whether it is enabled, use getUrlListState.

Returns

String - The name of the window or frame in which the URL list will be displayed.

Error Codes

none

See Also

setUrlListTarget, UrlListTarget

MGMap.getVersion

Syntax

```
string getVersion()
```

Description

Gets the version string of the Autodesk MapGuide Viewer.

Returns

String - The Autodesk MapGuide version.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays a dialog box showing the Autodesk MapGuide version number:

```
function whatsTheVersion()
{
    var msg = getMap().getVersion();
    alert("Viewer is version " + msg);
}
```

Here's another way to write the same function:

```
function whatsTheVersion()
{
    alert("Viewer is version " + getMap().getVersion());
}
```

See Also

Version, getViewerType, getApiVersion

MGMap.getViewChangedObserver

Syntax

```
MGViewChangedObserver getViewChangedObserver()
```

Description

Returns a reference to the Java object being used to observe view change events, or null if no observer has been established.

Applies To

Plug-In and Java Edition only

Returns

MGViewChangedObserver - The observer interface implementation.

Error Codes

none

See Also

setViewChangedObserver, onViewChanged

MGMap.getViewChangingObserver

Syntax

```
MGViewChangingObserver getViewChangingObserver()
```

Description

Returns a reference to the Java object that observes when a map is about to be refreshed, or null if no such observer has been established.

Applies To

Plug-In and Java Edition only

Returns

MGViewChangingObserver - The observer interface implementation.

Error Codes

none

See Also

setViewChangingObserver, onViewChanging

MGMap.getViewDistanceExObserver

Syntax

```
MGViewDistanceExObserver getViewDistanceExObserver()
```

Description

Returns a reference to the Java object that observes view distance events, or null if no such observer has been established.

Applies To

Plug-In and Java Edition only

Returns

MGViewDistanceExObserver - The observer interface implementation.

Error Codes

none

See Also

setViewDistanceExObserver, onViewedDistanceEx

MGMap.getViewDistanceObserver

Syntax

```
MGViewDistanceObserver getViewDistanceObserver()
```

Description

Returns a reference to the Java object that observes view distance events, or null if no such observer has been established.

Applies To

Plug-In and Java Edition only

Returns

MGViewDistanceObserver - The observer interface implementation.

Error Codes

none

See Also

setViewDistanceObserver, onViewedDistance

MGMap.getViewerType

Syntax

```
String getViewerType()
```

Description

Returns a string indicating whether the current Autodesk MapGuide Viewer is the ActiveX Control, Java Edition, or the Autodesk MapGuide Viewer Plug-In.

Returns

String - "ActiveX Control" if the Autodesk MapGuide Viewer is the ActiveX Control, "Plug-In" if the Autodesk MapGuide Viewer is the Autodesk MapGuide Viewer Plug-In, "Java" if the Autodesk MapGuide Viewer is the Java edition.

Error Codes

none

See Also

getVersion, ViewerType

MGMap.getWidth

Syntax

```
double getWidth(String units)
```

Description

Gets the current width of the map in the specified unit of measure.

Note that because of the floating-point calculation inaccuracy in operating systems, if you set the width of the map using setWidth before calling getWidth, the value returned by getWidth may be inaccurate, especially if the width is set to a very high number.

Parameters

units - The unit type, which will be MI (miles), FT (feet), KM (kilometers), (meters), null, or an empty string ("").

Note that null and "" do not work with the ActiveX Control. If you pass in null or empty string, Autodesk MapGuide uses the current unit.

Returns

double - The width of the map.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It uses the "KM " argument to display a dialog box showing the height and width of the map in kilometers:

```
function showSize()
{
    var map = getMap();

    var width = map.getWidth("KM");
    var height = map.getHeight("KM");
    alert("width: " + width + ", height: " + height);
}
```

Here's a more complex use of getHeight() and getWidth(). In the following example, an HTML form lets the user select the unit of measurement from a drop-down list:

```
<FORM NAME="myForm">
    <SELECT NAME="optionList" size="1" onChange="showSize();">
        <OPTION VALUE="MI">miles</OPTION>
        <OPTION VALUE="FT">feet</OPTION>
        <OPTION VALUE="KM">kilometers</OPTION>
        <OPTION VALUE="M">meters</OPTION>
    </SELECT>
    <INPUT TYPE="button" VALUE="Go..." ONCLICK="showSize();">
</FORM>
```

The form calls a JavaScript function, passing it the value and text properties of the selected form option:

```
function showSize()
{
    var map = getMap();
    // For selected form option, get array index, VALUE
    // attribute, and option name as it appears in the browser
    var idx = window.document.myForm.optionList.selectedIndex;
    var val = window.document.myForm.optionList.options[idx].value;
    var type = window.document.myForm.optionList.options[idx].text;
    // Use val and type to call height/width methods and display alert
    alert( "Size is " + map.getWidth(val) + " by " + map.getHeight(val) + " " +
type);
}
```

If the user selected kilometers from the form, showSize would read the selected item's value and text properties ("KM" and "kilometers", respectively). In that case, the line:

```
    alert( "Size is " + map.getWidth(val) + " by " + map.getHeight(val) + " " +
type);
```

Would evaluate as:

```
    alert( "Size is " + map.getWidth("KM") + " by " + map.getHeight("KM") + " " +
"kilometers");
```

See Also

setWidth, MapWidth, MapUnits

MGMap.isBusy

Syntax

```
boolean isBusy()
```

Description

Returns True if Autodesk MapGuide Viewer is busy loading or drawing the map.

Before calling other API methods, you should use the isBusy method in an if statement to test whether the Autodesk MapGuide Viewer is busy.

Do not attempt to use isBusy in a while loop to wait until Autodesk MapGuide Viewer becomes available. This may cause the Web browser and Autodesk MapGuide Viewer to freeze up.

Returns

boolean - Returns True if busy; otherwise, returns False.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays a dialog box showing whether the map is busy or not:

```
function getBusy()
{
    map = getMap();
    if (map.isBusy())
        alert("busy!");
    else
        alert("not busy!");
}
```

Here's a more meaningful use of isBusy(). The following example counts the features on each map layer and adds that count to the legend:

```
// This global variable tells us if showFeatureCount() has already
// added the legend info; if the function hasn't run successfully,
// the value will remain false
legendSet = false;
// Function starts here
function showFeatureCount() {
    // Exit function if legend info has already been added
    if (legendSet)
        return;
    var map = getMap();
    // If map is NOT busy, add legend info; otherwise, exit
    if (!map.isBusy())
    {
        var layers = map.getMapLayersEx();
        var cnt = layers.size();
        var i;
        var msg = "";
        for (i = 0; i < cnt; i++)
        {
```

```
            var layer = layers.item(i);
            var featureCount = layer.getMapObjectsEx().size();
            var label = layer.getLegendLabel();
            label = label + " " + featureCount + " features";
            layer.setLegendLabel(label);
        }
        // Change variable to true, showing that
        // legend info has been added
        legendSet = true;
    }
}
```

If the map is busy, the function will be unable to get the feature count and add it to the legend. By using isBusy(), you can make sure the legendSet variable updates only if the legend has updated successfully.

See Also

Busy

MGMap.lonLatToMcs

Syntax

```
MGPoint lonLatToMcs(double lon,
  double lat)
```

Description

Converts the specified lat/lon coordinate pair to a point using the Mapping Coordinate System (MCS) defined for the MWF by the map author.

Note that this method takes the longitude as its first argument and the latitude as its second agument. For more information on Mapping Coordinate Systems, refer to the *Autodesk MapGuide User's Guide*.

Parameters

lon - Longitude coordinate to convert. *lat* - Latitude coordinate to convert.

Returns

MGPoint - The coordinate in MCS units.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It converts a lat/lon coordinate pair to an MCS point and displays the *x* and *y* values of that point:

```
function convertToMcs()
{
    var map = getMap();
    var lat = map.getLat();
    var lon = map.getLon();
```

```
    // display point in lat/lon
    alert( "Lat: " + lat + "\nLon: " + lon );
    // get MCS coords and display
    var xyPt = map.lonLatToMcs(lon, lat);
    alert( "MCS X: " + xyPt.getX() + "\nMCS Y:" + xyPt.getY() );
}
```

See Also

mcsToLonLat

MGMap.mcsToLonLat

Syntax

`MGPoint mcsToLonLat(double x, double y)`

Description

Converts the specified Mapping Coordinate System (MCS) coordinate pair to lat/lon coordinates.

The MCS for the map file is set by the map author. For more information on Mapping Coordinate Systems, refer to the *Autodesk MapGuide User's Guide*.

Parameters

x - MCS *X* coordinate to convert. *y* - MCS *Y* coordinate to convert.

Returns

MGPoint - Coordinate in longitude/latitude units.

Error Codes

none

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It converts a set of lat/lon points to MCS, and then uses mcsToLonLat() to convert them back again:

```
{
    var map = getMap();
    // Get lat/lon and convert to MCS
    var xyPt = map.lonLatToMcs(map.getLon(), map.getLat());
    // pull out X and Y values and display in dialog
    x = xyPt.getX();
    y = xyPt.getY();
    alert( "MCS X: " + x + "\nMCS Y:" + y );
    // Convert MCS to lat/lon
    var latLonPt = map.mcsToLonLat(x, y);
    // pull out lat/lon vals and display in dialog
    lon = latLonPt.getX();
    lat = latLonPt.getY();
    alert( "Lat: " + lat + "\nLon: " + lon );
}
```

See Also

lonLatToMcs

MGMap.pageSetupDlg

Syntax

```
voidpageSetupDlg()
```

Description

Displays the Page Setup dialog box, where the user can specify preferences for printing the map.

This method is the equivalent of the Printing > Page Setup menu command in the user interface.

Note that this method does not work during a busystate.

Error Codes

-1 (busy)

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the Page Setup dialog box:

```
function showPageSetup()
{
    getMap().pageSetupDlg();
}
```

See Also

getPageSetup, PageSetup, printDlg

MGMap.panMode

Syntax

```
void panMode()
```

Description

Switches to Pan mode.

In Pan mode, the hand cursor is displayed, and the user can drag the map to pan in any direction. This method is the equivalent of the Pan button in the user interface.

Error Codes

-1 (busy)

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It switches to Pan mode:

```
function pan()
{
    alert("Drag the mouse to pan...");
        getMap().panMode();
}
```

MGMap.preferencesDlg

Syntax

```
void preferencesDlg()
```

Description

Displays the Preferences dialog box, where the user can specify the map units, selection mode, and options for the status bar.

This method is the equivalent of the Help > Preferences menu command in the user interface.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Error Codes

-1 (busy)

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the Preferences dialog box:

```
function showPreferences()
{
    getMap().preferencesDlg();
}
```

MGMap.printDlg

Syntax

```
void printDlg()
```

Description

Uses the page setup properties, then displays the Print dialog box.

This method is the equivalent of the Printing > Print menu command in the user interface.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Error Codes

-1 (busy)

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the Print dialog box:

```
function showPrint()
{
    getMap().printDlg();
}
```

See Also

getPageSetup, PageSetup, pageSetupDlg

MGMap.redrawExtent

Syntax

`boolean redrawExtent(MGExtentEx extents, boolean useMCS)`

Description

Redraws the specified portion of the map.

Parameters

extents - The extents of the window area to redraw.

useMCS - A boolean specifying whether the extent coordinates are in MCS or lat/lon.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

MGMap.redrawLegend

Syntax

`void redrawLegend()`

Description

Redraws the map legend.

This method operates only on the legend; the map is not affected.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Error Codes

-1 (busy)

See Also

refresh, setAutoRefresh, AutoRefresh

MGMap.refresh

Syntax

```
void refresh()
```

Description

Refreshes (redraws) the map.

If a layer is flagged for rebuild, the data will be requested from the server again before it is redrawn. Use this method after changing properties that do not invoke an automatic refresh (redraw) of the map.

This method overrides autoRefresh and always refreshes the map. Subsequent method calls adhere to the original setting of the autoRefresh flag.

This method does not work during a busy state.

Refresh does not work when called from event-handling code for onViewChanging or onMapLoaded. If you call refresh from these events, the method will fail and will set the error code to -14 (refresh disabled).

Calling refresh always sets the mouse pointer to Select mode.

Note: By default, intermediate updates of maps every 1.5 seconds during a drawing operation are enabled for all Autodesk MapGuide Viewers. This allows users to progressively see a map as it is being rendered, rather than waiting until the entire map is complete. To change this behavior, you can disable intermediate updates using the setIntermediateUpdatesEnabled method or IntermediateUpdatesEnabled property.

Error Codes

-1 (busy)

-14 (refresh disabled)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It toggles the visibility of a layer called "World Countries". The refresh() method forces a map refresh, making the change immediately visible to the user:

```
function toggleAndRefresh()
{
    var map = getMap();
    var layer = map.getMapLayer("World Countries");

    if (layer == null)
        alert("layer not found.");
    else
        layer.setVisibility(!layer.getVisibility());

    // force refresh to make layer change visible
    map.refresh();
}
```

See Also

getAutoRefresh, setAutoRefresh, AutoRefresh, getIntermediateUpdatesEnabled
setIntermediateUpdatesEnabled, IntermediateUpdatesEnabled

MGMap.removeMapLayer

Syntax

`boolean removeMapLayer(String layerName)`

Description

Removes the specified map layer from the map.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state.

Parameters

layerName - The name of the map layer to remove from the map.

Returns

boolean - Returns True if layer successfully removed; False if layer does not exist.

Error Codes

-1 (busy)

-3 (illegal argument)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It takes a layer name as its argument. If the layer exists, it is removed from the map. Otherwise, an alert displays:

```
function removeLayer(name)
{
    if ( getMap().removeMapLayer(name) == true )
        alert(name + " layer removed.");
    else
        alert(name + " layer not found!");
}
```

You might use the following HTML to call the function on a layer called "World Countries":

```
<FORM>
<INPUT TYPE="button" VALUE="Toggle" ONCLICK="removeLayer('World Countries');">
</FORM>
```

See Also

addMapLayer, viewBuffer

MGMap.removeMapLayerGroup

Syntax

`boolean removeMapLayerGroup(MGMapLayerGroup object)`

Description

Removes the specified map layer group from the map.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

object - The map layer group to remove from the map.

Returns

boolean - Returns True if layer group successfully removed; False if layer group does not exist.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

createMapLayerGroup

MGMap.saveMWF

Syntax

`boolean saveMWF(String pathFileName, String password)`

Description

Saves the map to a local MWF.

Note that the name of the MWF will not change after the save is complete, so getURL will return the same value after a call to saveMWF as it did before. This method is not implemented for the Autodesk MapGuide Viewer, Java Edition.

Applies to

ActiveX Control and Plug-In only

Parameters

pathFileName - Local path and file name for the saved MWF; this can be specified as a relative or absolute path.

password - The password to unlock the MWF file.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-5 (security violation)

-16 (write permission denied)

MGMap.selectMapObjectsDlg

Syntax

```
void selectMapObjectsDlg()
```

Description

Displays the Select Map Features dialog box, where the user can select map features by name from a specific map layer.

This method is the equivalent of the Select > Select Map Features menu command in the user interface.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It invokes the Select Map Features dialog box.

Note that the dialog displays only if one or more selectable layers is visible.

```
function showMapFeatures()
{
    getMap().selectMapObjectsDlg();
}
```

The next example checks for selectable map features before calling selectMapObjectsDlg(). If one or more selectable layers is visible, the Select Map Features dialog box displays. Otherwise an alert displays.

```
// first function calls selectMapObjectsDlg()
function showMapFeatures()
{
    map = getMap();

    // call function that tests selectability
    var test = checkSelectability();

    // selectable map features?
    if (test == true)
```

```
        // ....if yes, display dialog
        map.selectMapObjectsDlg();
    else
        // ...if no, display alert
        alert("Sorry, no selectable features found!");
}
// second function checks map feature selectability
function checkSelectability()
{
    var map = getMap();
    var layers = map.getMapLayersEx();

    // false means a selectable layer has not (yet) been found
    var selectable = false;
    // cycle through each layer in map...
    for (var i = 0; i < layers.size(); i++)
    {
        var layer = layers.item(i);
        // is the layer visible?
        if ( layer.isVisible() )
        {
            // is it selectable?
            if (layer.getSelectability())
            {
                // first time a visible/selectable layer is
                // found, set flag to true and exit loop
                selectable = true;
                break;
            }
        }
    }
    return selectable;
}
```

See Also

MGSelection

MGMap.selectMode

Syntax

```
void selectMode()
```

Description

Switches to Select mode.

This method is equivalent to the Select button on the Autodesk MapGuide Viewer toolbar.

You can use selectMode to mimic the behavior of the Release 4 and earlier versions of the Autodesk MapGuide Viewer, which always returned to the selection cursor after a Pan, Zoom, or Zoom Out. In most cases, a call to selectMode is not necessary, as methods that set the cursor type do so without requiring a specific type to be set beforehand.

Error Codes

-1 (busy)

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It switches to Select mode:

```
function select()
{
    alert("Select a feature...");
        getMap().selectMode();
}
```

See Also

panMode, zoomInMode, zoomOutMode

MGMap.selectPolygon

Syntax

ActiveX Control: `boolean selectPolygon(MGMapLayer layer, MGCollection latlon)`

Plug-In/Java Edition: `boolean selectPolygon(MGMapLayer layer, Vector latlon)`

Plug-In/Java Edition: `boolean selectPolygon(MGMapLayer layer, Vector lat, Vector lon)`

Description

Selects all map features that fall within the specified polygon.

For compatibility between the Autodesk MapGuide Viewer Plug-In, Java Edition, and ActiveX Control, use selectPolygonEx instead of selectPolygon.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

layer - Map layer containing the features to select, or null to select from all layers. A map layer must be selectable for any features on it to be selected.

latlon - A java.util.Vector (or MGCollection) object used to stored the list of latitude/longitude pairs represented by MGPoint objects.

lat - A java.util.Vector object used to store the list of latitude (*Y*) values of polygon vertices represented by java.lang.Double objects (Plug-In and Java Edition only).

lon - A java.util.Vector object used to store the list of longitude (*X*) values of polygon vertices represented by java.lang.Double objects (Plug-In and Java Edition only).

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

selectPolygonEx

MGMap.selectPolygonEx

Syntax

```
boolean selectPolygonEx(MGMapLayer
  layer, MGCollection points)
```

Description

Selects all map features that fall within the specified polygon.

This method replaces the selectPolygon method in earlier versions of the Autodesk MapGuide Viewer API and supports the Autodesk MapGuide Viewer Plug-In, Java Edition, and ActiveX Control with the same syntax. The older method will still work in Release 6, but you should use selectPolygonEx in all future application development.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

layer - Map layer containing the features to select, or null to select from all layers. A map layer must be selectable for any features on it to be selected.

points - Collection of MGPoints containing longitude/latitude pairs.

Error Codes

-1 (busy)

-3 (illegal argument)

Returns

boolean - Returns True if successful; otherwise, returns False.

See Also

selectPolygonMode, digitizePolygonEx

MGMap.selectPolygonMode

Syntax

```
void selectPolygonMode()
```

Description

Enters the select polygon mode.

In this mode, Autodesk MapGuide allows the user to digitize a polygon and select all map features that fall within the polygon.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It invokes polygon selection mode.

Note that the user can make a polygon selection only if one or more selectable layers is visible.

```
function polySelect()
{
    getMap().selectPolygonMode();
}
```

The next example checks for selectable map features before calling selectPolygonMode(). If one or more selectable layers is visible, the user can make a selection. Otherwise an alert displays.

```
// first function calls selectPolygonMode()
function polySelect()
{
    map = getMap();

    // call function that tests selectability
    var test = checkSelectability();

    // selectable map features?
    if (test == true)
    {
        // ...if yes, user can make polygon selection
        alert("Make a polygon selection now...");
        map.selectPolygonMode();
    }
    else
    {
        // ...if no, display alert
        alert("Sorry, no selectable features found!");
    }
}
// second function checks map feature selectability
function checkSelectability()
{
    var map = getMap();
    var layers = map.getMapLayersEx();

    // false means a selectable layer has not (yet) been found
    var selectable = false;
    // cycle through each layer in map...
    for (var i = 0; i < layers.size(); i++)
    {
        var layer = layers.item(i);
        // is the layer visible?
        if ( layer.isVisible() )
        {
            // is it selectable?
            if (layer.getSelectability())
```

```
            {
                // first time a visible/selectable layer is
                // found, set flag to true and exit loop
                selectable = true;
                break;
            }
        }
    }
    return selectable;
}
```

See Also

selectPolygonEx

MGMap.selectRadius

Syntax

```
boolean selectRadius(MGMapLayer layer, double lat,
  double lon, double radius, String units)
```

Description

Selects all map features that fall within the specified radius.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

layer - Map layer on which to select features. If null, all selectable layers are considered.

lat - Latitude (*Y*) center of search circle.

lon - Longitude (*X*) center of search circle. *radius* - Radius of search circle.

units - Unit of measure for the radius, either M (meters), KM (kilometers), FT (feet), MI (miles), "", or null. "" and null only work with the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition. They use the current default units.

Error Codes

-1 (busy)

-3 (illegal argument)

Returns

boolean - Returns True if successful; otherwise, returns False.

See Also

selectRadiusMode

MGMap.selectRadiusMode

Syntax

```
void selectRadiusMode()
```

Description

Selects Select Radius mode.

In this mode, Autodesk MapGuide allows the user to digitize a circle and select all map features that fall within the circle.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It invokes radius selection mode.

Note that the user can make a radius selection only if one or more selectable layers is visible.

```
function radSelect()
{
    getMap().selectRadiusMode();
}
```

Note: The selectPolygonMode() example shows how to check for selectable map features before calling a Viewer API selection method.

See Also

selectRadius

MGMap.selectWithin

Syntax

```
boolean selectWithin(MGCollection
  layers)
```

Description

Selects features from the specified map layers that fall within the currently selected polygon map features.

After this method is complete, you can use getSelection to retrieve the contents of the selected features.

Currently, maps must be authored with the selection mode preference set to Centroid, not Intersection, for selectWithin to have consistent results. If your application is also selecting points in adjacent polygons that are near the selected polygon, chances are the Autodesk MapGuide selection mode preference is set to "intersection" instead of "centroid".

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

layers - Collection of MGMapLayer objects to use in the selection.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-12 (invalid selection)

See Also

selectWithinDlg

MGMap.selectWithinDlg

Syntax

```
void selectWithinDlg()
```

Description

Displays the Select Within dialog box, where the user chooses the map layer containing the features that will be selected if they fall within the currently selected features.

This method is the equivalent of the Select > Select Within menu command in the user interface.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Error Codes

-1 (busy)

-12 (invalid selection)

See Also

selectWithin

MGMap.setAutoLinkDelay

Syntax

```
void setAutoLinkDelay(int delay)
```

Description

Sets the number of seconds before the next HTML page associated with a map feature from the AutoLinkLayer will be displayed in the AutoLinkTarget window.

If the delay value is 0, the AutoLinkTarget window will be updated whenever the mouse pauses over a map feature on the AutoLinkLayer. Negative numbers are not allowed, and setting a value less than 5 is not advisable, because requests will likely be issued before the previous requests can be serviced, resulting in links being passed over and overburdening the server. The default value is 20 seconds.

Parameters

delay - Delay in seconds between showing another link, or 0 to display the HTML page associated with a feature whenever the user positions the mouse cursor over it.

Error Codes

-3 (illegal argument)

See Also

getAutoLinkDelay, AutoLinkDelay

MGMap.setAutoLinkLayers

Syntax

```
void setAutoLinkLayers(String layer)
```

Description

Sets the name of the layer that will use the AutoLink feature.

AutoLink automatically displays HTML pages associated with map features from the AutoLinkLayer when the user pauses the mouse pointer over those features. The HTML pages display in the AutoLinkTarget window. AutoLinks apply only to those map features visible in the current view.

Parameters

layer - Name of the layer that will use AutoLinks. Use an empty string to disable the AutoLink feature.

Error Codes

-3 (illegal argument)

See Also

getAutoLinkLayers, AutoLinkLayers

MGMap.setAutoLinkTarget

Syntax

```
void setAutoLinkTarget(String target)
```

Description

Sets the name of the target window where Autodesk MapGuide will display the HTML pages associated with map features on the AutoLinkLayer.

Parameters

target - The name of the window/frame where you want to display the HTML pages. Use an empty string to specify the default target.

Error Codes

none

See Also

getAutoLinkTarget, AutoLinkTarget

MGMap.setAutoRefresh

Syntax

`void setAutoRefresh(boolean autoRefresh)`

Description

Enables or disables the automatic refresh flag.

This method does not initiate a display refresh. You must call the refresh method to do that.

Calling the refresh, setURL, zoomGotoDlg, or zoomGotoLocation methods always invoke a display refresh, regardless of the automatic refresh flag setting. Also note that setAutoRefresh does not work when called from event-handling code for onViewChanging or onMapLoaded. If you call setAutoRefresh from these events, the method will fail and will set the error code to -14 (refresh disabled).

By default, intermediate updates of maps every 1.5 seconds during a drawing operation are enabled for all Autodesk MapGuide Viewers. This allows users to progressively see a map as it is being rendered, rather than waiting until the entire map is complete. To change this behavior, you can disable intermediate updates using the setIntermediateUpdatesEnabled method or IntermediateUpdatesEnabled property.

Parameters

autoRefresh - Automatic refresh parameter. True to enable it; False to disable it.

Error Codes

-14 (refresh disabled)

See Also

getAutoRefresh, refresh, AutoRefresh, getIntermediateUpdatesEnabled setIntermediateUpdatesEnabled, IntermediateUpdatesEnabled

MGMap.setBusyStateChangedObserver

Syntax

`void setBusyStateChangedObserver(MGBusyStateChangedObserver observer)`

Description

Activates the Java object that observes the Autodesk MapGuide Viewer busy state.

You cannot use the onBusyStateChanged event without first calling this method.

Applies To

Plug-In and Java Edition only

Parameters

observer - MGBusyStateChangedObserver, the observer interface implementation, or null to disable event observation.

Error Codes

none

See Also

getBusyStateChangedObserver,

MGMap.setDefaultTarget

Syntax

`void setDefaultTarget(String target)`

Description

Sets the name of the default target window where Autodesk MapGuide will display HTML documents.

Parameters

target - The name of the window/frame where you want to display HTML documents. Use an empty string to specify the default target.

Error Codes

none

See Also

getDefaultTarget, DefaultTarget

MGMap.setDoubleClickObserver

Syntax

`void setDoubleClickObserver(MGDoubleClickObjectObserver observer)`

Description

Sets the Java object for observing double-click events.

Applies To

Plug-In and Java Edition only

Parameters

observer - MGDoubleClickObjectObserver observer object, or null to disable event observation.

Error Codes

none

See Also

getDoubleClickObserver, onDoubleClickObject

MGMap.setErrorTarget

Syntax

```
void setErrorTarget(String target)
```

Description

Sets the name of the target window where Autodesk MapGuide will display error messages.

Parameters

target - The name of the window/frame in which to display error messages. Use an empty string to specify the default error target.

Error Codes

none

See Also

getErrorTarget, ErrorTarget

MGMap.setIntermediateUpdatesEnabled

Syntax

```
setIntermediateUpdatesEnabled (boolean intermediateUpdatesEnabled)
```

Description

Specifies whether or not the map window should update every 1.5 seconds during a drawing operation or only at the end when all data has been loaded and rendered.

By default, intermediate updates are enabled for all Autodesk MapGuide Viewers. This allows users to progressively see a map as it is being rendered, rather than waiting until the entire map is complete.

Parameters

intermediateUpdatesEnabled - True to specify that intermediate updates should be performed, or False to disable intermediate updates.

Returns

none

Refresh

The setting takes effect on the next redraw or refresh operation.

Error Codes

none

Example

The following functions determine the current status of intermediate updates using getIntermediateUpdatesEnabled. They then use setIntermediateUpdatesEnabled to either disable and enable intermediate updates.

```
function disableIntermediateUpdates()
{
   var map = getMap();
   var status = map.getIntermediateUpdatesEnabled();
   if (status == true) map.setIntermediateUpdatesEnabled(false);
}

function enableIntermediateUpdates()
{
   var map = getMap();
   var status = map.getIntermediateUpdatesEnabled();
   if (status == false) map.setIntermediateUpdatesEnabled(true);
}
```

See Also

getIntermediateUpdatesEnabled, IntermediateUpdatesEnabled

MGMap.setMapLoadedObserver

Syntax

`void setMapLoadedObserver(MGMapLoadedObserver *observer*)`

Description

Activates the Java object that observes when Autodesk MapGuide loads MWFs, enabling the onMapLoaded event.

Applies To

Plug-In and Java Edition only

Parameters

MGMapLoadedObserver - A valid MGMapLoadedObserver object, or a null to disable the onMapLoaded event.

Error Codes

none

See Also

getMapLoadedObserver, onMapLoaded

MGMap.setName

Syntax

```
void setName(String name)
```

Description

Sets the name of the current map.

This is equivalent to setting the name of the map on the General tab of the Properties dialog box in Autodesk MapGuide Author.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

name - Name of the current map.

Error Codes

-3 (illegal argument)

See Also

getName, MapName

MGMap.setObjectLinkTarget

Syntax

```
void setObjectLinkTarget(String target)
```

Description

Sets the name of the target window where Autodesk MapGuide will display the URL associated with a map feature.

Parameters

target - The name of the window/frame in which to display a map feature's URL. Use an empty string to specify no window/frame for object link target.

Error Codes

none

See Also

getObjectLinkTarget, ObjectLinkTarget

MGMap.setPrintingObserver

Syntax

`boolean setPrintingObserver(MGPrintingObserver observer)`

Description

Sets the Java object that observes printing events.

Applies To

Plug-In and Java Edition only

Returns

obs - An MGPrintingObserver object that will handle the onBeginLayout and onEndLayout printing events.

Error Codes

none

See Also

getPrintingObserver, onBeginLayout, onEndLayout

MGMap.setReportTarget

Syntax

`void setReportTarget(String target)`

Description

Sets the name of the target window where Autodesk MapGuide will display reports from View Reports command.

Parameters

target - The name of the window/frame to display reports in, or an empty string if no window/frame is set for report target.

Error Codes

none

See Also

getReportTarget, ReportTarget

MGMap.setScale

Syntax

```
void setScale(double scale)
```

Description

Sets the current map display scale in terms of 1:scale.

The map will refresh centered at the current longitude/latitude using the new scale value.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state.

Parameters

scale - The new map scale. The value must be greater than 0.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getScale, MapScale

MGMap.setSelectionChangedObserver

Syntax

```
void setSelectionChangedObserver(MGSelectionChangedObserver observer)
```

Description

Sets the Java object for observing selection change events.

Applies To

Plug-In and Java Edition only

Parameters

observer - MGSelectionChangedObserver observer object, or null to disable event observation.

Error Codes

none

See Also

getSelectionChangedObserver, onSelectionChanged

MGMap.setSelectionMode

Syntax

```
void setSelectionMode(String mode)
```

Description

Sets the selection mode, which controls how map features are selected when a feature is not fully contained in the selection area.

Parameters

mode - The selection mode, either Centroid (selects features whose center points fall within the selection area) or Intersection (selects features that are completely or partially contained within the selection area).

Error Codes

none

See Also

SelectionMode, getSelectionMode

MGMap.setTrackingID

Syntax

```
boolean setTrackingID(String trackID)
```

Description

Sets the customer tracking ID associated with a map.

This method is available only if the map was created with the *Allow Access To Tracking ID From API* option selected in Autodesk MapGuide Author.

Parameters

trackID - The tracking ID string, which must be 255 characters or less.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-5 (security violation)

See Also

getTrackingID, TrackingID

MGMap.setUrl

Syntax

```
void setUrl(String URL)
```

Description

Sets the URL of the map to display.

The URL may contain the following optional Autodesk MapGuide Viewer parameters: Lat, Lon, Width, Units, and Scale.

Important Notes

- If the map you are specifying with setUrl is outside of the firewall, and the user is attempting to view the map with the Autodesk MapGuide Viewer, Java Edition in Netscape Navigator, Netscape will not display the authentication dialog box that allows the user to enter a user name and password for proxy authentication. Thus, if the user has not already viewed a page outside of the firewall and authenticated with the proxy server, the map will not load. This does not happen when you embed or link to a map in the HTML page; it happens only with setUrl. Therefore, to work around this problem, make sure that the first map the user loads is outside of the firewall so that the user can authenticate with the proxy server.
- This method overrides the autoRefresh flag and always refreshes the map. Subsequent method calls adhere to the original setting of the autoRefresh flag.
- This method does not work during a busy state and does not work if called from within event-handling code for the onViewChanging or onMapLoaded events.

Parameters

URL - The new map URL to display.

Error Codes

-1 (busy)

-3 (illegal argument)

-14 (refresh disabled)

See Also

getUrl, URL, FullUrl

MGMap.setUrlListState

Syntax

```
void setUrlListState(boolean state)
```

Description

Enables or disables automatic generation of an HTML document listing the names and URLs of map features associated with URLs.

The HTML document is displayed in the UrlListTarget window.

Parameters

state - True to enable the URL list feature; False to disable.

Error Codes

none

See Also

getUrlListState, UrlListState

MGMap.setUrlListTarget

Syntax

`void setUrlListTarget(String target)`

Description

Sets the name of the target window where Autodesk MapGuide will display the names and URLs associated with a map feature as the user pauses the mouse pointer over that feature.

The URL lists are generated only if this functionality is enabled through setUrlListState. To verify whether it is enabled, use getUrlListState.

Parameters

target - The name of the window/frame in which to display the URL list.

Error Codes

none

See Also

getUrlListTarget, UrlListTarget

MGMap.setViewChangedObserver

Syntax

`void setViewChangedObserver(MGViewChangedObserver observer)`

Description

Sets the Java object for observing view change events.

Applies To

Plug-In and Java Edition only

Parameters

observer - MGViewChangedObserver observer object, or null to disable event observation.

Error Codes

none

See Also

getViewChangedObserver, onViewChanged

MGMap.setViewChangingObserver

Syntax

`void setViewChangingObserver(MGViewChangingObserver observer)`

Description

Sets the Java object for observing view change events.

Applies To

Plug-In and Java Edition only

Parameters

observer - MGViewChangingObserver observer object, or null to disable event observation.

Error Codes

none

See Also

getViewChangingObserver, onViewChanging

MGMap.setViewDistanceExObserver

Syntax

`setViewDistanceExObserver(MGViewDistanceExObserver observer)`

Description

Sets the Java object for observing view distance events.

Applies To

Plug-In and Java Edition only

Parameters

observer - The Observer object, or null to disable event observation.

Error Codes

none

See Also

getViewDistanceExObserver, onViewedDistanceEx

MGMap.setViewDistanceObserver

Syntax

```
void setViewDistanceObserver(MGViewDistanceObserver observer)
```

Description

Sets the Java object for view distance events.

Applies To

Plug-In and Java Edition only

Parameters

observer - MGViewDistanceObserver observer object, or null to disable event observation.

Error Codes

none

See Also

getViewDistanceObserver, onViewedDistance

MGMap.setWidth

Syntax

```
void setWidth(double width, String units)
```

Description

Sets the current width of the map in the specified unit of measure.

The map will refresh centered at the same position but will display at the new width.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state. Also, note that because of the floating-point calculation inaccuracy in operating systems, if you set the width of the map using setWidth before calling

getWidth, the value returned by getWidth may be inaccurate, especially if the width is set to a very high number.

Parameters

width - New width of the map. The value must be greater than 0.

units - One of MI, FT, KM, M, "" or null. "" and null only work with the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition. They use the current default units.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getWidth, MapWidth, MapUnits

MGMap.stop

Syntax

```
void stop()
```

Description

Aborts any downloading or redrawing that is currently taking place.

This method is the API equivalent of pressing the Stop toolbar button or selecting the Stop command on the popup menu. The stop method works only with the ActiveX Control control and Plug-In; for the Autodesk MapGuide Viewer, Java Edition, use stopEx instead.

Note that Autodesk MapGuide Viewer might not be able to size highlighted features correctly due to the abrupt halt of its downloading/redrawing process when stop is invoked.

This method functions only during a busy state. For additional information about busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Applies To

ActiveX Control and Plug-In only

Error Codes

none

MGMap.stopEx

Syntax

```
void stopEx()
```

Description

Aborts any downloading or redrawing that is currently taking place.

This method is the API equivalent of pressing the Stop toolbar button or selecting the Stop command on the popup menu. The stopEx method works only with the Java Edition; for the ActiveX Control or Plug-In, use stop instead. (the Autodesk MapGuide Viewer, Java Edition requires this method name because "stop" is a reserved Java keyword.)

Note that Autodesk MapGuide Viewer might not be able to size highlighted features correctly due to the abrupt halt of its downloading/redrawing process when stop is invoked.

Note that this method functions only during a busy state. For additional information about busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Applies To

Java Edition only.

Error Codes

none

MGMap.viewBuffer

Syntax

```
MGMapLayer viewBuffer()
```

Description

Creates buffers around the selected map features.

Use MGBufferSetup methods to specify the buffering parameters.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

MGMapLayer - An MGMapLayer object for the new buffer map layer that was created or the existing buffer map layer to which the buffer objects were added. Returns null if a non-buffer map layer was specified or if no features were selected before calling this method.

Error Codes

-1 (busy)

-12 (invalid selection)

See Also

getBufferSetup, viewBufferDlg, removeMapLayer

MGMap.viewBufferDlg

Syntax

```
MGMapLayer viewBufferDlg()
```

Description

Displays the View Buffer dialog box, where the user specifies the settings for creating the buffers.

This method is the equivalent of the View > View Buffers menu command in the user interface.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

MGMapLayer - The buffer map layer that will contain the new buffer objects.

Error Codes

-1 (busy)

-12 (invalid selection)

See Also

getBufferSetup, viewBuffer

MGMap.viewDistance

boolean viewDistance(String *units*)

Description

Enters View Distance mode and fires the onViewedDistance event when complete.

When viewDistance is invoked, the cursor changes, allowing the user to view distances by clicking points on the map. When the user exits View Distance mode, the onViewedDistance event is fired. This method is the equivalent of the View > Distance menu command in the user interface.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

units - The units used to quantify the distance. The following values are possible: CM, M;, KM, IN, FT, MI, "" or null. "" and null do not work with the ActiveX Control. "" and null use the current default units.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

setViewDistanceObserver, getViewDistanceObserver

MGMap.viewDistanceEx

Syntax

`boolean viewDistanceEx(MGPoint startPt, String units, boolean useMcs)`

Description

Enters View Distance mode and fires the onViewedDistanceEx event when complete.

When viewDistanceEx is invoked, the cursor changes, allowing the user to view distances by clicking points on the map. When the user exits View Distance mode, the onViewedDistanceEx event is fired. This method is similar to viewDistance, but it differs by beginning its measurement from *startPt*, instead of from a user-specified point.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

startPt - An MGPoint object consisting of MCS coordinates representing an optional starting point for the view distance command. This starting point is not necessarily in the current view.

units - The units used to quantify the distance. The following values are possible: CM (centimeters), M (meters), KM (kilometers), IN (inches), FT (feet), MI (miles), "" or null. "" and null use the current default units. Note that null does not work with the ActiveX Control.

useMcs - Specifies the coordinate system of the start point and the points returned by onViewedDistanceEx. True if MCS is to be used; False if lat\lon is to be used.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

setViewDistanceExObserver, getViewDistanceExObserver

MGMap.viewReport

Syntax

```
void viewReport(String report)
```

Description

Invokes the specified report as authored in the map window.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Parameters

report - Name of the report to view.

