Autodesk MapGuide®

autodesk®

What's New in Release 6.5

15607-010000-5000A April 2004

Copyright © 2004 Autodesk, Inc.

All Rights Reserved

This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

AUTODESK, INC. MAKES NO WARRANTY, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, REGARDING THESE MATERIALS AND MAKES SUCH MATERIALS AVAILABLE SOLELY ON AN "AS-IS" BASIS.

IN NO EVENT SHALL AUTODESK, INC. BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF PURCHASE OR USE OF THESE MATERIALS. THE SOLE AND EXCLUSIVE LIABILITY TO AUTODESK, INC., REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE OF THE MATERIALS DESCRIBED HEREIN.

Autodesk, Inc. reserves the right to revise and improve its products as it sees fit. This publication describes the state of this product at the time of its publication, and may not reflect the product at all times in the future. This work is protected by U.S. and international copyright laws. Its use is subject to the terms of the license agreement included with this software and agreed to during the installation process.(c) 1982-2003 Autodesk, Inc. All rights reserved.

Autodesk Trademarks

The following are registered trademarks of Autodesk, Inc., in the USA and/or other countries: 3D Props, 3D Studio, 3D Studio MAX, 3D Studio VIZ, 3DSurfer, 3ds max, ActiveShapes, ActiveShapes (logo), Actrix, ADI, AEC Authority (logo), AEC-X, Animator Pro, Animator Studio, ATC, AUGI, AutoCAD, AutoCAD LT, AutoCAD Map, Autodesk, Autodesk Inventor, Autodesk (logo), Autodesk MapGuide, Autodesk Streamline, Autodesk University (logo), Autodesk View, Autodesk WalkThrough, Autodesk World, AutoLISP, AutoSketch, backdraft, Biped, bringing information down to earth, Buzzsaw, CAD Overlay, Character Studio, Cinepak, Cinepak (logo), Codec Central, Combustion, Design Your World, Design Your World (logo), Discreet, EditDV, Education by Design, gmax, Heidi, HOOPS, Hyperwire, i-drop, Inside Track, IntroDV, Kinetix, MaterialSpec, Mechanical Desktop, NAAUG, ObjectARX, PeopleTracker, Physique, Planix, Powered with Autodesk Technology (logo), ProjectPoint, RadioRay, Reactor, Revit, Softdesk, Texture Universe, The AEC Authority, The Auto Architect, VISION*, Visual, Visual Construction, Visual Drainage, Visual Hydro, Visual Landscape, Visual Roads, Visual Survey, Visual Toolbox, Visual Tugboat, Visual LISP, Volo, WHIP!, and WHIP! (logo).

The following are trademarks of Autodesk, Inc., in the USA and/or other countries: AutoCAD Learning Assistance, AutoCAD LT Learning Assistance, AutoCAD Simulator, AutoCAD SQL Extension, AutoCAD SQL Interface, Autodesk Envision, Autodesk Map, AutoSnap, AutoTrack, Built with ObjectARX (logo), Burn, Buzzsaw.com, CAiCE, Cinestream, Cleaner, Cleaner Central, ClearScale, Colour Warper, Content Explorer, Dancing Baby (image), DesignCenter, Design Doctor, Designer's Toolkit, DesignProf, DesignServer, Design Web Format, DWF, DWFwriter, DWG Linking, DXF, Extending the Design Team, GDX Driver, gmax (logo), gmax ready (logo),Heads-up Design, jobnet, lustre, ObjectDBX, onscreen onair online, Plans & Specs, Plasma, PolarSnap, Real-time Roto, Render Queue, Visual Bridge, Visual Syllabus, and Where Design Connects.

Autodesk Canada Inc. Trademarks

The following are registered trademarks of Autodesk Canada Inc. in the USA and/or Canada, and/or other countries: discreet, fire, flame, flint, flint RT, frost, glass, inferno, MountStone, riot, river, smoke, sparks, stone, stream, vapour, wire. The following are trademarks of Autodesk Canada Inc., in the USA, Canada, and/or other countries: backburner, Multi-Master Editing.

Third Party Trademarks

Microsoft, PowerPoint, and Windows, are registered trademarks of Microsoft Corporation in the United States and other countries. Oracle is a registered trademark of Oracle Corporation. ESRI is a trademark of Environmental Systems Research Institute, Inc., in the United States and other countries. All other brand names, product names or trademarks belong to their respective holders.

Third Party Software Program Credits

Copyright © 2002 Microsoft Corporation. All rights reserved. InstallShield™ Copyright © 2002 InstallShield Software Corporation. All rights reserved. Typefaces from the Bitstream (R) typeface library (c) 1992. Typefaces from Payne Loving Trust (c) 1996. All rights reserved. International CorrectSpell(TM) Spelling Correction System (c) 1995 by Lernout & Hauspie Speech Products, N.V. All rights reserved. ACIS (c) 1989-2001 Spatial Corp. All rights reserved. Active Delivery(TM) 2.0. (c) 1999-2000 Inner Media, Inc. All rights reserved. Portions (c) 1991-1996 Arthur D. Applegate. All rights reserved. Portions of this software are based on the work of the Independent JPEG Group. AnswerWorks 4.0 (c) 1997-2002 WexTech Systems, Inc. Portions of this software (c) Lernout & Hauspie, Inc. All Rights Reserved. This product includes software developed by the Apache Software Foundation .subject to its license terms and conditions (http://xml.apache.org/dist/LICENSE.txt). Copyright (c) 1999-2000 The Apache Software Foundation. All rights reserved. PANTONE(R) Colors displayed here may not match PANTONE-identified standards. Consult current PANTONE Color Publications for accurate color. PANTONE(R) and other Pantone, Inc. trademarks are the property of Pantone, Inc. (c) Pantone, Inc., 2002. The Director General of the Geographic Survey Institute has issued the approval for the coordinates exchange numbered TKY2JGD for Japan Geodetic Datum 2000, also known as technical information No H1-N0.2 of the Geographic Survey Institute, to be installed and used within this software product (Approval No.: 646 issued by GSI, April 8, 2002). OSTN97 (c) Crown copyright 1997. All rights reserved. OSTN02 (c) Crown copyright 2002. All rights reserved. OSGM02 (c) Crown copyright 2002, (c) Ordnance Survey Ireland, 2002. RAL DESIGN (c) RAL, Sankt Augustin, 2002. RAL CLASSIC (c) RAL, Sankt Augustin, 2002. Representation of the RAL Colors is done with the approval of RAL Deutsches Institut für Gütesicherung und Kennzeichnung e.V. (RAL - German Institute for Quality Assurance and Certification, reg. Assoc.), D-53757 Sankt Augustin. Please note that the representation of colors on monitors can only approximate the actual color shades as they are registered as lacquered samples. Neither the quality of the present software nor the hardware used is responsible for it. (c) Stade de France - Macary, Zublena et Regembal, Costantini - Architectes, ADAGP - Paris - 2003.

GOVERNMENT USE

Use, duplication, or disclosure by the U. S. Government is subject to restrictions as set forth in FAR 12.212 (Commercial Computer Software-Restricted Rights) and DFAR 227.7202 (Rights in Technical Data and Computer Software), as applicable.

1 2 3 4 5 6 7 8 9 10

Contents

Part I

Autodesk MapGuide Server and Author Enhancements

The chapters in Part I document the changes and additions made to Autodesk MapGuide Server and Autodesk MapGuide Author to support the new features in this release.

Introduction

1

This document describes new and changed features in Autodesk MapGuide® Release 6.5.

In this chapter

- Installation notes
- Upgrading from an earlier release
- Major new features

Installation Notes

For complete installation and licensing information, please refer to the *Installation Instructions* PDF file, which is available from the menu that appears when you insert the Autodesk MapGuide installation CD.

For late-breaking installation updates and known issues, please refer to the *Autodesk MapGuide Readme* (*MGReadme.htm*) located on the Autodesk MapGuide installation CD and to the *Autodesk MapGuide LiteView Readme* (*LVReadme.htm*) located on the Autodesk MapGuide LiteView installation CD.