Error Codes

-1 (busy)

See Also

viewReportsDlg

MGMap.viewReportsDlg

Syntax

```
void viewReportsDlg()
```

Description

Displays the View Reports dialog box, where the user selects the report to display.

When using this method with the ActiveX Control, no more than one request to viewReportsDlg can be processed at a time. To work around this, use the isBusy method to verify that previous requests have been completed before you call viewReportsDlg.

The viewReportsDlg method is the equivalent of the View > View Reports menu command in the user interface.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Error Codes

-1 (busy)

See Also

viewReport

MGMap.zoomGotoDlg

Syntax

```
void zoomGotoDlg()
```

Description

Displays the Zoom Goto dialog box, where the user specifies the zoom width as well as either a place (location) or an address or a ZIP code to zoom to.

This method is the equivalent of the Zoom > Zoom Goto menu command in the user interface.

This method overrides the autoRefresh flag and always refreshes the map. Subsequent method calls adhere to the original setting of the autoRefresh flag.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.. Also note that zoomGotoDlg does not work when called from event-handling code for onViewChanging or onMapLoaded. If you call zoomGotoDlg from these events, the method will fail and will set the error code to -14 (refresh disabled).

Error Codes

-1 (busy)

-14 (refresh disabled)

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the Zoom GoTo dialog box:

```
function showZoomGoTo()
{
    getMap().zoomGoToDlg();
}
```

See Also

zoomGotoLocation

MGMap.zoomGotoLocation

Syntax

boolean zoomGotoLocation(String *category*, String *location*, double *scale*)

Description

Zooms to the specified category, location, and scale.

This method overrides the autoRefresh flag and always refreshes the map. Subsequent method calls adhere to the original setting of the autoRefresh flag.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.. Also note that zoomGotoLocation does not work when called from event-handling code for onViewChanging or

onMapLoaded. If you call zoomGotoLocation from these events, the method will fail and will set the error code to -14 (refresh disabled).

Parameters

category - Category to zoom to.

location - Location to zoom to.

scale - Map scale factor to zoom to if the feature is a point. If the feature is not a point, the map will zoom to the extents of the feature.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-14 (refresh disabled)

See Also

zoomGotoDlg

MGMap.zoomInMode

Syntax

```
void zoomInMode()
```

Description

Enters Zoom In mode.

Users can then use the mouse to zoom by a factor of two or drag a rectangular area to zoom on. This method is the equivalent of the Zoom > Zoom menu command in the user interface.

Note that this method does not work during a busy state.

Error Codes

-1 (busy)

See Also

zoomOutMode

MGMap.zoomOut

Syntax

```
void zoomOut()
```

Description

Zooms to the full extent of the map window as defined in the map window properties.

This method is the equivalent of the Zoom > Unzoom menu command in the user interface.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Error Codes

-1 (busy)

See Also

zoomOutMode

MGMap.zoomOutMode

Syntax

```
void zoomOutMode()
```

Description

Enters the zoom out mode.

Users can then use the mouse to zoom out by a factor of two. This method is the equivalent of the Zoom > Zoom Out menu command in the user interface.

Note that this method does not work during a busy state.

Error Codes

-1 (busy)

See Also

zoomOut, zoomInMode

MGMap.zoomPrevious

Syntax

```
boolean zoomPrevious()
```

Description

Sets the current view of the map back to the previous view of the map.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

See Also

zoomOut, zoomInMode

MGMap.zoomScale

Syntax

```
boolean zoomScale(double lat, double lon, double scale)
```

Description

Zooms to the specified map center (lat, lon) and scale.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state.

Parameters

lat - Latitude (*Y*) center of map window.

lon - Longitude (*X*) center of map window. *scale* - Map scale to zoom to.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

zoomScaleDlg

MGMap.zoomScaleDlg

Syntax

```
void zoomScaleDlg()
```

Description

Displays the Zoom Scale dialog box, where the user enters the scale to zoom to.

This method is the equivalent of the Zoom > Zoom Scale menu command in the user interface.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state.

Error Codes

-1 (busy)

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the Zoom Scale dialog box:

```
function showZoomScale()
{
    getMap().zoomScaleDlg();
}
```

See Also

zoomScale

MGMap.zoomSelected

Syntax

```
void zoomSelected()
```

Description

Zooms to an extent covering the selected map features.

If no features are selected, the method has no effect. This method is the equivalent of the Zoom > Zoom Selected menu command in the user interface.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state.

Error Codes

-1 (busy)

See Also

getSelection, Selection

MGMap.zoomWidth

Syntax

```
boolean zoomWidth(double lat, double lon, double width,
  String units)
```

Description

Zooms to the specified map center and width.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state.

Parameters

lat - Latitude (*Y*) center of map window.

lon - Longitude (*X*) center of map window.

width - Map width to zoom to.

unit - One of the following: M (meters), KM (kilometers), FT ('feet'), MI ('miles'), "", and null.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

zoomWidthDlg

MGMap.zoomWidthDlg

Syntax

```
void zoomWidthDlg()
```

Description

Displays the Zoom Width dialog box, where the user enters the width of the map window to zoom to.

This method is the equivalent of the Zoom > Zoom Width menu command in the user interface.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onMapLoaded events.

Note that this method does not work during a busy state.

Error Codes

-1 (busy)

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the Zoom Width dialog box.

```
function showZoomWidth()
{
   getMap().zoomWidthDlg();
}
```

See Also

zoomWidth

MGMap.ApiVersion

Description

Gets the API version for the Autodesk MapGuide ActiveX Control.

The API version corresponds to the installed version of the control as follows:

- Autodesk MapGuide Release 2.5 uses API version *1.0*
- Autodesk MapGuide Release 2.5.5 uses API version *1.1*
- Autodesk MapGuide Release 3.0 uses API version *1.2*
- Autodesk MapGuide Release 4.0.3.4 uses API version *1.3*
- Autodesk MapGuide Release 4.0.3.6 uses API version *1.4*
- Autodesk MapGuide Release 4.1 uses API version *1.5*
- Autodesk MapGuide Release 5.0 uses API version *5.0*
- Autodesk MapGuide Release 6.0 uses API version *6.0*

Type

String (read-only)

Error Codes

none

See Also

getApiVersion

MGMap.AutoLinkDelay

Description

Gets or sets the number of seconds before the next HTML page associated with a map feature from an AutoLinkLayer will be displayed in the AutoLinkTarget window.

If the delay value is 0, the automatic timed update feature is disabled, and the AutoLinkTarget window will be updated whenever the mouse passes over a map feature on an AutoLinkLayer. Negative numbers are not allowed, and setting a value less than 5 is not advisable because requests will likely be issued before the previous requests can be serviced, resulting in links being passed over and overburdening the server. The default value is 20 seconds.

Type

int (read/write)

Error Codes

-3 (illegal argument)

See Also

getAutoLinkDelay, setAutoLinkDelay

MGMap.AutoLinkLayers

Description

Gets or sets the name of the layer that you want to use as the AutoLink layer.

When the user pauses the mouse pointer over a map feature on this layer, and that map feature has a URL associated with it, the HTML page at that URL will automatically display in the AutoLinkTarget window. AutoLinks apply only to map features that are visible in the current view.

Type

String (read/write)

Error Codes

-3 (illegal argument)

See Also

getAutoLinkLayers, setAutoLinkLayers

MGMap.AutoLinkTarget

Description

Gets or sets the name of the target window where Autodesk MapGuide will display the HTML pages associated with map features on AutoLinkLayers as the user pauses the mouse pointer over those map features.

Type

String (read/write)

Error Codes

none

See Also

getAutoLinkTarget, setAutoLinkTarget

MGMap.AutoRefresh

Description

Gets or sets the automatic refresh flag.

Note: By default, intermediate updates of maps every 1.5 seconds during a drawing operation are enabled for all Autodesk MapGuide Viewers. This allows users to progressively see a map as it is being rendered, rather than waiting until the entire map is complete. To change this behavior, you can disable intermediate updates using the setIntermediateUpdatesEnabled method or IntermediateUpdatesEnabled property.

Type

boolean (read/write)

Error Codes

-14 (refresh disabled)

See Also

getAutoRefresh, setAutoRefresh, refresh, getIntermediateUpdatesEnabled setIntermediateUpdatesEnabled, IntermediateUpdatesEnabled

MGMap.BufferSetup

Description

Gets an object that contains buffering properties, such as the distance from the selected features to the edge of the buffers.

Autodesk MapGuide uses these properties to create buffer objects.

Type

MGBufferSetup (read-only)

Error Codes

none

See Also

getBufferSetup

MGMap.Busy

Description

Returns True if Autodesk MapGuide Viewer is busy loading or drawing a map.

Before calling other API methods, you should use the Busy property to test whether Autodesk MapGuide Viewer is busy in an if statement. However, do not attempt to use it in a while loop to wait until Autodesk

MapGuide Viewer becomes available. This may cause the Web browser and Autodesk MapGuide Viewer to freeze up.

Type

boolean (read-only)

Error Codes

none

See Also

isBusy

MGMap.DefaultTarget

Description

Gets or sets the name of the default target window where Autodesk MapGuide will display HTML documents.

This window will be used as the AutoLinkTarget, ErrorTarget, and ReportTarget if these targets are not otherwise set.

Type

String (read/write)

Error Codes

none

See Also

getDefaultTarget, setDefaultTarget

MGMap.DisplayUnits

Description

Gets or sets the current display units, either M (meters), KM (kilometers), FT (feet), or MI (miles), as seen in the map width of the status bar.

Type

String (read/write)

Error Codes

none

See Also

getDisplayUnits

MGMap.ErrorTarget

Description

Gets or sets the name of the target window where Autodesk MapGuide will display error messages.

If this property is not set, the DefaultTarget setting will be used.

Type

String (read/write)

Error Codes

none

See Also

getErrorTarget, setErrorTarget

MGMap.FullUrl

Description

Gets the full URL of the map that is currently displayed by Autodesk MapGuide, including parameters specifying the current location and any non-default options.

Type

String (read-only)

Error Codes

none

See Also

URL, getUrl, setUrl

MGMap.IntermediateUpdatesEnabled

Description

Specifies whether or not the map window should update every 1.5 seconds during a drawing operation or only at the end when all data has been loaded and rendered.

By default, intermediate updates are enabled for all Autodesk MapGuide Viewers. This allows users to progressively see a map as it is being rendered, rather than waiting until the entire map is complete.

Type

boolean (read/write)

Error Codes

none

See Also

setIntermediateUpdatesEnabled, getIntermediateUpdatesEnabled

MGMap.LastError

Description

Gets the MGError object that contains the error value and message for the most recent API error.

Type

MGError (read-only)

Error Codes

none

See Also

getLastError

MGMap.Lat

Description

Gets the latitude (*Y*) value in decimal degrees of the center of the map area currently displayed.

Latitudes are positive north of the equator and negative south of the equator.

Type

double (read-only)

Error Codes

-10 (design time only)

See Also

getLat

MGMap.LayersViewWidth

Description

Gets or sets the width of the legend. In order to change this property at runtime, it must be non-zero when the MGMap object is created.

Type

int (read/write)

Error Codes

none

MGMap.Lon

Description

Gets the longitude (*X*) value in decimal degrees of the center of the map area currently displayed.

Longitudes are positive east of the Greenwich meridian and negative west of the meridian.

Type

double (read-only)

Error Codes

-10 (design time only)

See Also

getLon

MGMap.MapHeight

Description

Gets the height of the map. This value is expressed in the units set with the MapUnits property.

Type

double (read-only)

Error Codes

none

See Also

getHeight

MGMap.MapLayerGroups

Description

Gets a collection containing all the map layer groups defined in the map.

Type

Array/Vector of MGMapLayer (read-only)

Error Codes

none

See Also

getMapLayerGroup, getMapLayerGroups

MGMap.MapLayers

Description

Gets a collection containing all the map layers defined in the map.

Type

Array/Vector of MGMapLayer (read-only)

Error Codes

none

See Also

getMapLayersEx, getMapLayer, getMapLayers

MGMap.MapName

Description

Gets or sets the name of the map.

You can also set the name of the map in the Preferences dialog box in Autodesk MapGuide Author.

Type

String (read/write)

Error Codes

none

See Also

getName, setName

MGMap.MapScale

Description

Gets or sets the current map display scale in terms of 1:scale.

Type

double (read/write)

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getScale, setScale

MGMap.MapUnits

Description

Gets or sets the current units that will be used when getting or setting the MapWidth property, one of M (meters), KM (kilometers), FT (feet), or MI (miles).

Not to be confused with the coordinate system units, which the user can set to control the distance units in the Autodesk MapGuide Viewer, or default width units, which the map author uses to set the default width of the MWF in the Properties dialog box in Autodesk MapGuide Author. When using methods getWidth and setWidth instead of using the MapWidth property, you can specify this map unit as a parameter of the method.

Type

String (read/write)

Error Codes

-3 (illegal argument)

See Also

getUnits, getWidth, setWidth

MGMap.MapWidth

Description

Gets or sets the current map width.

This value is expressed in the units set with the MapUnits property.

Type

double (read/write)

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getWidth, setWidth

MGMap.McsScaleFactor

Description

Gets the conversion factor used to convert from the current Mapping Coordinate System (MCS) to meters.

The MCS can vary based upon factors such as the current projection of the map.

Type

double (read-only)

Error Codes

none

See Also

getMCSScaleFactor

MGMap.Mode

Gets the current viewer mode of the map, such as zoomIn, digitize, pan, and so on.

Types

String (read-only)

Returns

Busy, DigitizeCircle, DigitizeLine , DigitizePoint, DigitizePolyline, DigitizePolygon, DigitizeRectangle, Pan, Select, SelectPolygon, SelectRadius, ViewDistance, ZoomIn, ZoomOut, and Unknown.

Error Codes

None

See Also

getMode

MGMap.ObjectLinkTarget

Description

Gets or sets the name of the target window where Autodesk MapGuide will display the URL associated with a map feature.

If this property is not set, the DefaultTarget setting will be used.

Type

String (read/write)

Error Codes

none

See Also

getObjectLinkTarget, setObjectLinkTarget

MGMap.PageSetup

Description

Gets an object that contains properties used for printing maps.

Type

MGPageSetup (read-only)

Error Codes

none

See Also

getPageSetup, pageSetupDlg, printDlg

MGMap.RedlineSetup

Description

Gets an object containing all properties for drawing redline features.

This object provides access to the properties used to draw each of the individual feature types.

Type

MGRedlineSetup (read-only)

Error Codes

none

See Also

getRedlineSetup

MGMap.ReportTarget

Description

Gets or sets the name of the target window where Autodesk MapGuide will display reports from View Reports command.

If this property is not set, the DefaultTarget setting will be used.

Type

String (read/write)

Error Codes

none

See Also

getReportTarget, setReportTarget

MGMap.Selection

Description

Gets the currently selected map features.

Type

MGSelection (read-only)

Error Codes

none

See Also

getSelection

MGMap.SelectionMode

Description

Gets or sets the selection mode, either Centroid or Intersection.

Type

String (read/write)

Error Codes

-3 (illegal argument)

See Also

getSelectionMode, setSelectionMode

MGMap.StatusBar

Description

Gets or sets the state of the status bar, either "Off" or "On".

Type

String (read/write)

Error Codes

none

MGMap.Symbols

Description

Gets a collection of strings representing the symbols available for use as point attributes.

Type

MGCollection (read-only)

Error Codes

none

See Also

getSymbols

MGMap.ToolBar

Description

Gets or sets the state of the toolbar, either "Off", "On", or "Big".

Type

String (read/write)

Error Codes

none

MGMap.TrackingID

Description

Gets or sets the customer tracking ID associated with a map.

The tracking ID must be 255 characters or less. This property is available only if the map was created with the *Allow Access to Tracking ID from API* option selected in Autodesk MapGuide Author.

Type

String (write)

Error Codes

-1 (busy)

-3 (illegal argument)

-5 (security violation)

See Also

getTrackingID, setTrackingID

MGMap.Url

Description

Gets or sets the URL of the map that is currently displayed by Autodesk MapGuide.

Type

String (read/write)

Error Codes

-1 (busy)

-3 (illegal argument)

-14 (refresh disabled)

See Also

getUrl, setUrl, FullUrl

MGMap.UrlListState

Description

Specifies whether Autodesk MapGuide will generate an HTML document that lists the names and URLs (if any) that are associated with the selected map feature(s).

The HTML document is displayed in the UrlListTarget window.

Type

boolean (read/write)- True if the document will be displayed; otherwise, returns False.

Error Codes

none

See Also

getUrlListState, setUrlListState

MGMap.UrlListTarget

Description

Gets or sets the name of the target window where Autodesk MapGuide will display the names and URLs of map features associated with URLs.

The URL lists are only generated if this feature is enabled through UrlListState or setUrlListState. If UrlListTarget is not set, the DefaultTarget setting will be used instead.

Type

String (read/write)

Error Codes

none

See Also

getUrlListTarget, setUrlListTarget

MGMap.Version

Description

Gets the version string of the Autodesk MapGuide Viewer.

Type

String (read-only)

Error Codes

none

See Also

getVersion

MGMap.ViewerType

Description

Returns a string indicating whether the Autodesk MapGuide Viewer is the Autodesk MapGuide Viewer ActiveX Control, the Autodesk MapGuide Viewer Plug-In, or the Autodesk MapGuide Viewer, Java Edition.

Type

String (read-only)

Error Codes

none

See Also

getViewerType

MGMapApplet Object

The MGMapApplet is an applet that implements the MGMap interface.

You can use MGMapApplet to instantiate the Autodesk MapGuide Viewer, Java Edition in your HTML page by setting the CODE value of the APPLET tag to MGMapApplet.

Methods

getStaticVersion Gets the Autodesk MapGuide Viewer, Java Edition version number as a string.

MGMapApplet.getStaticVersion

Syntax

```
public static String getStaticVersion()
```

Description

Gets the Autodesk MapGuide Viewer, Java Edition version number as a string.

Applies To

Java Edition only.

Returns

String - The Autodesk MapGuide version.

MGMapComponent Object

The MGMapComponent class instantiates Autodesk MapGuide Viewer as a user interface object. MGMapComponent extends the Component class and implements the MGMap interface.

Methods

MGMapComponent	This method constructs a new instance of MGMapComponent.
getStaticVersion	Gets the Autodesk MapGuide Viewer, Java Edition version number as a string. The version number is the same as the number returned by the getVersion method of MGMap.

MGMapComponent.MGMapComponent

Syntax

`MGMapComponent(MGMapContext` *`context`*`)`
or

`MGMapComponent()`

Description

This method constructs a new instance of MGMapComponent.

Applies To

Java Edition only

Parameters

context - Optional. An MGMapContext instance that provides various contextual services to MGMapComponent.

MGMapComponent.getStaticVersion

Syntax

```
public static String getStaticVersion()
```

Description

Gets the Autodesk MapGuide Viewer, Java Edition version number as a string. The version number is the same as the number returned by the getVersion method of MGMap.

Applies To

Java Edition only

Returns

String - The Autodesk MapGuide version.

MGMapContext Object

The MGMapContext is a Java interface whose methods provide browser services to MGMapComponent.

MGMapComponent can utilize these methods to communicate with any applications that drive MGMapComponent.

Methods

getCodeBase	Gets the Autodesk MapGuide Viewer component's code base URL.
getDocumentBase	Gets the Autodesk MapGuide Viewer component's base document URL.
getParameter	Retrieves the value of a parameter.
showDocument	Displays a new document in an HTML window or frame, with or without a specified target.
showStatus	Displays a message in the application context.

MGMapContext.getCodeBase

Syntax

```
public URL getCodeBase()
```

Description

Gets the Autodesk MapGuide Viewer component's code base URL.

Applies To

Java Edition only.

Returns

URL - URL prefix for the code that defines the Autodesk MapGuide Viewer component.

MGMapContext.getDocumentBase

Syntax

```
public URL getDocumentBase()
```

Description

Gets the Autodesk MapGuide Viewer component's base document URL.

Applies To

Java Edition only.

Returns

URL - URL prefix for the document displayed by the Autodesk MapGuide Viewer component.

MGMapContext.getParameter

Syntax

```
public String getParameter(String paramName)
```

Description

Retrieves the value of a parameter.

Applies To

Java Edition only.

Returns

String - The value of the parameter defined by the name *paramName*. Null is returned if the parameter does not exist.

MGMapContext.showDocument

Syntax

```
public void showDocument(URL url)
  public void showDocument(URL url, String target)
```

Description

Displays a new document in an HTML window or frame, with or without a specified target.

Applies To

Java Edition only.

MGMapContext.showStatus

Syntax

```
public void showStatus(String status)
```

Description

Displays a message in the application context.

Where (and if) the message displays varies by application context.

Applies To

Java Edition only.

MGMapLayer Object

The MGMapLayer object represents a layer on the map.

Note that instances of the class MGMapLayer cannot be publicly created.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGMapLayer extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGMapLayer is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

compressThemeCats	Displays the list of theme categories shown in the legend for the specified map layer in collapsed form, so that only one category from the list is displayed.
createMapObject	Adds an empty feature to the layer. This method works only with redline layers.
equals	Compares this MGMapLayer object to the specified object.
expandThemeCats	Displays the list of theme categories shown in the legend for the specified map layer in expanded form, so that all categories in the list are displayed.
getLayerSetup	Gets an MGMapLayerSetup object.
getLayerType	Returns a string indicating the layer type.
getLegendLabel	Gets the string used to label the map layer in the legend.
getMap	Gets the map that contains the map layer.
getMapLayerGroup	Gets the map layer group to which the layer belongs.

getMapLayerStyles	Gets a collection containing all of the MGMapLayerStyle objects associated with the layer.
getMapObject	Gets the map feature with the specified key from the layer.
getMapObjects	Returns a Vector (or MGCollection object) containing the map features of the layer that fall within the map extent (the part of the map that is displayed).
getMapObjectsEx	Gets an MGCollection object containing the map features of the layer that fall within the map extent.
getName	Gets the name of the map layer.
getPriority	Gets the priority of the map layer.
getRebuild	Gets the map layers rebuild flag.
getSelectability	Returns True if map features within the map layer may be selected.
getShowInLegend	Returns a boolean value indicating whether the map layer should be displayed in the legend.
getSQLWhere	Gets the SQL Where clause for the layer.
getType	Returns a string indicating the type of features the layer contains.
getVisibility	Returns True if the map layer is potentially visible.
isDynamic	Returns a boolean value indicating whether the map layer is dynamic.
isLocked	Indicates whether Autodesk MapGuide Viewer has access to either coordinate values or map layer setup data, depending on the value of the function parameter that is passed in.
isStatic	Returns a boolean value indicating whether the map layer is static.
isVisible	Returns True if the map layer is currently visible on the map.
lock	Prevents Autodesk MapGuide Viewer from having access to either coordinate values or map layer setup data, depending on the value of the function parameter that is passed in.
removeAllObjects	Deletes all features from the layer.
removeObject	Deletes a map feature on the layer.
removeObjects	Deletes the specified map features from the layer.
setAccessKey	Sets the access key associated with a map layer.
setLegendLabel	Sets the string used to label the map layer in the legend. After calling this method, the legend is automatically redrawn.
setMapLayerGroup	Specifies which map layer group the current layer should belong to.
setName	Sets the name of the map layer.
setPriority	Sets the priority of the map layer.

setRebuild	Changes the map layer's rebuild flag.
setSelectability	Sets the selectability of map features within the map layer.
setShowInLegend	Sets a flag indicating whether the map layer should be displayed in the legend.
setSQLWhere	Sets the SQL Where clause for the layer.
setVisibility	Sets the visibility flag indicating whether the map layer is potentially visible.
unlock	Gives Autodesk MapGuide Viewer access to either coordinate values or map layer setup data, depending on the value of the function parameter that is passed in.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

AccessKey	Sets the access key for the layer. This value must be 255 characters or less.
Dynamic	Returns True if the map layer is dynamic.
LayerSetup	Gets the MGMapLayerSetup object for this layer. Applies to all layer types except Drawing layers or layers that have been secured.
LayerType	Returns a string indicating the layer type.
LegendLabel	Gets or sets the legend label used by the map layer.
Map	Gets the map containing the map layer.
MapLayerGroup	Specifies the name of the map layer group that the current layer belongs to.
MapLayerStyles	Gets a collection containing all of the MGMapLayerStyle objects associated with the layer.
MapObjects	Gets an MGCollection object containing the map features (MGMapObject objects) of the layer that fall within the map extent.
Name	Gets or sets the name of the map layer.
Priority	Gets or sets the priority of the map layer.
Rebuild	Gets or sets the map layer's rebuild flag.
Selectability	Gets or sets whether features within the map layer may be selected.
ShowInLegend	Specifies whether or not to show the map layer in the legend.
SQLWhere	Gets or sets the SQL Where clause for the dynamic layer.
Static	Returns True if the map layer is static.
Type	A string indicating the type of features the layer contains.

Visibility	Gets or sets whether the map layer is potentially visible.
Visible	Indicates whether the map layer is visible at the current map scale.

MGMapLayer.compressThemeCats

Syntax

```
void compressThemeCats()
```

Description

Displays the list of theme categories shown in the legend for the specified map layer in collapsed form, so that only one category from the list is displayed.

After calling this method, the legend is automatically redrawn. This method is the equivalent of the Compress menu command in the user interface.

Error Codes

none

See Also

expandThemeCats

MGMapLayer.createMapObject

Syntax

```
MGMapObject createMapObject(String key, String name, String url)
```

Description

Adds an empty feature to the layer. This method works only with redline layers.

Parameters

key - The feature key, which must be either a unique value within the layer, or an empty string ("""). If you specify an empty string, a non-permanent key is generated automatically for the feature. If you specify an explicit key value and that key already exists, an error is set and null is returned.

name - The feature name (or """).

url - The URL link for the feature (or """).

Returns

MGMapObject - An MGMapObject if successful; null otherwise.

Error Codes

-1 (busy)

-3 (illegal argument)

-15 (does not apply -- not a redline layer)

MGMapLayer.equals

Syntax

```
boolean equals(Object obj)
```

Description

Compares this MGMapLayer object to the specified object.

Parameters

obj - The object with which to compare this MGMapLayer.

Error Codes

-3 (illegal argument)

Returns

boolean - Returns True if the two objects are identical; otherwise, returns False.

MGMapLayer.expandThemeCats

Syntax

```
void expandThemeCats()
```

Description

Displays the list of theme categories shown in the legend for the specified map layer in expanded form, so that all categories in the list are displayed.

After calling this method, the legend is automatically redrawn. This method is the equivalent of the Expand menu command in the user interface.

Error Codes

none

See Also

compressThemeCats

MGMapLayer.getLayerSetup

Syntax

```
MGMapLayerSetup getLayerSetup()
```

Description

Gets an MGMapLayerSetup object.

The MGMapLayerSetup object provides run-time access to this layer's setup attributes. Applies to all layer types except "Drawing" and any layers that are secured. A layer is secured by the map author in Autodesk MapGuide Author. The map author can prevent access of layer setup data completely or can secure it with a passkey that the API can pass in with the unlock method. If your layer setup data has been secured with a passkey, you will need to first call unlock with the correct passkey to enable getLayerSetup.

Returns

MGMapLayerSetup - The MGMapLayerSetup object for this layer.

Error Codes

-5 (security violation)

-15 (does not apply)

See Also

unlock

MGMapLayer.getLayerType

Syntax

```
String getLayerType()
```

Description

Returns a string indicating the layer type.

The older method getType will continue to exist for backward compatibility and will return the types of features on the layer (e.g., "Points", "Polylines", "Polygons", "Drawing", and so on).

Returns

String - The layer type, either Unknown, Point, Text, Polyline, Polygon, Raster, Buffer, GIS Desgin Server Theme, Autodesk DWG, or Redline.

Error Codes

none

See Also

LayerType, getType in MGMapObject

MGMapLayer.getLegendLabel

Syntax

```
String getLegendLabel()
```

Description

Gets the string used to label the map layer in the legend.

Returns

String - The label that appears in the legend for the map layer.

Error Codes

none

See Also

setLegendLabel

MGMapLayer.getMap

Syntax

```
MGMap getMap()
```

Description

Gets the map that contains the map layer.

Returns

MGMap - The map that contains the map layer.

Error Codes

none

See Also

Map

MGMapLayer.getMapLayerGroup

Syntax

```
MGMapLayerGroup getMapLayerGroup()
```

Description

Gets the map layer group to which the layer belongs.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Returns

MGMapLayerGroup - The group to which the current layer belongs. If the layer does not belong to a group, null is returned.

Error Codes

-1 (busy)

See Also

setMapLayerGroup, mapLayerGroup

MGMapLayer.getMapLayerStyles

Syntax

```
MGCollection getMapLayerStyles()
```

Description

Gets a collection containing all of the MGMapLayerStyle objects associated with the layer.

Returns

MGCollection - An MGCollection of MGMapLayerStyle objects. Returns an empty MGCollection if unsuccessful.

Error Codes

none

See Also

mapLayerStyles, MGMapLayerStyle object

MGMapLayer.getMapObject

Syntax

```
MGMapObject getMapObject(String key)
```

Description

Gets the map feature with the specified key from the layer.

Each map feature within a layer must have a unique key.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

key - Key of the map feature to retrieve.

Returns

MGMapObject - The map feature, or null if the key does not exist.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getMapObjectsEx

MGMapLayer.getMapObjects

Syntax

ActiveX Control: `MGCollection getMapObjects()`

Plug-In/Java Edition: `Vector getMapObjects()`

Description

Returns a Vector (or MGCollection object) containing the map features of the layer that fall within the map extent (the part of the map that is displayed).

If the pan function is used, a subsequent call to getMapObjects will return the displayed map features. However, if the zoom function is used, a subsequent call to getMapObjects will return the highest number of map features that were ever displayed.

For compatibility between the Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer ActiveX Control of Autodesk MapGuide Viewer, use getMapObjectsEx instead of getMapObjects.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

ActiveX Control: MGCollection - A collection of MGMapObject objects. Returns empty MGCollection if unsuccessful.

Plug-In: Vector - A java.util.Vector of MGMapObject objects. Returns empty Vector if unsuccessful.

Error Codes

-1 (busy)

See Also

getMapObjectsEx

MGMapLayer.getMapObjectsEx

Syntax

```
MGCollection getMapObjectsEx()
```

Description

Gets an MGCollection object containing the map features of the layer that fall within the map extent (the part of the map that is displayed).

If the pan function is used, a subsequent call to getMapObjectsEx will return the displayed map features. However, if the zoom function is used, a subsequent call to getMapObjectsEx will return the highest number of map features that were ever displayed.

This method replaces the getMapObjects method in earlier versions of the Autodesk MapGuide Viewer API and supports both the Autodesk MapGuide Viewer Plug-In and ActiveX Control with the same syntax. The older method will still work in Release 6, but you should use getMapObjectsEx for all future development.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

MGCollection- A collection containing the layer's MGMapObject map features.

Error Codes

-1 (busy)

See Also

MapObjects, getMapObject, getMapObjectsEx in MGSelection

MGMapLayer.getName

Syntax

```
String getName()
```

Description

Gets the name of the map layer.

Returns

String - The map layer's name.

Error Codes

none

See Also

setName, Name

MGMapLayer.getPriority

Syntax

```
double getPriority()
```

Description

Gets the priority of the map layer.

Priorities range from 0 to 100. Layers with higher priorities are drawn on top of those with lower priorities.

Returns

double - The map layer's priority.

Error Codes

none

See Also

setPriority, Priority

MGMapLayer.getRebuild

Syntax

```
boolean getRebuild()
```

Description

Gets the map layers rebuild flag.

If the rebuild flag is set, the map layer will be rebuilt from its original data source the next time the map layer is refreshed.

Only dynamic layers are rebuilt when the rebuild flag is set to True; static layers are not rebuilt when you refresh the layer.

Returns

boolean - The map layer's current rebuild flag setting.

Error Codes

none

See Also

setRebuild, Rebuild

MGMapLayer.getSelectability

Syntax

```
boolean getSelectability()
```

Description

Returns True if map features within the map layer may be selected.

Returns

boolean - The selectability of features in the map layer. True if features are selectable; otherwise, returns False.

Error Codes

none

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It toggles the selectability of features on a layer called "World Countries":

```
function toggleSelectability()
{
    map = getMap();
    var layer = map.getMapLayer("World Countries");
    if ( layer.getSelectability() == false )
        layer.setSelectability(true);
    else
        layer.setSelectability(false);
}
```

See Also

setSelectability, Selectability

MGMapLayer.getShowInLegend

Syntax

```
boolean getShowInLegend()
```

Description

Returns a boolean value indicating whether the map layer should be displayed in the legend.

Returns

boolean - Returns True if map layer is to be displayed in legend; otherwise, returns False.

Error Codes

none

See Also

setShowInLegend

MGMapLayer.getSQLWhere

Syntax

```
String getSQLWhere()
```

Description

Gets the SQL Where clause for the layer.

This method is functionally equivalent to the MGDatabaseSetup getWhereClause method.

Returns

String - The SQL Where clause currently used by the layer to retrieve dynamic data.

Error Codes

none

See Also

setSQLWhere, SQLWhere

MGMapLayer.getType

Syntax

```
String getType()
```

Description

Returns a string indicating the type of features the layer contains.

Returns

String - One of the following values to represent the layer type: Unknown, Point, Text, Polyline, Polygon, Raster, or Drawing. Because each layer can contain only one type of feature, each layer is associated with a single type. A new method, getLayerType, replaces getType, but getType still works for backward compatibility.

Error Codes

none

See Also

Type, getType in MGMapObject

MGMapLayer.getVisibility

Syntax

```
boolean getVisibility()
```

Description

Returns True if the map layer is potentially visible.

The isVisible method indicates whether the map layer is visible at the current map scale.

Returns

boolean - Current visibility of features in the map layer.

Error Codes

none

See Also

setVisibility, Visibility

MGMapLayer.isDynamic

Syntax

```
boolean isDynamic()
```

Description

Returns a boolean value indicating whether the map layer is dynamic.

Returns

boolean - Returns True if the layer is dynamic; False if it is not.

Error Codes

none

See Also

isStatic, Dynamic

MGMapLayer.isLocked

Syntax

boolean isLocked(String *function*)

Description

Indicates whether Autodesk MapGuide Viewer has access to either coordinate values or map layer setup data, depending on the value of the function parameter that is passed in.

A value of "Geometry" is for coordinate data, a value of "LayerSetup" is for layer setup data.

These functions can be locked if the author of the map checked any of the following check boxes on the Security tab of the Map Layer Properties dialog box:

- *Allow access to map layer setup from API with the following passkey*
- *Allow access to coordinate values from API with the following passkey*
- *Do not allow access to coordinate values from API*
- *Do not allow access to map layer setup from API*

Parameters

function - The function to enable: either "Geometry" to access coordinate values, or "LayerSetup" to access layer setup data.

Returns

boolean - Returns True if the method is locked; False if it is not.

Error Codes

none

See Also

lock, unLock

MGMapLayer.isStatic

Syntax

```
boolean isStatic()
```

Description

Returns a boolean value indicating whether the map layer is static.

Returns

boolean - Returns True if the layer is static; otherwise, returns False.

Error Codes

none

See Also

isDynamic, Static

MGMapLayer.isVisible

Syntax

```
boolean isVisible()
```

Description

Returns True if the map layer is currently visible on the map.

If the current scale of the map is outside of the map layer's display range, or if the user has turned off the map layer, the map layer is not displayed on the map, so isVisible returns False. You can determine the display range of a map layer using getMinDisplayRange and getMaxDisplayRange from MGMapLayerStyle.

Returns

boolean - Returns True if the layer is currently visible; otherwise, returns False.

Error Codes

none

See Also

getVisibility, setVisibility, Visibility, Visible

MGMapLayer.lock

Syntax

```
boolean lock(String function)
```

Description

Prevents Autodesk MapGuide Viewer from having access to either coordinate values or map layer setup data, depending on the value of the function parameter that is passed in.

A value of "Geometry" prevents access to coordinate data, a value of "LayerSetup" prevents access to layer setup data. This method is required if the author of the map checked the *Enable access to map layer setup with the following key* or the *Enable access to coordinate values with the following key* check box on the Security tab of the Map Layer Properties dialog box, and the if API unlocked either function with the unlock method.

Parameters

function - The function to enable: either "Geometry" to prevent access to coordinate values, or "LayerSetup" to prevent access to layer setup data.

Returns

boolean - Returns True if the method was able to lock the function; otherwise, returns False.

Error Codes

-3 (illegal argument)

See Also

unlock, isLocked

MGMapLayer.removeAllObjects

Syntax

```
boolean removeAllObjects()
```

Description

Deletes all features from the layer.

This method works only with redline layers.

Depending on the state of the selection changed flag, calling this method on features in the current selection may fire the onSelectionChanged event.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events. Note that this method does not work during a busy state.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-15 (does not apply -- not a redline layer)

See Also

enableSelectionChangedEvent, disableSelectionChangedEvent

MGMapLayer.removeObject

Syntax

`boolean removeObject(MGMapObject object)`

Description

Deletes a map feature on the layer.

This method works only with redline layers.

Depending on the state of the selection changed flag, calling this method on features in the current selection may fire the onSelectionChanged event.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events. Note that this method does not work during a busy state.

Parameters

object - An MGMapObject.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-15 (does not apply -- not a redline layer)

See Also

enableSelectionChangedEvent, disableSelectionChangedEvent

MGMapLayer.removeObjects

Syntax

```
boolean removeObjects(MGCollection objects)
```

Description

Deletes the specified map features from the layer.

This method works only with redline layers.

Depending on the state of the selection changed flag, calling this method on features in the current selection may fire the onSelectionChanged event.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events. Note that this method does not work during a busy state.

Parameters

objects - An MGCollection containing MGMapObjects.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-15 (does not apply -- not a redline layer)

See Also

enableSelectionChangedEvent, disableSelectionChangedEvent

MGMapLayer.setAccessKey

Syntax

```
boolean setAccesskey(String accessKey)
```

Description

Sets the access key associated with a map layer.

Parameters

accessKey - The access key, which must be 255 characters or less.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-15 (does not apply)

See Also

AccessKey

MGMapLayer.setLegendLabel

Syntax

```
void setLegendLabel(String label)
```

Description

Sets the string used to label the map layer in the legend. After calling this method, the legend is automatically redrawn.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

label - The label that appears in the legend for the map layer.

Error Codes

-1 (busy)

See Also

getLegendLabel

MGMapLayer.setMapLayerGroup

Syntax

```
boolean setMapLayerGroup(MGMapLayerGroup layergroup)
```

Description

Specifies which map layer group the current layer should belong to.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

layergroup - The map layer group you want the current layer to be a part of. If you do not want to layer to be part of a group, specify null as an argument.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getMapLayerGroup, mapLayerGroup

MGMapLayer.setName

Syntax

`void setName(String name)`

Description

Sets the name of the map layer.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

name - The new name for the map layer.

Error Codes

-1 (busy)

See Also

getName, Name

MGMapLayer.setPriority

Syntax

`void setPriority(double priority)`

Description

Sets the priority of the map layer.

Priorities range from 0 to 100. Layers with higher priorities are drawn on top of those with lower priorities. setPriority does not automatically redraw the map or legend. You must force a redraw with a call to the MGMap refresh method.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

priority - The new priority for the map layer.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getPriority, Priority

MGMapLayer.setRebuild

Syntax

```
void setRebuild(boolean rebuild)
```

Description

Changes the map layer's rebuild flag.

If the rebuild flag is set to True, the map layer will be rebuilt from its original data source the next time it is refreshed.

Note that only dynamic layers are rebuilt when the rebuild flag is set to True; static layers are not rebuilt when you refresh the layer. To rebuild a static map layer, set the rebuild flag to True and call the refresh method in MGMap.

If you set the rebuild flag to false, the method returns error code -15 ("does not apply") and does not rebuild the layer upon refresh.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

rebuild - The new setting for the map layer's rebuild flag, either True or False.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getRebuild, Rebuild

MGMapLayer.setSelectability

Syntax

`void setSelectability(boolean selectability)`

Description

Sets the selectability of map features within the map layer.

If set to True, map features within the map layer can be selected.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

selectability - New selectability status of features in the map layer.

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It toggles the selectability of features on a layer called "World Countries":

```
function toggleSelectability()
{
    map = getMap();
    var layer = map.getMapLayer("World Countries");
    if ( layer.getSelectability() == false )
        layer.setSelectability(true);
    else
        layer.setSelectability(false);
}
```

See Also

getSelectability, Selectability

MGMapLayer.setShowInLegend

Syntax

`void setShowInLegend(boolean flag)`

Description

Sets a flag indicating whether the map layer should be displayed in the legend.

setShowInLegend does not automatically redraw the map or legend. You must force a redraw with a call to the MGMap refresh method.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

flag - Set to True to display the map layer in legend; otherwise, set to false.

Error Codes

-1 (busy)

See Also

getShowInLegend

MGMapLayer.setSQLWhere

Syntax

```
void setSQLWhere(String whereClause)
```

Description

Sets the SQL Where clause for the layer.

This method automatically sets the layer's rebuild flag to True. See setRebuild for details.

This method is functionally equivalent to the MGDatabaseSetup setWhereClause method.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*. Also note that in some cases, a very long SQL Where clause will cause a data request error.

Parameters

whereClause - The new SQL Where clause to be used by the layer to retrieve dynamic data.

Error Codes

-1 (busy)

See Also

getSQLWhere, SQLWhere

MGMapLayer.setVisibility

Syntax

```
void setVisibility(boolean visibility)
```

Description

Sets the visibility flag indicating whether the map layer is potentially visible.

The isVisible method indicates whether the map layer is visible at the current map scale. This method does not automatically redraw the map and legend. To redraw the map, use the MGMap refresh method.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

visibility - The new setting for the visibility flag.

Error Codes

-1 (busy)

See Also

getVisibility, Visibility

MGMapLayer.unlock

Syntax

```
boolean unlock(String function, String key)
```

Description

Gives Autodesk MapGuide Viewer access to either coordinate values or map layer setup data, depending on the value of the function parameter that is passed in.

A value of "Geometry" accesses coordinate data, a value of "LayerSetup" accesses layer setup data. This method is required if the author of the map checked the *Enable access to map layer setup with the following key* or the *Enable access to coordinate values with the following key* check box on the Security tab of the Map Layer Properties dialog box.

Parameters

function - The function to enable: either Geometry to access coordinate values, or LayerSetup to access layer setup data.

key - The key that was specified in the Autodesk MapGuide Author when the author of the map selected either of these checkboxes: *Allow access to coordinate values from the API with the following passkey* or *Allow access to map layer setup from the API with the following passkey*.

Returns

boolean - Returns True if the method was able to unlock the function; otherwise, returns False.

Error Codes

-3 (illegal argument)

See Also

lock, isLocked

MGMapLayer.AccessKey

Description

Sets the access key for the layer. This value must be 255 characters or less.

Type

String (write)

Error Codes

-1 (busy)

-3 (illegal argument)

-15 (does not apply)

See Also

setAccessKey

MGMapLayer.Dynamic

Description

Returns True if the map layer is dynamic.

Type

boolean (read-only)

Error Codes

none

See Also

static, isDynamic

MGMapLayer.LayerSetup

Description

Gets the MGMapLayerSetup object for this layer. Applies to all layer types except "Drawing" or layers that have been secured.

Type

MGMapLayerSetup (read-only)

Error Codes

-5 (security violation)

-15 (does not apply)

See Also

getLayerSetup

MGMapLayer.LayerType

Description

Gets or sets the layer type.

The older property Type will continue to exist for backward compatibility and will return the types of features on the layer (e.g., Points, Polylines, Polygons, Drawing, and so on).

Type

String (read/write). The following are valid layer types: Unknown, Point, Text, Polyline, Polygon, Raster, Buffer, GIS Design Server Theme, Autodesk DWG, and Redline.

Error Codes

none

See Also

getLayerType

MGMapLayer.LegendLabel

Description

Gets or sets the legend label used by the map layer.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getLegendLabel, setLegendLabel

MGMapLayer.Map

Description

Gets the map containing the map layer.

Type

MGMap (read-only)

Error Codes

none

See Also

getMap

MGMapLayer.MapLayerGroup

Description

Specifies the name of the map layer group that the current layer belongs to.

A value of "null" means the layer is not part of a map layer group.

This property adheres to the setAutoRefresh method except when called from the onViewChanging or onViewChanged events.

Note that for write operations, this property is not available during a busy state. For additional information about the busy state, see "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Type

MGMapLayerGroup (read/write)

Error Codes

-1 (busy): For write operations only.

-3 (illegal argument)

See Also

getMapLayerGroup, setMapLayerGroup

MGMapLayer.MapLayerStyles

Description

Gets a collection containing all of the MGMapLayerStyle objects associated with the layer.

Type

MGCollection (read-only)

Error Codes

none

See Also

getMapLayerStyles, MGMapLayerStyle object

MGMapLayer.MapObjects

Description

Gets a collection containing all the map features (MGMapObject) on the layer.

Note that this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

MGCollection (read-only)

Error Codes

-1 (busy)

See Also

getMapObjectsEx

MGMapLayer.Name

Description

Gets or sets the name of the map layer.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getName, setName

MGMapLayer.Priority

Description

Gets or sets the priority of the map layer.

Priorities range from 0 to 100. Layers with higher priorities are drawn on top of layers with lower priorities.

Type

double (read/write)

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getPriority, setPriority

MGMapLayer.Rebuild

Description

Gets or sets the map layer's rebuild flag.

If the rebuild flag is set, the map layer will be rebuilt from its original data source the next time it is rendered.

If you set the Rebuild property to False, the False value is ignored and the setting remains true. This will be corrected in a future release.

Type

boolean (read/write)

Error Codes

-1 (busy)

See Also

getRebuild, setRebuild

MGMapLayer.Selectability

Description

Gets or sets whether features within the map layer may be selected.

Only dynamic layers are rebuilt when the rebuild flag is set to True; static layers are not rebuilt when you refresh the layer.

Type

boolean (read/write)

Error Codes

-1 (busy)

See Also

getSelectability, setSelectability

MGMapLayer.ShowInLegend

Description

Specifies whether or not to show the map layer in the legend.

Type

boolean (read/write)

Error Codes

-1 (busy)

See Also

getShowInLegend, setShowInLegend

MGMapLayer.SQLWhere

Description

Gets or sets the SQL Where clause for the dynamic layer.

This property is equivalent to the MGDatabaseSetup WhereClause property.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getSQLWhere, setSQLWhere

MGMapLayer.Static

Description

Returns True if the map layer is static.

Type

boolean (read-only)

Error Codes

none

See Also

dynamic, isStatic

MGMapLayer.Type

Description

Gets the type of features the layer contains.

The type can be one of the following:"Point", "Text", "Polyline", "Polygon", "Raster", or "Drawing". Because each layer can contain only one type of feature, each layer is associated with a single type. A new property, LayerType, replaces Type, but Type still works for backward compatibility.

Type

String (read-only)

Error Codes

none

See Also

getType

MGMapLayer.Visibility

Description

Gets or sets whether the map layer is potentially visible.

The Visible property indicates whether the map layer is visible at the current map scale.

Type

boolean (read/write)

Error Codes

-1 (busy)

See Also

getVisibility, setVisibility, Visible, isVisible

MGMapLayer.Visible

Description

Returns True if the map layer is visible at the current map scale.

Type

boolean (read-only)

Error Codes

none

See Also

isVisible

MGMapLayerGroup Object

The MGMapLayerGroup object represents a map layer group.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGMapLayerGroup extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGMapLayerGroup is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

addMapLayer	Adds an existing map layer to the group. A map layer can only belong to one group at any time.
collapse	Collapses the list of map layers under a group so that only the group name heading is visible in the legend.
expand	Expands the list of map layers under a group in the legend so that all potentially visible map layers appear under the group name heading.
getLegendLabel	Gets the string used to label the map layer group in the legend.
getMapLayer	Gets the specified map layer from the group.
getMapLayers	Gets an MGCollection object containing all map layers in the group.
getName	Gets the name of the map layer group.
getShowInLegend	Returns boolean value indicating whether or not the map layer group should be displayed in the legend.
getVisibility	Returns a boolean value indicating if the map layers in this group are potentially visible.
isCollapsed	Determines whether the map layer group is collapsed or expanded.
removeMapLayer	Removes a single map layer from the group.
setLegendLabel	Sets the label used by the map layer group in the legend.
setMapLayers	Clears the contents of a map layer group and adds the specified map layers.
setName	Sets name of the map layer group.

setShowInLegend Sets a flag indicating whether or not the map layer group should be displayed in the legend.

setVisibility Sets a flag that affects the visibility of all map layers in the group.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

Collapsed Gets the state of the map layer group, telling whether it is collapsed or expanded.

LegendLabel Gets or sets the legend label used by the map layer group.

MapLayers Gets or sets an MGCollection containing all the map layers defined in the group.

Name Gets or sets the name of the map layer group.

ShowInLegend Specifies whether or not the map layer group should be shown in the legend.

Visibility Specifies whether or not map layers in the group are potentially visible.

MGMapLayerGroup.addMapLayer

Syntax

```
boolean addMapLayer(MGMapLayer obj)
```

Description

Adds an existing map layer to the group. A map layer can only belong to one group at any time.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

obj - An MGMapLayer object.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

removeMapLayer

MGMapLayerGroup.collapse

Syntax

```
boolean collapse()
```

Description

Collapses the list of map layers under a group so that only the group name heading is visible in the legend.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

See Also

expand, Collapsed

MGMapLayerGroup.expand

Syntax

```
boolean expand()
```

Description

Expands the list of map layers under a group in the legend so that all potentially visible map layers appear under the group name heading.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

See Also

collapse, Collapsed

MGMapLayerGroup.getLegendLabel

Syntax

```
String getLegendLabel()
```

Description

Gets the string used to label the map layer group in the legend.

Returns

String - The name of the map layer group as it appears in the legend.

Error Codes

none

See Also

setLegendLabel, LegendLabel

MGMapLayerGroup.getMapLayer

Syntax

```
MGMapLayer getMapLayer(String layername)
```

Description

Gets the specified map layer from the group.

Parameters

layername - The name of the map layer to return.

Returns

MGMapLayer - The specified map layer, or null if no layer exists with the given name.

Error Codes

-3 (illegal argument)

See Also

getMapLayers, setMapLayers, MapLayers

MGMapLayerGroup.getMapLayers

Syntax

```
MGCollection getMapLayers()
```

Description

Gets an MGCollection object containing all map layers in the group.

Returns

MGCollection- An MGCollection of MGMapLayer objects. Returns an empty MGCollection if there are no layers in the group.

Error Codes

none

See Also

getMapLayers, setMapLayers, MapLayers

MGMapLayerGroup.getName

Syntax

```
String getName()
```

Description

Gets the name of the map layer group.

Returns

String - The map layer group name.

Error Codes

none

See Also

setName, Name

MGMapLayerGroup.getShowInLegend

Syntax

```
boolean getShowInLegend()
```

Description

Returns a boolean value indicating whether or not the map layer group should be displayed in the legend.

Returns

boolean - Returns True if the map layer group should be displayed in the legend; otherwise returns False.

Error Codes

none

See Also

setShowInLegend, ShowInLegend

MGMapLayerGroup.getVisibility

Syntax

```
boolean getVisibility()
```

Description

Returns a boolean value indicating if the map layers in this group are potentially visible.

For a map layer to actually be visible on the map, this property and the layer's visibility property must be True and the current view must be within the visible scale range defined for the group.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

none

See Also

setVisibility, Visibility

MGMapLayerGroup.isCollapsed

Syntax

```
boolean isCollapsed()
```

Description

Determines whether the map layer group is collapsed or expanded.

Use the collapse and expand methods to actually change the state of the layer group.

Returns

boolean - Returns True if collapsed, otherwise, returns False.

Error Codes

none

See Also

collapse, expand

MGMapLayerGroup.removeMapLayer

Syntax

```
boolean removeMapLayer(MGMapLayer obj)
```

Description

Removes a single map layer from the group.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

obj - An MGMapLayer object.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

addMapLayer

MGMapLayerGroup.setLegendLabel

Syntax

```
boolean setLegendLabel(String label)
```

Description

Sets the label used by the map layer group in the legend.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

label - The legend label.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

See Also

getLegendLabel, LegendLabel

MGMapLayerGroup.setMapLayers

Syntax

`boolean setMapLayers(MGCollection objects)`

Description

Clears the contents of a map layer group and adds the specified map layers.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

object - A collection of MGMapLayer objects.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getMapLayers, MapLayers

MGMapLayerGroup.setName

Syntax

`boolean setName(String name)`

Description

Sets name of the map layer group.

Map layer group names must be unique within a map.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

name - The map layer group name.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getName, Name

MGMapLayerGroup.setShowInLegend

Syntax

`boolean setShowInLegend(boolean showInLegend)`

Description

Sets a flag indicating whether or not the map layer group should be displayed in the legend.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

showInLegend - True if the map layer group should be displayed in the legend; otherwise, returns False.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

See Also

getShowInLegend, ShowInLegend

MGMapLayerGroup.setVisibility

Syntax

```
boolean setVisibility(String visi)
```

Description

Sets a flag that affects the visibility of all map layers in the group.

For the map layer to actually be visible on the map, this property and the layer's visibility property must be True and the current view must be within the visible scale range defined for the group.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

visibility - Visibility flag.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

See Also

getVisibility, Visibility

MGMapLayerGroup.Collapsed

Description

Gets the state of the map layer group, telling whether it is collapsed or expanded.

To change the state, use the collapse and expand methods.

Type

boolean (read-only)

Error Codes

none

See Also

collapse, expand

MGMapLayerGroup.LegendLabel

Description

Gets or sets the legend label used by the map layer group.

This property adheres to the setAutoRefresh method except when called from the onViewChanging or onViewChanged events.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getLegendLabel, setLegendLabel

MGMapLayerGroup.MapLayers

Description

Gets or sets an MGCollection containing all the map layers defined in the group.

This property adheres to the setAutoRefresh method except when called from the onViewChanging or onViewChanged events.

Note that for write operations, this property is not available during a busy state. For additional information about the busy state, see "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Type

MGCollection of MGMapLayer objects (read/write)

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getMapLayer, getMapLayers, setMapLayers

MGMapLayerGroup.Name

Description

Gets or sets the name of the map layer group.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getName, setName

MGMapLayerGroup.ShowInLegend

Description

Specifies whether or not the map layer group should be shown in the legend.

This property adheres to the setAutoRefresh method except when called from the onViewChanging or onViewChanged events.

Note that this property is not available during a busy state. For additional information about the busy state, see "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Type

boolean (read/write)

Error Codes

-1 (busy)

See Also

getShowInLegend, setShowInLegend

MGMapLayerGroup.Visibility

Description

Specifies whether or not map layers in the group are potentially visible.

For the map layer to actually be visible on the map, this property and the layer's visibility property must be true, and the current view must be within the visible scale range defined for the group.

This property adheres to the setAutoRefresh method except when called from the onViewChanging or onViewChanged events.

Note that this property is not available during a busy state. For additional information about the busy state, see "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Type

boolean (read/write)

Error Codes

-1 (busy)

See Also

getVisibility, setVisibility

MGMapLayerSetup Object

The MGMapLayerSetup object provides run-time access to map layer setup through methods that let you change data source names and alter the way map features are linked to databases.

You can access all map layer setup attributes of all types of map layers with MGMapLayerSetup. If you attempt to query or modify an attribute that does not apply to the map feature you are working on, the method call fails and sets an error code indicating so in the MGError object.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGMapLayerSetup extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGMapLayerSetup is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getDatabaseSetup	Gets the MGDatabaseSetup object for this layer.
getDataFile	CHANGED Returns the name of the data file from which Autodesk MapGuide Server retrieves data for this layer.
getDwgDataSources	NEW Returns the MGDwgDataSources object for this layer.
getLinkedToDatabase	Indicates whether or not the map features in a spatial data file are linked to a table in a database.
getNameFromSdpFeatureTable	Indicates whether the feature name should come from the SDP feature table or from a secondary table in an OLE DB data source.
getSdpDataSource	Gets the spatial data source name for this map layer.
getSdpFeatureTable	Gets the feature table name for this map layer.
getSdpGeometryColumn	Gets the SDP feature table geometry column name.
getSdpKeyColumn	Gets the SDP feature table key column.
getSdpKeyColumnType	Gets the SDP feature table key column type.
getServerUrl	Gets the Autodesk MapGuide Server URL, including the name of the map agent file.

getSourceType	Gets the source of the map data on this layer.
getURLFromSdpFeatureTable	Indicates whether the feature URL should come from the SDP feature table or from a secondary table in an OLE DB data source.
getWhereFromSdpFeatureTable	Indicates whether the feature SQL Where clause should come from the SDP feature table or from a secondary table in an OLE DB data source.
setDataFile	CHANGED Sets the name of the data file from which Autodesk MapGuide Server retrieves data for this layer.
setLinkedToDatabase	Sets the NameFromSdpFeatureTable, UrlFromSdpFeatureTable, and WhereFromSdpFeatureTable properties to True or false.
setNameFromSdpFeatureTable	Indicates whether the feature name should come from the SDP feature table or from a secondary table in an OLE DB data source.
setSdpDataSource	Sets the Spatial Data Source name for this map layer.
setSdpFeatureTable	Sets the Spatial Data Source name for this map layer.
setSdpGeometryColumn	Sets the name of the SDP feature table geometry column.
setSdpKeyColumn	Sets the name of the SDP feature table key column.
setSdpKeyColumnType	Sets the SDP feature table key column type.
setServerUrl	Sets the Autodesk MapGuide Server URL to the string value that you pass in.
setURLFromSdpFeatureTable	Indicates whether the feature URL should come from the SDP feature table or from a secondary table in an OLE DB data source.
setWhereFromSdpFeatureTable	Indicates whether the feature SQL Where clause should come from the SDP feature table or from a secondary table in an OLE DB data source.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

Properties that set layer setup attributes set the layer's rebuild flag, but do not update the display. To update the display, call the refresh method of MGMap.

DatabaseSetup	Gets the MGDatabaseSetup object for this layer.
DataFile	CHANGED Gets or sets the name of the data file from which Autodesk MapGuide Server retrieves map data for this layer.
LinkedToDatabase	Indicates whether or not the map features in a spatial data file are linked to a table in a database.
MGDwgDataSources	NEW Returns the MGDwgDataSources object for this layer.

NameFromSdpFeatureTable	Indicates whether the feature name should come from the SDP feature table or from a secondary table in an OLE DB data source.
SdpDataSource	Gets or sets the SDP name for this map layer.
SdpFeatureTable	Gets or sets the SDP feature table name for this map layer.
SdpGeometryColumn	Gets or sets the SDP feature table geometry column name.
SdpKeyColumn	Gets or sets the Spatial Data Provider feature table key column name.
SdpKeyColumnType	Gets or sets the SDP feature table key column type.
ServerUrl	Gets or sets the Autodesk MapGuide Server URL.
SourceType	Gets the source of the map data on this layer.
URLFromSdpFeatureTable	Indicates whether the feature URL should come from the SDP feature table or from a secondary table in an OLE DB data source.
WhereFromSdpFeatureTable	Indicates whether the SQL Where clause should come from the SDP feature table or from a secondary table in an OLE DB data source.

MGMapLayerSetup.getDatabaseSetup

Syntax

```
MGDatabaseSetup getDatabaseSetup()
```

Description

Gets the MGDatabaseSetup object for this layer.

This method only applies to layers that are from either an OLE DB data source or an SDP data source.

Returns

MGDatabaseSetup - The MGDatabaseSetup object for this layer.

Error Codes

-15 (does not apply)

See Also

DatabaseSetup

MGMapLayerSetup.getDataFile

Syntax

```
String getDataFile()
```

Description

Gets the name of the data file from which Autodesk MapGuide Server retrieves data for this layer.

The path of the file is relative to the search path set in the Autodesk MapGuide Server Administrator.

This method does not apply to Text and Point map layers from OLE DB data sources, or to Buffer or Redline map layers.

For Text, Point, Polyline and Polygon map layers from Spatial Data Provider (SDP) Data Sources, this method returns the same value as getSdpFeatureTable. We strongly encourage you to use getSdpFeatureTable instead.

For GIS Design Server layers, this method returns the database and theme name, separated by a vertical bar delimiter (for example, configuration name| theme name).

For DWG layers, this methods returns the Autodesk DWG file. This use of this method is retained for backward compatibility and will be deprecated in the next release. Instead, you should use the MGDwgDataSources.getDwg method.

For raster layers, this method returns the raster image file or catalog.

Returns

String - The data file name.

Error Codes

-15 (does not apply)

See Also

setDataFile, DataFile, getSdpFeatureTable, setSdpFeatureTable, SdpFeatureTable, getDwg, setDwg, Dwg

MGMapLayerSetup.getDwgDataSources

Syntax

```
MGDwgDataSources getDwgDataSources()
```

Description

Gets the MGDwgDataSources object for this layer.

Parameters

none

Returns

MGDwgDataSources - The MGDwgDataSources object for this layer.

Error Codes

-15 (does not apply)

See Also

MGDwgDataSources

MGMapLayerSetup.getLinkedToDatabase

Syntax

```
boolean getLinkedToDatabase()
```

Description

Indicates whether or not the map features in a spatial data file are linked to a table in a database.

This method is retained only for backward compatibility with Autodesk MapGuide 4.*x*; it will be deprecated in the next release. Developers are encouraged to use getNameFromSdpFeatureTable, getUrlFromSdpFeatureTable, and getWhereFromSdpFeatureTable instead.

Returns

boolean - Returns True if NameFromSdpFeatureTable, UrlFromSdpFeatureTable, or WhereFromSdpFeatureTable is False. Returns False if all are True, or if the method does not apply.

Error Codes

-15 (does not apply)

See Also

setLinkedToDatabase, LinkedToDatabase, getNameFromSdpFeatureTable, setNameFromSdpFeatureTable, NameFromSdpFeatureTable, getUrlFromSdpFeatureTable, setUrlFromSdpFeatureTable, UrlFromSdpFeatureTable, getWhereFromSdpFeatureTable, setWhereFromSdpFeatureTable, WhereFromSdpFeatureTable

MGMapLayerSetup.getNameFromSdpFeatureTable

Syntax

```
boolean getNameFromSdpFeatureTable()
```

Description

Indicates whether the feature name should come from the SDP feature table or from a secondary table in an OLE DB data source.

This is useful because getNameColumn, setNameColumn, and NameColumn specify the name of the feature name column without specifying its source.

Returns

boolean - Returns True if the name is from the SDP feature table. Returns False if the name comes from a secondary table in an OLE DB data source, or if the method does not apply.

Error Codes

-15 (does not apply)

See Also

setNameFromSdpFeatureTable, NameFromSdpFeatureTable

MGMapLayerSetup.getSdpDataSource

Syntax

```
String getSdpDataSource()
```

Description

Gets the spatial data source name for this map layer.

Returns

String - A Microsoft UDL file name representing the name of the spatial data source.

Error Codes

-15 (does not apply)

See Also

setSdpDataSource, SdpDataSource

MGMapLayerSetup.getSdpFeatureTable

Syntax

```
String getSdpFeatureTable()
```

Description

Gets the feature table name for this map layer.

Returns

String - The name of the feature table.

Error Codes

-15 (does not apply)

See Also

setSdpFeatureTable, SdpFeatureTable

MGMapLayerSetup.getSdpGeometryColumn

Syntax

```
String getSdpGeometryColumn()
```

Description

Gets the SDP feature table geometry column name.

Returns

String - The geometry column name, or an empty string if this method does not apply.

Error Codes

-15 (does not apply)

See Also

setSdpGeometryColumn, SdpGeometryColumn

MGMapLayerSetup.getSdpKeyColumn

Syntax

```
String getSdpKeyColumn()
```

Description

Gets the SDP feature table key column.

Returns

String - The name of the key column.

Error Codes

-15 (does not apply)

See Also

setSdpKeyColumn, SdpKeyColumn

MGMapLayerSetup.getSdpKeyColumnType

Syntax

```
String getSdpKeyColumnType()
```

Description

Gets the SDP feature table key column type.

Returns

String - The key column data type. Values can be: Boolean, Byte, Date, Decimal, Double, Float, Integer, Numeric, SmallInt, String, Time, TimeStamp, UnsignedByte, UnsignedInteger, and UnsignedSmallInt.

We recommend that you do not use Float, Double, and Real types because of the inherent inaccuracies in comparing them.

Note that the Autodesk MapGuide Viewer API uses string values while the SDPFeatureDataSource KeyColumnType element in Map Window XML (MWX) files uses enumerated constants. For more information about MWX, refer to the *Autodesk MapGuide Dynamic Authoring Toolkit Developer's Guide*.

Error Codes

-15 (does not apply)

See Also

setSdpKeyColumnType, SdpKeyColumnType

MGMapLayerSetup.getServerUrl

Syntax

```
String getServerUrl()
```

Description

Gets the Autodesk MapGuide Server URL, including the name of the MapAgent file.

Any requests for map data go to Autodesk MapGuide Server at this URL.

Returns

String - The the Autodesk MapGuide Server URL.

Error Codes

none

See Also

setServerUrl, ServerUrl

MGMapLayerSetup.getSourceType

Syntax

```
String getSourceType()
```

Description

Gets the source of the map data on this layer.

For backward compatibility with release 4.*x*, this method will continue to return Database for map layers from OLE DB data sources and SpatialDataFile for map layers from SDP data sources.

Returns

String - The source of the map data, either Database, DWG, RasterImageFile, SpatialDataFile, or VisionTheme. A DoesNotApply error is set for buffer and redlining layers.

Error Codes

-15 (does not apply)

See Also

SourceType

MGMapLayerSetup.getURLFromSdpFeatureTable

Syntax

```
boolean getUrlFromSdpFeatureTable()
```

Description

Indicates whether the feature URL should come from the SDP feature table or from a secondary table in an OLE DB data source.

This is useful because getUrlLinkColumn, setUrlLinkColumn, and UrlLinkColumn specify the name of the URL column without specifying its source.

Returns

boolean - Returns True if the URL is from the SDP feature table. Returns False if the URL comes from a secondary table in an OLE DB data source, or if the method does not apply.

Error Codes

-15 (does not apply)

See Also

setUrlFromSdpFeatureTable, UrlFromSdpFeatureTable

MGMapLayerSetup.getWhereFromSdpFeatureTable

Syntax

```
boolean getWhereFromSdpFeatureTable()
```

Description

Indicates whether the feature SQL Where clause should come from the SDP feature table or from a secondary table in an OLE DB data source.

This is useful because getWhereClause, setWhereClause, and WhereClause specify the SQL Where clause without specifying its source.

This method does not apply to Text and Point map layers from OLE DB data sources, or to Buffer or Redline map layers.

For Text, Point, Polyline, and Polygon map layers from SDP data sources, getWhereFromSdpFeatureTable returns the same value as setSdpFeatureTable; developers are strongly encouraged to use that method instead. For GIS Design Server layers, getWhereFromSdpFeatureTable returns the database and theme name, separated by a delimiter.

Parameters

dataFile - The path and name of the data file, relative to the search path set in Autodesk MapGuide Server Admin.

Returns

boolean - Returns True if successful. Returns False if busy, or if the method does not apply.

Error Codes

-15 (does not apply)

See Also

getDataFile, DataFile, getSdpFeatureTable, setSdpFeatureTable

MGMapLayerSetup.setDataFile

Syntax

```
boolean setDataFile(String dataFile)
```

Description

Sets the name of the data file from which Autodesk MapGuide Server retrieves data for this layer.

If this method is successful, it automatically sets the layer's rebuild flag to True, causing the application to rebuild the map. setDataFile always rebuilds the map, even if you call it twice in a row with the same file name. To avoid this, you can check to see what the current data file is by calling getDataFile. If the current data file is the same as the data file that you wanted to set, you can choose not to call setDataFile in your script, or you can set the rebuild flag to false.

This method does not apply to Text and Point map layers from OLE DB Data Sources,or to Buffer or Redline map layers.

For Text, Point, Polyline and Polygon map layers from Spatial Data Provider (SDP) Data Sources this method is the same as setSdpFeatureTable. We strongly encourage you to use setSdpFeatureTable instead.

For GIS Design Server layers, this will set the database and theme name, separated by a vertical bar delimiter (for example, configuration name| theme name).

For DWG layers, this will set the Autodesk DWG file. This use of this method is retained for backward compatibility and will be deprecated in the next release. Instead, you should use the MGDwgDataSources.setDwg method.

Parameters

dataFile - The path and name of the data file. The path of the file is relative to the search path set in Autodesk MapGuide Server Administrator. In this case, this is not the ADL.

Returns

boolean - Returns True if successful. Returns False if busy state or does not apply.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getDataFile, DataFile, getSdpFeatureTable, setSdpFeatureTable, SdpFeatureTable, getDwg, setDwg, Dwg

MGMapLayerSetup.setLinkedToDatabase

Syntax

`boolean setLinkedToDatabase(boolean linkedToDatabase)`

Description

Sets the NameFromSdpFeatureTable, UrlFromSdpFeatureTable, and WhereFromSdpFeatureTable properties to True or false.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

This method is retained only for backward compatibility with Autodesk MapGuide 4.*x*; it will be deprecated in the next release. Developers are encouraged to use setNameFromSdpFeatureTable, setUrlFromSdpFeatureTable, and setWhereFromSdpFeatureTable instead.

Parameters

linkedToDatabase - The boolean value, either True or False, that you want to set the LinkedToDatabase flag to.

Returns

boolean - Returns True if successful. Returns False if not successful, or does not apply.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getLinkedToDatabase, LinkedToDatabase, getNameFromSdpFeatureTable, getUrlFromSdpFeatureTable, getWhereFromSdpFeatureTable,

MGMapLayerSetup.setNameFromSdpFeatureTable

Syntax

`boolean setNameFromSdpFeatureTable(boolean fromFeatureTable)`

Description

Indicates whether the feature name should come from the SDP feature table or from a secondary table in an OLE DB data source.

This is useful because getNameColumn, setNameColumn, and NameColumn specify the name of feature name column without specifying its source.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Parameters

fromFeatureTable - A boolean value, either True or False.

Returns

boolean - Returns True if successful. Returns False if not successful, or does not apply.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getNameFromSdpFeatureTable, NameFromSdpFeatureTable

MGMapLayerSetup.setSdpDataSource

Syntax

`boolean setSdpDataSource(String dataSource)`

Description

Sets the Spatial Data Source name for this map layer.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

You might need to update other SDP-related properties after changing the SDP data source. What those properties are varies depending on your data.

Parameters

dataSource - A Microsoft UDL file name representing the name of the spatial data source. This name might or might not include the *.udl* file extension.

Returns

boolean - Returns True if successful.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getSdpDataSource, SdpDataSource

MGMapLayerSetup.setSdpFeatureTable

Syntax

`boolean setSdpFeatureTable(String featureTable)`

Description

Sets the feature table name for this map layer.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

You might need to update other SDP-related properties after changing the feature table. What those properties are varies depending on your data.

Parameters

featureTable - The name of the feature table.

Returns

boolean - Returns True if successful.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getSdpFeatureTable, SdpFeatureTable

MGMapLayerSetup.setSdpGeometryColumn

Syntax

`boolean setSdpGeometryColumn(String column)`

Description

Sets the name of the SDP feature table geometry column.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

column - The name you want the column set to.

Returns

boolean - Returns True if successful.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getSdpGeometryColumn, getSdpGeometryColumn

MGMapLayerSetup.setSdpKeyColumn

Syntax

```
boolean setSdpKeyColumn(String column)
```

Description

Sets the name of the SDP feature table key column.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

column - The name you want the column set to.

Returns

boolean - Returns True if successful.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getSdpKeyColumn, SdpKeyColumn

MGMapLayerSetup.setSdpKeyColumnType

Syntax

```
boolean setSdpKeyColumnType(String keyColumnType)
```

Description

Sets the SDP feature table key column type.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

We recommend that you do not use Float, Double, and Real types because of the inherent inaccuracies in comparing them.

Parameters

keyColumnType - The key column data type, which will be one of the following: String, Numeric, Decimal, Integer, SmallInt, Float, Double, Date, Time, TimeStamp, Boolean, UnsignedInteger, or UnsignedSmallInt.

Returns

boolean - Returns True if successful.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getSdpKeyColumnType, SdpKeyColumnType

MGMapLayerSetup.setServerUrl

Syntax

```
void setServerUrl(String serverUrl)
```

Description

Sets the Autodesk MapGuide Server URL to the string value that you pass in.

Any requests for map data go to Autodesk MapGuide Server at this URL. Before coding your application to change the server URL, be sure to check the new server's SDF files, data sources, and the various database column settings of the database setup, including the Theme settings if applicable (note that you cannot modify Theme settings with the API).

Parameters

serverUrl - The URL of Autodesk MapGuide Server containing data for this layer.

Error Codes

-1 (busy)

See Also

getServerUrl, ServerUrl

MGMapLayerSetup.setURLFromSdpFeatureTable

Syntax

```
boolean setUrlFromSdpFeatureTable(boolean fromFeatureTable)
```

Description

Indicates whether the feature URL should come from the SDP feature table or from a secondary table in an OLE DB data source.

This is useful because getUrlLinkColumn, setUrlLinkColumn, and UrlLinkColumn specify the name of the URL column without specifying its source.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

fromFeatureTable - A boolean value, either True or False.

Returns

boolean - Returns True if successful. Returns False if not successful, or does not apply.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getUrlFromSdpFeatureTable, UrlFromSdpFeatureTable

MGMapLayerSetup.setWhereFromSdpFeatureTable

Syntax

```
boolean setWhereFromSdpFeatureTable(boolean fromFeatureTable)
```

Description

Indicates whether the feature SQL Where clause should come from the SDP feature table or from a secondary table in an OLE DB data source.

This is useful because getWhereClause, setWhereClause, and WhereClause specify the SQL Where clause without specifying its source.

If this method is successful, it automatically sets the layer's rebuild flag to True. See setRebuild for more information.

Note that this method does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

fromFeatureTable - A boolean value, either True or False.

Returns

boolean - Returns True if successful. Returns False not successful, or does not apply.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getWhereFromSdpFeatureTable, WhereFromSdpFeatureTable

MGMapLayerSetup.DatabaseSetup

Description

Gets the MGDatabaseSetup object for this layer.

Type

MGDatabaseSetup (read-only)

Error Codes

-15 (does not apply)

See Also

getDatabaseSetup

MGMapLayerSetup.DataFile

Description

Gets or sets the name of the data file from which Autodesk MapGuide Server retrieves map data for this layer.

Returns an empty string if this property does not apply.

If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

This property does not apply to Text and Point map layers from OLE DB data sources, or to Buffer or Redline map layers.

For Text, Point, Polyline and Polygon map layers from Spatial Data Provider (SDP) data sources, this property is the same as the SdpFeatureTable property. We strongly encourage you to use SdpFeatureTable instead.

For GIS Design Server layers, this will get or set the database and theme name, separated by a vertical bar delimiter (for example, configuration name| theme name).

For DWG layers, this property will get or set the Autodesk DWG file. This use of this property is retained for backward compatibility and will be deprecated in the next release. Instead, you should use the MGDwgDataSources.Dwg property.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getDataFile, setDataFile, getSdpFeatureTable, setSdpFeatureTable, SdpFeatureTable, getDwg, setDwg, Dwg

MGMapLayerSetup.LinkedToDatabase

Description

Indicates whether or not the map features in a spatial data file are linked to a table in a database.

If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

This property is retained only for backward compatibility with Autodesk MapGuide 4.*x*; it will be deprecated in the next release. Developers are encouraged to use NameFromSdpFeatureTable, UrlFromSdpFeatureTable, and WhereFromSdpFeatureTable instead.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getLinkedToDatabase, setLinkedToDatabase

MGMapLayerSetup.MGDwgDataSources

Description

Gets the MGDwgDataSources object for this layer.

Types

MGDwgDataSources (read-only)

Error Codes

-15 (does not apply)

See Also

getDataSource

MGMapLayerSetup.NameFromSdpFeatureTable

Description

Indicates whether the feature name should come from the SDP feature table or from a secondary table in an OLE DB data source. This is useful because getNameColumn, setNameColumn, and NameColumn specify the name of the feature name column without specifying its source.

If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getNameFromSdpFeatureTable, setNameFromSdpFeatureTable, NameColumn

MGMapLayerSetup.SdpDataSource

Description

Gets or sets the SDP name for this map layer.

This is a Microsoft UDL file name which might or might not include the *.udl* file extension.

If this method is successful, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

You might need to update other SDP-related properties after changing the SDP data source. What those properties are varies depending on your data.

Type

String (read/write)

Error Codes

-15 (does not apply)

See Also

getSdpDataSource, setSdpDataSource

MGMapLayerSetup.SdpFeatureTable

Description

Gets or sets the SDP feature table name for this map layer.

If this property is successful, it automatically sets the layer's rebuild flag to True. See Rebuild for more information.

You might need to update other SDP-related properties after changing the SDP data source. What those properties are varies depending on your data.

Type

String (read/write)

Error Codes

-15 (does not apply)

See Also

getSdpFeatureTable, setSdpFeaturetable

MGMapLayerSetup.SdpGeometryColumn

Description

Gets or sets the SDP feature table geometry column name.

Returns an empty string if the property does not apply. If SdpGeometryColumn is modified successfully, it automatically sets the layer's rebuild flag to true.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getSdpGeometryColumn, setSdpGeometryColumn

MGMapLayerSetup.SdpKeyColumn

Description

Gets or sets the Spatial Data Provider feature table key column name.

If SdpKeyColumn is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getSdpKeyColumn, setSdpKeyColumn

MGMapLayerSetup.SdpKeyColumnType

Description

Gets or sets the SDP feature table key column type.

If this property is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for more information.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Values can be: Boolean, Byte, Date, Decimal, Double, Float, Integer, Numeric, SmallInt, String, Time, TimeStamp, UnsignedByte, UnsignedInteger, and UnsignedSmallInt.

We recommend that you do not use Float, Double, and Real types because of the inherent inaccuracies in comparing them.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-15 (does not apply)

See Also

getSdpKeyColumnType, setSdpKeyColumnType

MGMapLayerSetup.ServerUrl

Description

Gets or sets the Autodesk MapGuide Server URL.

Requests for map data go to Autodesk MapGuide Server at this URL. If this property is modified successfully, it automatically sets the layer's rebuild flag to True. Before coding your application to change the server URL, be sure to check the new server's SDF files, data sources, and the various database column settings, including the Theme settings if applicable (note that you cannot modify Theme settings with the API).