Upgrading From an Earlier Release

The MWF file format has changed for this release. If you open MWF files from Release 6.3 or earlier in this release of Autodesk MapGuide Author, they will be converted to the new format. If you want to keep the old files, be sure to make backup copies before you save the new files.

If you are using DWG data created by Autodesk Map 3D 2005 (released in April 2004), note that the default location of the *acadmap.ini* file is now *C:\Documents and Settings\All Users\Application Data\Autodesk\Autodesk Map 3D 2005\R16.1\acadmap.ini*

If you are upgrading Autodesk MapGuide from a release earlier than Release 6.3, you may also want to read the following documentation:

- *What's New in Release 6.3* (PDF file)
- *What's New in Release 6* section in the *Autodesk MapGuide User's Guide*

Major New Features

New features in Autodesk MapGuide Release 6.5 include:

- Enhanced support for DWG format, including layers created directly from DWG queries and themes. See page 5.
- Enhanced support for DWF™ format, including layers created from DWF files. See page 23.
- Enhanced layer functionality, with direct access to the analytical power of databases such as Oracle®. See page 39.

Improved DWG Support

2

This chapter describes the enhanced support for DWG files provided in this release of Autodesk® MapGuide. Most of the changes covered in this chapter apply to Autodesk MapGuide Author—there is only one change to the Autodesk MapGuide Server Admin program. Follow the procedures in this chapter to learn the new methods of authoring layers directly from DWG files.

To review the new API functions for the DWG format, see "DWG API Additions" on page 81.

In this chapter

Summary of Enhancements

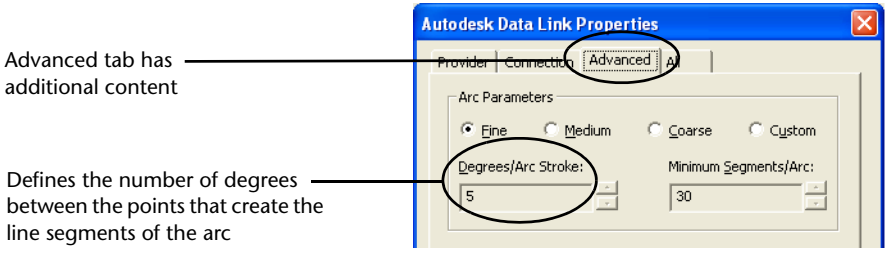
This release of Autodesk MapGuide provides greatly improved support for DWG files, particularly those created in Autodesk Map. In previous releases, you could only bring in the entire DWG or certain layers from it. To extract geometry of a particular type from the DWG file, it was necessary to convert the polygons, points, or polylines to individual SDF files first.

The new release gives you much greater flexibility in working with DWG files. The following is a complete list of DWG-related enhancements, showing what you can now do in Autodesk MapGuide:

- Create a new layer directly from an Autodesk Map query. See "Creating a Layer from a Query" on page 10.
- Create a new layer directly from an Autodesk Map theme. See "Creating a Layer from a Theme" on page 13.
- Create a new layer directly from objects of a particular geometry type. See "Creating a Layer By Geometry Type" on page 15.
- Theme layers created from objects of a particular geometry type. Autodesk MapGuide now provides the same support for DWG data sources as it does for SDF. See "Theming a DWG Layer" on page 17.
- Link secondary tables to layers created from objects of a particular geometry type. Autodesk MapGuide now provides the same support for DWG data sources as it does for SDF. See "Using a Secondary Table" on page 19.
- Access linked-template external data. All columns of linked tables are available for selection. See "Accessing Data Using a Link Template" on page 20.

Dialog Box Changes: Autodesk MapGuide Server

New arc parameters have been added to the Autodesk Data Link Properties dialog box. You access this dialog box by editing the properties of a DWG data source in the Autodesk MapGuide Server Admin program. Using the parameters, you can now specify how finely arcs are rendered as polylines when they are brought into Autodesk MapGuide (note that finer arcs slow performance).



New Advanced tab for Autodesk Data Link Properties dialog box

Dialog Box Changes: Autodesk MapGuide Author

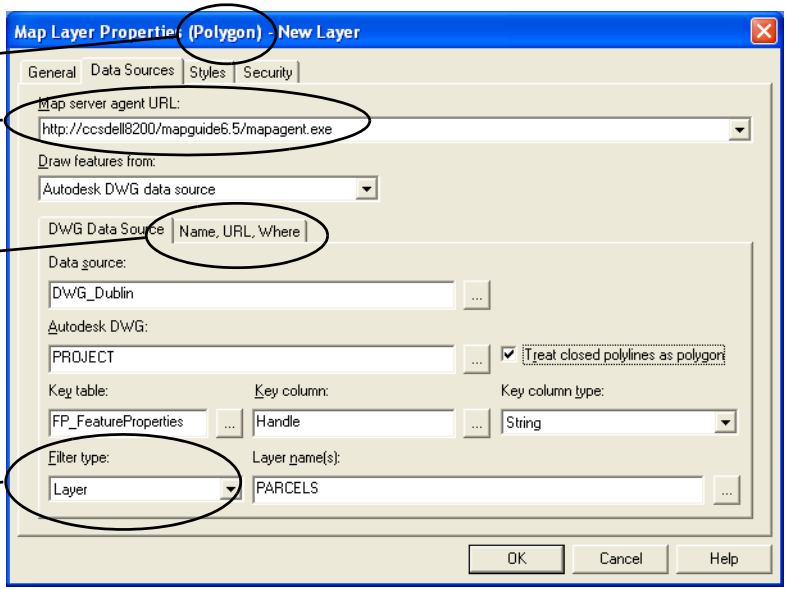
The Map Layer Properties dialog box has been rearranged and updated to make it easier to use and to provide a logical interface for the new DWG features.

Layer type appears in the title bar

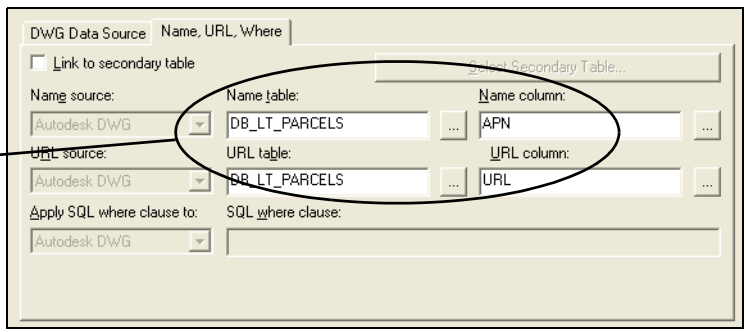
Map Server Agent URL has been moved to top of work area

Name, URL, Where tab has been added. (See the illustration below.)

For Filter Type, you can select Layer, Query, or Theme.



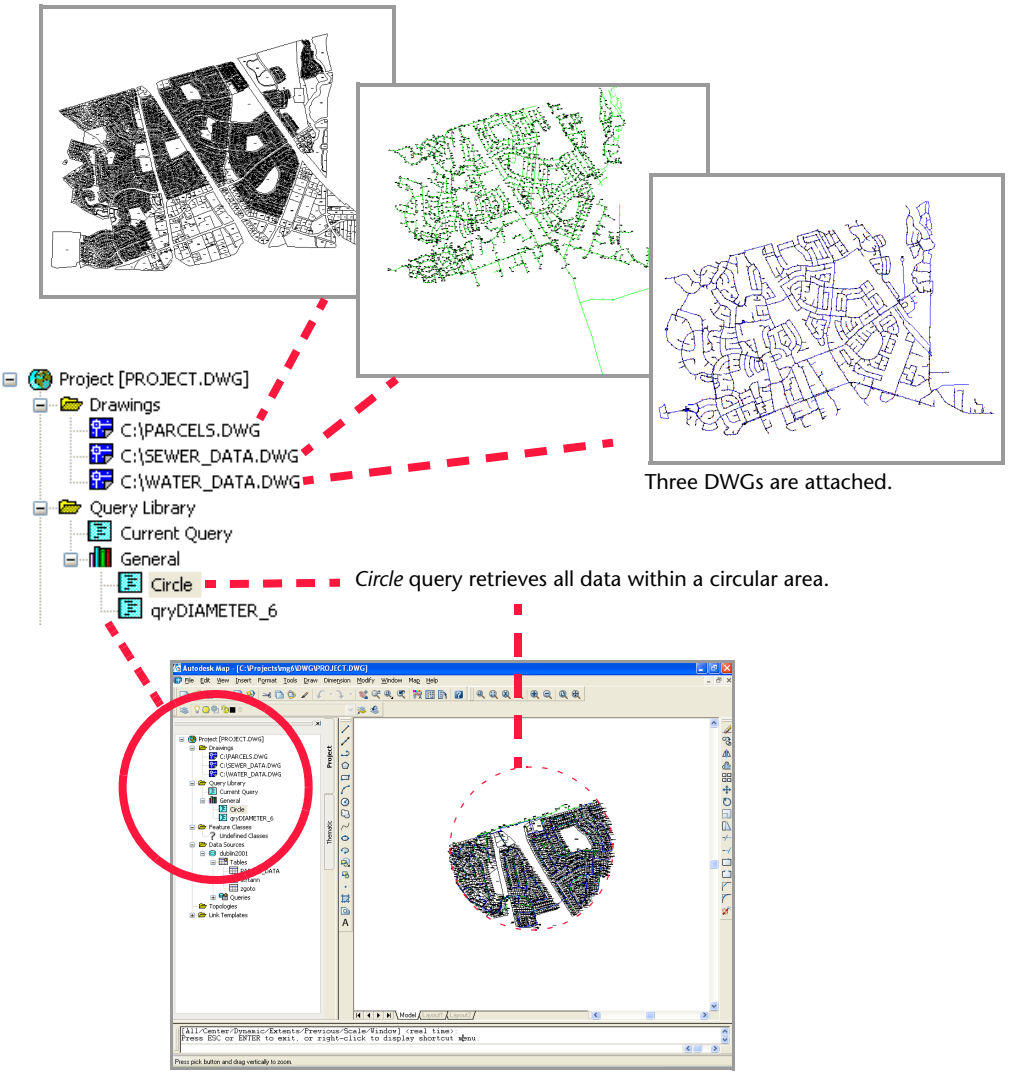
Name and URL fields to access tables stored in the DWG or linked by a link template



New Map Layer Properties dialog box for DWG layer

How Queries Work in Autodesk Map

In Autodesk Map, queries are used to extract a subset of data from a series of DWG files. Queries can be saved in the DWG file. You can use these saved queries to create new layers in Autodesk MapGuide. The following diagram illustrates how a simple location query is created in Autodesk Map:



A typical query created in Autodesk Map

In the illustration on the facing page, the drawing file *PROJECT.DWG* has three drawings attached to it: *PARCELS.DWG*, *SEWER_DATA.DWG*, and *WATER_DATA.DWG*. When the the *Circle* query is executed, it extracts all the data that falls within a pre-defined circle from all three attached files.

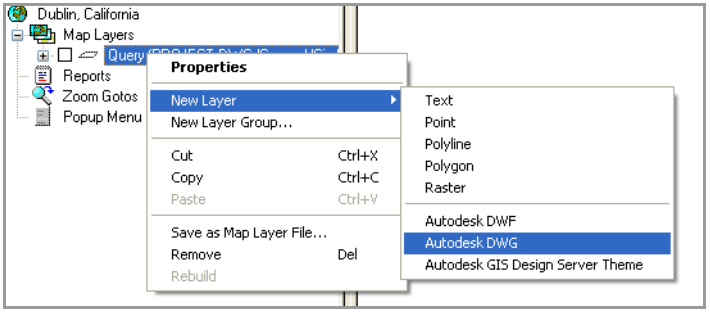
The example in the illustration is a very simple query, called a location query because it is based on a location in the drawing (in this case, a circular area). Queries can be much more complex than this one, with multiple lines that include SQL statements. However, to bring data from any Autodesk Map query into Autodesk MapGuide Author, the procedure is the same.

Creating a Layer from a Query

Use the following steps to create a new layer from a query in Autodesk MapGuide.

To create a new layer from a query

1 In Autodesk MapGuide Author, right-click Map Layers and then click New Layer ➤ Autodesk DWG.



2 On the General tab, enter a name for the new layer.

3 Click the Data Sources tab, and make sure that the map server agent path is pointing to the correct location.

4 Click the Data Source Browse button, and select a DWG data source.

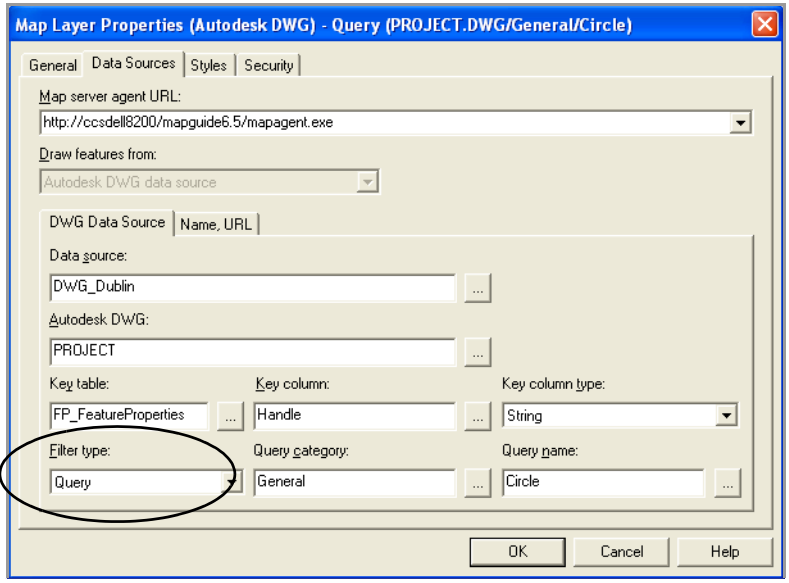
5 Click the Autodesk DWG Browse button, and select a DWG file.

6 Optionally, select a key table and key column if you want to associate table data with the geometry on this layer. For more information, see the topic “About DWG Data Sources” in the Autodesk MapGuide Help.

7 For Filter Type, select Query.

8 Click the Query Name Browse button, and select a query.

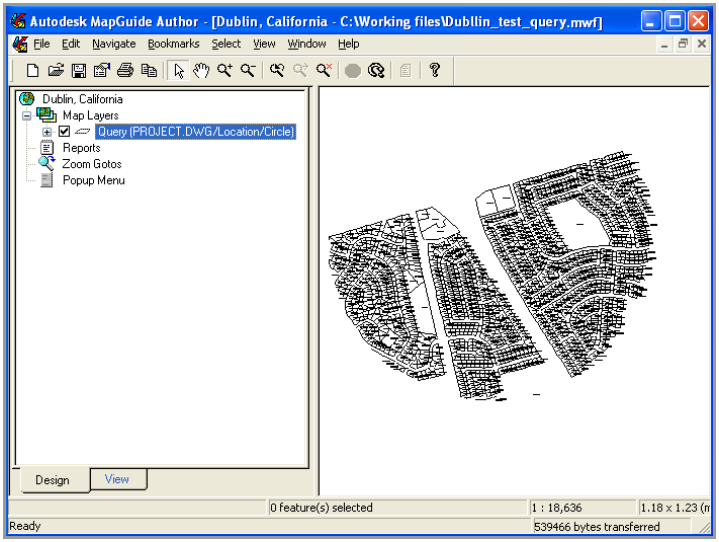
You can also click the Query Category Browse button to select a particular query category. If there are many queries defined in the drawing, this can make the list more manageable.