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getServerUrl, setServerUrl

MGMapLayerSetup.SourceType

Description

Gets the source of the map data on this layer.

The return value is "Database", "DWG", "RasterImageFile", "SpatialDataFile", or "VisionTheme".

For backwards compatibility with release 4.*x*, this method will continue to return "Database" for map layers from OLE DB data sources and "SpatialDataFile" for map layers from SDP data sources.

Type

String (read-only)

Error Codes

-15 (does not apply)

See Also

getSourceType

MGMapLayerSetup.URLFromSdpFeatureTable

Description

Indicates whether the feature URL should come from the SDP feature table or from a secondary table in an OLE DB data source.

This is useful because getUrlLinkColumn, setUrlLinkColumn, and UrlLinkColumn specify the name of the URL column without specifying its source.

If UrlFromSdpFeatureTable is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that this property is not available during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Type

boolean (read/write)

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getUrlFromSdpFeatureTable, setUrlFromSdpFeatureTable

MGMapLayerSetup.WhereFromSdpFeatureTable

Description

Indicates whether the SQL Where clause should come from the SDP feature table or from a secondary table in an OLE DB data source.

This is useful because getWhereClause, setWhereClause, and WhereClause specify the SQL Where clause without specifying its source.

If WhereFromSdpFeatureTable is modified successfully, it automatically sets the layer's rebuild flag to True. See Rebuild for details.

Note that this property is not available during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Type

boolean (read/write)

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getWhereFromSdpFeatureTable, setWhereFromSdpFeatureTable, WhereClause in MGDatabaseSetup

MGMapLayerStyle Object

The MGMapLayerStyle object lets you access a map layer style's display scale range.

In future releases, this object will be extended to provide access to the theme properties, display attribute styles, and theme categories associated with a map layer style.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGMapLayerStyle extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGMapLayerStyle is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getMaxDisplayRange	Gets the maximum value of the display range scale.
getMinDisplayRange	Gets the minimum value of the display range scale.
setDisplayRange	Sets the display scale range associated with a style for a particular map layer.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

MaxDisplayRange	Gets the maximum value of the display range scale.
MinDisplayRange	Gets the minimum value of the display range scale.

MGMapLayerStyle.getMaxDisplayRange

Syntax

```
double getMaxDisplayRange()
```

Description

Gets the maximum value of the display range scale.

Returns

double - The maximum display range scale value.

Error Codes

none

See Also

getMinDisplayRange, setDisplayRange, MinDisplayRange

MGMapLayerStyle.getMinDisplayRange

Syntax

```
double getMinDisplayRange()
```

Description

Gets the minimum value of the display range scale.

Returns

double - The minimum display range scale value.

Error Codes

none

See Also

getMaxDisplayRange, setDisplayRange, MaxDisplayRange

MGMapLayerStyle.setDisplayRange

Syntax

boolean setDisplayRange(double *maxScale*, double *minScale*)

Description

Sets the display scale range associated with a style for a particular map layer.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

maxScale - The maximum display range scale value. *minScale* -

The minimum display range scale value.

Returns

boolean - Returns True if successful and sets the layer's rebuild flag to True. Returns False if the minimum value is greater then the maximum value.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getMaxDisplayRange, getMinDisplayRange, MaxDisplayRange, MinDisplayRange

MGMapLayerStyle.MaxDisplayRange

Description

Gets the maximum value of the display range scale.

Type

double (read-only)

Error Codes

none

See Also

getMaxDisplayRange

MGMapLayerStyle.MinDisplayRange

Description

Gets the minimum value of the display range scale.

Type

double (read-only)

Error Codes

none

See Also

getMinDisplayRange

MGMapObject Object

The MGMapObject object represents a feature on the map, such as a point, polyline, polygon, or text.

Note that instances of the class MGMapObject cannot be publicly created.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGMapObject extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGMapObject is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

addCirclePrimitive	Adds a polygon primitive, composed of individual line segments representing a circle, to the current map feature.
addPolygonPrimitive	Adds a polygon or poly-polygon primitive to the current map feature.
addPolylinePrimitive	Adds a polyline or poly-polyline primitive to the current map feature.
addSymbolPrimitive	Adds a point or predefined symbol primitive to the current map feature.
addTextPrimitive	Adds a text primitive to the current map feature.
equals	Compares this MGMapObject object to the specified object.
getArea	Gets the area of the specified polygon map feature.
getExtent	Gets the extent of the map feature, that is, the smallest rectangle that can be drawn around all the map feature's elements.
getExtentEx	Gets the extent of the map feature, that is, the smallest rectangle that can be drawn around all the map feature's elements.
getKey	Gets the key of the map feature.
getLength	Gets the length of the specified polyline map feature.
getLink	Gets the URL link associated with the map feature.
getMapLayer	Gets the map layer that contains the map feature.
getName	Gets the name of the map feature.
getPrimitives	Gets a collection of primitives describing the map feature.
getType	Gets the type of the map feature.
getVertices	Gets the vertices used to describe the geometry of the map feature.
setLink	Sets the URL link associated with the map feature.

setName	Sets the name of the map feature.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

Area	The area of the specified polygon in Mapping Coordinate System (MCS) units.
DoubleClickHandled	Indicates whether the onDoubleClickObject event has been handled for this object.
Extent	The extent of the map feature.
Key	The key of the map feature.
Length	The length of the specified polyline in Mapping Coordinate System (MCS) units.
Link	Gets or sets the URL link associated with the map feature.
MapLayer	The map layer that contains the map feature.
Name	The name of the map feature.
Primitives	Gets a collection of primitives describing the map feature.
Type	The type of the map feature.

MGMapObject.addCirclePrimitive

Syntax

`boolean addCirclePrimitive(MGPoint center, boolean`*`useMCS`*`, double`*`radius`*`, String`*`units`*`, integer`*`vertices`*`)`

Description

Adds a polygon primitive, composed of individual line segments representing a circle, to the current map feature.

The primitive is positioned so that its northern-most and southern-most points are the vertices (that is, if a four point circle were created, it would be positioned like a baseball diamond). The rotation cannot be altered.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events. Note that this method does not work during a busy state.

Parameters

center - An MGPoint object representing the center of the primitive, in MCS units.

useMCS - A boolean specifying whether the vertex coordinates are in MCS or lat/lon.

radius - The radius in specified units.

units - The units used by *radius* (values can be M, KM, FT, MI)

vertices - The number of vertices used to polygonize the circle. You must specify at least four vertices (a number of 200 or greater is recommended). All numbers are rounded up to the nearest multiple of four.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-15 (does not apply -- not a redline layer)

MGMapObject.addPolygonPrimitive

Syntax

```
boolean addPolygonPrimitive(MGCollection vertices,
MGCollection numberOfVertices,
boolean useMCS)
```

Description

Adds a polygon or poly-polygon primitive to the current map feature.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events. Note that this method does not work during a busy state.

Parameters

vertices - a collection of MGPoint objects, each representing a vertex in MCS units. *numberOfVertices* - An MGCollection object. For a polygon, the collection contains a single integer representing the number of vertices. For a poly-polygon, the collection contains integer objects, each representing the number of vertices contained a component polygon. *useMCS* - a boolean specifying whether the vertex coordinates are in MCS or lat/lon.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-15 (does not apply -- not a redline layer)

MGMapObject.addPolylinePrimitive

Syntax

```
boolean addPolylinePrimitive(MGCollection vertices,
 MGCollection numberOfVertices,
 boolean useMCS)
```

Description

Adds a polyline or poly-polyline primitive to the current map feature.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events. Note that this method does not work during a busy state.

Parameters

vertices - a collection of MGPoint objects, each representing a vertex in MCS units

numberOfVertices - An MGCollection object; for a polyline, the collection contains a single integer representing the number of vertices; for a poly-polyline, the collection contains integer objects, each representing the number of vertices contained a component polyline.

useMCS - a boolean specifying whether the vertex coordinates are in MCS or lat/lon.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-15 (does not apply -- not a redline layer)

MGMapObject.addSymbolPrimitive

Syntax

```
boolean addSymbolPrimitive(MGPoint vertex, booleanuseMCS)
```

Description

Adds a point or predefined symbol primitive to the current map feature.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events. Note that this method does not work during a busy state.

Parameters

vertex - An MGPoint object representing the symbol insertion point in MCS units.

useMCS - a boolean specifying whether the insertion coordinates parameter are in MCS or lat/lon.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-15 (does not apply -- not a redline layer)

MGMapObject.addTextPrimitive

Syntax

```
boolean addTextPrimitive(MGPoint vertex, booleanuseMCS, String textValue)
```

Description

Adds a text primitive to the current map feature.

This method always refreshes the map when the autoRefresh flag is set to True or when called from the onViewChanging or onViewChanged events. Note that this method does not work during a busy state.

Parameters

vertex - An MGPoint object representing the text insertion point in MCS units. *useMCS* - A boolean specifying whether the insertion coordinates are in MCS or lat/lon.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-15 (does not apply -- not a redline layer)

MGMapObject.equals

Syntax

```
boolean equals(Object obj)
```

Description

Compares this MGMapObject object to the specified object.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

obj - The object to compare to this MGMapObject.

Returns

boolean - Returns True if the two objects are identical; otherwise, returns False.

Error Codes

-3 (illegal argument)

MGMapObject.getArea

Syntax

```
double getArea()
```

Description

Gets the area of the specified polygon map feature.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.. Also note that the area may not be calculated correctly for overlapping polypolygons.

Returns

double - The area of the specified polygon in Mapping Coordinate System (MCS) units. If the specified map feature is not a polygon, returns 0.

Error Codes

-1 (busy)

See Also

getLength, getMCSScaleFactor, Area

MGMapObject.getExtent

Syntax

```
MGExtent getExtent()
```

Description

Gets the extent of the map feature, that is, the smallest rectangle that can be drawn around all the map feature's elements.

Note that for point features, this method does not return the coordinates of the point; it returns the four points representing the extent of the *symbol* that is displayed for the point. To get the coordinates of the point, not the symbol, use getVertices.

This method always returns the extent in latitude/longitude coordinates. To return the extent in the map's coordinate system, use getExtentEx instead.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

MGExtent - The feature's extent.

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It takes a feature name as its argument. If the feature is found, its center point is displayed:

```
function showCenterPt(key)
{
    var map = getMap()
    var layer = map.getMapLayer("US States");
    if (layer.isVisible())
    {
        // Get feature, using key that was passed
        // to function; then get extents of that feature
        var obj = layer.getMapObject(key);
        var ext = obj.getExtent();
        // Get average latitude and average longitude;
        // then display alert showing center point
        var avLat = ext.getMinLat() + ext.getMaxLat() / 2;
        var avLon = ext.getMaxLon() +  ext.getMaxLon() / 2
        alert (key + " center point: " + avLat + ", " + avLon);
    }
}
```

You might use the following HTML to call the function on a feature whose key is "CA":

```
<FORM>
<INPUT TYPE="button" VALUE="Toggle" ONCLICK="showCenterPt('CA');">
</FORM>
```

See Also

getExtentEx, Extent

MGMapObject.getExtentEx

Syntax

```
MGExtent getExtentEx(boolean useMcs)
```

Description

Gets the extent of the map feature, that is, the smallest rectangle that can be drawn around all the map feature's elements.

Note that for point features, this method does not return the coordinates of the point; it returns the four points representing the extent of the *symbol* that is displayed for the point. To get the coordinates of the point, not the symbol, use getVertices.

This method is identical to getExtent, except that you specify whether to return the extent in the mapping coordinate system (MCS) or latitude/longitude, whereas getExtent always returns in latitude/longitude.

Parameters

useMcs - A boolean specifying whether the coordinates will be returned in MCS or lat/lon.

Returns

MGExtent - The feature's extent.

Error Codes

-1 (busy)

See Also

getExtent, Extent

MGMapObject.getKey

Syntax

```
String getKey()
```

Description

Gets the key of the map feature.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

String - The map feature's key.

Error Codes

-1 (busy)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the keys for all map features on a layer called "World Countries":

```
function getTheKeys()
{
    // Get map, layer, and features
    var features = getMap().getMapLayer('World Countries').getMapObjectsEx();
    // Cycle through features, getting key for each member
    // and adding it to msg
    var msg = "";
    for (i = 0; i < features.size(); i++)
```

```
        msg += features.item(i).getKey() + " "
    // Display the keys
    alert (msg);
}
```

See Also

Key

MGMapObject.getLength

Syntax

```
double getLength()
```

Description

Gets the length of the specified polyline map feature.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

double - The length of the specified polyline in Mapping Coordinate System (MCS) units. If the polyline is composed of more than one polyline internally, the sum of all the distances of the polylines will be returned. If the specified map feature is not a polyline, returns 0.

Error Codes

-1 (busy)

See Also

getArea, getMCSScaleFactor, Length

MGMapObject.getLink

Syntax

```
String getLink()
```

Description

Gets the URL link associated with the map feature.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

String - The map feature's linked URL.

Error Codes

-1 (busy)

See Also

setLink, Link

MGMapObject.getMapLayer

Syntax

```
MGMapLayer getMapLayer()
```

Description

Gets the map layer that contains the map feature.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

MGMapLayer - The layer that contains the map feature.

Error Codes

-1 (busy)

See Also

MapLayer

MGMapObject.getName

Syntax

```
String getName()
```

Description

Gets the name of the map feature.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

String - The map feature's name.

Error Codes

-1 (busy)

See Also

setName, Name

MGMapObject.getPrimitives

Syntax

```
MGCollection getPrimitives()
```

Description

Gets a collection of primitives describing the map feature.

If the map feature consists of a single primitive, the collection will have only one item.

Returns

MGCollection - A collection of MGPrimitives describing the map feature.

Error Codes

none

MGMapObject.getType

Syntax

```
String getType()
```

Description

Gets the type of the map feature.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

String - The map feature's type, which will be "Unknown", "Point", "Text", "Polyline", or "Polygon".

Error Codes

-1 (busy)

See Also

Type, getType in MGMapLayer

MGMapObject.getVertices

Syntax

int getVertices(MGCollection *vertices*, MGCollection *numberOfVertices*)

Description

Gets the vertices used to describe the geometry of the map feature.

This method works only if the author of the map enabled access by selecting the *Allow Access To Coordinate Values From API* check box or the *Allow Access To Coordinate Values From API with the Following Passkey* check box in Autodesk MapGuide Author. If the map author selected the *Allow Access To Coordinate Values From API with the Following Passkey* check box, then you must first call the unLock method with the correct passkey to enable getVertices.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

vertices - An MGCollection of MGPoint objects. The coordinates of the vertices are in Mapping Coordinate System (MCS) units.

numberOfVertices - For simple map features (text, point, polyline, and polygon), the collection contains a single Integer that represents the number of vertices in the collection *vertices*. For complex poly-polyline and poly-polygons, the collection of Integer objects represents the number of vertices contained in each component (polyline or polygon) of the complex map feature (poly-polyline or poly-polygon).

Returns

int - The number of internal objects, which is the same as the length of the collection returned by *numberOfVertices*.

Returns one of the following:

1 The map feature is a polyline, polygon, point, or text.

>1 The map feature is a poly-polyline or poly-polygon.

0 The map does not allow access to its coordinates from the API, or the Autodesk MapGuide Viewer is Release 4.0 or later and the map was authored with Autodesk MapGuide Author Release 3.0 or earlier.

Error Codes

-1 (busy)

-5 (security violation)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the type and, if applicable, number of vertices of a selected feature.

```
function getVertexInfo()
{
    var map = getMap();
    // Get selected features
    var selected = map.getSelection().getMapObjectsEx(null);
    // Make sure just one is selected
    if (selected.size() != 1)
        alert ("Please select one, and only one, feature.");
    else
    {
        // Create collections to hold vertex info
        verts = map.createObject('MGCollection');
        vertNum = map.createObject('MGCollection');
        // Get vertices of first (and only) item in selected collection
        result = selected.item(0).getVertices(verts, vertNum);
        // Display results
```

```
        if (result == 0)
            alert ("Unable to access feature's coordinate information");
        if (result == 1)
            alert ("This is a polyline, polygon, point, or text feature.");
        else
            alert ("This is a poly-polyline or poly-polygon. It has " + result + "
vertices.");
    }
}
```

MGMapObject.setLink

Syntax

`void setLink(String link)`

Description

Sets the URL link associated with the map feature.

This method should be used on static layers only, because a rebuild of the layer will overwrite the new link with the original data from the server.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*..

Parameters

link - The map feature's new URL. If the URL is anything other than a complete, valid URL, the browser handles the URL in the following ways:

- If the URL does not contain a valid protocol identifier such as http://, it is treated as a relative URL and is appended to the end of the URL of the page containing the map.
- If the URL is complete but invalid, the browser will attempt to load that URL without success.
- If the URL is an empty string, the link is deactivated.

Error Codes

-1 (busy)

See Also

getLink, Link

MGMapObject.setName

Syntax

`void setName(String name)`

Description

Sets the name of the map feature.

This method should be used on static layers only, because a rebuild of the layer will overwrite the new name with the original data from the server.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

name - The map feature's new name.

Error Codes

-1 (busy)

See Also

getName, Name

MGMapObject.Area

Description

Gets the area of the specified polygon in Mapping Coordinate System (MCS) units.

If the specified map feature is not a polygon, this value will be 0.

Note that this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

double (read-only)

Error Codes

-1 (busy)

See Also

getArea, Length

MGMapObject.DoubleClickHandled

Description

Indicates whether the onDoubleClickObject event has been handled for this object.

This property may also be set to True in the onDoubleClickObject event to inform Autodesk MapGuide not to perform the default double-click action, which displays the URL link associated with the feature. By default, this property is False, which causes the default double-click action to be executed.

Type

boolean (read-only)

Error Codes

none

See Also

onDoubleClickObject

MGMapObject.Extent

Description

Gets the extent of the map feature.

Note that for point features, this property does not contain the coordinates of the point itself; it contains the coordinates of the four points representing the extent of the *symbol* that is displayed for the point. To get the coordinates of the point, not the symbol, use the getVertices method.

Note that this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

MGExtent (read-only)

Error Codes

-1 (busy)

See Also

getExtent, getExtentEx

MGMapObject.Key

Description

Gets the key of the map feature.

Note that this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read-only)

Error Codes

-1 (busy)

See Also

getKey

MGMapObject.Length

Description

Gets the length of the specified polyline in Mapping Coordinate System (MCS) units.

If the specified polyline is composed of more than one polyline internally, the Length value will be the sum of all the distances of the polylines. If the specified map feature is not a polyline, the value will be 0.

Type

double (read-only)

Error Codes

none

See Also

getLength, Area

MGMapObject.Link

Description

Gets or sets the URL link associated with the map feature.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

See Also

getLink, setLink

MGMapObject.MapLayer

Description

Gets the map layer that contains the map feature.

Note that this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

MGMapLayer (read-only)

Error Codes

-1 (busy)

See Also

getMapLayer

MGMapObject.Name

Description

Gets or sets the name of the map feature.

Type

String (read/write)

See Also

getName, setName

MGMapObject.Primitives

Description

Gets a collection of primitives describing the map feature.

If the map feature consists of a single primitive, the collection will have only one item.

Type

MGCollection (read-only)

Error Codes

none

See Also

getPrimitives

MGMapObject.Type

Description

Gets the type of the map feature.

Note that this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read-only)

Error Codes

-1 (busy)

See Also

getType

MGPageElement Object

The MGPageElement object represents a page element on the printout.

A page element can be the map, legend, title, URL, time stamp, scale bar, north arrow, or custom symbols. You can use this object to modify the extents, print priority, and background mode of the page element.

Note

Instances of this class are valid only within the onBeginLayout and onEndLayout event handler methods. You cannot save a copy of this object for use outside of the event handler at a later time.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGPageElement extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGPageElement is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getBackMode	Gets the background mode of this element. The supported background modes are *opaque* and *transparent*.
getExtent	Gets the extents of this element on the printout.
getPrintPriority	Gets the print priority of this page element.
getSymbolAttr	Gets an object that groups the current page element's symbol attributes.
setBackMode	Sets the background mode of this element.
setExtent	Modifies the extents of this element on the printout.
setPrintPriority	Sets the print priority of this element.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

BackMode	The background mode of this page element.
Extent	The extents of the current page element on the printout.
PrintPriority	The print priority of this page element.

MGPageElement.getBackMode

Syntax

```
int getBackMode()
```

Description

Gets the background mode of this element. The supported background modes are *opaque* and *transparent*.

Returns

int - A number representing the background mode: 0 for transparent; 1 for opaque.

Error Codes

none

See Also

setBackMode, BackMode

MGPageElement.getExtent

Syntax

```
MGExtentEx getExtent()
```

Description

Gets the extents of this element on the printout.

Returns

MGExtentEx - A non-null MGExtentEx object storing the extents of the page element in pixel-based page coordinates.

Error Codes

none

See Also

setExtent, Extent

MGPageElement.getPrintPriority

Syntax

```
double getPrintPriority()
```

Description

Gets the print priority of this page element.

Page elements with lower priority numbers are printed first (i.e., a low number means a high priority). If two elements have the same number, the system prints the element with the larger extents first.

The following table lists page elements and their priorities:

Element	*Priority*
map	10
legend	20
title	30
URL	40
time stamp	50
scale bar	60
north arrow	70
all custom symbols	80

Returns

double - A number between 0.0 and 100.00, representing the print priority of the current page element.

Error Codes

none

See Also

setPrintPriority, PrintPriority

MGPageElement.getSymbolAttr

Syntax

```
MGSymbolAttr getSymbolAttr()
```

Description

Gets an object that groups the current page element's symbol attributes.

If the page element is a custom symbol, modifying the symbol attributes also modifies the appearance of the page element.

Returns

MGSymbolAttr - An MGSymbolAttr object representing the symbol attributes of the current page element, or null if the current page element is not a custom symbol.

Error Codes

none

Example

This example tests a printed page element to see if it is a custom symbol. The function takes two arguments: an MGPrintLayout object, and the ID associated with the page element:

```
function isSymbolObject(layout, ID)
{
    var ele = layout.getPageElement(ID);
    if (ele == null) {
        alert("The specified page element does not exist!");
    }
    else
    {
        var symbolAttr = ele.getSymbolAttr();
        if (symbolAttr == null)
            alert("It is NOT a symbol!");
        else
            alert("It is a symbol!");
    }
}
```

MGPageElement.setBackMode

Syntax

```
boolean setBackMode(int bgMode)
```

Description

Sets the background mode of this element.

The supported background modes are *opaque* and *transparent*. You can set the background mode of any of the page elements except the map; setting the background mode of the map has no effect.

Parameters

bgMode - An integer representing the background mode: 0 for transparent; 1 for opaque.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-3 (illegal argument)

See Also

getBackMode, BackMode

MGPageElement.setExtent

Syntax

`boolean setExtent(MGExtentEx extents)`

Description

Modifies the extents of this element on the printout.

Parameters

extents - An MGExtentEx object storing the new extents of this element on the print out in pizel-based page coordinates.

Returns

boolean - Returns True if the extents are set successfully; otherwise, returns False.

Error Codes

-3 (illegal argument)

See Also

getExtent, Extent

MGPageElement.setPrintPriority

Syntax

`boolean setPrintPriority(double printOrder)`

Description

Sets the print priority of this element.

Page elements with lower priority numbers are printed first (i.e., a low number means a high priority). If two elements have the same number, the system prints the element with the larger extents first.

The following table lists page elements and their priorities:

Element	*Priority*
map	10
legend	20
title	30
URL	40
time stamp	50
scale bar	60
north arrow	70
all custom symbols	80

Parameters

printOrder - A positive integer representing the print priority of the page element.

Returns

boolean - Returns True if the print priority is set successfully; otherwise, returns False.

Error Codes

-3 (illegal argument)

See Also

getPrintPriority, PrintPriority

MGPageElement.BackMode

Description

Gets or sets the background mode of this page element.

You can set the background mode of any of the page elements except the map; setting the background mode of the map has no effect.

Type

integer (read/write)

See Also

getBackMode, setBackMode

MGPageElement.Extent

Description

Gets or sets the extents of the current page element on the printout.

Type

MGExtentEx object (read/write)

Error Codes

none

See Also

getExtent, setExtent

MGPageElement.PrintPriority

Description

Gets or sets the print priority of this page element.

Type

double (read/write)

Error Codes

none

See Also

getPrintPriority, setPrintPriority

MGPageSetup Object

The MGPageSetup object organizes the Page Setup properties, which control how Autodesk MapGuide prints maps.

To get an instance of the MGPageSetup class, use the getPageSetup method from MGMap.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGPageSetup extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGPageSetup is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getInclude	Determines whether the specified element will be included on the printout.
getIncludeLegend	Returns a value indicating whether the legend will be printed next to the map.
getPageMargins	Gets the page margins used by the current page layout.
getPlotScale	Gets the scale if the map is to be printed at a specific map scale.
getTitle	Gets the title to be printed above the map at the top of the page.
getTitleTextAttr	Gets an object that groups the title text attributes.
getUsePlotScale	Returns a value indicating whether the map will be printed at a specific map scale.
setInclude	Changes the visibility state of the page element on the print out.
setIncludeLegend	Sets a value indicating whether the legend will be printed next to the map.
setPageMargins	Sets the page margins used by the current page layout.
setPlotScale	Sets the scale if the map is to be printed at a specific map scale.
setTitle	Sets the title to print at the top of the page when you print a map.
setUsePlotScale	Sets a value indicating whether the map will be printed at a specific scale.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

IncludeLegend	Gets or sets whether to print the legend next to the map.
PlotScale	Gets or sets the map scale at which the map will print.
Title	Gets or sets the title that will print at the top of the page when you print a map.
UsePlotScale	Gets or sets whether to print the map at a specific map scale.

MGPageSetup.getInclude

Syntax

`boolean getInclude(String elementID)`

Description

Determines whether the specified element will be included on the printout.

Parameters

elementID - A unique, case-sensitive string ID representing possible printed page elements. The following IDs are currently supported: mg_legend, mg_northarrow, mg_scalebar, mg_title, mg_timestamp, and mg_url.

Returns

boolean - Returns True if the element will be included on the printout; otherwise, returns False.

Error Codes

-3 (illegal argument)

Example

This simple example tests whether the legend will be visible on the printout. The function takes an MGPageSetup object as its argument:

```
function testLegend(pgsetup)
{
    if (pgsetup.getInclude("mg_legend") == true)
        alert("Legend will be visible!");
    else
        alert("Legend will not be visible!");
}
```

See Also

setInclude

MGPageSetup.getIncludeLegend

Syntax

```
boolean getIncludeLegend()
```

Description

Returns a value indicating whether the legend will be printed next to the map.

This method is retained for compatibility with previous versions of Autodesk MapGuide; it will be deprecated in the next release. Developers are encouraged to use getInclude() instead.

Returns

boolean - Returns True if the legend is to be printed; False if the legend is to be omitted from the printed page.

Error Codes

none

See Also

getInclude

MGPageSetup.getPageMargins

Syntax

```
MGCollection getPageMargins()
```

Description

Gets the page margins used by the current page layout.

Returns

MGCollection - An MGCollection object storing the four page margin values in pixel-based device coordinates. The left, right, top, and bottom margin values are stored in the collection as array items 0 through 3, respectively.

Error Codes

none

Example

This function is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It displays the current page margins. The function takes an MGPageSetup object as its argument:

```
function displayPageMargins(pgsetup)
{
    var margins = pgsetup.getPageMargins();
    alert("Left margin = " + margins.item(0));
    alert("Top margin = " + margins.item(1));
    alert("Right margin = " + margins.item(2));
    alert("Bottom margin = " + margins.item(3));
}
```

See Also

setPageMargins

MGPageSetup.getPlotScale

Syntax

```
double getPlotScale()
```

Description

Gets the scale if the map is to be printed at a specific map scale.

When Autodesk MapGuide Viewer loads a map, getPlotScale returns the initial scale of the map.

Returns

double - The map scale at which the map is to be printed.

Error Codes

none

See Also

setUsePlotScale, setPlotScale, getUsePlotScale

MGPageSetup.getTitle

Syntax

```
String getTitle()
```

Description

Gets the title to be printed above the map at the top of the page.

Returns

String - The title to be printed at the top of the page.

Error Codes

none

See Also

setTitle

MGPageSetup.getTitleTextAttr

Syntax

```
MGTextAttr getTitleTextAttr()
```

Description

Gets an object that groups the title text attributes.

By modifying the values of this object, you can change the appearance of the printed title.

Returns

MGTextAttr - A non-null MGTextAttr object.

Error Codes

none

Example

This function sets the font of the printed title. It takes an MGPageSetup object as its argument:

```
function changeTitle(pgsetup)
{
    var attr = pgsetup.getTitleTextAttr();
    attr.setFont("Impact");
}
```

MGPageSetup.getUsePlotScale

Syntax

```
boolean getUsePlotScale()
```

Description

Returns a value indicating whether the map will be printed at a specific map scale.

Returns

boolean - Returns True if the map will be printed at a specific scale, False if the map will be scaled to fit the printed page.

Error Codes

none

See Also

setUsePlotScale, setPlotScale, getPlotScale

MGPageSetup.setInclude

Syntax

```
boolean setInclude(String elementID, boolean bInclude)
```

Description

Changes the visibility state of the page element on the print out.

Parameters

elementID - A unique, case-sensitive string ID representing possible printed page elements. The following IDs are currently supported: mg_legend, mg_northarrow, mg_scalebar, mg_title, mg_timestamp, and mg_url.

bInclude - A boolean value indicating whether the page element will be included on the printout.

Returns

boolean - Returns True if the operation is successful; otherwise, returns False.

Error Codes

-3 (illegal argument)

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It turns off the Include Scale Bar and Include North Arrow checkboxes on the Page Setup dialog and then initiates the printing process:

```
// map is a reference to the MapGuide Viewer on the HTML page
function printCurrentMap(MGMap map)
```

```
{
    var pgSetup = map.getPageSetup();
    pgSetup.setInclude("mg_scalebar", false);
    pgSetup.setInclude("mg_northarrow", false);
    map.printDlg();
}
```

See Also

getInclude

MGPageSetup.setIncludeLegend

Syntax

`void setIncludeLegend(boolean flag)`

Description

Sets a value indicating whether the legend will be printed next to the map.

Note: This method is retained for compatibility with previous versions of Autodesk MapGuide; it will be deprecated in the next release. Developers are encouraged to use getInclude() instead.

Parameters

flag - True to print the legend with the map; False to omit the legend when printing the map.

Error Codes

none

See Also

setInclude

MGPageSetup.setPageMargins

Syntax

`boolean setPageMargins(int left, int right, int top, int bottom)`

Description

Sets the page margins used by the current page layout.

Parameters

left - The left margin, in pixel-based device coordinates.

right - The right margin, in pixel-based device coordinates.

top - The top margin, in pixel-based device coordinates.

bottom - The bottom margin, in pixel-based device coordinates.

Returns

boolean - Returns True if the operation is successful; otherwise, returns False.

Error Codes

-3 (illegal argument)

Example

This function is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It disables the page margins. It takes an MGPageSetup object as its argument:

```
function disablePageMargins(pgsetup)
{
    pgsetup.setPageMargins(0, 0, 0, 0);
}
```

See Also

getPageMargins

MGPageSetup.setPlotScale

Syntax

`void setPlotScale(double scale)`

Description

Sets the scale if the map is to be printed at a specific map scale.

The scale is honored only if setUsePlotScale or UsePlotScale is set to True.

Parameters

scale - The map scale at which the map will print. For example: 50000.0 to print the map at a scale of 1:50,000.

Error Codes

-3 (illegal argument)

See Also

setUsePlotScale, getUsePlotScale, getPlotScale

MGPageSetup.setTitle

Syntax

`void setTitle(String title)`

Description

Sets the title to print at the top of the page when you print a map.

Parameters

title - The title to print at the top of the printed page.

Error Codes

none

See Also

getTitle

MGPageSetup.setUsePlotScale

Syntax

`void setUsePlotScale(boolean flag)`

Description

Sets a value indicating whether the map will be printed at a specific scale.

Parameters

flag - True to print the map at a specific scale; False to scale the map to fit the printed page.

Error Codes

none

See Also

setPlotScale, getUsePlotScale, getPlotScale

MGPageSetup.IncludeLegend

Description

Gets or sets whether to print the legend next to the map.

Type

boolean (read/write)

Error Codes

none

See Also

getIncludeLegend, setIncludeLegend

MGPageSetup.PlotScale

Description

Gets or sets the map scale at which the map will print.

When Autodesk MapGuide Viewer loads a map, the value of PlotScale is the initial scale of the map.

Type

double (read/write)

Error Codes

-3 (illegal argument)

See Also

getPlotScale, setPlotScale

MGPageSetup.Title

Description

Gets or sets the title that will print at the top of the page when you print a map.

Type

String (read/write)

Error Codes

none

See Also

getTitle, setTitle

MGPageSetup.UsePlotScale

Description

Gets or sets whether to print the map at a specific map scale.

Type

boolean (read/write)

Error Codes

none

See Also

getUsePlotScale, setUsePlotScale

MGPoint Object

The MGPoint object represents a point on the map expressed in MCS or latitude/longitude coordinates.

For a point expressed in latitude/longitude, *X* represents longitude, while *Y* represents latitude. You can create an MGPoint object using the createObject method in MGMap.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGPoint extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGPoint is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

equals Compares this MGPoint object to the specified object.

getX Returns the *X* value (longitude) of the point.

getY Returns the *Y* value (latitude) of the point.

setX Sets the *X* value (longitude) of the point.

setY Returns the *Y* value (latitude) of the point.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

X The *X* value (longitude) of the point.

Y The *Y* value (latitude) of the point.

MGPoint.equals

Syntax

```
boolean equals(Object obj)
```

Description

Compares this MGPoint object to the specified object.

Parameters

obj - The object to compare to the MGPoint object.

Returns

boolean - Returns True if the two objects are identical; False if they are not.

Error Codes

-3 (illegal argument)

MGPoint.getX

Syntax

```
double getX()
```

Description

Gets the *X* value (longitude) of the point.

Returns

double - The *X* value of the point.

Error Codes

none

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It gets the lat/lon coordinates of the map's center point, and then uses those coordinates to create a new MGPoint object.

```
function getThePoint()
{
    var map = getMap();

    // Get and display map's center point, in lat/lon
    lat = map.getLat();
    lon = map.getLon();
    alert( "Lat: " +lat + "\nLon: " + lon );
    // Create new MGPoint, using same lat/lon values
    var ctrPt = map.CreateObject('MGPoint');
    ctrPt.setX(lon);
    ctrPt.setY(lat);
    // Display point again, this time using MGPoint object
    alert( "Lat: " + ctrPt.getY() + "\nLon: " + ctrPt.getX() );
}
```

The next example shows a function that lets the user pan the screen in one of four directions. The direction is supplied to the function as an argument:

```
function moveIt(direction)
{
```

```
    var map = getMap();
    var delta;

    // Get width or height of map, in meters
    if (direction == "Left" | direction == "Right")
        delta = map.getWidth("M");
    if (direction == "Up" | direction == "Down")
        delta = 0.60 * map.getWidth("M");
    delta = 0.90 * delta;
    // Compute center point of map in MCS units
    var xyPt = map.lonLatToMcs(map.getLon(), map.getLat());
    // Convert delta from meters to MCS units.
    var MCStoMeters = map.getMCSScaleFactor();
    delta = delta / MCStoMeters;
    // Adjust center point by width/height of map
    if (direction == "Left")
        xyPt.setX(xyPt.getX() - delta);
    if (direction == "ight")
        xyPt.setX(xyPt.getX() + delta);
    if (direction == "Up")
        xyPt.setY(xyPt.getY() + delta);
    if (direction == "Down")
        xyPt.setY(xyPt.getY() - delta);
    // Convert center point back to lat/lon coordinates
    xyPt = map.mcsToLonLat(xyPt.getX(), xyPt.getY());
    // Zoom to the new location
    map.zoomScale(xyPt.getY(), xyPt.getX(), map.getScale());
}
```

You might use the following HTML to call the function from your page:

```
<FORM>
<INPUT TYPE="button" VALUE="Left" ONCLICK="moveIt('Left');">
<INPUT TYPE="button" VALUE="Right" ONCLICK="moveIt(Right);">
<INPUT TYPE="button" VALUE="Up" ONCLICK="moveIt('Up');">
<INPUT TYPE="button" VALUE="Down" ONCLICK="moveIt('Down');">
</FORM>
```

See Also

X

MGPoint.getY

Syntax

```
double getY()
```

Description

Gets the *Y* value (latitude) of the point.

Returns

double - The *Y* value of the point.

Error Codes

none

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It gets the lat/lon coordinates of the map's center point, and then uses those coordinates to create a new MGPoint object.

```
function getThePoint()
{
    var map = getMap();

    // Get and display map's center point, in lat/lon
    lat = map.getLat();
    lon = map.getLon();
    alert( "Lat: " +lat + "\nLon: " + lon );
    // Create new MGPoint, using same lat/lon values
    var ctrPt = map.CreateObject('MGPoint');
    ctrPt.setX(lon);
    ctrPt.setY(lat);
    // Display point again, this time using MGPoint object
    alert( "Lat: " + ctrPt.getY() + "\ " n: " + ctrPt.getX() );
}
```

The next example shows a function that lets the user pan the screen in one of four directions. The direction is supplied to the function as an argument:

```
function moveIt(direction)
{
    var map = getMap();
    var delta;

    // Get width or height of map, in meters
    if (direction == "Left" | direction == "Right")
        delta = map.getWidth("M");
    if (direction == "Up" | direction == "Down")
        delta = 0.60 * map.getWidth("M");
    delta = 0.90 * delta;
    // Compute center point of map in MCS units
    var xyPt = map.lonLatToMcs(map.getLon(), map.getLat());
    // Convert delta from meters to MCS units.
    var MCStoMeters = map.getMCSScaleFactor();
    delta = delta / MCStoMeters;
    // Adjust center point by width/height of map
    if (direction == "Left")
        xyPt.setX(xyPt.getX() - delta);
    if (direction == "Right")
        xyPt.setX(xyPt.getX() + delta);
    if (direction == "Up")
        xyPt.setY(xyPt.getY() + delta);
    if (direction == "Down")
        xyPt.setY(xyPt.getY() - delta);
    // Convert center point back to lat/lon coordinates
    xyPt = map.mcsToLonLat(xyPt.getX(), xyPt.getY());
    // Zoom to the new location
    map.zoomScale(xyPt.getY(), xyPt.getX(), map.getScale());
}
```

You might use the following HTML to call the function from your page:

```
<FORM>
<INPUT TYPE="button" VALUE="Left" ONCLICK="moveIt('Left');">
<INPUT TYPE="button" VALUE="Right" ONCLICK="moveIt(Right);">
```

```
<INPUT TYPE="button" VALUE="Up" ONCLICK="moveIt('Up');">
<INPUT TYPE="button" VALUE="Down" ONCLICK="moveIt('Down');">
</FORM>
```

See Also

Y

MGPoint.setX

Syntax

`void setX(double newValue)`

Description

Sets the *X* value (longitude) of the point.

Parameters

newValue - The new *X* value of the point.