Filter type set to Query

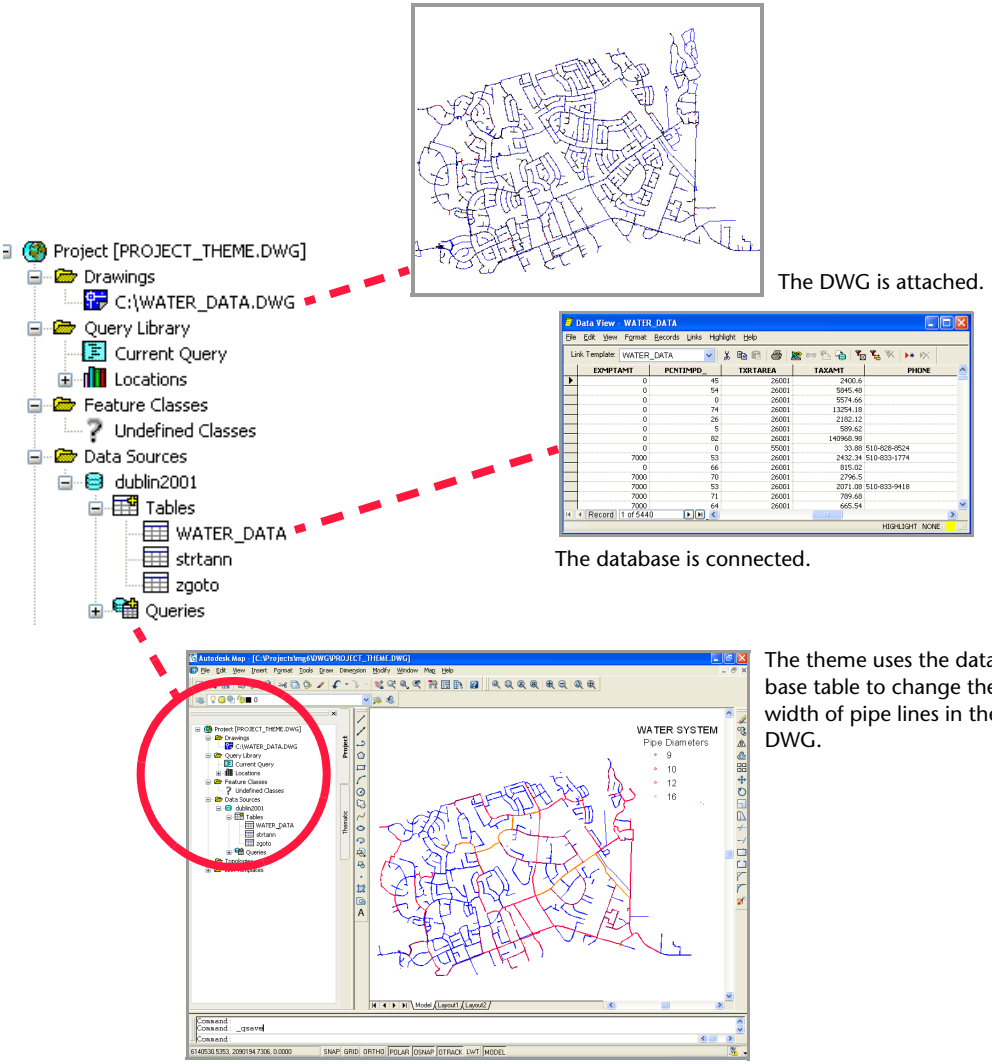
9 Click OK.

The data specified by the query is retrieved from the server and displayed in Autodesk MapGuide Author.



How Themes Work in Autodesk Map

In Autodesk Map, themes are used to create thematic maps from data contained in, or linked to, a series of DWG files. You can use these themes to create new layers in Autodesk MapGuide. The following diagram illustrates how a simple theme is created in Autodesk Map:



A typical theme created in Autodesk Map

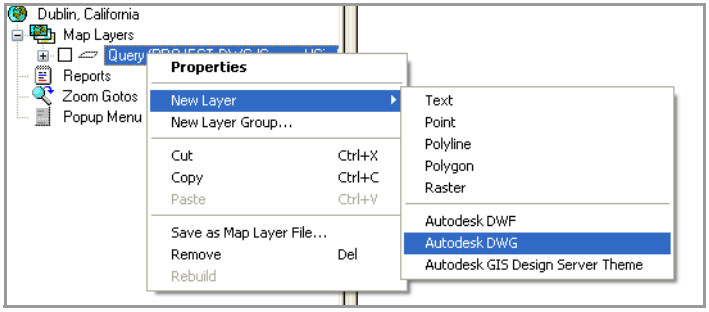
In the illustration on the previous page, the drawing file *PROJECT_THEME.DWG* has the *WATER_DATA.DWG* drawing attached to it. The *PROJECT_THEME.DWG* drawing file is also linked to a database that contains information about the pipes that make up the city's water system. A theme has been created to show the different pipe diameters. When the theme is executed, it reads the data from the database and redraws the thickness of the polylines in the drawing according to the diameter of the pipes. The theme also has a legend, which was placed manually in the drawing after the theme was created.

Creating a Layer from a Theme

Follow these steps to create a new layer from a theme in Autodesk MapGuide.

To create a new layer from a theme

1 In Autodesk MapGuide Author, right-click Map Layers and then click New Layer ➤ Autodesk DWG.



2 On the General tab, enter a name for the new layer.

3 Click the Data Sources tab, and make sure that the map server agent path is pointing to the correct location.

4 Click the Data Source Browse button, and select a DWG data source.

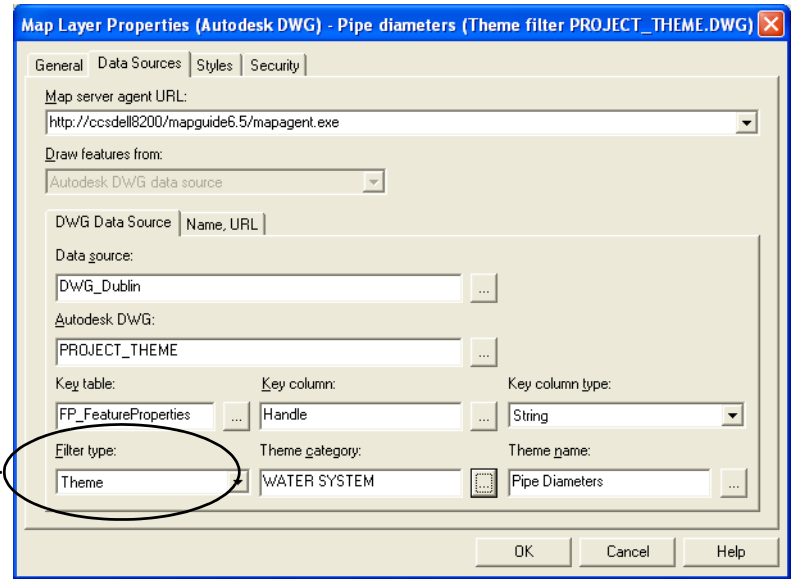
5 Click the Autodesk DWG Browse button, and select a DWG file.

6 Optionally, select a key table and key column if you want to associate table data with the geometry on this layer. For more information, see the topic "About DWG Data Sources" in the Autodesk MapGuide Help.

7 For Filter Type, select Theme.

8 Click the Theme Name Browse button, and select a theme.

You can also click the Theme Category Browse button to select a particular category of theme. If there are many themes defined in the drawing, this can make the list more manageable.



Filter type set to Theme

9 Click OK.

The data specified by the theme is retrieved from the server and displayed in Autodesk MapGuide Author.

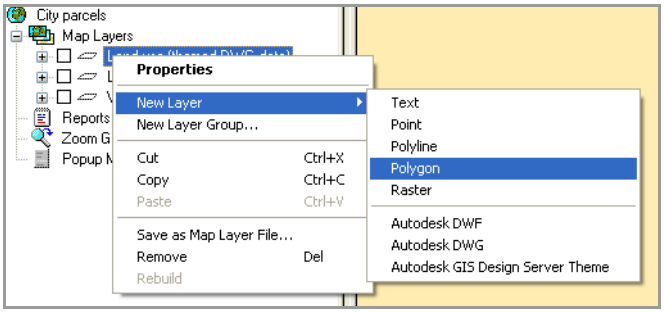
Creating a Layer By Geometry Type

You can create new layers of a particular geometry type, such as polyline, polygon, or point, directly from a DWG file. You create these layers in the usual way, by selecting the type of geometry that the layer contains; the only difference in using DWG files is that when you select the data source, you select Autodesk DWG.

Note Text layers cannot be created directly from DWG files. If you want to extract text objects from a DWG file, and want to retain attributes such as rotation, height, justification, import those layers into Autodesk MapGuide as regular Autodesk DWG layers. For more information, see the topic "Creating a DWG Map Layer," in the Autodesk MapGuide Help.

To create a new layer by geometry type

1 In Autodesk MapGuide Author, right-click Map Layers and then click either New Layer ➤ Point, New Layer ➤ Polyline, or New Layer ➤ Polygon.



2 On the General tab, enter a name for the new layer.

3 Click the Data Sources tab, and make sure that the map server agent path is pointing to the correct location.

4 For Draw Features From, select Autodesk DWG Data Source.

5 Click the Data Source Browse button, and select a DWG data source.

6 Click the Autodesk DWG Browse button, and select a DWG file.

If you are working with polygons, you can click Treat Closed Polylines as Polygons. If you select this option, the polygon layer treats the closed polylines brought in from the DWG as polygons. If you do not select this option, the clos ed polylines are ignored.

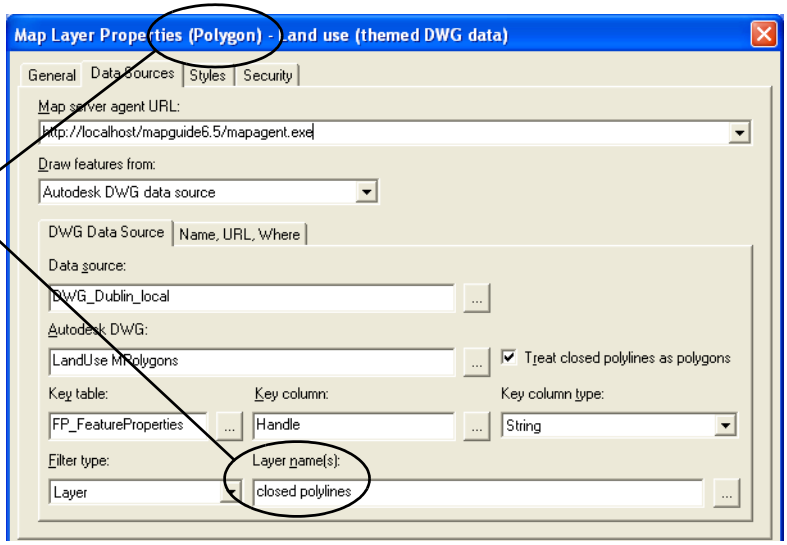
If you are working with points, you can click Treat Blocks as Points. If you select this option, the point layer treats the blocks brought in from the DWG as points. If you do not select this option, the blocks are ignored.

7 Optionally, select a key table and key column if you want to associate table data with the geometry on this layer. For more information, see the topic "About DWG Data Sources" in the Autodesk MapGuide Help.

8 For Filter Type, select either Layer, Query, or Theme.

You can use any of the three filtering methods to specify the geometry that you want to bring into your map.

When you are creating a geometry layer, make sure that the geometry you are bringing in consists of appropriate objects.



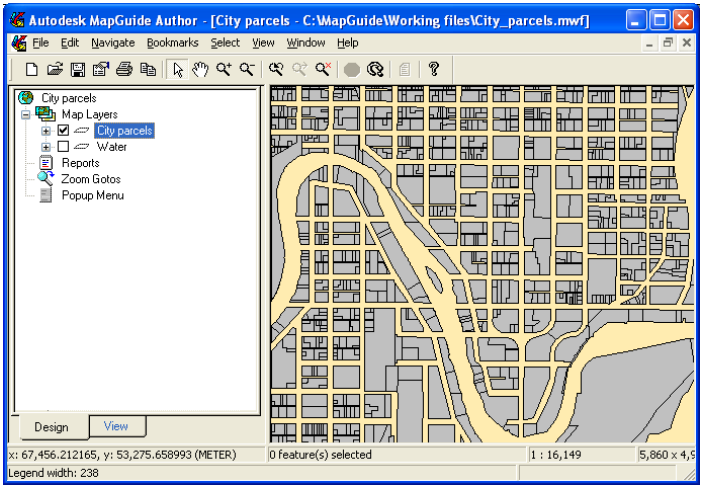
If you want to bring in the entire drawing, leave the Layer Name field blank.

9 Click OK.

Note For performance reasons, Autodesk MapGuide does not check the geometry types of the objects you specify. Therefore, after you have created a new layer from selected layers in the DWG, you should double-check that the objects that have been placed on the layer are what you expected.

Also, be aware that if the query or theme has changed since you authored your map, the geometry on the layer may not be as expected.

The geometry you specified is retrieved from the server and displayed in Autodesk MapGuide Author.



Theming a DWG Layer

You can theme new layers of a particular geometry type that you are bringing into Autodesk MapGuide Author from a DWG file. The steps to theme geometry from a DWG file are exactly the same as the steps to theme SDF geometry. The following procedure assumes that you have already brought in the geometry (see "Creating a Layer By Geometry Type" on page 15), and you now wish to apply a theme based on object data stored in the DWG file.

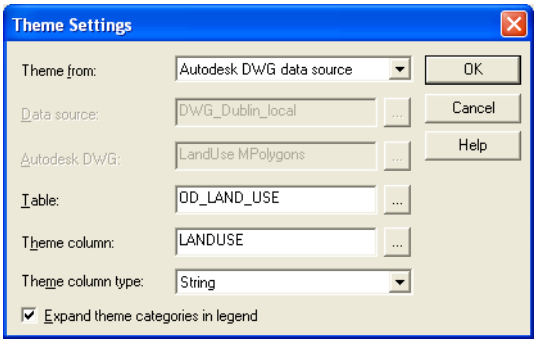
To theme a DWG layer from object data

1. In Autodesk MapGuide Author, double-click the DWG layer to display the Map Layer Properties dialog box.

   Note If you need more information at any point in this procedure, click Help in the current dialog box.

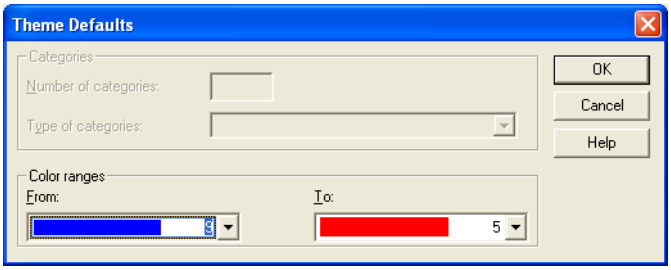
2. In the Map Layer Properties dialog box, click the Styles tab,
3. On the Styles tab, select Theme, and then click Theme Settings.
4. Click the Table Browse button, and select the table.

5 Click the Theme Column Browse button, and select the Theme Column.



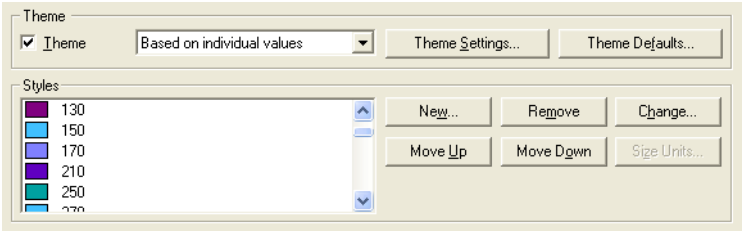
6 Click OK to return to the Map Layer Properties dialog box.