Error Codes

none

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It gets the lat/lon coordinates of the map's center point, and then uses those coordinates to create a new MGPoint object.

```
function getThePoint()
{
    var map = getMap();

    // Get and display map's center point, in lat/lon
    lat = map.getLat();
    lon = map.getLon();
    alert( "Lat: " +lat + "\nLon: " + lon );
    // Create new MGPoint, using same lat/lon values
    var ctrPt = map.CreateObject('MGPoint');
    ctrPt.setX(lon);
    ctrPt.setY(lat);
    // Display point again, this time using MGPoint object
    alert( "Lat: " + ctrPt.getY() + "\nLon: " + ctrPt.getX() );
}
```

The next example shows a function that lets the user pan the screen in one of four directions. The direction is supplied to the function as an argument:

```
function moveIt(direction)
{
    var map = getMap();
    var delta;

    // Get width or height of map, in meters
    if (direction == "Left" | direction == "Right")
```

```
        delta = map.getWidth("M");
    if (direction == "Up" | direction == "Down")
        delta = 0.60 * map.getWidth("M");
    delta = 0.90 * delta;
    // Compute center point of map in MCS units
    var xyPt = map.lonLatToMcs(map.getLon(), map.getLat());
    // Convert delta from meters to MCS units.
    var MCStoMeters = map.getMCSScaleFactor();
    delta = delta / MCStoMeters;
    // Adjust center point by width/height of map
    if (direction == "Left")
        xyPt.setX(xyPt.getX() - delta);
    if (direction == "Right")
        xyPt.setX(xyPt.getX() + delta);
    if (direction == "Up")
        xyPt.setY(xyPt.getY() + delta);
    if (direction == "Down")
        xyPt.setY(xyPt.getY() - delta);
    // Convert center point back to lat/lon coordinates
    xyPt = map.mcsToLonLat(xyPt.getX(), xyPt.getY());
    // Zoom to the new location
    map.zoomScale(xyPt.getY(), xyPt.getX(), map.getScale());
}
```

You might use the following HTML to call the function from your page:

```
<FORM>
<INPUT TYPE="button" VALUE="Left" ONCLICK="moveIt('Left');">
<INPUT TYPE="button" VALUE="Right" ONCLICK="moveIt(Right);">
<INPUT TYPE="button" VALUE="Up" ONCLICK="moveIt('Up');">
<INPUT TYPE="button" VALUE="Down" ONCLICK="moveIt('Down');">
</FORM>
```

See Also

X

MGPoint.setY

Syntax

`void setY(double newValue)`

Description

Gets the *Y* value (latitude) of the point.

Parameters

newValue - The new *Y* value of the point.

Error Codes

none

Example

This example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It gets the lat/lon coordinates of the map's center point, and then uses those coordinates to create a new MGPoint object.

```
function getThePoint()
{
    var map = getMap();

    // Get and display map's center point, in lat/lon
    lat = map.getLat();
    lon = map.getLon();
    alert( "Lat: " +lat + "\nLon: " + lon );
    // Create new MGPoint, using same lat/lon values
    var ctrPt = map.CreateObject('MGPoint');
    ctrPt.setX(lon);
    ctrPt.setY(lat);
    // Display point again, this time using MGPoint object
    alert( "Lat: " + ctrPt.getY() + "\nLon: " + ctrPt.getX() );
}
```

The next example shows a function that lets the user pan the screen in one of four directions. The direction is supplied to the function as an argument:

```
function moveIt(direction)
{
    var map = getMap();
    var delta;

    // Get width or height of map, in meters
    if (direction == "Left" | direction == "Right")
        delta = map.getWidth("M");
    if (direction == "Up" | direction == "Down")
        delta = 0.60 * map.getWidth("M");
    delta = 0.90 * delta;
    // Compute center point of map in MCS units
    var xyPt = map.lonLatToMcs(map.getLon(), map.getLat());
    // Convert delta from meters to MCS units.
    var MCStoMeters = map.getMCSScaleFactor();
    delta = delta / MCStoMeters;
    // Adjust center point by width/height of map
    if (direction == "Left")
        xyPt.setX(xyPt.getX() - delta);
    if (direction == "Right")
        xyPt.setX(xyPt.getX() + delta);
    if (direction == "Up")
        xyPt.setY(xyPt.getY() + delta);
    if (direction == "Down")
        xyPt.setY(xyPt.getY() - delta);
    // Convert center point back to lat/lon coordinates
    xyPt = map.mcsToLonLat(xyPt.getX(), xyPt.getY());
    // Zoom to the new location
    map.zoomScale(xyPt.getY(), xyPt.getX(), map.getScale());
}
```

You might use the following HTML to call the function from your page:

```
<FORM>
<INPUT TYPE="button" VALUE="Left" ONCLICK="moveIt('Left');">
<INPUT TYPE="button" VALUE="Right" ONCLICK="moveIt(Right);">
```

```
<INPUT TYPE="button" VALUE="Up" ONCLICK="moveIt('Up');">
<INPUT TYPE="button" VALUE="Down" ONCLICK="moveIt('Down');">
</FORM>
```

See Also

Y

MGPoint.X

Description

Gets or sets the *X* value (longitude) of the point.

Type

double (read/write)

Error Codes

none

See Also

getX, setX

MGPoint.Y

Description

Gets or sets the *Y* value (latitude) of the point.

Type

double (read/write)

Error Codes

none

See Also

getY, setY

MGPrimitive Object

The MGPrimitive object allows you to query the individual primitives that make up a multi-primitive feature such as a GIS Design Server or redline feature.

You can also query simple, single-primitive features (point, text, or polyline features, for example), but doing so yields the same results as querying the entire feature.

MGPrimitive objects cannot be publicly created. To access this object, use the getPrimitives method or Primitives property of MGMapObject.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGPrimitive extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGPrimitive is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getArea	Gets the area of the specified polygon map feature primitive.
getExtent	Gets the extents of the map feature primitive.
getLength	Gets the length of the specified polyline map feature primitive.
getParentObject	Gets the parent object of the primitive.
getTextValue	Gets the text value for a text primitive.
getType	Gets the type of map feature primitive.
getVertices	Gets the vertices used to describe the geometry of the map feature primitive.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

Area	The area of the specified polygon primitive in Mapping Coordinate System (MCS) units.
Length	The length of the specified polyline primitive in Mapping Coordinate System (MCS) units.
ParentObject	Gets the map feature that the primitive belongs to.
TextValue	Gets the string value of a text primitive.
Type	The map feature primitive type.

MGPrimitive.getArea

Syntax

```
double getArea()
```

Description

Gets the area of the specified polygon map feature primitive.

Returns

double - The area of the specified polygon primitive in Mapping Coordinate System (MCS) units, or 0.0 if the specified primitive is not a polygon primitive.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

Area

MGPrimitive.getExtent

Syntax

```
MGExtentEx getExtent(boolean useMcs)
```

Description

Gets the extents of the map feature primitive.

Note that for point feature primitives, this property does not contain the posting point; instead, it contains the four corners representing the extent of the symbol posted for the point. To get the original posting point, use the getVertices method.

Parameters

useMCS - a boolean specifying whether the extent coordinates are in MCS or lat/lon.

Returns

MGExtentEx - The extents of map feature primitive.

Error Codes

-1 (busy)

MGPrimitive.getLength

Syntax

```
double getLength()
```

Description

Gets the length of the specified polyline map feature primitive.

Returns

double - The area of the specified polyline primitive in Mapping Coordinate System (MCS) units, or 0.0 if the specified primitive is not a polyline primitive. If the polyline primitive is composed of more than one polyline internally, the sum of all the polyline distances will be returned.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

Length

MGPrimitive.getParentObject

Syntax

```
MGMapObject getParentObject()
```

Description

Gets the parent object of the primitive.

Returns

MGMapObject - The parent map feature of the primitive.

Error Codes

-1 (busy)

See Also

ParentObject

MGPrimitive.getTextValue

Syntax

```
String getTextValue()
```

Description

Gets the text value for a text primitive.

Returns

String - The text value for the primitive, or an empty string (and a does not apply error) if the primitive is not a text primitive.

Error Codes

-1 (busy)

-15 (does not apply)

See Also

TextValue

MGPrimitive.getType

Syntax

```
String getType()
```

Description

Gets the type of map feature primitive.

Returns

String - The map feature primitive type, which will be "Point", "Text", "Polyline", or "Polygon".

Error Codes

-1 (busy)

-15 (does not apply)

See Also

Type

MGPrimitive.getVertices

Syntax

```
int getVertices(MGCollection vertices, MGCollection numberOfVertices)
```

Description

Gets the vertices used to describe the geometry of the map feature primitive.

This method works only if the author of the map enabled access by checking the *Allow Access To Coordinate Values From API* check box in Autodesk MapGuide Author. If this option is not enabled, getVertices returns 0.

Parameters

vertices - An MGCollection of MGPoint objects, with coordinates in MCS units.

numberOfVertices - For simple feature primitives (text, point, polyline, polygon), the collection contains a single integer representing the number of vertices in the collection object. For poly-polylines and poly-polygons, the collection of integer objects represents the number of vertices contained in each component polyline or polygon.

Returns

int - The number of internal objects, which is the same as the length of the collection returned by *numberOfVertices*. Returns one of the following:

1 The feature is a polyline, polygon, point, or text.

>1 The feature is a poly-polyline or poly-polygon.

0 The map does not allow access to its coordinates from the API, or the Autodesk MapGuide Viewer is Release 4.0 or later and the map was authored with Autodesk MapGuide Author Release 3.0 or earlier.

Error Codes

-1 (busy)

-3 (illegal argument)

-5 (security violation)

-15 (does not apply)

MGPrimitive.Area

Description

Gets the area of the specified polygon primitive in Mapping Coordinate System (MCS) units.

If the specified primitive is not a polygon primitive, Area returns zero.

Note that this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

double (read-only)

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getArea, Length

MGPrimitive.Length

Description

Gets the length of the specified polyline primitive in Mapping Coordinate System (MCS) units.

If the specified primitive is not a polyline primitive, Length returns zero. If the specified primitive is composed of more than one polyline feature internally, Length returns the sum of all the lengths of the component polylines.

Note that this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

double (read-only)

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getLength, Area

MGPrimitive.ParentObject

Description

Gets the map feature that the primitive belongs to.

Note that this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

MGMapObject (read-only)

Error Codes

-1 (busy)

See Also

getParentObject

MGPrimitive.TextValue

Description

Gets the string value of a text primitive.

Note that this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read-only)

Error Codes

-1 (busy)

-15 (does not apply)

See Also

getTextValue

MGPrimitive.Type

Description

Gets the map feature primitive type.

The type can be one of the following: "Point", "Text", "Polyline", or "Polygon".

Note that this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read-only)

Error Codes

-1 (busy)

See Also

getType

MGPrintInfo Object

The MGPrintInfo object provides information about the current print operation.

This object is passed as an argument to the onBeginLayout and onEndLayout events, allowing you to write event handler code specific to a particular print operation.

Instances of this class are valid only within the event handler methods. You cannot save a copy of this object for use outside of the event handler at a later time.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGPrintInfo extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGPrintInfo is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getPageExtent	Gets the size of the printable area of the page.
getPageResolution	Gets the resolution of the output device to be used for the current print operation.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same

get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

PageExtent	The size of the printable area of the page.
PageResolution	The resolution of the output device to be used for the current print operation.

MGPrintInfo.getPageExtent

Syntax

```
int getPageExtent()
```

Description

Gets the size of the printable area of the page.

Returns

int - The size of the printable area of the paper in pixel-based page coordinates.

Error Codes

none

See Also

PageExtent

MGPrintInfo.getPageResolution

Syntax

```
int getPageResolution()
```

Description

Gets the resolution of the output device to be used for the current print operation.

Returns

int - An integer value representing the resolution in pixels per inch.

Error Codes

none

See Also

PageResolution

MGPrintInfo.PageExtent

Description

Gets the size of the printable area of the page.

Type

MGExtentEx object (read-only)

Error Codes

none

See Also

getPageExtent

MGPrintInfo.PageResolution

Description

Gets the resolution of the output device to be used for the current print operation.

Type

int (read-only)

Error Codes

none

See Also

getPageResolution

MGPrintLayout Object

The MGExtentEx object controls the layout of the page elements on the printout.

The MGPrintLayout object is passed as an argument to the onEndLayout event, allowing you to write event handler code specific to a particular print operation.

Instances of this class are valid only within the event handler method. You cannot save a copy of this object for use outside of the event handler at a later time.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGPrintLayout extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGPrintLayout is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

addSymbol	Adds a symbol to the printed page. Symbols are defined in the API Symbol List; refer to the *Autodesk MapGuide User's Guide* for more information
getPageElement	Gets the page element with the specified element ID.

MGPrintLayout.addSymbol

Syntax

```
MGPageElement addSymbol(String symbolID)
```

Description

Adds a symbol to the printed page. Symbols are defined in the API Symbol List; refer to the *Autodesk MapGuide User's Guide* for more information

Parameters

symbolID - A unique, case-sensitive string ID that identifies a symbol in the API Symbol List.

Returns

MGPageElement - An MGPageElement object representing the symbol, or null if the specified symbol was not found.

Error Codes

-7 (no such element)

MGPrintLayout.getPageElement

Syntax

```
MGPageElement getPageElement(String elementID)
```

Description

Gets the page element with the specified element ID.

Parameters

elementID - A unique, case-sensitive string ID representing possible printed page elements. The ID will be a custom ID specified by the user, or it will be one of the following built-in values: mg_legend, mg_northarrow, mg_scalebar, mg_title, mg_timestamp, and mg_url.

Returns

MGPageElement - An MGPageElement object containing the specified element ID, or null if the element is not found.

Error Codes

-7 (no such element)

MGRedlineSetup Object

The MGRedlineSetup object provides access to the style information used to create features on a redline layer.

The MGRedlineSetup object is accessed by calling the MGMap.getRedlineSetup method.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGRedlineSetup extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGRedlineSetup is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getEdgeAttr	Gets an object that groups the redline style's polygon edge attributes.
getFillAttr	Gets an object that groups the redline style's polygon fill attributes.
getLineAttr	Gets an object that groups the redline style's line or polyline attributes.
getSymbolAttr	Gets an object that groups the redline style's symbol attributes.
getTextAttr	Gets an object that groups the redline style's text attributes.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

EdgeAttr	Gets an object that groups the redline style's polygon edge attributes.
FillAttr	Gets an object that groups the redline style's polygon fill attributes.
LineAttr	Gets an object that groups the redline style's line or polyline attributes.
SymbolAttr	Gets an object that groups the redline style's symbol attributes.
TextAttr	Gets an object that groups the redline style's text attributes.

MGRedlineSetup.getEdgeAttr

Syntax

```
MGEdgeAttr getEdgeAttr()
```

Description

Gets an object that groups the redline style's polygon edge attributes.

Returns

MGEdgeAttr - An MGEdgeAttr object that controls how polygon edges are created on the redline layer.

Error Codes

none

See Also

EdgeAttr

MGRedlineSetup.getFillAttr

Syntax

```
MGFillAttr getFillAttr()
```

Description

Gets an object that groups the redline style's polygon fill attributes.

Returns

MGFillAttr - An MGFillAttr object that controls how polygon fills are created on the redline layer.

Error Codes

none

See Also

FillAttr

MGRedlineSetup.getLineAttr

Syntax

```
MGLineAttr getLineAttr()
```

Description

Gets an object that groups the redline style's line or polyline attributes.

Returns

MGLineAttr - An MGLineAttr object that controls how lines and polylines are created on the redline layer.

Error Codes

none

See Also

LineAttr

MGRedlineSetup.getSymbolAttr

Syntax

```
MGSymbolAttr getSymbolAttr()
```

Description

Gets an object that groups the redline style's symbol attributes.

Returns

MGSymbolAttr - An MGSymbolAttr object that controls how symbol primitives are created on the redline layer.

Error Codes

none

See Also

SymbolAttr

MGRedlineSetup.getTextAttr

Syntax

```
MGTextAttr getTextAttr()
```

Description

Gets an object that groups the redline style's text attributes.

Returns

MGTextAttr - An MGTextAttr object that controls how text and note features are created on the redline layer.

Error Codes

none

See Also

TextAttr

MGRedlineSetup.EdgeAttr

Description

Gets an object that groups the redline style's polygon edge attributes.

Type

MGEdgeAttr (read-only)

Error Codes

none

See Also

getEdgeAttr

MGRedlineSetup.FillAttr

Description

Gets an object that groups the redline style's polygon fill attributes.

Type

MGFillAttr (read-only)

Error Codes

none

See Also

getFillAttr

MGRedlineSetup.LineAttr

Description

Gets an object that groups the redline style's line or polyline attributes.

Type

MGLineAttr (read-only)

Error Codes

none

See Also

getLineAttr

MGRedlineSetup.SymbolAttr

Description

Gets an object that groups the redline style's symbol attributes.

Type

MGSymbolAttr (read-only)

Error Codes

none

See Also

getSymbolAttr

MGRedlineSetup.TextAttr

Description

Gets an object that groups the redline style's text attributes.

Type

MGTextAttr (read-only)

Error Codes

none

See Also

getTextAttr

MGSelection Object

The MGSelection object represents the list of selected map features.

Note that instances of the class MGSelection cannot be publicly created.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGSelection extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGSelection is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

addObject	Adds a single map feature to the selection and updates the display.
addObjects	Adds the specified map features to the selection and updates the display.
addObjectsEx	Adds multiple map features to the selection and updates the display.
clear	Removes all the map features from the selection and updates the display.
getAsString	Returns a string that contains the keys of the selected map features.
getMapObjects	Returns a Vector (or MGCollection) of selected map features.
getMapObjectsEx	Gets a collection containing the selected map features.

getNumObjects	Gets the number of map features in the current selection.
removeObject	Removes a single map feature from the selection and updates the display.
removeObjects	Removes the specified map features from the selection and updates the display.
removeObjectsEx	Removes multiple map features from the selection and updates the display.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

NumObjects The number of map features in the current selection.

MGSelection.addObject

Syntax

`void addObject(MGMapObject obj, boolean notify)`

Description

Adds a single map feature to the selection and updates the display.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

obj - The map feature to add to the selection.

notify - If true, fires the onSelectionChanged event after the addition.

Error Codes

-1 (busy)

-3 (illegal argument)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It takes two strings -- a layer name and a feature key -- as its arguments. The function arguments are used with getMapLayer() and getMapObject() to get a map feature. That feature is then added to the selection and zoomed to:

```
function zoomTo(layerName, featureKey)
{
    var map = getMap();
    var sel = map.getSelection();
    // Get feature, using arguments passed down to function
    var feature = map.getMapLayer(layerName).getMapObject(featureKey);
```

```
    // Prevent busy state from happening when
    // zoomSelected() is called
    map.setAutoRefresh(false);
    // Clear existing selection (if any); then add feature to it
    sel.clear();
    sel.addObject(feature, false);

    // Zoom to feature, which is selected
    map.zoomSelected();
    map.setWidth(5000, "KM");
    // Reset autoRefresh flag; then update display
    map.setAutoRefresh(true);
    map.refresh();
}
```

You might use the following HTML to zoom to the New Zealand feature on the World Countries layer:

```
<FORM>
    <INPUT TYPE="button" VALUE="Toggle" ONCLICK="zoomTo(NZ">
</FORM>
```

Or, you could let the user pick the layer and feature:

```
    <FORM NAME="form01">
    Name:
    <INPUT TYPE="text" NAME="layer" VALUE="">
    Feature:
    <INPUT TYPE="text" NAME="feature" VALUE="">
    <INPUT TYPE="button" VALUE="Go!" ONCLICK="zoomTo(document.form01.layer.value,
document.form01.feature.value);">
</FORM>
```

See Also

addObjectsEx

MGSelection.addObjects

Syntax

ActiveX Control: `void addObjects(MGCollection objects, boolean notify)`

Plug-In/Java Edition: `void addObjects(Vector objects, boolean notify)`

Description

Adds the specified map features to the selection and updates the display.

For compatibility between the Autodesk MapGuide Viewer Plug-In, Java Edition, and ActiveX Control, use addObjectsEx instead of addObjects.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

objects - A java.util.Vector or MGCollection of MGMapObjects. *notify* - If true, fires the onSelectionChanged event after the addition.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

addObject, addObjectsEx

MGSelection.addObjectsEx

Syntax

```
void addObjectsEx(MGCollection objects, boolean notify)
```

Description

Adds multiple map features to the selection and updates the display.

This method replaces the addObjects method in earlier versions of the Autodesk MapGuide Viewer API and supports the Autodesk MapGuide Viewer Plug-In, Java Edition, and ActiveX Control with the same syntax. The older method will still work in Release 6, but you should use addObjectsEx for all future development.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

objects - Collection of MGMapObjects to add to the selection.

notify - If true, fires the onSelectionChanged event after the addition.

Error Codes

-1 (busy)

-3 (illegal argument)

MGSelection.clear

Syntax

```
void clear()
```

Description

Removes all the map features from the selection and updates the display.

This method is the equivalent of the Select > Clear menu command in the user interface.

Error Codes

none

Example

This simple but useful example removes all features from the current selection. You might attach it to a Clear Selection button on your HTML page:

```
function clearASel()
{
    getMap().getSelection().clear();
}
```

MGSelection.getAsString

Syntax

```
String getAsString(MGMapLayer layer,
        String quoteChar)
```

Description

Returns a string that contains the keys of the selected map features.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

layer - The layer from which feature keys will be returned (if null, feature keys are returned from all layers).

quoteChar - Character used to quote character-based keys. Numeric keys are never quoted.

Returns

String - Comma-delimited list of map feature keys.

Error Codes

-1 (busy)

MGSelection.getMapObjects

Syntax

ActiveX Control: `MGCollection getMapObjects(MGMapLayer layer)`

Plug-In/Java Edition: `Vector getMapObjects(MGMapLayer layer)`

Description

Returns a Vector (or MGCollection) of selected map features.

For compatibility between the Autodesk MapGuide Viewer Plug-In, Java Edition, and ActiveX Control, use getMapObjectsEx instead of getMapObjects.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

layer - The layer from which features will be returned (if null, features are returned from all layers).

Returns

ActiveX Control: An MGCollection of MGMapObject objects. Returns empty MGCollection if unsuccessful.

Plug-In and Java Edition: Vector - A *java.util.Vector* of MGMapObject objects. Returns empty Vector if unsuccessful.

Error Codes

-1 (busy)

MGSelection.getMapObjectsEx

Syntax

```
MGCollection getMapObjectsEx(MGMapLayer
  layer)
```

Description

Gets a collection containing the selected map features.

This method replaces the getMapObjects method in earlier versions of the Autodesk MapGuide Viewer API and supports the Autodesk MapGuide Viewer Plug-In, Java Edition, and ActiveX Control Control with the same syntax. The older method will still work in Release 6, but you should use getMapObjectsEx for all future development.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

layer - If non null, only feature keys within the specified layer are returned; otherwise, all feature keys are returned.

Returns

MGCollection - A collection of the selected MGMapObjects, or empty collection if unsuccessful.

Error Codes

-1 (busy)

MGSelection.getNumObjects

Syntax

```
int getNumObjects()
```

Description

Gets the number of map features in the current selection.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Returns

int - The number of selected map features.

Error Codes

-1 (busy)

See Also

NumObjects

MGSelection.removeObject

Syntax

`void removeObject(MGMapObject obj, boolean notify)`

Description

Removes a single map feature from the selection and updates the display.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

obj - The map feature to remove from the selection.

notify - If true, fires the onSelectionChanged event after the removal.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

removeObjectsEx

MGSelection.removeObjects

Syntax

ActiveX Control: `void removeObjects(MGCollection objects, boolean notify)`

Plug-In/Java Edition: `void removeObjects(Vector objects, boolean notify)`

Description

Removes the specified map features from the selection and updates the display.

For compatibility between the Autodesk MapGuide Viewer Plug-In, Java Edition, and Autodesk MapGuide Viewer ActiveX Control, use removeObjectsEx instead of removeObjects.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

objects - A java.util.Vector or MGCollection containing the map features to remove from the selection.

notify - If true, fires the onSelectionChanged event after the removal.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

removeObject, removeObjectsEx

MGSelection.removeObjectsEx

Syntax

`void removeObjectsEx(MGCollection objects, boolean notify)`

Description

Removes multiple map features from the selection and updates the display.

This method replaces the removeObjects method in earlier versions of the Autodesk MapGuide Viewer API and supports the Autodesk MapGuide Viewer Plug-In, Java Edition, and ActiveX Control with the same syntax. The older method will still work in Release 6, but you should use removeObjectsEx for all future development.

Note that this method does not work during a busy state. For additional information about the busy state, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

Parameters

objects - A collection of the map features to remove from the selection.

notify - If true, fires the onSelectionChanged event after the removal.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

removeObject

Description

Gets the number of map features in the current selection.

Note that this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

int (read-only)

Error Codes

-1 (busy)

See Also

getNumObjects

MGSymbolAttr Object

The MGSymbolAttr object contains the point attributes used to draw symbols on redline layers.

Currently, the API can only access symbols in the API symbol list. The API symbol list is an MWF or EMF file containing a small set of predefined symbols. You can use the Autodesk MapGuide user interface to create API symbols for redlining and printing. For more information, refer to "Adding Symbols for Use with the Autodesk MapGuide Viewer API" and "API Symbol Manager dialog box" in the *MapGuide Help*.

You cannot currently create redline features (except for symbols) with the Autodesk MapGuide user interface. You can create these features only through the API. To get an instance of the MGSymbolAttr object for use as a redline feature, first use the MGMap.getRedlineSetup method to get an instance of the MGRedlineSetup object. Then use the MGRedlineSetup.getSymbolAttr method.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGSymbolAttr extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGSymbolAttr is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getHeight	Gets the display height of the point in the specified unit type.
getRotation	Gets the rotation of the point, in degrees.
getSymbol	Gets the symbol to be used as the point attribute.
getWidth	Gets the display width of the point in the specified unit type.
setHeight	Sets the display height of the point in the specified unit type.
setRotation	CHANGED Sets the rotation of the point.

setSymbol CHANGED Sets the symbol to be used as the symbol attribute.

setWidth Sets the display width of the point in the specified unit type.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

Height The display height of the point in the specified unit type.

Rotation CHANGED The rotation of the point, in degrees.

Symbol CHANGED The symbol to be used as the point attribute.

Width The display width of the point in the specified unit type.

MGSymbolAttr.getHeight

Syntax

```
double getHeight(String units)
```

Description

Gets the display height of the point in the specified unit type.

Parameters

units - The display size unit type, which can be CM, M, KM, IN, FT, or MI.

Returns

double - The display height.

Error Codes

-3 (illegal argument)

See Also

setHeight, Height

MGSymbolAttr.getRotation

Syntax

```
double getRotation()
```

Description

Gets the rotation of the point, in degrees.

Returns

double - The rotation, in degrees.

Error Codes

none

See Also

setRotation, Rotation

MGSymbolAttr.getSymbol

Syntax

```
String getSymbol()
```

Description

Gets the symbol to be used as the point attribute.

Currently, the API can only access symbols in the API symbol list. The API symbol list is an MWF or EMF file containing a small set of predefined symbols. You can use the Autodesk MapGuide user interface to create additional API symbols. For more information, refer to "Adding Symbols for Use with the Autodesk MapGuide Viewer API" and "API Symbol Manager dialog box" in the *MapGuide Help*.

Returns

String - The symbol name.

Error Codes

-1 (busy)

See Also

setSymbol, Symbol

MGSymbolAttr.getWidth

Syntax

```
double getWidth(String units)
```

Description

Gets the display width of the point in the specified unit type.

Parameters

units - The display size unit type, which can be CM, M, KM, IN, FT, or MI.

Returns

double - The display width.

Error Codes

-3 (illegal argument)

See Also

setWidth, Width

MGSymbolAttr.setHeight

Syntax

```
boolean setHeight(double height, String units)
```

Description

Sets the display height of the point in the specified unit type.

Parameters

height - The display height

units - The display size unit type, which can be CM, M, KM, IN, FT, or MI.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getHeight, Height

MGSymbolAttr.setRotation

Syntax

```
boolean setRotation(double rotation)
```

Description

Sets the rotation of the point.

Parameters

rotation - The rotation of the point in degrees; values can range from -360.0º to 360.0º.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-17 (can't rotate symbols with bitmaps): The rotation angle cannot be non-zero because the current symbol contains one or more bitmaps.

See Also

getRotation, Rotation

MGSymbolAttr.setSymbol

Syntax

```
boolean setSymbol(String symbol)
```

Description

Sets the symbol to be used as the symbol attribute.

If the specified symbol contains bitmaps, and the current rotation angle is non-zero, the symbol will be set, and the rotation angle will be set to zero.

Currently, the API can only access symbols in the API symbol list. The API symbol list is an MWF or EMF file containing a small set of predefined symbols. You can use the Autodesk MapGuide user interface to create additional API symbols. For more information, refer to "Adding Symbols for Use with the Autodesk MapGuide Viewer API" and "API Symbol Manager dialog box" in the *MapGuide Help*.

Parameters

symbol - The symbol name.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

-17 (can't rotate symbols with bitmaps): This return code is acts as a warning rather than an error.

See Also

getSymbol, Symbol

MGSymbolAttr.setWidth

Syntax

`boolean setWidth(double width, String units)`

Description

Sets the display width of the point in the specified unit type.

Parameters

height - The display width.

units - The display size unit type, which can be CM, M, KM, IN, FT, or MI.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getWidth, Width

MGSymbolAttr.Height

Description

Gets or sets the display height of the point in the specified unit type.

Type

double (read/write)

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getHeight, setHeight

MGSymbolAttr.Rotation

Description

Gets or sets the rotation of the point, in degrees.

Type

double (read/write)

Error Codes

-1 (busy)

-3 (illegal argument)

-17 (can't rotate symbols with bitmaps): The rotation angle cannot be non-zero because the current symbol contains one or more bitmaps.

See Also

getRotation, setRotation

MGSymbolAttr.Symbol

Description

Gets or sets the symbol to be used as the point attribute.

If the specified symbol contains bitmaps, and the current rotation angle is non-zero, the symbol will be set, and the rotation angle will be set to zero.

Currently, the API can only access symbols in the API symbol list. The API symbol list is an MWF or EMF file containing a small set of predefined symbols. You can use the Autodesk MapGuide user interface to create additional API symbols. For more information, refer to "Adding Symbols for Use with the Autodesk MapGuide Viewer API" and "API Symbol Manager dialog box" in the *MapGuide Help*.

Note that for write operations, this property does not work during a busy state. For additional information, refer to "Handling Busy State and Map Refresh" chapter in the *Autodesk MapGuide Developer's Guide*.

Type

String (read/write)

Error Codes

-1 (busy): For write operations only.

-3 (illegal argument)

-17 (can't rotate symbols with bitmaps): This return code is acts as a warning rather than an error.

See Also

getSymbol, setSymbol

MGSymbolAttr.Width

Description

Gets or sets the display width of the point in the specified unit type.

Type

double (read/write)

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getWidth, setWidth

MGTextAttr Object

The MGTextAttr object contains the attributes used to draw text on redline layers.

Redline features can't currently be created through the Autodesk MapGuide user interface, only through the API. To get an instance of the MGTextAttr object for use as a redline feature, first use the MGMap.getRedlineSetup method to get an instance of the MGRedlineSetup object. Then use the MGRedlineSetup.getTextAttr method.

Java Development for Autodesk MapGuide Viewer Plug-In: For Autodesk MapGuide Viewer Plug-In, the Java class MGTextAttr extends java.lang.Object and can be found in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Java Development for Autodesk MapGuide Viewer, Java Edition: For Autodesk MapGuide Viewer, Java Edition, MGTextAttr is a Java interface that resides in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

Methods

getBackColor	Gets the background color.
getBackMode	Gets the display mode for the background behind text features.
getColor	Gets the text color.
getFontName	Gets the font name.
getFontNames	Gets a collection containing the available font names.
getFontStyle	Gets the font style.
getFontStyles	Gets a collection containing the font styles for the specified font. Styles vary by font family, but are typically values like Regular, Bold, and Italic.
getHeight	Gets the display height of the text in the specified unit type.
getHorizAlign	Gets the horizontal placement of the text features in relation to an imaginary vertical axis.

getRotation	Gets the rotation of the text, in degrees.
getVertAlign	Gets the vertical placement of the text features in relation to an imaginary horizontal baseline.
setBackColor	Sets the background color.
setBackMode	Sets the display mode for the background behind text features.
setColor	Sets the text color.
setFontName	Sets the font name.
setFontStyle	Sets the font style.
setHeight	Sets the display height of the text in the specified unit type.
setHorizAlign	Sets the horizontal placement of the text features in relation to an imaginary vertical axis.
setRotation	Sets the rotation of the text.
setVertAlign	Sets the vertical placement of the text features in relation to an imaginary horizontal baseline.

Properties

Note that direct access to properties is only available in Autodesk MapGuide Viewer ActiveX Control. In Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer, Java Edition APIs, get/set methods provide access to an object's properties. For compatibility, the ActiveX Control API supports the same get/set methods as the Autodesk MapGuide Viewer Plug-In and the Autodesk MapGuide Viewer, Java Edition APIs.

BackColor	Gets or sets the background color.
BackMode	The display mode for the background behind text features.
Color	Gets or sets the color index used to draw the text.
FontName	The font name. Font names are not case-sensitive.
FontNames	A collection containing the available font names.
FontStyle	The font style. Names of font styles are not case-sensitive.
HorizAlign	Sets the horizontal placement of the text features in relation to an imaginary vertical axis.
Rotation	The rotation of the text, in degrees.
VertAlign	Sets the vertical placement of the text features in relation to an imaginary vertical baseline.

MGTextAttr.getBackColor

Syntax

```
int getBackColor()
```

Description

Gets the background color.

Returns

int - The color index, based on the color controls in Autodesk MapGuide Author; values range from 1 to 256.

Error Codes

none

See Also

setBackColor, BackColor

MGTextAttr.getBackMode

Syntax

```
int getBackMode()
```

Description

Gets the display mode for the background behind text features.

Returns

int - The background mode, which will be one of the following values: 0 (transparent), 1 (opaque), or 2 (ghosted).

Error Codes

none

See Also

setBackMode, BackMode

MGTextAttr.getColor

Syntax

```
int getColor()
```

Description

Gets the text color.

Returns

int - The color index, based on the color controls in Autodesk MapGuide Author; values range from 1 to 256.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

setColor, Color

MGTextAttr.getFontName

Syntax

```
String getFontName()
```

Description

Gets the font name.

Returns

String - The font name.

Error Codes

none

See Also

setFontName, FontName

MGTextAttr.getFontNames

Syntax

```
MGCollection getFontNames()
```

Description

Gets a collection containing the available font names.

Returns

MGCollection - The available fonts.

Error Codes

none

See Also

FontNames

MGTextAttr.getFontStyle

Syntax

```
String getFontStyle()
```

Description

Gets the font style.

Returns

String - The font style, typically a value like Regular, Bold, or Italic.

Error Codes

none

See Also

getFontStyles, setFontStyle, FontStyle

MGTextAttr.getFontStyles

Syntax

```
MGCollection getFontStyles(String font)
```

Description

Gets a collection containing the font styles for the specified font. Styles vary by font family, but are typically values like Regular, Bold, and Italic.

Parameters

font - The font name.

Returns

MGCollection - An MGCollection containing the available font styles.

Error Codes

none

See Also

getFontStyle, setFontStyle

MGTextAttr.getHeight

Syntax

```
double getHeight(String units)
```

Description

Gets the display height of the text in the specified unit type.

Parameters

units - The display size unit type, which can be CM, M, KM, IN, FT, or MI.

Returns

double - The display height.

Error Codes

-3 (illegal argument)

See Also

setHeight

MGTextAttr.getHorizAlign

Syntax

```
int getHorizAlign()
```

Description

Gets the horizontal placement of the text features in relation to an imaginary vertical axis.

Returns

int - The horizontal alignment, which will be one of the following: 0 (left), 1 (center), or 2 (right).

Error Codes

none

See Also

setHorizAlign, HorizAlign

MGTextAttr.getRotation

Syntax

```
double getRotation()
```

Description

Gets the rotation of the text, in degrees.

Returns

double - The rotation, in degrees.

Error Codes

none

See Also

setRotation, Rotation

MGTextAttr.getVertAlign

Syntax

```
int getVertAlign()
```

Description

Gets the vertical placement of the text features in relation to an imaginary horizontal baseline.

Returns

int - The vertical alignment, which will be one of the following: 0 (bottom), 1 (baseline), 2 (halfline), 3 (capline), or 4 (top).

Error Codes

none

See Also

setVertAlign, VertAlign

MGTextAttr.setBackColor

Syntax

```
boolean setBackColor(int color)
```

Description

Sets the background color.

Parameters

int - The color index, based on the color controls in Autodesk MapGuide Author; values range from 1 to 256.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getBackColor, BackColor

MGTextAttr.setBackMode

Syntax

```
boolean setBackMode(int mode)
```

Description

Sets the display mode for the background behind text features.

Parameters

mode - The background mode, which will be one of the following values: 0 (transparent), 1 (opaque), or 2 (ghosted).

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getBackMode, BackMode

MGTextAttr.setColor

Syntax

```
void setColor(int color)
```

Description

Sets the text color.

Parameters

int - The color index, based on the color controls in Autodesk MapGuide Author; values range from 1 to 256.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getColor, Color

MGTextAttr.setFontName

Syntax

`boolean setFontName(String font)`

Description

Sets the font name.

Font names are not case-sensitive.

Parameters

font - The font name.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It shows a print event handler, written in JavaScript, that prints a left-aligned map title in 14pt Verdana:

```
function onBeginLayout(pageSetup, info)
{
    if (pageSetup.getInclude("mg_title"))
    {
        var attr = pageSetup.getTitleTextAttr();
        attr.setFontName("Verdana");
        attr.setHeight(14.0, "");
        attr.setHorizAlign(0);
    }
}
```

See Also

getFontName, FontName

MGTextAttr.setFontStyle

Syntax

`boolean setFontStyle(String style)`

Description

Sets the font style.

Names of font styles are not case-sensitive.

Parameters

style - The font style, typically a value like Regular, Bold, or Italic.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getFontStyle, FontStyle

MGTextAttr.setHeight

Syntax

```
boolean setHeight(double height, String units)
```

Description

Sets the display height of the text in the specified unit type.

Parameters

height - The display height.

units - The display size unit type, which can be CM, M, KM, IN, FT, or MI.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It shows a print event handler, written in JavaScript, that prints a left-aligned map title in 14pt Verdana:

```
function onBeginLayout(pageSetup, info)
```

```
{
    if (pageSetup.getInclude("mg_title"))
    {
        var attr = pageSetup.getTitleTextAttr();
        attr.setFontName("Verdana");
        attr.setHeight(14.0, "");
        attr.setHorizAlign(0);
    }
}
```

See Also

getHeight

MGTextAttr.setHorizAlign

Syntax

`boolean setHorizAlign(int alignment)`

Description

Sets the horizontal placement of the text features in relation to an imaginary vertical axis.

Parameters

alignment - The horizontal alignment, which can be one of the following: 0 (left), 1 (center), or 2 (right).

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

Example

The following example is designed to work with the *Tutorial.mwf* file that is included with Autodesk MapGuide Author. It shows a print event handler, written in JavaScript, that prints a left-aligned map title in 14pt Verdana:

```
function onBeginLayout(pageSetup, info)
{
    if (pageSetup.getInclude("mg_title"))
    {
        var attr = pageSetup.getTitleTextAttr();
        attr.setFontName("Verdana");
        attr.setHeight(14.0, "");
        attr.setHorizAlign(0);
    }
}
```

See Also

getHorizAlign, HorizAlign

MGTextAttr.setRotation

Syntax

```
boolean setRotation(double rotation)
```

Description

Sets the rotation of the text.