7 Click Theme Defaults and then define the categories for your theme.



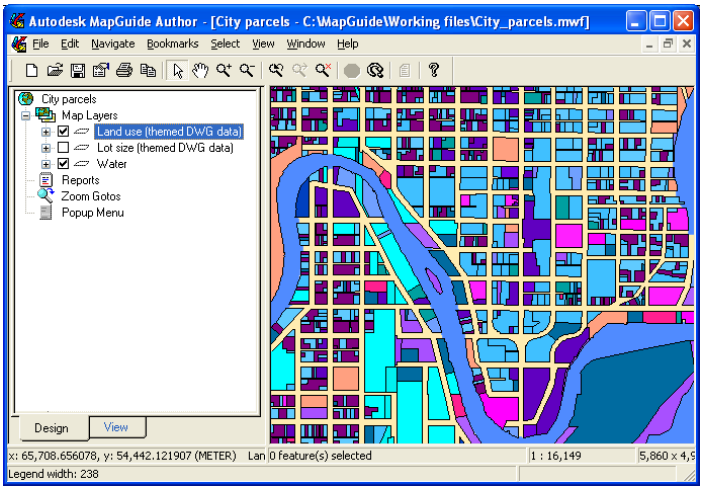
8 Click OK to return to the Map Layer Properties dialog box.

9 Click Change to modify any of the styles displayed in the list.



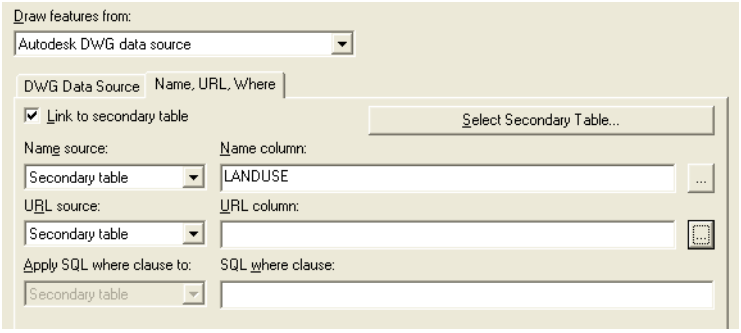
10 Click OK when you have finished.

The geometry is themed according to the styles you specified and displayed in Autodesk MapGuide Author.



Using a Secondary Table

You can use a secondary table to associate additional data with the geometry you are bringing into Autodesk MapGuide Author from a DWG file. The steps to use a secondary table with a DWG file are almost exactly the same as the steps to use a secondary table with SDF geometry. The Map Layer Properties dialog box looks slightly different, as shown in the following illustration.



For more information about secondary tables, see the topic “Data Sources tab: Name, URL, Where tab (Map Layer Properties dialog box)” in the Autodesk MapGuide Help.

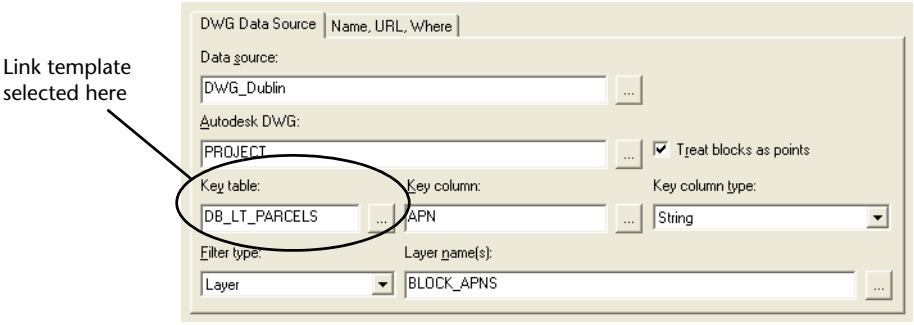
Accessing Data Using a Link Template

In Autodesk Map, a link template associates objects in the drawing with records in an external database. In this release of Autodesk MapGuide, you can access all the columns of any database table that is linked to the drawing by a link template. For information about the other kinds of database tables that you can link, see the topic "About DWG Data Sources" in the Autodesk MapGuide Help.

Note When using DWG link templates in Autodesk MapGuide, the UDL files for the link template data sources must be copied from the Autodesk Map Data Links folder to the Autodesk MapGuide Server Data Sources folder. Otherwise, MapGuide will be unable to access the data source. You can find the location of the UDL files used by Autodesk Map on the Files tab of the Options dialog box.

To access external database tables using a link template

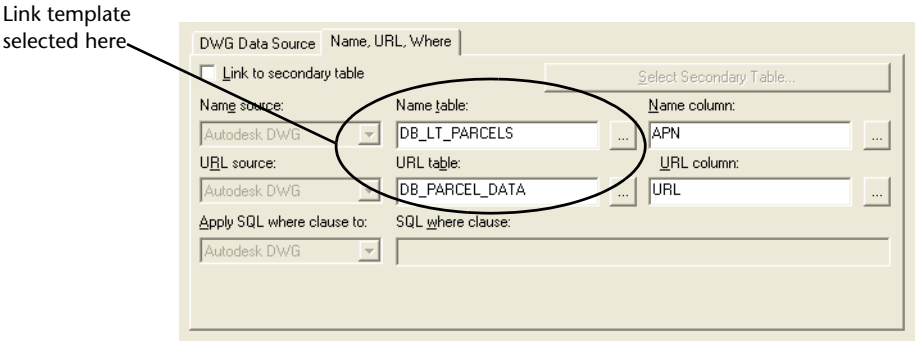
1. In Autodesk MapGuide Author, right-click Map Layers and then click any of the DWG layer-creation options on the shortcut menu.
2. On the General tab, enter a name for the new layer.
3. For Draw Features From, select Autodesk DWG Data Source.
4. Click the Data Source Browse button, and select a DWG data source.
5. Click the Autodesk DWG Browse button, and select a DWG file.
6. Click the Key Table Browse button, and select a link template (link template tables begin with DB_).
7. Click the Key Column Browse button, and select a column.



8. For Filter Type, select Layer, Query, or Theme.

   You can use any of the three filtering methods to specify the geometry that you want to bring into your map.

9 Optionally, click the Name, URL, Where tab, and select a link template for the Name and/or URL, if you want to access database columns linked to the DWG to label the geometry you are bringing into Autodesk MapGuide or to associate URLs.



10 Click OK.

The selected link template associates the data table with the objects on the layer.

DWF Support

3

This chapter describes the new support for the DWF™ file format in this release of Autodesk MapGuide. DWF stands for Design Web Format™ and is usually pronounced "DWIF."

The DWF features in this release apply to both Autodesk MapGuide Server, for data source setup, and Autodesk MapGuide Author, for reading and writing DWF files.

For information about the new API functions for the DWF format, see "DWF API Additions" on page 123.

In this chapter

Summary of DWF Support

You can think of a DWF file as a container for a design package that is comprised of various kinds of design information published in a print-ready drawing set. DWF is an open format that can be published by many different design applications. The DWF format has been created by Autodesk as a way for team members, who may be separated geographically and who may not all have the same software programs, to share and distribute design data.

With this release, Autodesk MapGuide now supports reading and writing of DWF files. Here is the list of DWF-related features:

- Set up a data source specifically for DWF files. See "Setting Up a Data Source for DWF Files" on page 25.
- Create a new layer directly from a DWF-file sheet. See "Creating a Layer from a DWF File Sheet" on page 29.
- Write a DWF file from an Autodesk MapGuide view. See "Saving a View As a DWF File" on page 32.

Note DWF Version 6 files are supported. However, Autodesk MapGuide Author does not display markup created by Volo® View 3 and saved in the DWF file.