Parameters

rotation - The rotation of the text in degrees; values can range from -360.0º to 360.0º.

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getRotation, Rotation

MGTextAttr.setVertAlign

Syntax

```
boolean setVertAlign(int alignment)
```

Description

Sets the vertical placement of the text features in relation to an imaginary horizontal baseline.

Parameters

alignment - The horizontal alignment, which can be one of the following: 0 (bottom), 1 (baseline), 2 (halfline), 3 (capline), or 4 (top).

Returns

boolean - Returns True if successful; otherwise, returns False.

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getVertAlign, VertAlign

MGTextAttr.BackColor

Description

Gets or sets the background color.

Type

int (read/write)

Error Codes

-3 (illegal argument)

See Also

getBackColor, setBackColor

MGTextAttr.BackMode

Description

Gets or sets the display mode for the background behind text features.

BackMode supports the following values: 0 (transparent), 1 (opaque), and 2 (ghosted).

Type

int (read/write)

Error Codes

-3 (illegal argument)

See Also

getBackMode, setBackMode

MGTextAttr.Color

Description

Gets or sets the color index used to draw the text.

The color index is based on the color controls in Autodesk MapGuide Author; values range from 1 to 256.

Type

int (read/write)

Error Codes

-3 (illegal argument)

See Also

getColor, setColor

MGTextAttr.FontName

Description

Gets or sets the font name. Font names are not case-sensitive.

Type

String (read/write)

Error Codes

-3 (illegal argument)

See Also

getFontname, setFontName

MGTextAttr.FontNames

Description

Gets or sets a collection containing the available font names.

Type

MGCollection (read/write)

Error Codes

none

See Also

getFontNames

MGTextAttr.FontStyle

Description

Gets or sets the font style. Names of font styles are not case-sensitive.

Type

String (read/write)

Error Codes

-3 (illegal argument)

See Also

getFontStyle, setFontStyle

MGTextAttr.HorizAlign

Description

Sets the horizontal placement of the text features in relation to an imaginary vertical axis.

HorizAlign supports the following values: 0 (left), 1 (center), and 2 (right).

Type

int (read/write)

Error Codes

-3 (illegal argument)

See Also

getHorizAlign, setHorizAlign

MGTextAttr.Rotation

Description

TGets or sets the rotation of the text, in degrees.

Type

double (read/write)

Error Codes

-1 (busy)

-3 (illegal argument)

See Also

getRotation, setRotation

MGTextAttr.VertAlign

Description

Sets the vertical placement of the text features in relation to an imaginary vertical baseline.

VertAlign supports the following values: 0 (bottom), 1 (baseline), 2 (halfline), 3 (capline), and 4 (top).

Type

int (read/write)

Error Codes

-3 (illegal argument)

See Also

getVertAlign, setVertAlign

Events

The Autodesk MapGuide Viewer API fires a number of events that can be used to determine when the selection changes, when the map is redrawn, or when map features are double-clicked. In addition, events are used upon completion of certain API requests. For additional information about events, observers, and setting up event handler interfaces, refer to "Handling Busy State and Map Refresh" in the *Autodesk MapGuide Developer's Guide*.

onBusyStateChanged, onMapLoaded, onViewChanged, and onViewChanging might not work properly under Netscape 4.x when the map is first loaded. This is because the observer applet loads on a different thread from the map, so the map might load before the observer applet has finished loading. These events work correctly after the map and the observer applet have both finished fully loading. The workaround to this limitation is to use the ActiveX Control, or to avoid relying on these events when the map first loads.

ActiveX Control	Plug-In and Java Edition	Description
onAddMapLayer	onAddMapLayer	This event is fired when a new map layer has been added.
onBeginLayout	onBeginLayout	This event is fired just before the Autodesk MapGuide Viewer uses the internal algorithm to lay out the page elements.
onBusyStateChanged	onBusyStateChanged	This event is fired when Autodesk MapGuide Viewer enters and exits busy states.
onDigitizedCircle	onDigitizedCircle	This event is fired when the user has finished digitizing a circle.
onDigitizedPoint	onDigitizedPoint	This event is fired when the user has finished digitizing a point.
onDigitizedPolygon	onDigitizedPolygon	This event is fired when the user has finished digitizing a polygon.
onDigitizedPolyline	onDigitizedPolyline	This event is fired when the user has finished digitizing a polyline.
onDigitizedRectangle	onDigitizedRectangle	This event is fired when the user has finished digitizing a rectangle.
onDoubleClickObject	onDoubleClickObject	This event is fired when the user double-clicks on a map feature.
onEndLayout	onEndLayout	This event is fired after the Autodesk MapGuide Viewer has finished the internal layout of the page elements, but before the map is sent to the printer.

onMapLoaded	onMapLoaded	This event is fired when Autodesk MapGuide loads an MWF into the Autodesk MapGuide Viewer, but before the first refresh.
onSelectionChanged	onSelectionChanged	This event is fired when the map selection has changed.
onViewChanged	onViewChanged	This event is fired after the view of the map has been updated or refreshed.
onViewChanging	onViewChanging	This event is fired just before each map refresh.
onViewedDistance	onViewedDistanceEx	This event is fired when a user had finished viewing distances via the viewDistance method or the user interface menu commands.
onViewedDistanceEx	onViewedDistance	This event is fired when a user has finished viewing distances via the viewDistanceEx method.

onAddMapLayer

Syntax

`onAddMapLayer` *`URL`*, *`layer`*

Description

This event is fired when a new map layer has been added. It is initiated by a call to addMapLayer.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGAddMapLayerObserver.onAddMapLayer.

Applies To

ActiveX Control only

Parameters

URL - A string that will contain the URL of the map layer file as passed to the addMapLayer method. This can be used to differentiate between multiple onAddMapLayer events if several layers are added to the map in succession.

layer - An MGMapLayer object for the new map layer. If unsuccessful, this parameter will be null.

Error Codes

none

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define VBScript and JavaScript functions like the following:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onAddMapLayer(URL, layer)
onAddMapLayer URL, layer
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onAddMapLayer (URL, layer)
{
// event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

See Also

addMapLayer

onBeginLayout

Syntax

```
onBeginLayout pgSetup, printInfo
```

Description

This event is fired right before the Autodesk MapGuide Viewer uses the internal algorithm to lay out the page elements.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGPrintingObserver.onBeginLayout.

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define the following VBScript and JavaScript functions:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onBeginLayout (pgSetup, printInfo)
onBeginLayout pgSetup, printInfo
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JAVASCRIPT">
function onBeginLayout (pgSetup, printInfo)
```

```
{
... Event Handling Code Here ...
}
</SCRIPT>
```

In the example above, *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" chapter in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

Applies To

ActiveX Control only

Parameters

pgSetup - An MGPageSetup object representing the current page layout settings.

info - An MGPrintInfo object containing information about the current printing operation.

Error Codes

none

See Also

onEndLayout, enablePrintingEvents, disablePrintingEvents, getPrintingObserver, setPrintingObserver

onBusyStateChanged

Syntax

```
onBusyStateChanged map, busyState
```

Description

This event is fired when the Autodesk MapGuide Viewer enters and exits busy states.

Autodesk MapGuide enters busy states when users pan or zoom to view more map data.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGBusyStateChangedObserver.onBusyStateChanged.

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define the following VBScript and JavaScript functions:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onBusyStateChanged(map, busyState)
onBusyStateChanged map, busyState
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onBusyStateChanged(map, busyState)
```

```
{
// event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

Parameters

map - The map object that this event applies to.

busyState - The current busy state of Autodesk MapGuide Viewer.

Error Codes

none

See Also

getBusyStateChangedObserver, setBusyStateChangedObserver

onDigitizedCircle

Syntax

`onDigitizedCircle` *`map, units, center, radius`*

Description

This event is fired when the user has finished digitizing a circle.

It is initiated by a call to digitizeCircle.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGDigitizeCircleObserver.onDigitizedCircle.

Applies To

ActiveX Control only

Parameters

map - The map object for which the event occurred.

units - A string containing the units in which the radius is provided, one of the following codes: M (meters), KM (kilometers), FT (feet), or MI (miles).

center - An MGPoint object for the longitude/latitude center of the digitized circle.

radius - The radius of the digitized circle (a String).

Error Codes

-1 (busy)

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define the following VBScript and JavaScript functions:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onDigitizedCircle(map, units, center, radius)
onDigitizedCircle map, units, center, radius
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onDigitizedCircle(map, units, center, radius)
{
 // event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

Warning: Because the radius parameter provided by onDigitizedCircle is of type String, you may need to explicitly convert its type to float before passing it to other methods within the event handling code, or before performing mathematical manipulations to it. For example, you must convert its type to float before passing it to the MGMapObject.addCirclePrimitive method as follows:

```
<SCRIPT LANGUAGE="JavaScript">
function onDigitizedCircle(map, units, center, radius)
{
   ...
   radius = parseFloat(radius);
   obj.addCirclePrimitive(center, useMCS, radius, units, numLines);
   ...
}
</SCRIPT>
```

See Also

digitizeCircle

onDigitizedPoint

Syntax

onDigitizedPoint *map*, *point*

Description

This event is fired when the user has finished digitizing a point. It is initiated by a call to digitizePoint.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGDigitizePointObserver.onDigitizedPoint.

Applies To

ActiveX Control only

Parameters

map - The map object for which the event occurred.

point - The MGPoint object for the digitized longitude/latitude coordinate pair.

Error Codes

none

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define the following VBScript and JavaScript functions:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onDigitizedPoint(map, point)
onDigitizedPoint map, point
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onDigitizedPoint(map, point)
{
// event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

See Also

digitizePoint

onDigitizedPolygon

Syntax

```
onDigitizedPolygon map, numPoints, latlon
```

Description

This event is fired when the user has finished digitizing a polygon. It is initiated by a call to digitizePolygonEx.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGDigitizePolygonObserverEx.onDigitizedPolygon.

Applies To

ActiveX Control only

Parameters

map - Specifies the MGMap instance.

numPoints - The number of vertices of the polygon.

latlon - An MGCollection object that stores a list of MGPoint objects, each of which represents a longitude-latitude pair.

Error Codes

none

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define the following VBScript and JavaScript functions:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onDigitizedPolygon(map, numPoints, points)
onDigitizedPolygon map, numPoints, points
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onDigitizedPolygon(map, numPoints, points)
{
// event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

See Also

digitizePolygonEx

onDigitizedPolyline

Syntax

```
onDigitizedPolyline map, numPoints, latlon
```

Description

This event is fired when the user has finished digitizing a polyline. It is initiated by a call to digitizePolylineEx.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGDigitizePolylineObserverEx.onDigitizedPolyline.

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define the following VBScript and JavaScript functions:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onDigitizedPolyline(map, numPoints, latlon)
onDigitizedPolyline map, numPoints, latlon
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onDigitizedPolyline(map, numPoints, latlon)
{
 // event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

Parameters

map - Specifies the MGMap instance.

numPoints - The number of vertices of the polyline.

latlon - An MGCollection object that stores a list of MGPoint objects, each of which represents a vertex of the polyline.

Error Codes

none

See Also

digitizePolylineEx

onDigitizedRectangle

Syntax

`onDigitizedRectangle map, anchorPt, endPt`

Description

This event is fired when the user has finished digitizing a rectangle. It is initiated by a call to digitizeRectangle.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGDigitizeRectangleObserver.onDigitizedRectangle.

Applies To

ActiveX Control only

Parameters

map - The map object for which the event occurred.

anchorPt - The MGPoint object containing the longitude and latitude of the digitized rectangle's starting (anchor) point.

endPt - The MGPoint object containing the longitude and latitude of the digitized rectangle's ending point.

Error Codes

none

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define the following VBScript and JavaScript functions:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onDigitizedRectangle(map, anchorPt, endPt)
onDigitizedRectangle map, anchorPt, endPt
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onDigitizedRectangle(map, anchorPt, endPt)
{
// event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

See Also

digitizeRectangle

onDoubleClickObject

Syntax

`onDoubleClickObject(mapobject)`

Description

This event is fired when the user double-clicks on a map feature.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGDoubleClickObjectObserver.onDoubleClickObject.

Applies To

ActiveX Control only

Parameters

obj - The MGMapObject map feature that was double-clicked.

Error Codes

none

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define the following VBScript and JavaScript functions:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onDoubleClickObject(mapobject)
mapObject.DoubleClickHandled = onDoubleClickObject(mapobject)
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onDoubleClickObject(mapobject)
{
// event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

Note that you can use the MGMapObject property DoubleClickHandled in the event handler to prevent Autodesk MapGuide from executing the default double-click behavior.

See Also

getDoubleClickObserver, setDoubleClickObserver, DoubleClickHandled

onEndLayout

Syntax

```
onEndLayout layout, printInfo
```

Description

This event is fired after the Autodesk MapGuide Viewer has finished the internal layout of the page elements but before the map is sent to the printer.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGPrintingObserver.onEndLayout.

Applies To

ActiveX Control only

Parameters

layout - An MGPrintLayout object that represents the current page layout.

info - An MGPrintInfo object that contains information about the current printing operation.

Error Codes

none

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define the following VBScript and JavaScript functions:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onEndLayout (layout, printInfo)
onEndLayout layout, printInfo
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JAVASCRIPT">
function onEndLayout (layout, printInfo)
{
// event handling code here...
}
</SCRIPT>
```

In the example above, *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" chapter in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

See Also

onBeginLayout, enablePrintingEvents, disablePrintingEvents, getPrintingObserver, setPrintingObserver

onMapLoaded

Syntax

`onMapLoaded map`

Description

This event is fired when Autodesk MapGuide loads an MWF into the Autodesk MapGuide Viewer, before the first refresh.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGMapLoadedObserver.onMapLoaded.

A typical use of onMapLoaded is to set certain MWF properties, such as a SQL where clause, based on external variables, for example, the user name. If you want to change settings every time the map is about to be refreshed, use the onViewChanging event instead.

Note that Autodesk MapGuide disables the autoRefresh flag, set by setAutoRefresh, when it executes the onMapLoaded event. Therefore, calling zoomGotoDlg, zoomGotoLocation, setUrl, or refresh in onMapLoaded has no effect. Additionally, setAutoRefresh has no effect when called from onMapLoaded.

Applies To

ActiveX Control only

Parameters

map - The MGMap map object that this event applies to.

Error Codes

none

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define VBScript and JavaScript functions like the following:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onMapLoaded(map)
onMapLoaded map
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onMapLoaded(map)
{
// event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

onSelectionChanged

Syntax

`onSelectionChanged map`

Description

This event is fired when the map selection has changed.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGSelectionChangedObserver.onSelectionChanged.

Applies To

ActiveX Control only

Parameters

map - The map object for which the event occurred.

Error Codes

none

Note

If your application has *notify* (the second argument of addObject) set to True, writing code in the onSelectionChanged event that changes the selection will result in an infinite loop. For example:

```
var mapSel = map.getSelection();
var obj = layer.getMapObject('CA');
mapSel.addObject(obj, true);
```

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define VBScript and JavaScript functions like the following:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onSelectionChanged(map)
onSelectionChanged map
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onSelectionChanged(map)
{
// event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

See Also

getSelectionChangedObserver, setSelectionChangedObserver

onViewChanged

Syntax

```
onViewChanged map
```

Description

This event is fired after the view of the map has been updated or refreshed.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGViewChangedObserver.onViewChanged.

Applies To

ActiveX Control only

Parameters

map - The map object for which the event occurred.

Error Codes

none

Note

Writing code in the onViewChanged event that changes the map view, such as map.refresh(); results in an infinite loop.

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define VBScript and JavaScript functions like the following:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onViewChanged(map)
onViewChanged map
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onViewChanged(map)
{
// event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

onViewChanging

Syntax

`onViewChanging map`

Description

This event is fired just before the map is refreshed.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGViewChangingObserver.onViewChanging.

There are many useful actions you can execute with onViewChanging, for instance, customizing layer settings such as visibility, the SQL Where clause, or the data file to which a layer refers. If you just want to initialize layer settings the first time that Autodesk MapGuide Viewer displays a map, as opposed to every time that the map is refreshed, use the onMapLoaded event instead.

Note that Autodesk MapGuide disables the autoRefresh flag, set by setAutoRefresh, when it executes the onViewChanging event. Therefore, calling zoomGotoDlg, zoomGotoLocation, setUrl, or refresh in onViewChanging has no effect. Additionally, setAutoRefresh has no effect when called from within onViewChanging.

Applies To

ActiveX Control only

Parameters

map - The map object for which the event occurred.

Error Codes

none

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define VBScript and JavaScript functions like the following:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onViewChanging(map)
onViewChanging map
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onViewChanging(map)
{
// event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

See Also

getViewChangingObserver, setViewChangingObserver

onViewedDistance

Syntax

`onViewedDistance map, totalDistance, distances, units`

Description

This event is fired when a user has finished viewing distances.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGViewDistanceObserver.onViewedDistance.

Note that the event is fired *only* if the distances were measured using the viewDistance method or the UI menu commands. If the distances were measured using the viewDistanceEx method, onViewedDistanceEx is fired instead.

Applies To

ActiveX Control only

Parameters

map - The MGMap object for which the event occurred.

totalDistance - The total distance viewed.

distances - An MGCollection object containing all the sub-distances.

units - The units used to quantify the distance.

Error Codes

-1 (busy)

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define the following VBScript and JavaScript functions:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onViewedDistance(map, totalDistance, distances, units)
onViewedDistance map, totalDistance, distances, units
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onViewedDistance(map, totalDistance, distances, units)
{
 // event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

See Also

viewDistance

onViewedDistanceEx

Syntax

```
onViewedDistanceEx map, totalDistance, distances, units, useMcs
```

Description

This event is fired when a user has finished viewing distances.

This topic describes this event for ActiveX Control. For Plug-In and Java Edition, see MGViewDistanceExObserver.onViewedDistanceEx.

Note that the event is fired *only* if the distances were measured using the viewDistanceEx method. If the distances were measured using the viewDistance method (or the user interface menu commands), onViewedDistance is fired instead.

Applies To

ActiveX Control only

Parameters

map - The MGMap object for which the event occurred

totalDistance - The total distance viewed.

distances - An MGCollection object containing type-double objects representing all the sub-distances.

points - An MGCollection containing MGPoint objects representing all the points used in the view distance action; the points will be in MCS or lat/lon coordinates, based on the setting of *useMcs*.

units - The units used to quantify the distance.

useMcs - A flag to indicate whether the points are in MCS or lat/lon.

Error Codes

none

Example

To handle this event from the Autodesk MapGuide ActiveX Control, define the following VBScript and JavaScript functions:

```
<!--
The VBScript function handles the event from the ActiveX Control
Control and then passes it to the JavaScript function.
-->
<SCRIPT LANGUAGE="VBScript">
Sub ObjectId_onViewedDistanceEx(map, totalDistance, distances, units, useMcs)
onViewedDistanceEx map, totalDistance, distances, units, useMcs
End Sub
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
function onViewedDistanceEx(map, totalDistance, distances, units, useMcs)
{
 // event handling code here...
}
</SCRIPT>
```

Where *ObjectId* is the ID of the embedded map specified as part of the HTML OBJECT tag. Refer to "Displaying Maps" in the *Autodesk MapGuide Developer's Guide* for details on how to embed the ActiveX Control using the OBJECT tag.

See Also

viewDistanceEx

Observers and Event Methods

This section describes the observer objects provided by Autodesk MapGuide. These observers are available for the Autodesk MapGuide Viewer Plug-In and Autodesk MapGuide Viewer,Java Edition. Observers for the Autodesk MapGuide Viewer, Java Edition are provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). Observers for the Autodesk MapGuide Viewer Plug-In are provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

MGAddMapLayerObserver	Implements the map layer-addition observer method onAddMapLayer. This method is called by when addMapLayer has completed its operation.
MGBusyStateChangedObserver	Implements the busy state observer event onBusyStateChanged. The Autodesk MapGuide Plug-In calls onBusyStateChanged when the user activates a change in the busy state, usually by panning or zooming on a map.
MGDigitizeCircleObserver	Implements the circle-digitizing observer method onDigitizedCircle. This method is called by when digitizeCircle in MGMap has completed its operation.
MGDigitizePointObserver	Implements the point-digitizing observer method onDigitizedPoint. This method is called by when digitizePoint in MGMap has completed its operation.
MGDigitizePolygonObserver	Replaced: Use MGDigitizePolygonObserverEx instead. Implements the polygon-digitizing observer method onDigitizedPolygon.
MGDigitizePolygonObserverEx	Implements the polygon-digitizing observer method onDigitizedPolygon. This method is called by and Java Editions when digitizePolygonEx in MGMap has completed its operation. MGDigitizePolygonObserverEx replaces MGDigitizePolygonObserver.
MGDigitizePolylineObserver	Replaced: Use MGDigitizePolylineObserverEx instead. Implements the polyline-digitizing observer method onDigitizedPolyline.
MGDigitizePolylineObserverEx	Implements the polyline-digitizing observer method onDigitizedPolyline. This method is called by when digitizePolyline in MGMap has completed its operation. MGDigitizePolylineObserverEx replaces MGDigitizePolylineObserver.
MGDigitizeRectangleObserver	Implements the rectangle-digitizing observer method onDigitizedRectangle. This method is called by when digitizeRectangle in MGMap has completed its operation.
MGDoubleClickObjectObserver	Implements the double-click observer method onDoubleClickObject. This method is called by when the user has double-clicked on a map feature.
MGMapLoadedObserver	Implements the map loaded observer method onMapLoaded. This method is called by when it loads an MWF file into the Autodesk MapGuide Viewer, prior to the first refresh. The Autodesk MapGuide Viewer, Java Edition provides this interface in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms).

MGPrintingObserver	Implements the printing observer methods onBeginLayout and onEndLayout.
MGSelectionChangedObserver	Implements the changed selection observer method onSelectionChanged. This method is called by when the selection of map features on the map is changed.
MGViewChangedObserver	Implements the changed view observer method onViewChanged. This method is called by when the map view has been updated or refreshed.
MGViewChangingObserver	Implements the changing view observer method onViewChanging. This method is called by before each refresh.
MGViewDistanceExObserver	Implements the view distance observer method onViewedDistanceEx. This method is called by the viewDistanceEx method.
MGViewDistanceObserver	Implements the view distance observer method onViewedDistance. This method is called by the viewDistance method.

MGAddMapLayerObserver

Implements the map layer-addition observer event method onAddMapLayer.

MGAddMapLayerObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGAddMapLayerObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

addMapLayer	The map layer-addition observer method onAddMapLayer.

MGAddMapLayerObserver.onAddMapLayer

Syntax

```
public void onAddMapLayer(String URL, MGMapLayer layer)
```

Description

The layer-addition observer method. This event is fired when a new map layer has been added. It is initiated by a call to addMapLayer.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onAddMapLayer.

Applies To

Plug-In and Java Edition only

Parameters

URL - A string that will contain the URL of the map layer file as passed to the addMapLayer method. This can be used to differentiate between multiple onAddMapLayer events if several layers are added to the map in succession.

layer - An MGMapLayer object for the new map layer. If unsuccessful, this parameter will be null.

Error Codes

none

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGAddMapLayerObserver interface must be passed to the addMapLayer method. The observer Java object should include the following:

```
<class_definition> implements MGAddMapLayerObserver
{
// ...
public void onAddMapLayer(String URL, MGMapLayer layer)
{
// event handling code here...
}
// ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

addMapLayer

MGBusyStateChangedObserver

Implements the busy state observer event onBusyStateChanged.

MGBusyStateChangedObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGBusyStateChangedObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onBusyStateChanged The busy state observer event method.

MGBusyStateChangedObserver.onBusyStateChanged

Syntax

```
void onBusyStateChanged(MGMap map, boolean busyState)
```

Description

The busy state observer event method.

Autodesk MapGuide enters busy states when users pan or zoom to view more map data.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onBusyStateChanged.

Applies To

Plug-In and Java Edition only

Parameters

map - The map object that this event applies to.

busyState - The current busy state of Autodesk MapGuide Viewer.

Error Codes

none

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGBusyStateChangedObserver interface must be established with a call to the setBusyStateChanged method. The observer Java object should include the following:

```
<class_definition> implements MGBusyStateChangedObserver
{
        ...
public void onBusyStateChanged(MGMap map, boolean busyState)
{
// event handling code here...
}
        // ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

getBusyStateChangedObserver, setBusyStateChangedObserver

MGDigitizeCircleObserver

Implements the circle-digitizing observer method onDigitizedCircle.

MGDigitizeCircleObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGDigitizeCircleObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onDigitizedCircle The circle-digitizing observer event method.

MGDigitizeCircleObserver.onDigitizedCircle

Syntax

```
public abstract void onDigitizedCircle(MGMap map, String units,
MGPoint center, double radius)
```

Description

The circle-digitizing observer event method.

This event is fired when the user has finished digitizing a circle. It is initiated by a call to digitizeCircle.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onDigitizedCircle.

Applies To

Plug-In and Java Edition only

Parameters

map - The map object for which the event occurred.

units - A string containing the units in which the radius is provided, one of the following codes: M (meters), KM (kilometers), FT (feet), or MI (miles)

center - An MGPoint object for the longitude/latitude center of the digitized circle.

radius - The radius of the digitized circle.

Error Codes

-1 (busy)

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGDigitizeCircleObserver interface must be passed to the digitizeCircle method. The observer Java object should include the following:

```
<class_definition> implements MGDigitizeCircleObserver
{
// ...
public void onDigitizedCircle(MGMap map,
 String units,
 MGPoint center,
```

```
                                                        double radius)
{
// event handling code here...
}
// ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

digitizeCircle

MGDigitizePointObserver

Implements the point-digitizing observer method onDigitizedPoint.

MGDigitizePointObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGDigitizePointObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onDigitizedPoint	The point-digitizing observer method.

MGDigitizePointObserver.onDigitizedPoint

Syntax

```
public abstract void onDigitizedPoint(MGMap map, MGPoint point)
```

Description

The point-digitizing observer method. It is initiated by a call to digitizePoint.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onDigitizedPoint.

Applies To

Plug-In and Java Edition only

Parameters

map - The map object for which the event occurred.

point - The MGPoint object for the digitized longitude/latitude coordinate pair.

Error Codes

none

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGDigitizePointObserver interface must be passed to the digitizePoint method. The observer Java object should include the following:

```
<class_definition> implements MGDigitizePointObserver
{
        // ...
        public void onDigitizedPoint(MGMap map,
MGPoint point)
{
                // event handling code here...
        }
        // ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

digitizePoint

MGDigitizePolygonObserver

Implements the polygon-digitizing observer method onDigitizedPolygon.

MGDigitizePolygonObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGDigitizePolygonObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onDigitizedPolygon The polygon-digitizing observer method

MGDigitizePolygonObserver.onDigitizedPolygon

Syntax

```
public abstract void onDigitizedPolygon(MGMap map, int npts, Vector latlon)
```

Description

The polygon-digitizing observer method.

This method is called by when digitizePolygon in MGMap has completed its operation. MGDigitizePolygonObserver has been replaced by the interface MGDigitizePolygonObserverEx.

MGDigitizePolygonObserver will still work in Release 6, but you should use MGDigitizePolygonObserverEx for all future development.

Applies To

Plug-In and Java Edition only

Parameters

map - Specifies the MGMap instance.

npts - The number of vertices of the polygon.

latlon - A java.util.Vector object that stores a list of MGPoint objects, each of which represents a longitude-latitude pair.

MGDigitizePolygonObserverEx

Implements the polygon-digitizing observer method onDigitizedPolygon.

MGDigitizePolygonObserverEx for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGDigitizePolygonObserverEx is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onDigitizedPolygon	The polygon-digitizing observer method onDigitizedPolygon

MGDigitizePolygonObserverEx.onDigitizedPolygon

Syntax

```
public abstract void onDigitizedPolygon(MGMap map,int numPoints, MGCollection
latlon)
```

Description

The polygon-digitizing observer method.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onDigitizedPolygon.

This method is called by and Java Editions when digitizePolygonEx in MGMap has completed its operation. MGDigitizePolygonObserverEx replaces the interface MGDigitizePolygonObserver from earlier versions of the Autodesk MapGuide Viewer API. The older interface will still work in Release 6, but you should use MGDigitizePolygonObserverEx for all future development.

Applies To

Plug-In and Java Edition only

Parameters

map - Specifies the MGMap instance.

numPoints - The number of vertices of the polygon.

latlon - An MGCollection object that stores a list of MGPoint objects, each of which represents a longitude-latitude pair.

Error Codes

none

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGDigitizePolygonObserverEx interface must be passed to the digitizePolygonEx method. The observer Java object should include the following:

```
<class_definition> implements MGDigitizePolygonObserverEx
{
// ...
public void onDigitizedPolygon(MGMap map,
int numPoints
MGCollection points)
{
// event handling code here...
}
        // ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

digitizePolygonEx

MGDigitizePolylineObserver

Implements the polyline-digitizing observer method onDigitizedPolyline.

MGDigitizePolylineObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGDigitizePolylineObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onDigitizedPolyline	The polyline-digitizing observer method

MGDigitizePolylineObserver.onDigitizedPolyline

Syntax

```
public abstract void onDigitizedPolyline(MGMap map, int npts, Vector latlon)
```

Description

The polyline-digitizing observer method.

This method is called by when digitizePolyline in MGMap has completed its operation. MGDigitizePolylineObserver has been replaced by the interface MGDigitizePolylineObserverEx. The MGDigitizePolylineObserver interface will still work in Release 6, but you should use MGDigitizePolylineObserverEx for all future development.

Applies To

Plug-In and Java Edition only

Parameters

map - Specifies the MGMap instance.

npts - The number of vertices of the polyline.

latlon - A java.util.Vector object that stores a list of MGPoint objects, each of which represents a longitude-latitude pair.

MGDigitizePolylineObserverEx

Implements the polyline-digitizing observer method onDigitizedPolyline.

MGDigitizePolylineObserverEx for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGDigitizePolylineObserverEx is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onDigitizedPolyline	The polyline-digitizing observer method

MGDigitizePolylineObserverEx.onDigitizedPolyline

Syntax

```
public abstract void onDigitizedPolyline(MGMap map, int numPoints, MGCollection
latlon)
```

Description

The polyline-digitizing observer method.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onDigitizedPolyline.

This method is called by when digitizePolyline in MGMap has completed its operation. MGDigitizePolylineObserverEx replaces the interface MGDigitizePolylineObserver from earlier versions of the Autodesk MapGuide Viewer API. The older interface will still work in version 6.0, but you should use MGDigitizePolylineObserverEx for all future development.

Applies To

Plug-In and Java Edition only

Parameters

map - Specifies the MGMap instance.

numPoints - The number of vertices of the polyline.

latlon - An MGCollection object that stores a list of MGPoint objects, each of which represents a vertex of the polyline.

Error Codes

none

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGDigitizePolylineObserverEx interface must be passed to the digitizePolylineEx method. The observer Java object should include the following:

```
<class_definition> implements MGDigitizePolylineObserverEx
{
// ...
public void onDigitizedPolyline(MGMap map, int numPoints
    MGCollection latlon)
{
// event handling code here...
}
// ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

digitizePolylineEx

MGDigitizeRectangleObserver

Implements the rectangle-digitizing observer method onDigitizedRectangle.

MGDigitizeRectangleObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGDigitizeRectangleObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onDigitizedRectangle The rectangle-digitizing observer method.

MGDigitizeRectangleObserver.onDigitizedRectangle

Syntax

```
public abstract void onDigitizedRectangle(MGMap map, MGPoint anchorPt, MGPoint
endPt)
```

Description

The rectangle-digitizing observer event method. This event is fired when the user has finished digitizing a rectangle. It is initiated by a call to digitizeRectangle.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onDigitizedRectangle.

This method is called by when digitizeRectangle in MGMap has completed its operation.

Applies To

Plug-In and Java Edition only

Parameters

map - The map object for which the event occurred.

anchorPt - The MGPoint object containing the longitude and latitude of the digitized rectangle's starting (anchor) point.

endPt - The MGPoint object containing the longitude and latitude of the digitized rectangle's ending point.

Error Codes

none

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGDigitizeRectangleObserver interface must be passed to the digitizeRectangle method. The observer Java object should include the following:

```
<class_definition> implements MGDigitizeRectangleObserver
{
// ...
public void onDigitizedRectangle(MGMap map,
MGPoint anchorPt
MGPoint endPt)
{
// event handling code here...
}
// ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

digitizeRectangle

MGDoubleClickObjectObserver

Implements the double-click observer method onDoubleClickObject.

MGDoubleClickObjectObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGDoubleClickObjectObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onDoubleClickObject The double-click observer method

MGDoubleClickObjectObserver.onDoubleClickObject

Syntax

```
public abstract boolean onDoubleClickObject(MGMapObject obj)
```

Description

The double-click observer event method. This event is fired when the user double clicks on a map feature.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onDoubleClickObject.

Applies To

Plug-In and Java Edition only

Parameters

obj - The MGMapObject map feature that was double-clicked.

Returns

boolean - Returns True if the event is completely handled in this method (i.e., no default action is required); otherwise, returns False.

Error Codes

none

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGDoubleClickObjectObserver interface must be established with a call to the setDoubleClickObserver method. The observer Java object should include the following:

```
<class_definition> implements MGDoubleClickObjectObserver
{
// ...
public boolean onDoubleClickObject(MGMapObject obj)
{
// event handling code here...
// Return True if event is handled by this function;
// False for default handling by Viewer
}
// ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

getDoubleClickObserver, setDoubleClickObserver, DoubleClickHandled

MGMapLoadedObserver

Implements the map loaded observer method onMapLoaded.

MGMapLoadedObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGMapLoadedObserver is provided in the Java *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onMapLoaded	The map loaded observer method.

MGMapLoadedObserver.onMapLoaded

Syntax

```
public abstract void onMapLoaded(MGMap map)
```

Description

The map loaded observer event method. This event is fired when Autodesk MapGuide loads an MWF into the Autodesk MapGuide Viewer, before the first refresh.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onMapLoaded.

A typical use of onMapLoaded is to set certain MWF properties, such as a SQL where clause, based on external variables, for example, the user name. If you want to change settings every time the map is about to be refreshed, use the onViewChanging event instead.

Note that Autodesk MapGuide disables the autoRefresh flag, set by setAutoRefresh, when it executes the onMapLoaded event. Therefore, calling zoomGotoDlg, zoomGotoLocation, setUrl, or refresh in onMapLoaded has no effect. Additionally, setAutoRefresh has no effect when called from onMapLoaded.

Applies To

Plug-In and Java Edition only

Parameters

map - The MGMap map object that this event applies to.

Error Codes

none

Example

To handle this event, a Java object that implements the MGMapLoadedObserver interface must be established with a call to the setMapLoadedObserver method. The observer Java object should include the following:

```
<class_definition> implements MGMapLoadedObserver
{
// ...
public void onMapLoaded(MGMap map)
{
// event Handling code here, for example:
var mapLayer = map.getMapLayer("ODBCPointLayer");
maplayer.setSQLWhere("Name = " + userName);
}
// ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

MGPrintingObserver

Implements the printing observer methods onBeginLayout and onEndLayout.

MGPrintingObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGPrintingObserver is provided in the Java *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onBeginLayout	The begin layout printing observer method.

onEndLayout The end layout printing observer method.

MGPrintingObserver.onBeginLayout

Syntax

```
public void onBeginLayout (MGPageSetup pgSetup, MGPrintInfo info)
```

Description

The begin layout printing observer method. This event is fired right before the Autodesk MapGuide Viewer uses the internal algorithm to lay out the page elements.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onBeginLayout.

Applies To

Plug-In and Java Edition only

Parameters

pgSetup - An MGPageSetup object representing the current page layout settings.

info - An MGPrintInfo object containing information about the current printing operation.

Error Codes

none

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGPrintingObserver interface must be established with a call to the setPrintingObserver method. The observer Java object should include the following:

```
<class_definition> implements MGPrintingObserver
{
// ...
public void onBeginLayout (MGPageSetup pgSetup, MGPrintInfo info)
{
// event handling code here...
}
// ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

onEndLayout, enablePrintingEvents, disablePrintingEvents, getPrintingObserver, setPrintingObserver

MGPrintingObserver.onEndLayout

Syntax

```
public void onEndLayout(MGPrintLayout layout, MGPrintInfo info)
```

Description

The end layout observer event method. This event is fired after the Autodesk MapGuide Viewer has finished the internal layout of the page elements but before the map is sent to the printer.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onEndLayout.

Applies To

Plug-In and Java Edition only

Parameters

layout - An MGPrintLayout object that represents the current page layout.

info - An MGPrintInfo object that contains information about the current printing operation.

Error Codes

none

Examples

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGPrintingObserver interface must be established with a call to the setPrintingObserver method. The observer Java object should include the following:

```
<class_definition> implements MGPrintingObserver
{
// ...
public void onEndLayout (MGPrintLayout layout, MGPrintInfo info)
{
// event handling code here...
}
// ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

onBeginLayout, enablePrintingEvents, disablePrintingEvents, getPrintingObserver, setPrintingObserver

MGSelectionChangedObserver

Implements the changed selection observer method onSelectionChanged.

MGSelectionChangedObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the

mgjava.jar file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGSelectionChangedObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onSelectionChanged The changed selection observer method

MGSelectionChangedObserver.onSelectionChanged

Syntax

```
public abstract void onSelectionChanged(MGMap map)
```

Description

The changed selection observer event method. This event is fired when the map selection has changed.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onSelectionChanged.

Applies To

Plug-In and Java Edition only

Parameters

map - The map object for which the event occurred.

Error Codes

none

Note

If your application has *notify* (the second argument of addObject) set to True, writing code in the onSelectionChanged event that changes the selection will result in an infinite loop. For example:

```
var mapSel = map.getSelection();
var obj = layer.getMapObject('CA');
mapSel.addObject(obj, true);
```

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGSelectionChangedObserver interface must be established with a call to the setSelectionChangedObserver method. The observer Java object should include the following:

```
<class_definition> implements MGSelectionChangedObserver
{
// ...
public void onSelectionChanged(MGMap map)
{
// event handling code here...
}
// ...
```

}

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

getSelectionChangedObserver, setSelectionChangedObserver

MGViewChangedObserver

Implements the changed view observer method onViewChanged.

MGViewChangedObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGViewChangedObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onViewChanged	The changed view observer method.

MGViewChangedObserver.onViewChanged

Syntax

```
public abstract void onViewChanged(MGMap map)
```

Description

The changed view observer method. This event is fired after the view of the map has been updated or refreshed.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onViewChanged.

Applies To

Plug-In and Java Edition only

Parameters

map - The map object for which the event occurred.

Error Codes

none

Note

Writing code in the onViewChanged event that changes the map view, such as map.refresh(); results in an infinite loop.

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGViewChangedObserver interface must be established with a call to the setViewChangedObserver method. The observer Java object should include the following:

```
<class_definition> implements MGViewChangedObserver
{
// ...
public void onViewChanged(MGMap map)
{
// event handling code here...
}
// ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

getViewChangedObserver, setViewChangedObserver

MGViewChangingObserver

Implements the changing view observer method onViewChanging.

MGViewChangingObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGViewChangingObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onViewChanging	The changing view observer method.

MGViewChangingObserver.onViewChanging

Syntax

```
public abstract void onViewChanging(MGMap map)
```

Description

The changing view observer method. This event is fired just before the map is refreshed.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onViewChanging.

There are many useful actions you can execute with onViewChanging, for instance, customizing layer settings such as visibility, the SQL Where clause, or the data file to which a layer refers. If you just want to initialize layer settings the first time that Autodesk MapGuide Viewer displays a map, as opposed to every time that the map is refreshed, use the onMapLoaded event instead.

Note that Autodesk MapGuide disables the autoRefresh flag, set by setAutoRefresh, when it executes the onViewChanging event. Therefore, calling zoomGotoDlg, zoomGotoLocation, setUrl, or refresh in onViewChanging has no effect. Additionally, setAutoRefresh has no effect when called from within onViewChanging.

Applies To

Plug-In and Java Edition only

Parameters

map - The map object for which the event occurred.

Error Codes

none

Example

To handle this event, a Java object that implements the MGViewChangingObserver interface must be established with a call to the setViewChangingObserver method. The observer Java object should include the following:

```
<class_definition> implements MGViewChangingObserver
{
// ...
public void onViewChanging(MGMap map)
{
        // event handling code here, for example:
var lat = map.getLat();
var lon = map.getLon();
var point = map.lonlatToMcs(lon, lat);
var sdf = "urban.sdf";
if (point.getX() > 1000) sdf = "rural.sdf";
if (point.getX() < -1000) sdf = "rural.sdf";
if (point.getY() > 1000) sdf = "rural.sdf";
if (point.getY() < -1000) sdf = "rural.sdf";
var mapLayer = map.getMapLayer("streets");
mapLayer.getlayerSetup().setDataFile(sdf);
}
// ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

getViewChangingObserver, setViewChangingObserver

MGViewDistanceExObserver

Implements the view distance observer method onViewedDistanceEx.

MGViewDistanceExObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGViewDistanceExObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onViewedDistanceEx The view distance observer method

MGViewDistanceExObserver.onViewedDistanceEx

Syntax

```
public abstract void onViewedDistanceEx(MGMap map, double totalDistance,
MGCollection distances,
MGCollection points, String units, boolean useMcs)
```

Description

The view distance observer event method. This event is fired when a user has finished viewing distances.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onViewedDistanceEx.

Note that the event is fired *only* if the distances were measured using the viewDistanceEx method. If the distances were measured using the viewDistance method (or the user interface menu commands), onViewedDistance is fired instead.

Applies To

Plug-In and Java Edition only

Parameters

map - The MGMap object for which the event occurred

totalDistance - The total distance viewed.

distances - An MGCollection object containing type-double objects representing all the sub-distances.

points - An MGCollection containing MGPoint objects representing all the points used in the view distance action; the points will be in MCS or lat/lon coordinates, based on the setting of *useMcs*.

units - The units used to quantify the distance.

useMcs - A flag to indicate whether the points are in MCS or lat/lon.

Error Codes

none

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGViewDistanceExObserver interface must be established with a call to the setViewDistanceExObserver method. The observer Java object should include the following:

```
<class_definition> implements MGViewDistanceExObserver
{
// ...
public void onViewedDistanceEx(MGMap map,
 double totalDistance,
 MGCollection distances,
 MGCollection points,
 String units,
 boolean useMcs)
{
<... Event Handling Code Here >
}
// ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

viewDistanceEx

MGViewDistanceObserver

Implements the view distance observer method onViewedDistance.

MGViewDistanceObserver for the Autodesk MapGuide Viewer, Java Edition is provided in the *com.autodesk.mgjava* package, which is located in the *mgjava.cab* file (for Windows systems only) and the *mgjava.jar* file (for all platforms). For Autodesk MapGuide Viewer Plug-In, MGViewDistanceObserver is provided in the *mapguide* package, located in the *npmapv32.zip* file in the Netscape *Plugins* folder.

Event Methods

onViewedDistance	The view distance observer method.

MGViewDistanceObserver.onViewedDistance

Syntax

```
public abstract void onViewedDistance(MGMap map, double totalDistance, MGCollection
distances, String units)
```

Description

The view distance observer event method. This event is fired when a user has finished viewing distances.

This topic describes this event for Plug-In and Java Edition only. For ActiveX Control, see onViewedDistance.

Note that the event is fired *only* if the distances were measured using the viewDistance method or the UI menu commands. If the distances were measured using the viewDistanceEx method, onViewedDistanceEx is fired instead.

Applies To

Plug-In and Java Edition only

Parameters

map - The MGMap object for which the event occurred.

totalDistance - The total distance viewed.

distances - An MGCollection object containing all the sub-distances.

units - The units used to quantify the distance.

Error Codes

-1 (busy)

Example

To handle this event from the Plug-In or Java Edition, a Java object that implements the MGViewDistanceObserver interface must be established with a call to the setViewDistanceObserver method. The observer Java object should include the following:

```
<class_definition> implements MGViewDistanceObserver
{
// ...
public void onViewedDistance(MGMap map,
 double totalDistance,
 MGCollection distances,
 String units)
{
<... Event Handling Code Here >
}
// ...
}
```

The simplest way to include the observer object in a Web page is to implement it as an applet. See Advanced Examples for a fully functional observer object that supports all of the observer interfaces by forwarding the events back to JavaScript functions that have identical semantics to the ActiveX Control event handlers.

See Also

viewDistance

Index

A

aboutDlg A method defined in MGMap

Displays the Autodesk MapGuide About dialog box, which shows information about Autodesk MapGuide Viewer.

AccessKey A property defined in MGMapLayer

Sets the access key for the layer. This value must be 255 characters or less.

add A method defined in MGCollection

Adds the specified object to the collection.

addCirclePrimitive A method defined in MGMapObject

Adds a polygon primitive, composed of individual line segments representing a circle, to the current map feature.

addMapLayer A method defined in MGMap

Downloads and adds the specified Map Layer File (*.mlf*) to the map.

addMapLayer A method defined in MGMapLayerGroup

Adds an existing map layer to the group. A map layer can only belong to one group at any time.

addMapLayers A method defined in MGMap

Downloads and adds the specified Map Layer Files to the map.

addObject A method defined in MGSelection

Adds a single map feature to the selection and updates the display.

addObjects A method defined in MGSelection

Adds the specified map features to the selection and updates the display.

addObjectsEx A method defined in MGSelection

Adds multiple map features to the selection and updates the display.

addPolygonPrimitive A method defined in MGMapObject

Adds a polygon or poly-polygon primitive to the current map feature.

addPolylinePrimitive A method defined in MGMapObject

Adds a polyline or poly-polyline primitive to the current map feature.

addSymbol A method defined in MGPrintLayout

Adds a symbol to the printed page. Symbols are defined in the API Symbol List; refer to the *Autodesk MapGuide User's Guide* for more information.

addSymbolPrimitive A method defined in MGMapObject

Adds a point or predefined symbol primitive to the current map feature.

addTextPrimitive A method defined in MGMapObject

Adds a text primitive to the current map feature.

ApiVersion A property defined in MGMap

Gets the API version for the Autodesk MapGuide Viewer.

Area A property defined in MGMapObject

Gets the area of the specified polygon in Mapping Coordinate System (MCS) units.

Area A property defined in MGPrimitive

Gets the area of the specified polygon primitive in Mapping Coordinate System (MCS) units.

Arg A property defined in MGError

Gets the number of the incorrect argument in the case of an illegal argument.

AutoLinkDelay A property defined in MGMap

Gets or sets the number of seconds before the next HTML page associated with a map feature from an AutoLinkLayer will be displayed in the AutoLinkTarget window.

AutoLinkLayers A property defined in MGMap

Gets or sets the name of the layer that you want to use as the AutoLink layer.

AutoLinkTarget A property defined in MGMap

Gets or sets the name of the target window where Autodesk MapGuide will display the HTML pages associated with map features on AutoLinkLayers as the user pauses the mouse pointer over those map features.

AutoRefresh A property defined in MGMap

Gets or sets the automatic refresh flag.

B

BackColor A property defined in MGFillAttr

Gets or sets the background color index used to draw the polygons.

BackColor A property defined in MGTextAttr

Gets or sets the background color.

BackMode A property defined in MGFillAttr

Gets or sets the background mode used to draw the polygons.

BackMode A property defined in MGPageElement

Gets or sets the background mode of this page element.

BackMode A property defined in MGTextAttr

Gets or sets the display mode for the background behind text features.

BufferSetup A property defined in MGMap

Gets an object that contains buffering properties, such as the distance from the selected features to the edge of the buffers.

Busy A property defined in MGMap

Returns True if Autodesk MapGuide Viewer is busy loading or drawing a map.

C

clear A method defined in MGSelection

Removes all the map features from the selection and updates the display.

Code A property defined in MGError

Gets the error code in the MGError object.

collapse A method defined in MGMapLayerGroup

Collapses the list of map layers under a group so that only the group name heading is visible in the legend.

Collapsed A property defined in MGMapLayerGroup

Gets the state of the map layer group, telling whether it is collapsed or expanded.

Color A property defined in MGEdgeAttr

Gets the edge color index used to draw the polygons.

Color A property defined in MGFillAttr

Gets or sets the fill color index used to draw the polygons.

Color A property defined in MGLineAttr

Gets or sets the color index used to draw the lines or polylines.

Color A property defined in MGTextAttr

Gets or sets the color index used to draw the text.

compressThemeCats A method defined in MGMapLayer

Displays the list of theme categories shown in the legend for the specified map layer in collapsed form, so that only one category from the list is displayed.

copyMap A method defined in MGMap

Copies the current view of the map to the clipboard as an Enhanced Windows Metafile.

copyUrl A method defined in MGMap

Copies the URL of the displayed map to the clipboard.

Count A property defined in MGCollection

Gets the number of map features in the collection that have been displayed in the view since the last reload or rebuild of the map layer.

createLayer A method defined in MGMap

Adds a layer of the specified type to the map.

createMapLayerGroup A method defined in MGMap

Creates a new map layer group and adds it to the map.

createMapObject A method defined in MGMapLayer

Adds an empty feature to the layer. This method works only with redline layers.

createObject A method defined in MGMap

Creates and returns an object of the specified type.

D

DatabaseSetup A property defined in MGMapLayerSetup

Gets the MGDatabaseSetup object for this layer.

DataFile A property defined in MGMapLayerSetup

CHANGED Gets or sets the name of the data file from which Autodesk MapGuide Server retrieves map data for this layer.

DataSource A property defined in MGDatabaseSetup

Gets or sets the name of the OLE DB data source from which Autodesk MapGuide Server retrieves data for this layer.

DataSource A property defined in MGDwgDataSources

NEW Gets or sets the DWG data source name that establishes a connection between Autodesk MapGuide Server and the data provider.

DefaultTarget A property defined in MGMap

Gets or sets the name of the default target window where Autodesk MapGuide will display HTML documents.

digitizeCircle A method defined in MGMap

Enables the user to define a circle on the map by clicking and dragging the mouse.

digitizeLine A method defined in MGMap

Enables the user to draw a line on the map by clicking and dragging the mouse.

digitizePoint A method defined in MGMap

Lets the user define a point on the map by clicking the mouse.

digitizePolygon A method defined in MGMap

Enables the user to use the mouse to define a polygon on the map, then fires the event onDigitizedPolygon upon completion.

digitizePolygonEx A method defined in MGMap

Enables the user to use the mouse to define a polygon on the map, then fires the event onDigitizedPolygon upon completion.

digitizePolyline A method defined in MGMap

Enables a user to use the mouse to define a polyline on the map, then fires the event onDigitizedPolyline upon completion.

digitizePolylineEx A method defined in MGMap

Enables a user to use the mouse to define a polyline on the map, then fires the event onDigitizedPolyline upon completion.

digitizeRectangle A method defined in MGMap

Enables the user to define a rectangle on the map by clicking and dragging the mouse, then fires the onDigitizedRectangle event upon completion.

disableApiExceptions A method defined in MGMap

Disables Autodesk MapGuide Viewer from throwing exceptions.

disablePrintingEvents A method defined in MGMap

Disables the Autodesk MapGuide Viewer from sending printing events to the associated event handler during a print process.

disableSelectionChangedEvent A method defined in MGMap

Disables the firing of the onSelectionChanged event during calls to the MGMapLayer methods removeObject, removeObjects, and removeAllObjects.

DisplayUnits A property defined in MGMap

Gets or sets the current display units, either M (meters), KM (kilometers), FT (feet), or MI (miles), as seen in the map width of the status bar.

Distance A property defined in MGBufferSetup

Gets or sets the buffer offset distance.

DoubleClickHandled A property defined in MGMapObject

Indicates whether the onDoubleClickObject event has been handled for this object.

Dwg A property defined in MGDwgDataSources

NEW Gets or sets the Autodesk DWG file name or project file for the Autodesk DWG layer.

Dynamic A property defined in MGMapLayer

Returns True if the map layer is dynamic.

E

EdgeAttr A property defined in MGBufferSetup

Gets the object that groups the polygon edge attributes used to draw the buffer polygon(s).

EdgeAttr A property defined in MGRedlineSetup

Gets an object that groups the redline style's polygon edge attributes.

enableApiExceptions A method defined in MGMap

Enables Autodesk MapGuide Viewer exception throwing.

enablePrintingEvents A method defined in MGMap

Enables the Autodesk MapGuide Viewer to send printing events to the associated event handler during a print process.

enableSelectionChangedEvent A method defined in MGMap

Enables the firing of the onSelectionChanged event during calls to the MGMapLayer methods removeObject, removeObjects, and removeAllObjects.

equals A method defined in MGExtent

Compares this MGExtent object to the specified object.

equals A method defined in MGExtentEx

Compares this MGExtentEx to the specified object.

equals A method defined in MGMapLayer

Compares this MGMapLayer object to the specified object.

equals A method defined in MGMapObject

Compares this MGMapObject object to the specified object.

equals A method defined in MGPoint

Compares this MGPoint object to the specified object.

Error A property defined in MGError

Gets the error status in the MGError object.

ErrorTarget A property defined in MGMap

Gets or sets the name of the target window where Autodesk MapGuide will display error messages.

expand A method defined in MGMapLayerGroup

Expands the list of map layers under a group in the legend so that all potentially visible map layers appear under the group name heading.

expandThemeCats A method defined in MGMapLayer

Displays the list of theme categories shown in the legend for the specified map layer in expanded form, so that all categories in the list are displayed.

Extent A property defined in MGMapObject

Gets the extent of the map feature.

Extent A property defined in MGPageElement

Gets or sets the extents of the current page element on the printout.

F

FillAttr A property defined in MGBufferSetup

Gets the object that groups the polygon fill attributes used to draw the buffer polygon(s).

FillAttr A property defined in MGRedlineSetup

Gets an object that groups the redline style's polygon fill attributes.

FontName A property defined in MGTextAttr

Gets or sets the font name. Font names are not case-sensitive.

FontNames A property defined in MGTextAttr

Gets or sets the collection containing the available font names.

FontStyle A property defined in MGTextAttr

Gets or sets the font style. Names of font styles are not case-sensitive.

FullUrl A property defined in MGMap

Gets the full URL of the map that is currently displayed by Autodesk MapGuide, including parameters specifying the current location and any non-default options.

G

getApiVersion A method defined in MGMap

Gets the Autodesk MapGuide Viewer API version.

getArea A method defined in MGMapObject

Gets the area of the specified polygon map feature.

getArea A method defined in MGPrimitive

Gets the area of the specified polygon map feature primitive.

getArg A method defined in MGError

Gets the number of an incorrect argument in the event that an illegal argument is passed.

getAsString A method defined in MGSelection

Returns a string that contains the keys of the selected map features.

getAutoLinkDelay A method defined in MGMap

Gets the number of seconds before the next HTML page associated with a map feature from the AutoLinkLayer will be displayed in the AutoLinkTarget window.

getAutoLinkLayers A method defined in MGMap

Gets the name of the layer that uses the AutoLink feature.

getAutoLinkTarget A method defined in MGMap

Gets the name of the target window where Autodesk MapGuide will display the HTML pages associated with map features on the AutoLinkLayer.

getAutoRefresh A method defined in MGMap

Returns a boolean value that indicates whether or not the automatic refresh feature is enabled.

getBackColor A method defined in MGFillAttr

Gets the background color index used to draw the polygons.

getBackColor A method defined in MGTextAttr

Gets the background color.

getBackMode A method defined in MGFillAttr

Gets the background mode used to draw the polygons.

getBackMode A method defined in MGPageElement

Gets the background mode of this element. The supported background modes are *opaque* and *transparent*.

getBackMode A method defined in MGTextAttr

Gets the display mode for the background behind text features.

getBufferSetup A method defined in MGMap

Gets an object that contains buffering properties, such as the distance from the selected features to draw the buffers.

getBusyStateChangedObserver A method defined in MGMap

Returns a reference to the Java object that observes busy state change events, or null if no observer has been established.

getCode A method defined in MGError

Gets the error code of the MGError object.

getCodeBase A method defined in MGMapContext

Gets the Autodesk MapGuide Viewer component's code base URL.

getColor A method defined in MGEdgeAttr

Gets the edge color index used to draw the polygons.

getColor A method defined in MGFillAttr

Gets the fill color index used to draw the polygons.

getColor A method defined in MGLineAttr

Gets the color index used to draw the lines or polylines.

getColor A method defined in MGTextAttr

Gets the text color.

getDatabaseSetup A method defined in MGMapLayerSetup

Gets the MGDatabaseSetup object for this layer.

getDataFile A method defined in MGMapLayerSetup

CHANGED Gets the name of the data file from which Autodesk MapGuide Server retrieves data for this layer.

getDataSource A method defined in MGDatabaseSetup

Gets the name of the OLE DB data source from which Autodesk MapGuide Server retrieves data.

getDataSource A method defined in MGDwgDataSources

NEW Gets the DWG data source name that establishes a connection between Autodesk MapGuide Server and the data provider.

getDefaultTarget A method defined in MGMap

Gets the name of the default target window where Autodesk MapGuide will display HTML documents, including autolinks, errors, feature links, reports, and URL lists.

getDisplayUnits A method defined in MGMap

Gets the current type of display units being used to display the map width, as seen in the status bar.

getDistance A method defined in MGBufferSetup

Gets the buffer offset distance.

getDocumentBase A method defined in MGMapContext

Gets the Autodesk MapGuide Viewer component's base document URL.

getDoubleClickObserver A method defined in MGMap

Returns a reference to the Java object being used to observe double-click events, or null if no observer has been established.

getDwg A method defined in MGDwgDataSources

NEW Gets the Autodesk DWG file name or project file for the Autodesk DWG layer.

getDwgDataSources A method defined in MGMapLayerSetup

NEW Gets the MGDwgDataSources object for this layer.

getEdgeAttr A method defined in MGBufferSetup

Gets an object that groups the polygon edge attributes used to draw the buffer polygon(s).

getEdgeAttr A method defined in MGRedlineSetup

Gets an object that groups the redline style's polygon edge attributes.

getErrorTarget A method defined in MGMap

Gets the name of the target window where Autodesk MapGuide will display error messages.

getExtent A method defined in MGMapObject

Gets the extent of the map feature, that is, the smallest rectangle that can be drawn around all the map feature's elements.

getExtent A method defined in MGPageElement

Gets the extents of this element on the printout.

getExtent A method defined in MGPrimitive

Gets the extents of the map feature primitive.

getExtentEx A method defined in MGMapObject

Gets the extent of the map feature, that is, the smallest rectangle that can be drawn around all the map feature's elements.

getFillAttr A method defined in MGBufferSetup

Gets an object that groups the polygon fill attributes used to draw the buffer polygon(s).

getFillAttr A method defined in MGRedlineSetup

Gets an object that groups the redline style's polygon fill attributes.

getFontName A method defined in MGTextAttr

Gets the font name.

getFontNames A method defined in MGTextAttr

Gets a collection containing the available font names.

getFontStyle A method defined in MGTextAttr

Gets the font style.

getFontStyles A method defined in MGTextAttr

Gets a collection containing the font styles for the specified font. Styles vary by font family, but are typically values like 'Regular', 'Bold', and 'Italic'

getHatchPattern A method defined in MGFillAttr

Gets the fill hatch pattern used to draw the polygons.

getHeight A method defined in MGMap

Gets the height of the map.

getHeight A method defined in MGSymbolAttr

Gets the display height of the point in the specified unit type.

getHeight A method defined in MGTextAttr

Gets the display height of the text in the specified unit type.

getHorizAlign A method defined in MGTextAttr

Gets the horizontal placement of the text features in relation to an imaginary vertical axis.

getInclude A method defined in MGPageSetup

Determines whether the specified element will be included on the printout.

getIncludeLegend A method defined in MGPageSetup

Returns a value indicating whether the legend will be printed next to the map.

getIntermediateUpdatesEnabled A method defined in MGMap

NEW Gets the current setting of the intermediate updates flag.

getKey A method defined in MGMapObject

Gets the key of the map feature.

getKeyColumn A method defined in MGDatabaseSetup

Gets the database column name that contains the keys of the map features on this layer.

getKeyColumn A method defined in MGDwgDataSources

NEW Gets the name of the column that contains the primary key for each map feature on the DWG layer.

getKeyColumnType A method defined in MGDatabaseSetup

Gets the key column type.

getKeyColumnType A method defined in MGDwgDataSources

NEW Gets the DWG key column data type.

getKeyTable A method defined in MGDwgDataSources

NEW Gets the name of the DWG key table to use as primary keys for either feature properties, external database links, or object data in the Autodesk DWG.

getLastError A method defined in MGMap

Returns a reference to the MGError object of the most recent Autodesk MapGuide Viewer API method call or property access.

getLat A method defined in MGMap

Gets the latitude (*Y*) value in decimal degrees of the center of the map area currently displayed.

getLatColumn A method defined in MGDatabaseSetup

Gets the name of the database column that contains the latitude (or *Y* coordinate) values of the map features on this layer.

getLayerFilter A method defined in MGDwgDataSources

NEW Gets the layer filter, which specifies which layers to extract from the specified Autodesk DWG or project file.

getLayerName A method defined in MGBufferSetup

Gets the name of the map layer that will contain the buffer objects generated from the viewBuffer method of MGMap.

getLayerSetup A method defined in MGMapLayer

Gets an MGMapLayerSetup object.

getLayerType A method defined in MGMapLayer

Returns a string indicating the layer type.

getLegendLabel A method defined in MGMapLayer

Gets the string used to label the map layer in the legend.

getLegendLabel A method defined in MGMapLayerGroup

Gets the string used to label the map layer group in the legend.

getLength A method defined in MGMapObject

Gets the length of the specified polyline map feature.

getLength A method defined in MGPrimitive

Gets the length of the specified polyline map feature primitive.

getLineAttr A method defined in MGRedlineSetup

Gets an object that groups the redline style's line or polyline attributes.

getLink A method defined in MGMapObject

Gets the URL link associated with the map feature.

getLinkedToDatabase A method defined in MGMapLayerSetup

Indicates whether or not the map features in a spatial data file are linked to a table in a database.

getLon A method defined in MGMap

Gets the longitude (*X*) value in decimal degrees of the center of the map area currently displayed.

getLonColumn A method defined in MGDatabaseSetup

Gets the name of the database column that contains the longitude (or *X* coordinate) values of the map features on this layer.

getMap A method defined in MGMapLayer

Gets the map that contains the map layer.

getMapExtent A method defined in MGMap

Gets the extents for the map or current map view.

getMapLayer A method defined in MGMap

Gets the specified map layer.

getMapLayer A method defined in MGMapLayerGroup

Gets the specified map layer from the group.

getMapLayer A method defined in MGMapObject

Gets the map layer that contains the map feature.

getMapLayerGroup A method defined in MGMap

Gets the specified map layer group.

getMapLayerGroup A method defined in MGMapLayer

Gets the map layer group to which the layer belongs.

getMapLayerGroups A method defined in MGMap

Gets an MGCollection object containing all the map layer groups defined in the map.

getMapLayers A method defined in MGMap

Returns a Vector (or MGCollection) object containing all the map layers defined in the map.

getMapLayers A method defined in MGMapLayerGroup

Gets an MGCollection object containing all map layers in the group.

getMapLayersEx A method defined in MGMap

Gets an MGCollection object containing all the map layers defined in the map.

getMapLayerStyles A method defined in MGMapLayer

Gets a collection containing all of the MGMapLayerStyle objects associated with the layer.

getMapLoadedObserver A method defined in MGMap

Returns a reference to an MGMapLoadedObserver object or null if no observer has been established.

getMapObject A method defined in MGMapLayer

Gets the map feature with the specified key from the layer.

getMapObjects A method defined in MGMapLayer

Returns a Vector (or MGCollection object) containing the map features of the layer that fall within the map extent (the part of the map that is displayed).

getMapObjects A method defined in MGSelection

Returns a Vector (or MGCollection object) of selected map features.

getMapObjectsEx A method defined in MGMapLayer

Gets an MGCollection object containing the map features of the layer that fall within the map extent (the part of the map that is displayed).

getMapObjectsEx A method defined in MGSelection

Gets an MGCollection object containing the selected map features.

getMaxDisplayRange A method defined in MGMapLayerStyle

Gets the maximum value of the display range scale.

getMaxLat A method defined in MGExtent

Gets the maximum latitude (or *Y* coordinate) of the extent in decimal degrees.

getMaxLon A method defined in MGExtent

Gets the maximum longitude (or *X* coordinate) of the extent in decimal degrees.

getMaxX A method defined in MGExtentEx

Gets the maximum *X* value, or longitude, of the extent.

getMaxY A method defined in MGExtentEx

Gets the maximum *Y* value, or latitude, of the extent.

getMcsScaleFactor A method defined in MGMap

Gets the conversion factor to convert from the current Mapping Coordinate System (MCS) to meters.

getMinDisplayRange A method defined in MGMapLayerStyle

Gets the minimum value of the display range scale.

getMinLat A method defined in MGExtent

Gets the minimum latitude (or *Y* coordinate) of the extent in decimal degrees.

getMinLon A method defined in MGExtent

Gets the minimum longitude (or *X* coordinate) of the extent in decimal degrees.

getMinX A method defined in MGExtentEx

Gets the minimum *X* value, or longitude, of the extent.

getMinY A method defined in MGExtentEx

Gets the minimum *Y* value, or latitude, of the extent.

getMode A method defined in MGMap

NEW Gets the current mode of the Autodesk MapGuide Viewer.

getName A method defined in MGMap

Gets the name of the map, as set in the Preferences dialog box in Autodesk MapGuide Author, or with setName.

getName A method defined in MGMapLayer

Gets the name of the map layer.

getName A method defined in MGMapLayerGroup

Gets the name of the map layer group.

getName A method defined in MGMapObject

Gets the name of the map feature.

getNameColumn A method defined in MGDatabaseSetup

Gets the name of the database column that contains the names of the map features on this layer.

getNameColumn A method defined in MGDwgDataSources

NEW Gets the name of the column that contains the name for each map feature.

getNameFromSdpFeatureTable A method defined in MGMapLayerSetup

Indicates whether the feature name should come from the SDP feature table or from a secondary table in an OLE DB data source.

getNameTable A method defined in MGDwgDataSources

NEW Gets the name of the DWG table for MapTips about feature properties, external database links, or object data in the Autodesk DWG.

getNumObjects A method defined in MGSelection

Gets the number of map features in the current selection.

getObjectLinkTarget A method defined in MGMap

Gets the name of the target window where Autodesk MapGuide will display the URL associated with a map feature.

getOneObject A method defined in MGBufferSetup

Returns a value indicating whether one large merged buffer object should be created for all buffered features.

getPageElement A method defined in MGPrintLayout

Gets the page element with the specified element ID.

getPageExtent A method defined in MGPrintInfo

Gets the size of the printable area of the page.

getPageMargins A method defined in MGPageSetup

Gets the page margins used by the current page layout.

getPageResolution A method defined in MGPrintInfo

Gets the resolution of the output device to be used for the current print operation.

getPageSetup A method defined in MGMap

Gets an object containing properties used by Autodesk MapGuide to print maps.

getParameter A method defined in MGMapContext

Retrieves the value of a parameter.

getParentObject A method defined in MGPrimitive

Gets the parent object of the primitive.

getPlotScale A method defined in MGPageSetup

Gets the scale if the map is to be printed at a specific map scale.

getPrimitives A method defined in MGMapObject

Gets a collection of primitives describing the map feature.

getPrintingObserver A method defined in MGMap

Returns a reference to the Java object being used to observe printing events.

getPrintPriority A method defined in MGPageElement

Gets the print priority of this page element.

getPriority A method defined in MGMapLayer

Gets the priority of the map layer.

getRebuild A method defined in MGMapLayer

Gets the map layers rebuild flag.

getRedlineSetup A method defined in MGMap

Gets an object containing all properties for drawing redline features.

getReportTarget A method defined in MGMap

Gets the name of the target window where Autodesk MapGuide will display reports from View Reports command.

getRotation A method defined in MGSymbolAttr

Gets the rotation of the point, in degrees.

getRotation A method defined in MGTextAttr

Gets the rotation of the text, in degrees.

getScale A method defined in MGMap

Gets the current map display scale in terms of 1:scale.

getSdpDataSource A method defined in MGMapLayerSetup

Gets the spatial data source name for this map layer.

getSdpFeatureTable A method defined in MGMapLayerSetup

Gets the feature table name for this map layer.

getSdpGeometryColumn A method defined in MGMapLayerSetup

Gets the SDP feature table geometry column name.

getSdpKeyColumn A method defined in MGMapLayerSetup

Gets the SDP feature table key column.

getSdpKeyColumnType A method defined in MGMapLayerSetup

Gets the SDP feature table key column type.

getSelectability A method defined in MGMapLayer

Returns True if map features within the map layer may be selected.

getSelection A method defined in MGMap

Gets the currently selected map features.

getSelectionChangedObserver A method defined in MGMap

Returns a reference to the Java object being used to observe selection change events, or null if no observer has been established.

getSelectionMode A method defined in MGMap

Gets the selection mode, which controls how map features are selected when a feature is not fully contained in the selection area.

getServerUrl A method defined in MGMapLayerSetup

Gets the Autodesk MapGuide Server URL, including the name of the MapAgent file.

getShowInLegend A method defined in MGMapLayer

Returns a boolean value indicating whether the map layer should be displayed in the legend.

getShowInLegend A method defined in MGMapLayerGroup

Returns a boolean value indicating whether or not the map layer group should be displayed in the legend.

getSourceType A method defined in MGMapLayerSetup

Gets the source of the map data on this layer.

getSQLWhere A method defined in MGMapLayer

Gets the SQL Where clause for the layer.

getStaticVersion A method defined in MGMapApplet

Gets the Autodesk MapGuide Viewer, Java Edition version number as a string.

getStaticVersion A method defined in MGMapComponent

Gets the Autodesk MapGuide Viewer, Java Edition version number as a string. The version number is the same as the number returned by the getVersion method of MGMap.

getStyle A method defined in MGEdgeAttr

Gets the edge style used to draw the polygons.

getStyle A method defined in MGFillAttr

Gets the fill style used to draw the polygons.

getStyle A method defined in MGLineAttr

Gets the style used to draw the lines or polylines.

getSymbol A method defined in MGSymbolAttr

Gets the symbol to be used as the point attribute.

getSymbolAngleColumn A method defined in MGDatabaseSetup

Gets the name of the column containing the rotation angle of symbols for point features.

getSymbolAttr A method defined in MGPageElement

Gets an object that groups the current page element's symbol attributes.

getSymbolAttr A method defined in MGRedlineSetup

Gets an object that groups the redline style's symbol attributes.

getSymbolHeightColumn A method defined in MGDatabaseSetup

Gets the name of the column containing the height of symbols for point features.

getSymbols A method defined in MGMap

Gets a collection of strings representing the symbols available for use as symbol attributes.

getSymbolWidthColumn A method defined in MGDatabaseSetup

Gets the name of the column containing the width of symbols for point features.

getTable A method defined in MGDatabaseSetup

Gets the database table name, within the OLE DB data source, from which Autodesk MapGuide Server retrieves data for this layer.

getTextAngleColumn A method defined in MGDatabaseSetup

Gets the name of the database column that contains the rotation angle for text map features on this layer.

getTextAttr A method defined in MGRedlineSetup

Gets an object that groups the redline style's text attributes.

getTextHeightColumn A method defined in MGDatabaseSetup

Gets the name of the database column that contains the height for text map features on this layer.

getTextHorizAlignColumn A method defined in MGDatabaseSetup

Gets the name of the column containing horizontal alignment values for text.

getTextValue A method defined in MGPrimitive

Gets the text value for a text primitive.

getTextVertAlignColumn A method defined in MGDatabaseSetup

Gets the name of the column containing vertical alignment values for text.

getThickness A method defined in MGEdgeAttr

Gets the edge thickness used to draw the polygons.

getThickness A method defined in MGLineAttr

Gets the thickness used to draw the lines or polylines.

getTitle A method defined in MGPageSetup

Gets the title to be printed above the map at the top of the page.

getTitleTextAttr A method defined in MGPageSetup

Gets an object that groups the title text attributes.

getTrackingID A method defined in MGMap

Gets the customer tracking ID associated with a map.

getType A method defined in MGMapLayer

Returns a string indicating the type of features the layer contains.

getType A method defined in MGMapObject

Gets the type of the map feature.

getType A method defined in MGPrimitive

Gets the type of map feature primitive.

getUnit A method defined in MGBufferSetup

Gets the units used to specify the size of the buffer.

getUnits A method defined in MGMap

Gets the current units that will be used when getting or setting the MapWidth property, or when using getWidth or setWidth without specifying the *Units* parameter.

getUrl A method defined in MGMap

Gets the URL of the map that is currently displayed by Autodesk MapGuide Viewer.

getUrlColumn A method defined in MGDwgDataSources

NEW Gets the name of the column that contains the URL associated with each map feature.

getURLFromSdpFeatureTable A method defined in MGMapLayerSetup

Indicates whether the feature URL should come from the SDP feature table or from a secondary table in an OLE DB data source.

getUrlLinkColumn A method defined in MGDatabaseSetup

Gets the name of the database column that contains URLs that are linked to map features on this layer.

getUrlListState A method defined in MGMap

Returns True if Autodesk MapGuide will generate an HTML document that lists the names and URLs associated with a map feature as the user pauses the mouse pointer over that feature.

getUrlListTarget A method defined in MGMap

Gets the name of the target window where Autodesk MapGuide will display the names and URLs associated with a map feature as the user pauses the mouse pointer over that feature.

getUrlTable A method defined in MGDwgDataSources

NEW Gets the name of the DWG URL table containing URLs to which Autodesk DWG map features are linked.

getUsePlotScale A method defined in MGPageSetup

Returns a value indicating whether the map will be printed at a specific map scale.

getVersion A method defined in MGMap

Gets the version string of the Autodesk MapGuide Viewer.

getVertAlign A method defined in MGTextAttr

Gets the vertical placement of the text features in relation to an imaginary horizontal baseline.

getVertices A method defined in MGMapObject

Gets the vertices used to describe the geometry of the map feature.

getVertices A method defined in MGPrimitive

Gets the vertices used to describe the geometry of the map feature primitive.

getViewChangedObserver A method defined in MGMap

Returns a reference to the Java object being used to observe view change events, or null if no observer has been established.

getViewChangingObserver A method defined in MGMap

Returns a reference to the Java object that observes when a map is about to be refreshed, or null if no such observer has been established.

getViewDistanceExObserver A method defined in MGMap

Returns a reference to the Java object that observes view distance events, or null if no such observer has been established.

getViewDistanceObserver A method defined in MGMap

Returns a reference to the Java object that observes view distance events, or null if no such observer has been established.

getViewerType A method defined in MGMap

Returns a string indicating whether the current Autodesk MapGuide Viewer is the ActiveX Control, Java Edition, or the Autodesk MapGuide Viewer Plug-In.

getVisibility A method defined in MGMapLayer

Returns True if the map layer is potentially visible.

getVisibility A method defined in MGMapLayerGroup

Returns a boolean value indicating if the map layers in this group are potentially visible.

getVisible A method defined in MGEdgeAttr

Returns a value indicating whether the edge should be drawn around the polygons.

getWhereClause A method defined in MGDatabaseSetup

Gets the SQL Where clause that Autodesk MapGuide Server uses to query the OLE DB data source that provides data for this layer.

getWhereFromSdpFeatureTable A method defined in MGMapLayerSetup

Indicates whether the feature SQL Where clause should come from the SDP feature table or from a secondary table in an OLE DB data source.

getWidth A method defined in MGMap

Gets the current width of the map in the specified unit of measure.

getWidth A method defined in MGSymbolAttr

Gets the display width of the point in the specified unit type.

getX A method defined in MGPoint

Gets the *X* value (longitude) of the point.

getY A method defined in MGPoint

Gets the *Y* value (latitude) of the point.

H

HatchPattern A property defined in MGFillAttr

Gets or sets the fill hatch pattern used to draw the polygons.

Height A property defined in MGSymbolAttr

Gets or sets the display height of the point in the specified unit type.

HorizAlign A property defined in MGTextAttr

Sets the horizontal placement of the text features in relation to an imaginary vertical axis.

I

IncludeLegend A property defined in MGPageSetup

Gets or sets whether to print the legend next to the map.

IntermediateUpdatesEnabled A property defined in MGMap

NEW Specifies whether or not the map window should update every 1.5 seconds during a drawing operation or only at the end when all data has been loaded and rendered.

isBusy A method defined in MGMap

Returns True if Autodesk MapGuide Viewer is busy loading or drawing the map.

isCollapsed A method defined in MGMapLayerGroup

Determines whether the map layer group is collapsed or expanded.

isDynamic A method defined in MGMapLayer

Returns a boolean value indicating whether the map layer is dynamic.

isEmpty A method defined in MGCollection

Returns a value indicating whether the collection is empty or contains any objects.

isError A method defined in MGError

Gets the error status of the MGError object.

isLocked A method defined in MGMapLayer

Indicates whether Autodesk MapGuide Viewer has access to either coordinate values or map layer setup data, depending on the value of the function parameter that is passed in.

isStatic A method defined in MGMapLayer

Returns a boolean value indicating whether the map layer is static.

isVisible A method defined in MGMapLayer

Returns True if the map layer is currently visible on the map.

item A method defined in MGCollection

Gets the object at the specified index in the collection.

K

Key A property defined in MGMapObject

Gets the key of the map feature.

KeyColumn A property defined in MGDatabaseSetup

Gets or sets the name of the database column that contains the keys of the map features on this layer.

KeyColumn A property defined in MGDwgDataSources

NEW Gets or sets the name of the column that contains the primary key for each map feature on the DWG layer.

KeyColumnType A property defined in MGDatabaseSetup

Gets or sets the data type of the database column containing the keys of map features.

KeyColumnType A property defined in MGDwgDataSources

NEW Gets or sets the DWG key column data type.

KeyTable A property defined in MGDwgDataSources

NEW Gets or sets the name of the DWG key table to use as primary keys for either feature properties, external database links, or object data in the Autodesk DWG.

L

LastError A property defined in MGMap

Gets the MGError object that contains the error value and message for the most recent API error.

Lat A property defined in MGMap

Gets the latitude (*Y*) value in decimal degrees of the center of the map area currently displayed.