Setting Up a Data Source for DWF Files

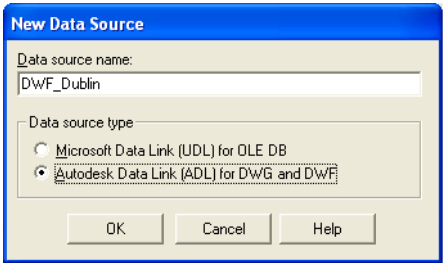
A DWF data source points to the folder where the DWF files that you want to use are stored.

The dialog boxes that you see when you set up a new data source for DWF files are almost identical to those you see when you set up a data source for DWG files (for more information, see the topic "Creating Autodesk DWG Data Sources" in the Autodesk MapGuide Author Help). The principal difference is an additional function on the Advanced tab where you specify the default Spatial Reference System (SRS). The ability to establish an SRS is needed because the DWF format has no support for projection or coordinate systems. The default SRS defines a projection and/or coordinate system for all the data in the DWF data source. Autodesk MapGuide can then transform the data from this SRS to whatever SRS a particular map file is using.

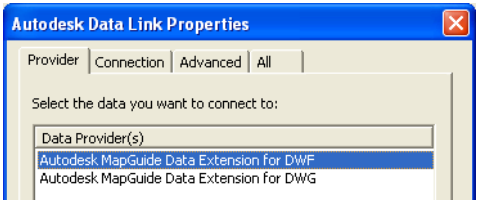
Note Password-protected DWF files are not supported in this release. Passwords are set when the DWF files are published, for example, in Autodesk Map. Check that the DWF files have no password protection before you start using them to author maps.

To create a new data source for DWF files

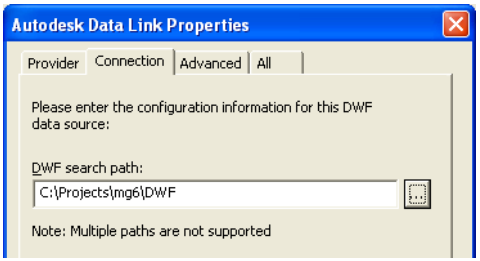
1. In the Autodesk MapGuide Server Admin program, from the Edit menu, choose Properties.
2. In the Properties dialog box, click the Data Sources tab, and then click New.
3. In the New Data Source dialog box, select Autodesk Data Link (ADL) For DWG And DWF.
4. Enter a name for the data source, and then click OK.



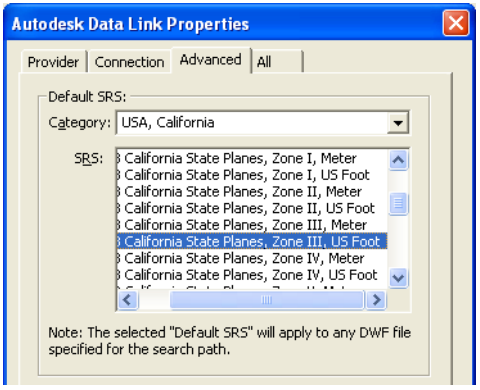
5 In the Autodesk Data Link Properties dialog box, select Autodesk MapGuide Data Extension For DWF, and then click the Connection tab.



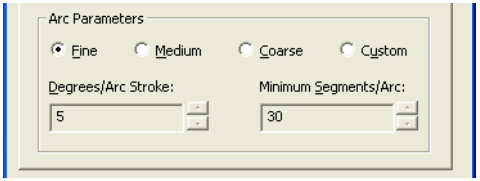
6 On the Connection tab, click the Browse button next to DWF Search Path and navigate to the folder where the DWF files are located. Click the Advanced tab.



7 On the Advanced tab, select the SRS (Spatial Reference System) that you want to use for the DWF files in the data source.



8 Optionally, on the Advanced tab, set the arc parameters. For a description of these parameters, see “Dialog Box Changes: Autodesk MapGuide Server” on page 7.

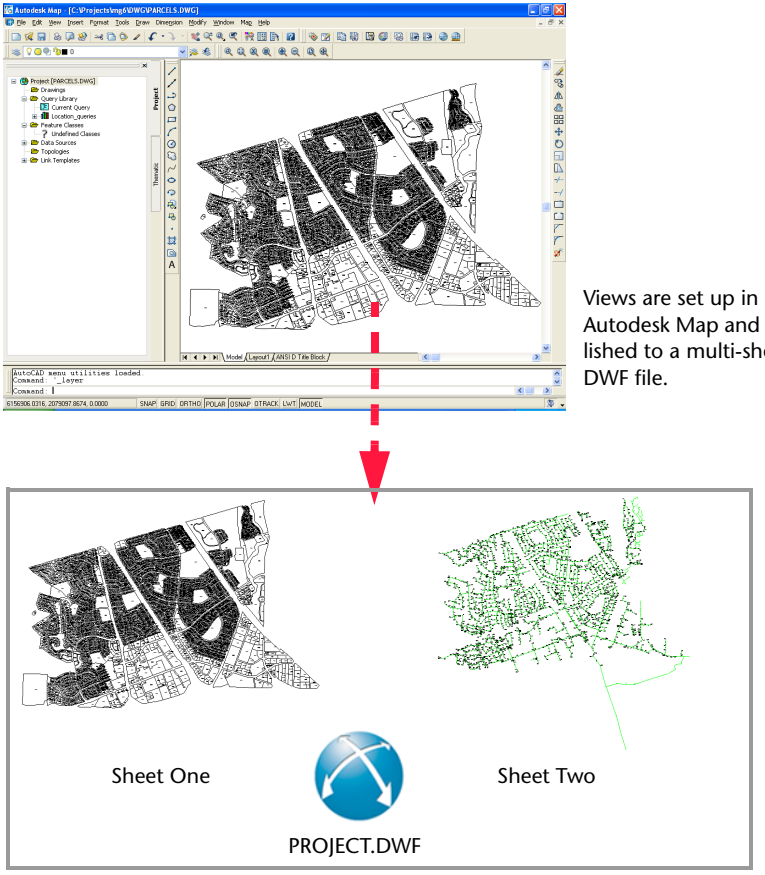


9 Click OK.

The new data source is added to the list in the Properties dialog box.

How DWFs are Published in Autodesk Map

You can create DWFs in various Autodesk products. A DWF file consists of a number of *sheets*, each of which can contain a view of a different DWG or a different view of the same DWG, for example the map with different layers turned on or off. The following diagram illustrates how a DWF file is published from Autodesk Map.



Publishing to a DWF file from Autodesk Map

In the illustration above, views from the drawing files *PARCELS.DWG* and *SEWER_DATA.DWG*, are published to the PROJECT.DWF file. The DWF file contains two sheets. Each of these sheets can be brought into Autodesk MapGuide Author as a separate layer.

Creating a Layer from a DWF File Sheet

DWFs are really intended to be used as background images, in much the same way that DWG files were used in previous releases. Because DWFs are essentially electronic plots, the level of precision depends on the plot settings. This is not an issue for most maps. However, if your application demands a higher degree of precision, you can increase the size of the virtual DWF paper by setting the appropriate DPI and paper size before you publish to DWF.

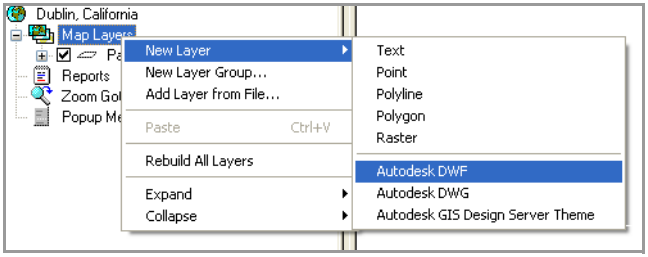
The following link goes to an article that explains DWF precision, why it is not the same as the DWG, and how to improve the precision.
http://autodesk.blogs.com/between_the_lines/2004/01/dwf_precision_a.html

As you could with a DWG layer, you can bring in specific layers from the DWF file. However, you cannot stylize or theme the features once they are part of the map. Also, font information is stored in the DWF and cannot be changed.