LatColumn A property defined in MGDatabaseSetup

Gets or sets the name of the database column that contains the latitude (or *Y* coordinate) values of the map features on this layer.

LayerFilter A property defined in MGDwgDataSources

NEW Gets or sets the layer filter, which specifies which layers to extract from the specified Autodesk DWG or project file.

LayerName A property defined in MGBufferSetup

Gets or sets the name of the buffer layer to use for the buffer objects.

LayerSetup A property defined in MGMapLayer

Gets the MGMapLayerSetup object for this layer. Applies to all layer types except Drawing layers or layers that have been secured.

LayersViewWidth A property defined in MGMap

Gets or sets the width of the legend. In order to change this property at runtime, it must be non-zero when the MGMap object is created.

LayerType A property defined in MGMapLayer

Gets the layer type.

LegendLabel A property defined in MGMapLayer

Gets or sets the legend label used by the map layer.

LegendLabel A property defined in MGMapLayerGroup

Gets or sets the legend label used by the map layer group.

Length A property defined in MGMapObject

Gets the length of the specified polyline in Mapping Coordinate System (MCS) units.

Length A property defined in MGPrimitive

Gets the length of the specified polyline primitive in Mapping Coordinate System (MCS) units.

LineAttr A property defined in MGRedlineSetup

Gets an object that groups the redline style's line or polyline attributes.

Link A property defined in MGMapObject

Gets or sets the URL link associated with the map feature.

LinkedToDatabase A property defined in MGMapLayerSetup

Indicates whether or not the map features in a spatial data file are linked to a table in a database.

lock A method defined in MGMapLayer

Prevents Autodesk MapGuide Viewer from having access to either coordinate values or map layer setup data, depending on the value of the function parameter that is passed in.

Lon A property defined in MGMap

Gets the longitude (*X*) value in decimal degrees of the center of the map area currently displayed.

LonColumn A property defined in MGDatabaseSetup

Gets or sets the name of the database column that contains the longitude (or *X* coordinate) values of the map features on this layer.

lonLatToMcs A method defined in MGMap

Converts the specified lat/lon coordinate pair to a point using the map coordinate system (MCS) defined for the MWF by the map author.

M

Map A property defined in MGMapLayer

Gets the map containing the map layer.

MapHeight A property defined in MGMap

Gets the height of the map. This value is expressed in the units set with the MapUnits property.

MapLayer A property defined in MGMapObject

Gets the map layer that contains the map feature.

MapLayerGroup A property defined in MGMapLayer

Specifies the name of the map layer group that the current layer belongs to.

MapLayerGroups A property defined in MGMap

Gets a collection containing all the map layer groups defined in the map.

MapLayers A property defined in MGMap

Gets a collection containing all the map layers defined in the map.

MapLayers A property defined in MGMapLayerGroup

Gets or sets an MGCollection containing all the map layers defined in the group.

MapLayerStyles A property defined in MGMapLayer

Gets a collection containing all of the MGMapLayerStyle objects associated with the layer.

MapName A property defined in MGMap

Gets or sets the name of the map.

MapObjects A property defined in MGMapLayer

Gets an MGCollection object containing the map features of the layer that fall within the map extent.

MapScale A property defined in MGMap

Gets or sets the current map display scale in terms of 1:scale.

MapUnits A property defined in MGMap

Gets or sets the current units that will be used when getting or setting the MapWidth property, one of M (meters), KM (kilometers), FT (feet), or MI (miles).

MapWidth A property defined in MGMap

Gets or sets the current map width.

MaxDisplayRange A property defined in MGMapLayerStyle

Gets the maximum value of the display range scale.

MaxLat A property defined in MGExtent

Gets the maximum latitude (or *Y* coordinate) of the extent in decimal degrees.

MaxLon A property defined in MGExtent

Gets the maximum longitude (or *X* coordinate) of the extent in decimal degrees.

MaxX A property defined in MGExtentEx

Gets the maximum *X* value, or longitude, of the extent.

MaxY A property defined in MGExtentEx

Gets the maximum *Y* value, or latitude, of the extent.

McsScaleFactor A property defined in MGMap

Gets the conversion factor used to convert from the current Mapping Coordinate System (MCS) to meters.

mcsToLonLat A method defined in MGMap

Converts the specified Mapping Coordinate System (MCS) coordinate pair to lat/lon coordinates.

MGAddMapLayerObserver An object.

Implements the map layer-addition observer event method onAddMapLayer.

MGBufferSetup An object.

Organizes the buffer setup properties, which control how Autodesk MapGuide Viewer creates buffers with the viewBuffer and viewBufferDlg methods and the popup menu View Buffer command.

MGBusyStateChangedObserver An object.

Implements the busy state observer event onBusyStateChanged.

MGCollection An object.

Represents collections of objects, such as MGPoint, MGMapObject, Double, Integer, or any other type of object.

MGDatabaseSetup An object.

Provides access to the database setup for map layers that are linked to database tables.

MGDigitizeCircleObserver An object.

Implements the circle-digitizing observer method onDigitizedCircle.

MGDigitizePointObserver An object.

Implements the point-digitizing observer method onDigitizedPoint.

MGDigitizePolygonObserver An object.

Implements the polygon-digitizing observer method onDigitizedPolygon.

MGDigitizePolygonObserverEx An object.

Implements the polygon-digitizing observer method onDigitizedPolygon.

MGDigitizePolylineObserver An object.

Implements the polyline-digitizing observer method onDigitizedPolyline.

MGDigitizePolylineObserverEx An object.

Implements the polyline-digitizing observer method onDigitizedPolyline.

MGDigitizeRectangleObserver An object.

Implements the rectangle-digitizing observer method onDigitizedRectangle.

MGDoubleClickObjectObserver An object.

Implements the double-click observer method onDoubleClickObject.

MGDwgDataSources A property defined in MGMapLayerSetup

NEW! Gets the MGDwgDataSources object for this layer.

MGDwgDataSources An object.

NEW! Implements methods and properties to support the use of DWG data sources.

MGEdgeAttr An object.

Contains the polygon edge attributes used to draw buffer or redline polygons.

MGError An object.

Contains error diagnostic information about the most recently executed API method.

MGExtent An object.

Represents an extent of a map in lat/lon (*X*/*Y*) coordinates.

MGExtentEx An object.

Represents spatial extents of a map or a bounding box.

MGFillAttr An object.

Contains the polygon fill attributes used to draw buffer or redline polygons.

MGLineAttr An object.

Contains the attributes used to draw lines and polylines on redline layers.

MGMap An object.

Top-level object of the API for Autodesk MapGuide Viewer ActiveX Control, Autodesk MapGuide Viewer Plug-In, and Autodesk MapGuide Viewer, Java Edition. For the ActiveX Control, MGMap is a COM interface.

MGMapApplet An object.

Applet that implements the MGMap interface in the Autodesk MapGuide Viewer, Java Edition API.

MGMapComponent An object.

Instantiates Autodesk MapGuide Viewer as a user interface object in the Autodesk MapGuide Viewer, Java Edition API. MGMapComponent extends the Component class and implements the MGMap interface.

MGMapComponent A method defined in MGMapComponent

This method constructs a new instance of MGMapComponent in the Autodesk MapGuide Viewer, Java Edition.

MGMapContext An object.

Java interface whose methods provide browser services to MGMapComponent in theAutodesk MapGuide Viewer, Java Edition API.

MGMapLayer An object.

Represents a layer on the map.

MGMapLayerGroup An object.

Represents a map layer group.

setVisibility A method defined in MGMapLayerGroup

Sets a flag that affects the visibility of all map layers in the group.

MGMapLayerSetup An object.

Provides run-time access to map layer setup through methods that let you change data source names and alter the way map features are linked to databases.

MGMapLayerStyle An object.

Provides access to the display scale range of a map layer style.

MGMapLoadedObserver An object.

Implements the map loaded observer method onMapLoaded.

MGMapObject An object.

Represents a feature on the map, such as a point, polyline, polygon, or text.

MGPrintingObserver An object.

Implements the printing observer methods onBeginLayout and onEndLayout.

MGPageElement An object.

Represents a page element on the printout.

MGPageSetup An object.

Organizes the Page Setup properties, which control how Autodesk MapGuide prints maps.

MGPoint An object.

Represents a point on the map expressed in Mapping Coordinate System (MCS) or latitude/longitude coordinates.

MGPrimitive An object.

Allows you to query the individual primitives that make up a multi-primitive feature such as a GIS Design Server or redline feature.

MGPrintInfo An object.

Provides information about the current print operation.

MGPrintLayout An object.

Controls the layout of the page elements on the printout.

MGRedlineSetup An object.

Provides access to the style information used to create features on a redline layer.

MGSelection An object.

Represents the list of selected map features.

MGSelectionChangedObserver An object.

Implements the changed selection observer method onSelectionChanged.

MGSymbolAttr An object.

Contains the point attributes used to draw symbols on redline layers.

MGTextAttr An object.

Contains the attributes used to draw text on redline layers.

MGViewChangedObserver An object.

Implements the changed view observer method onViewChanged.

MGViewChangingObserver An object.

Implements the changing view observer method onViewChanging.

MGViewDistanceExObserver An object.

Implements the view distance observer method onViewedDistanceEx.

MGViewDistanceObserver An object.

Implements the view distance observer method onViewedDistance.

MinDisplayRange A property defined in MGMapLayerStyle

Gets the minimum value of the display range scale.

MinLat A property defined in MGExtent

Gets the minimum latitude (or *Y* coordinate) of the extent.

MinLon A property defined in MGExtent

Gets the minimum longitude (or *X* coordinate) of the extent.

MinX A property defined in MGExtentEx

Gets the minimum *X* value, or longitude, of the extent.

MinY A property defined in MGExtentEx

Gets the minimum *Y* value, or latitude, of the extent.

Mode A property defined in MGMap

NEW Gets the current viewer mode of the map, such as zoomIn, digitize, pan, and so on.

N

Name A property defined in MGMapLayer

Gets or sets the name of the map layer.

Name A property defined in MGMapLayerGroup

Gets or sets the name of the map layer group.

Name A property defined in MGMapObject

Gets or sets the name of the map feature.

NameColumn A property defined in MGDatabaseSetup

Gets or sets the name of the database column containing the names of map features on this layer.

NameColumn A property defined in MGDwgDataSources

NEW Gets or sets the name of the column that contains the name for each map feature.

NameFromSdpFeatureTable A property defined in MGMapLayerSetup

Indicates whether the feature name should come from the SDP feature table or from a secondary table in an OLE DB data source.

NameTable A property defined in MGDwgDataSources

NEW Gets or sets the name of the DWG table for MapTips about feature properties, external database links, or object data in the Autodesk DWG.

NumObjects A property defined in MGSelection

Gets the number of map features in the current selection.

O

ObjectLinkTarget A property defined in MGMap

Gets or sets the name of the target window where Autodesk MapGuide will display the URL associated with a map feature.

onAddMapLayer An ActiveX Control event. Also an event method defined in MGAddMapLayerObserver

This event is fired when a new map layer has been added. It is initiated by a call to addMapLayer.

onBeginLayout An ActiveX Control event. Also an event method defined in MGPrintingObserver

This event is fired right before the Autodesk MapGuide Viewer uses the internal algorithm to lay out the page elements.

onBusyStateChanged An ActiveX Control event. Also an event method defined in MGBusyStateChangedObserver

This event is fired when the Autodesk MapGuide Viewer enters and exits busy states.

onDigitizedCircle An ActiveX Control event. Also an event method defined in MGDigitizeCircleObserver

This event is fired when the user has finished digitizing a circle.

onDigitizedPoint An ActiveX Control event. Also an event method defined in MGDigitizePointObserver

This event is fired when the user has finished digitizing a point. It is initiated by a call to digitizePoint.

onDigitizedPolygon An ActiveX Control event. Also an event method defined in MGDigitizePolygonObserver

This event is fired when the user has finished digitizing a polygon. It is initiated by a call to digitizePolygonEx.

onDigitizedPolygon An ActiveX Control event. Also an event method defined in MGDigitizePolygonObserverEx

This event is fired when the user has finished digitizing a polygon. It is initiated by a call to digitizePolygonEx.

onDigitizedPolyline An ActiveX Control event. Also an event method defined in MGDigitizePolylineObserver

This event is fired when the user has finished digitizing a polyline. It is initiated by a call to digitizePolylineEx.

onDigitizedPolyline An ActiveX Control event. Also an event method defined in MGDigitizePolylineObserverEx

This event is fired when the user has finished digitizing a polyline. It is initiated by a call to digitizePolylineEx.

onDigitizedRectangle An ActiveX Control event. Also an event method defined in MGDigitizeRectangleObserver

This event is fired when the user has finished digitizing a rectangle. It is initiated by a call to digitizeRectangle.

onDoubleClickObject An ActiveX Control event. Also an event method defined in MGDoubleClickObjectObserver

This event is fired when the user double-clicks on a map feature.

onEndLayout An ActiveX Control event. Also an event method defined in MGPrintingObserver

This event is fired after the Autodesk MapGuide Viewer has finished the internal layout of the page elements but before the map is sent to the printer.

OneObject A property defined in MGBufferSetup

Specifies whether or not one large merged buffer object should be created for all buffered features.

onMapLoaded An ActiveX Control event. Also an event method defined in MGMapLoadedObserver

This event is fired when Autodesk MapGuide loads an MWF into the Autodesk MapGuide Viewer, before the first refresh.

onSelectionChanged An ActiveX Control event. Also an event method defined in MGSelectionChangedObserver

This event is fired when the map selection has changed.

onViewChanged An ActiveX Control event. Also an event method defined in MGViewChangedObserver

This event is fired after the view of the map has been updated or refreshed.

onViewChanging An ActiveX Control event. Also an event method defined in MGViewChangingObserver

This event is fired just before the map is refreshed.

onViewedDistance An ActiveX Control event. Also an event method defined in MGViewDistanceObserver

This event is fired when a user has finished viewing distances.

onViewedDistanceEx An ActiveX Control event. Also an event method defined in MGViewDistanceExObserver

This event is fired when a user has finished viewing distances.

P

PageExtent A property defined in MGPrintInfo

Gets the size of the printable area of the page.

PageResolution A property defined in MGPrintInfo

Gets the resolution of the output device to be used for the current print operation.

PageSetup A property defined in MGMap

Gets an object that contains properties used for printing maps.

pageSetupDlg A method defined in MGMap

Displays the Page Setup dialog box, where the user can specify preferences for printing the map.

panMode A method defined in MGMap

Switches to Pan mode.

ParentObject A property defined in MGPrimitive

Gets the map feature that the primitive belongs to.

PlotScale A property defined in MGPageSetup

Gets or sets the map scale at which the map will print.

preferencesDlg A method defined in MGMap

Displays the Preferences dialog box, where the user can specify the map units, selection mode, and options for the status bar.

Primitives A property defined in MGMapObject

Gets a collection of primitives describing the map feature.

printDlg A method defined in MGMap

Uses the page setup properties, then displays the Print dialog box.

PrintPriority A property defined in MGPageElement

Gets or sets the print priority of this page element.

Priority A property defined in MGMapLayer

Gets or sets the priority of the map layer.

R

Rebuild A property defined in MGMapLayer

Gets or sets the map layer's rebuild flag.

RedlineSetup A property defined in MGMap

Gets an object containing all properties for drawing redline features.

redrawExtent A method defined in MGMap

Redraws the specified portion of the map.

redrawLegend A method defined in MGMap

Redraws the map legend.

refresh A method defined in MGMap

Refreshes (redraws) the map.

remove A method defined in MGCollection

Removes the object at the specified index from the collection.

removeAll A method defined in MGCollection

Removes all of the objects from the collection.

removeAllObjects A method defined in MGMapLayer

Deletes all features from the layer.

removeMapLayer A method defined in MGMap

Removes the specified map layer from the map.

removeMapLayer A method defined in MGMapLayerGroup

Removes a single map layer from the group.

removeMapLayerGroup A method defined in MGMap

Removes the specified map layer group from the map.

removeObject A method defined in MGMapLayer

Deletes a map feature on the layer.

removeObject A method defined in MGSelection

Removes a single map feature from the selection and updates the display.

removeObjects A method defined in MGMapLayer

Deletes the specified map features from the layer.

removeObjects A method defined in MGSelection

Removes the specified map features from the selection and updates the display.

removeObjectsEx A method defined in MGSelection

Removes multiple map features from the selection and updates the display.

ReportTarget A property defined in MGMap

Gets or sets the name of the target window where Autodesk MapGuide will display reports from View Reports command.

Rotation A property defined in MGSymbolAttr

CHANGED Gets or sets the rotation of the point, in degrees.

Rotation A property defined in MGTextAttr

Gets or sets the rotation of the text, in degrees.

S

saveMWF A method defined in MGMap

Saves the map to a local MWF.

SdpDataSource A property defined in MGMapLayerSetup

Gets or sets the SDP name for this map layer.

SdpFeatureTable A property defined in MGMapLayerSetup

Gets or sets the SDP feature table name for this map layer.

SdpGeometryColumn A property defined in MGMapLayerSetup

Gets or sets the SDP feature table geometry column name.

SdpKeyColumn A property defined in MGMapLayerSetup

Gets or sets the Spatial Data Provider feature table key column name.

SdpKeyColumnType A property defined in MGMapLayerSetup

Gets or sets the SDP feature table key column type.

Selectability A property defined in MGMapLayer

Gets or sets whether features within the map layer may be selected.

Selection A property defined in MGMap

Gets the currently selected map features.

SelectionMode A property defined in MGMap

Gets or sets the selection mode, either Centroid or Intersection.

selectMapObjectsDlg A method defined in MGMap

Displays the Select map features dialog box, where the user can select map features (MGMapObject objects) by name from a specific map layer.

selectMode A method defined in MGMap

Switches to Select mode.

selectPolygon A method defined in MGMap

Selects all map features that fall within the specified polygon.

selectPolygonEx A method defined in MGMap

Selects all map features that fall within the specified polygon.

selectPolygonMode A method defined in MGMap

Enters the select polygon mode.

selectRadius A method defined in MGMap

Selects all map features that fall within the specified radius.

selectRadiusMode A method defined in MGMap

Selects Select Radius mode.

selectWithin A method defined in MGMap

Selects features from the specified map layers that fall within the currently selected polygon map features.

selectWithinDlg A method defined in MGMap

Displays the Select Within dialog box, where the user chooses the map layer containing the features that will be selected if they fall within the currently selected features.

ServerUrl A property defined in MGMapLayerSetup

Gets or sets the Autodesk MapGuide Server URL.

set A method defined in MGExtentEx

Sets the extents of the object.

setAccessKey A method defined in MGMapLayer

Sets the access key associated with a map layer.

setAutoLinkDelay A method defined in MGMap

Sets the number of seconds before the next HTML page associated with a map feature from the AutoLinkLayer will be displayed in the AutoLinkTarget window.

setAutoLinkLayers A method defined in MGMap

Sets the name of the layer that will use the AutoLink feature.

setAutoLinkTarget A method defined in MGMap

Sets the name of the target window where Autodesk MapGuide will display the HTML pages associated with map features on the AutoLinkLayer.

setAutoRefresh A method defined in MGMap

Enables or disables the automatic refresh flag.

setBackColor A method defined in MGFillAttr

Specifies the background color index used to draw the polygons.

setBackColor A method defined in MGTextAttr

Sets the background color.

setBackMode A method defined in MGFillAttr

Specifies the background mode used to draw the polygons.

setBackMode A method defined in MGPageElement

Sets the background mode of this element.

setBackMode A method defined in MGTextAttr

Sets the display mode for the background behind text features.

setBusyStateChangedObserver A method defined in MGMap

Activates the Java object that observes the Autodesk MapGuide Viewer busy state.

setColor A method defined in MGEdgeAttr

Specifies the edge color index used to draw the polygons.

setColor A method defined in MGFillAttr

Specifies the fill color index used to draw the polygons.

setColor A method defined in MGLineAttr

Specifies the color index used to draw the lines or polylines.

setColor A method defined in MGTextAttr

Sets the text color.

setDataFile A method defined in MGMapLayerSetup

CHANGED Sets the name of the data file from which Autodesk MapGuide Server retrieves data for this layer.

setDataSource A method defined in MGDatabaseSetup

Sets the name of the OLE DB data source from which the Autodesk MapGuide Server retrieves its data.

setDataSource A method defined in MGDwgDataSources

NEW Sets the DWG data source name that establishes a connection between Autodesk MapGuide Server and the data provider.

setDefaultTarget A method defined in MGMap

Sets the name of the default target window where Autodesk MapGuide will display HTML documents.

setDisplayRange A method defined in MGMapLayerStyle

Sets the display scale range associated with a style for a particular map layer.

setDistance A method defined in MGBufferSetup

Specifies the buffer offset distance.

setDoubleClickObserver A method defined in MGMap

Sets the Java object for observing double-click events.

setDwg A method defined in MGDwgDataSources

NEW Sets the Autodesk DWG file name or project file for the Autodesk DWG layer.

setErrorTarget A method defined in MGMap

Sets the name of the target window where Autodesk MapGuide will display error messages.

setExtent A method defined in MGPageElement

Modifies the extents of this element on the printout.

setFontName A method defined in MGTextAttr

Sets the font name.

setFontStyle A method defined in MGTextAttr

Sets the font style.

setHatchPattern A method defined in MGFillAttr

Specifies the fill hatch pattern used to draw the polygons.

setHeight A method defined in MGSymbolAttr

Sets the display height of the point in the specified unit type.

setHeight A method defined in MGTextAttr

Sets the display height of the text in the specified unit type.

setHorizAlign A method defined in MGTextAttr

Sets the horizontal placement of the text features in relation to an imaginary vertical axis.

setInclude A method defined in MGPageSetup

Changes the visibility state of the page element on the print out.

setIncludeLegend A method defined in MGPageSetup

Sets a value indicating whether the legend will be printed next to the map.

setIntermediateUpdatesEnabled A method defined in MGMap

NEW Specifies whether or not the map window should update every 1.5 seconds during a drawing operation or only at the end when all data has been loaded and rendered.

setKeyColumn A method defined in MGDatabaseSetup

Sets the name of the column that contains the primary key for each map feature on the DWG layer.

setKeyColumn A method defined in MGDwgDataSources

NEW Sets the name of the column that contains the key of the features on this DWG layer.

setKeyColumnType A method defined in MGDatabaseSetup

Sets the data type of the key column to the data type you specify.

setKeyColumnType A method defined in MGDwgDataSources

NEW Sets the DWG key column data type.

setKeyTable A method defined in MGDwgDataSources

NEW Sets the name of the DWG key table to use as primary keys for either feature properties, external database links, or object data in the Autodesk DWG.

setLatColumn A method defined in MGDatabaseSetup

Sets the name of the database column that contains the latitude (or *Y* coordinate) value of map features on this layer.

setLayerFilter A method defined in MGDwgDataSources

NEW Sets the layer filter, which specifies which layers to extract from the specified Autodesk DWG or project file.

setLayerName A method defined in MGBufferSetup

Sets the name of the map layer that will contain the buffer objects generated from the viewBuffer method of MGMap.

setLegendLabel A method defined in MGMapLayer

Sets the string used to label the map layer in the legend. After calling this method, the legend is automatically redrawn.

setLegendLabel A method defined in MGMapLayerGroup

Sets the label used by the map layer group in the legend.

setLink A method defined in MGMapObject

Sets the URL link associated with the map feature.

setLinkedToDatabase A method defined in MGMapLayerSetup

Sets the NameFromSdpFeatureTable, UrlFromSdpFeatureTable, and WhereFromSdpFeatureTable properties to True or false.

setLonColumn A method defined in MGDatabaseSetup

Sets the name of the database column that contains the longitude (or *X* coordinate) values for the map features on this layer.

setMapLayerGroup A method defined in MGMapLayer

Specifies which map layer group the current layer should belong to.

setMapLayers A method defined in MGMapLayerGroup

Clears the contents of a map layer group and adds the specified map layers.

setMapLoadedObserver A method defined in MGMap

Activates the Java object that observes when Autodesk MapGuide loads MWFs, enabling the onMapLoaded event.

setName A method defined in MGMap

Sets the name of the current map.

setName A method defined in MGMapLayer

Sets the name of the map layer.

setName A method defined in MGMapLayerGroup

Sets name of the map layer group.

setName A method defined in MGMapObject

Sets the name of the map feature.

setNameColumn A method defined in MGDatabaseSetup

Sets the name of the database column containing the names of the map features on this layer.

setNameColumn A method defined in MGDwgDataSources

NEW Sets the name of the column that contains the name for each map feature.

setNameFromSdpFeatureTable A method defined in MGMapLayerSetup

Indicates whether the feature name should come from the SDP feature table or from a secondary table in an OLE DB data source.

setNameTable A method defined in MGDwgDataSources

NEW Sets the name of the DWG table for MapTips about feature properties, external database links, or object data in the Autodesk DWG.

setObjectLinkTarget A method defined in MGMap

Sets the name of the target window where Autodesk MapGuide will display the URL associated with a map feature.

setOneObject A method defined in MGBufferSetup

Specifies whether one large merged buffer object should be created for all buffered features.

setPageMargins A method defined in MGPageSetup

Sets the page margins used by the current page layout.

setPlotScale A method defined in MGPageSetup

Sets the scale if the map is to be printed at a specific map scale.

setPrintingObserver A method defined in MGMap

Sets the Java object that observes printing events.

setPrintPriority A method defined in MGPageElement

Sets the print priority of this element.

setPriority A method defined in MGMapLayer

Sets the priority of the map layer.

setRebuild A method defined in MGMapLayer

Changes the map layer's rebuild flag.

setReportTarget A method defined in MGMap

Sets the name of the target window where Autodesk MapGuide will display reports from View Reports command.

setRotation A method defined in MGSymbolAttr

CHANGED Sets the rotation of the point.

setRotation A method defined in MGTextAttr

Sets the rotation of the text.

setScale A method defined in MGMap

Sets the current map display scale in terms of 1:scale.

setSdpDataSource A method defined in MGMapLayerSetup

Sets the Spatial Data Source name for this map layer.

setSdpFeatureTable A method defined in MGMapLayerSetup

Sets the Spatial Data Source name for this map layer.

setSdpGeometryColumn A method defined in MGMapLayerSetup

Sets the name of the SDP feature table geometry column.

setSdpKeyColumn A method defined in MGMapLayerSetup

Sets the name of the SDP feature table key column.

setSdpKeyColumnType A method defined in MGMapLayerSetup

Sets the SDP feature table key column type.

setSelectability A method defined in MGMapLayer

Sets the selectability of map features within the map layer.

setSelectionChangedObserver A method defined in MGMap

Sets the Java object for observing selection change events.

setSelectionMode A method defined in MGMap

Sets the selection mode, which controls how map features are selected when a feature is not fully contained in the selection area.

setServerUrl A method defined in MGMapLayerSetup

Sets the Autodesk MapGuide Server URL to the string value that you pass in.

setShowInLegend A method defined in MGMapLayer

Sets a flag indicating whether the map layer should be displayed in the legend.

setShowInLegend A method defined in MGMapLayerGroup

Sets a flag indicating whether or not the map layer group should be displayed in the legend.

setSQLWhere A method defined in MGMapLayer

Sets the SQL Where clause for the layer.

setStyle A method defined in MGEdgeAttr

Specifies the edge style used to draw the polygons.

setStyle A method defined in MGFillAttr

Specifies the fill style used to draw the polygons.

setStyle A method defined in MGLineAttr

Specifies the style used to draw the lines or polylines.

setSymbol A method defined in MGSymbolAttr

CHANGED Sets the symbol to be used as the symbol attribute.

setSymbolAngleColumn A method defined in MGDatabaseSetup

Sets the name of the column containing the rotation angle of symbols for point features.

setSymbolHeightColumn A method defined in MGDatabaseSetup

Sets the name of the column containing the height of symbols for point features.

setSymbolWidthColumn A method defined in MGDatabaseSetup

Sets the name of the column containing the width of symbols for point features.

setTable A method defined in MGDatabaseSetup

Sets the name of the database table, within the OLE DB data source, from which Autodesk MapGuide Server retrieves data for this layer.

setTextAngleColumn A method defined in MGDatabaseSetup

Sets the name of the database column that contains the rotation angle for the text map features on this layer.

setTextHeightColumn A method defined in MGDatabaseSetup

Sets the name of the database column that contains the height of the text map features on this layer.

setTextHorizAlignColumn A method defined in MGDatabaseSetup

Sets the name of the column containing horizontal alignment values for text features.

setTextVertAlignColumn A method defined in MGDatabaseSetup

Sets the name of the column containing vertical alignment values for text features.

setThickness A method defined in MGEdgeAttr

Specifies the edge thickness used to draw the polygons.

setThickness A method defined in MGLineAttr

Specifies the thickness used to draw the lines or polylines.

setTitle A method defined in MGPageSetup

Sets the title to print at the top of the page when you print a map.

setTrackingID A method defined in MGMap

Sets the customer tracking ID associated with a map.

setUnit A method defined in MGBufferSetup

Sets the units used to specify the size of the buffer.

setUrl A method defined in MGMap

Sets the URL of the map to display.

setUrlColumn A method defined in MGDwgDataSources

NEW Sets the name of the column that contains the URL associated with each map feature.

setURLFromSdpFeatureTable A method defined in MGMapLayerSetup

Indicates whether the feature URL should come from the SDP feature table or from a secondary table in an OLE DB data source.

setUrlLinkColumn A method defined in MGDatabaseSetup

Sets the name of the database column that contains the URLs that are linked to map features on this layer.

setUrlListState A method defined in MGMap

Enables or disables automatic generation of an HTML document listing the names and URLs of map features associated with URLs.

setUrlListTarget A method defined in MGMap

Sets the name of the target window where Autodesk MapGuide will display the names and URLs associated with a map feature as the user pauses the mouse pointer over that feature.

setUrlTable A method defined in MGDwgDataSources

NEW Sets the name of the DWG URL table containing URLs to which Autodesk DWG map features are linked.

setUsePlotScale A method defined in MGPageSetup

Sets a value indicating whether the map will be printed at a specific scale.

setVertAlign A method defined in MGTextAttr

Sets the vertical placement of the text features in relation to an imaginary horizontal baseline.

setViewChangedObserver A method defined in MGMap

Sets the Java object for observing view change events.

setViewChangingObserver A method defined in MGMap

Sets the Java object for observing view change events.

setViewDistanceExObserver A method defined in MGMap

Sets the Java object for observing view distance events.

setViewDistanceObserver A method defined in MGMap

Sets the Java object for view distance events.

setVisibility A method defined in MGMapLayer

Sets the visibility flag indicating whether the map layer is potentially visible.

setVisibility A method defined in MGMapLayerGroup

Sets a flag that affects the visibility of all map layers in the group.

setVisible A method defined in MGEdgeAttr

Specifies whether the edge should be drawn around the polygons.

setWhereClause A method defined in MGDatabaseSetup

Sets the SQL Where clause that Autodesk MapGuide uses to query the OLE DB data source that provides data for this layer.

setWhereFromSdpFeatureTable A method defined in MGMapLayerSetup

Indicates whether the feature SQL Where clause should come from the SDP feature table or from a secondary table in an OLE DB data source.

setWidth A method defined in MGMap

Sets the current width of the map in the specified unit of measure.

setWidth A method defined in MGSymbolAttr

Sets the display width of the point in the specified unit type.

setX A method defined in MGPoint

Sets the *X* value (longitude) of the point.

setY A method defined in MGPoint

Sets the *Y* value (latitude) of the point.

showDocument A method defined in MGMapContext

Displays a new document in an HTML window or frame, with or without a specified target.

ShowInLegend A property defined in MGMapLayer

Specifies whether or not to show the map layer in the legend.

ShowInLegend A property defined in MGMapLayerGroup

Specifies whether or not the map layer group should be shown in the legend.

showStatus A method defined in MGMapContext

Displays a message in the application context.

size A method defined in MGCollection

Gets the number of map features in the collection that have been displayed in the view since the last reload or rebuild of the map layer.

SourceType A property defined in MGMapLayerSetup

Gets the source of the map data on this layer.

SQLWhere A property defined in MGMapLayer

Gets or sets the SQL Where clause for the dynamic layer.

Static A property defined in MGMapLayer

Returns True if the map layer is static.

StatusBar A property defined in MGMap

Gets or sets the state of the status bar, either Off or On.

stop A method defined in MGMap

Aborts any downloading or redrawing that is currently taking place.

stopEx A method defined in MGMap

Aborts any downloading or redrawing that is currently taking place.

Style A property defined in MGEdgeAttr

Gets or sets the edge style used to draw the polygons.

Style A property defined in MGFillAttr

Gets or sets the fill style used to draw the polygons.

Style A property defined in MGLineAttr

Gets or sets the style used to draw the lines or polylines.

Symbol A property defined in MGSymbolAttr

CHANGED Gets or sets the symbol to be used as the point attribute.

SymbolAngleColumn A property defined in MGDatabaseSetup

Gets or sets the name of the column containing the rotation angle of symbols for point features.

SymbolAttr A property defined in MGRedlineSetup

Returns an object that groups the redline style's symbol attributes.

SymbolHeightColumn A property defined in MGDatabaseSetup

Gets or sets the name of the column containing the height of symbols for point features.

Symbols A property defined in MGMap

Gets a collection of strings representing the symbols available for use as point attributes.

SymbolWidthColumn A property defined in MGDatabaseSetup

Gets or sets the name of the column containing the width of symbols for point features.

T

Table A property defined in MGDatabaseSetup

Gets or sets the name of the database table, within the OLE DB data source, from which Autodesk MapGuide Server retrieves data for this layer.

TextAngleColumn A property defined in MGDatabaseSetup

Gets or sets the name of the database column that contains the rotation angle of text map features on this layer.

TextAttr A property defined in MGRedlineSetup

Returns an object that groups the redline style's text attributes.

TextHeightColumn A property defined in MGDatabaseSetup

Gets or sets the name of the database column that contains the height of text map features on this layer.

TextHorizAlignColumn A property defined in MGDatabaseSetup

Gets or sets the name of the column containing horizontal alignment values for text.

TextValue A property defined in MGPrimitive

Returns the string value of a text primitive.

TextVertAlignColumn A property defined in MGDatabaseSetup

Gets or sets the name of the column containing vertical alignment values for text.

Thickness A property defined in MGEdgeAttr

Gets or sets the edge thickness used to draw the polygons.

Thickness A property defined in MGLineAttr

Gets or sets the thickness used to draw the lines or polylines.

Title A property defined in MGPageSetup

Gets or sets the title that will print at the top of the page when you print a map.

ToolBar A property defined in MGMap

Gets or sets the state of the toolbar, either Off, On, or Big.

TrackingID A property defined in MGMap

Gets or sets the customer tracking ID associated with a map.

Type A property defined in MGMapLayer

Gets the type of features the layer contains.

Type A property defined in MGMapObject

Gets the type of the map feature.

Type A property defined in MGPrimitive

Gets the map feature primitive type.

U

Unit A property defined in MGBufferSetup

Gets or sets the distance unit, which is used with setDistance and getDistance to determine the distance that buffers will extend from the selected features.

unlock A method defined in MGMapLayer

Gives Autodesk MapGuide Viewer access to either coordinate values or map layer setup data, depending on the value of the function parameter that is passed in.

Url A property defined in MGMap

Gets or sets the URL of the map that is currently displayed by Autodesk MapGuide.

UrlColumn A property defined in MGDwgDataSources

NEW Gets or sets the name of the column that contains the URL associated with each map feature.

URLFromSdpFeatureTable A property defined in MGMapLayerSetup

Indicates whether the feature URL should come from the SDP feature table or from a secondary table in an OLE DB data source.

UrlLinkColumn A property defined in MGDatabaseSetup

Gets or sets the name of the database column that contains the URLs that are linked to map features on this layer.

UrlListState A property defined in MGMap

Specifies whether Autodesk MapGuide will generate an HTML document that lists the names and URLs (if any) that are associated with the selected map feature(s).

UrlListTarget A property defined in MGMap

Gets or sets the name of the target window where Autodesk MapGuide will display the names and URLs of map features associated with URLs.

UrlTable A property defined in MGDwgDataSources

NEW Gets or sets the name of the DWG URL table containing URLs to which Autodesk DWG map features are linked.

UsePlotScale A property defined in MGPageSetup

Gets or sets whether to print the map at a specific map scale.

V

Version A property defined in MGMap

Gets the version string of the Autodesk MapGuide Viewer.

VertAlign A property defined in MGTextAttr

Sets the vertical placement of the text features in relation to an imaginary vertical baseline.

viewBuffer A method defined in MGMap

Creates buffers around the selected map features.

viewBufferDlg A method defined in MGMap

Displays the View Buffer dialog box, where the user specifies the settings for creating the buffers.

viewDistance A method defined in MGMap

Enters View Distance mode and fires the onViewedDistance event when complete.

viewDistanceEx A method defined in MGMap

Enters View Distance mode and fires the onViewedDistanceEx event when complete.

ViewerType A property defined in MGMap

Returns a string indicating whether the Autodesk MapGuide Viewer is the Autodesk MapGuide Viewer ActiveX Control, the Autodesk MapGuide Viewer Plug-In, or the Autodesk MapGuide Viewer, Java Edition.

viewReport A method defined in MGMap

Invokes the specified report as authored in the map window.

viewReportsDlg A method defined in MGMap

Displays the View Reports dialog box, where the user selects the report to display.

Visibility A property defined in MGMapLayer

Gets or sets whether the map layer is potentially visible.

Visibility A property defined in MGMapLayerGroup

Specifies whether or not map layers in the group are potentially visible.

Visible A property defined in MGEdgeAttr

Gets or sets whether the edge should be drawn around the polygons.

Visible A property defined in MGMapLayer

Returns True if the map layer is visible at the current map scale.

W

WhereClause A property defined in MGDatabaseSetup

Gets or sets the SQL Where clause that Autodesk MapGuide Server uses to query the OLE DB data source that provides data for this layer.

WhereFromSdpFeatureTable A property defined in MGMapLayerSetup

Indicates whether the SQL Where clause should come from the SDP feature table or from a secondary table in an OLE DB data source.

Width A property defined in MGSymbolAttr

Gets or sets the display width of the point in the specified unit type.

X

X A property defined in MGPoint

Gets or sets the *X* value (longitude) of the point.

Y

Y A property defined in MGPoint

Gets or sets the *Y* value (latitude) of the point.

Z

zoomGotoDlg A method defined in MGMap

Displays the Zoom Goto dialog box, where the user specifies the zoom width as well as either a place (location) or an address or a ZIP code to zoom to.

zoomGotoLocation A method defined in MGMap

Zooms to the specified category, location, and scale.

zoomInMode A method defined in MGMap

Enters Zoom In mode.

zoomOut A method defined in MGMap

Zooms to the full extent of the map window as defined in the map window properties.

zoomOutMode A method defined in MGMap

Enters the zoom out mode.

zoomPrevious A method defined in MGMap

Sets the current view of the map back to the previous view of the map.

zoomScale A method defined in MGMap

Zooms to the specified map center (lat, lon) and scale.

zoomScaleDlg A method defined in MGMap

Displays the Zoom Scale dialog box, where the user enters the scale to zoom to.

zoomSelected A method defined in MGMap

Zooms to an extent covering the selected map features.

zoomWidth A method defined in MGMap

Zooms to the specified map center and width.

zoomWidthDlg A method defined in MGMap

Displays the Zoom Width dialog box, where the user enters the width of the map window to zoom to.