Note By default, a DWF file created in Autodesk Map does not retain the layers from the original DWG file. If you want to bring in specific layers from the DWF file, you must modify the DWF plot-configuration file. For more information, see the topic "Overview of Creating or Modifying a DWF6 Configuration File" in the AutoCAD Help.

To create a new Autodesk MapGuide Author layer from a DWF file

1 In Autodesk MapGuide Author, right-click Map Layers, and then click New Layer ➤ Autodesk DWF.



2 On the General tab, enter a name for the new layer.

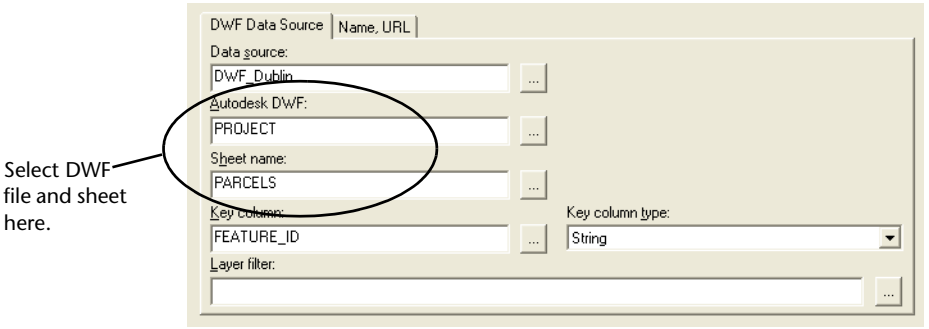
Note If you use a high-dpi DWF for a DWF layer, and you select the Make Map Features Selectable check box (on the General tab of the Map Layer Properties dialog box), it may take a long time to select all the map features.

3 Click the Data Sources tab, and make sure that the Map agent path is pointing to the correct location.

4 Click the Data Source Browse button, and select a DWF data source.

5 Click the Autodesk DWF Browse button, and select a DWF file.

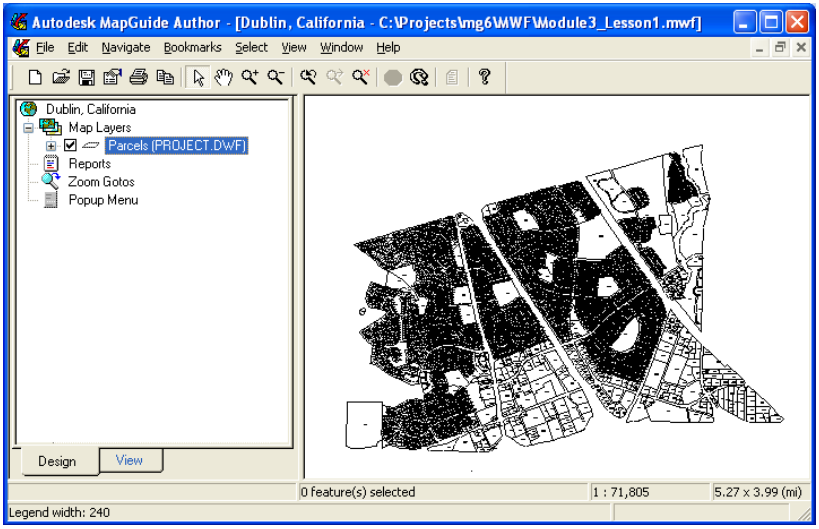
6 Click the Sheet Name Browse button and select a sheet from the list.



7 Optionally, click the Layer Filter Browse button, and select a layer from the list. If you want to bring in the entire sheet, leave the Layer Name field blank.

8 Click OK.

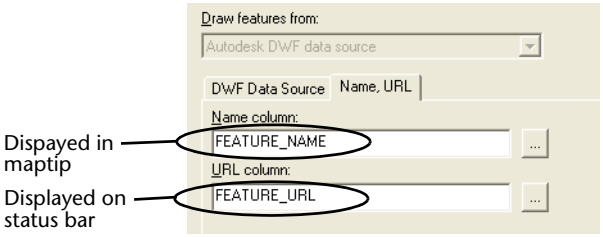
The DWF sheet is retrieved from the server and displayed in Autodesk MapGuide Author.



Displaying Friendly Names and URLs

If you want to access database columns defined for the DWF, you can click the Name, URL tab and select a column for the Name and/or URL. You can use the content of the columns to display a “friendly name” for the features or to display URLs. The friendly name replaces the URL in the maptip. For example. “MapGuide” could replace “http://www.mapguide.com.”

As with SDP data sources, FEATURE_URL is the keyword for accessing URLs in the DWF file. If you enter information in the URL column, that information will be available on the status bar when a user passes the mouse pointer over the related link. Only fully qualified URLs are recognized. Those not fully qualified (that is, those without http://) are ignored.



You may decide to enter the URL into the maptip instead of the DWF friendly name (that is, FEATURE_NAME), in which case you can also select FEATURE_URL for the Name column. You can also do this the other way and use the Name column to display a friendly name in the status bar.

You can add your own text in the Name field using the concatenation operator (||). (Note that concatenation for OLEDB uses + and concatenation in this field does not exist for SDP providers.) Single quotes are used for constant text (for example, ‘CountryID: ‘ || FEATURE_NAME).

Saving a View As a DWF File

In Autodesk MapGuide Author, you can save any view of your map as a DWF file. Then, anyone on your extended team can review these files even though they don't have a copy of Autodesk MapGuide Author. Team members can open the DWF in any of the Autodesk applications that support the DWF file format or in the free Autodesk® DWF™ Viewer.

You save a view to a DWF file works in the same way that you send a file to a printer. Before you can save views to DWF files, you need to download and install the Autodesk® DWF™ Writer print driver to the PC on which you will be authoring. The Autodesk DWFWriter driver can also be downloaded and installed for the Autodesk MapGuide Viewer ActiveX Control or Plug-In.

Note The Autodesk DWFWriter driver is supported on PCs running Windows XP, Windows 2000, or Windows 2003.

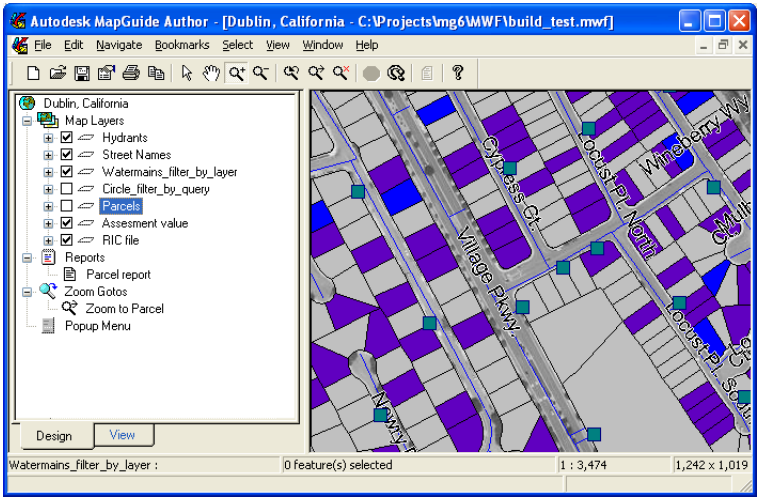
To download and install the Autodesk DWFWriter

1. In your Web browser, go to http://www.autodesk.com/mapguide-dwfwriter .
2. Enter your information on the form and then download the Autodesk DWFWriter.
3. When the download is complete, double-click the downloaded-file icon and then folllow the instructions onscreen to install the driver on your PC.

The default paper size for the print driver is 8.5 x 11 inches. The area of the map that is saved to the DWF file is the area that will fit on that size sheet at the current scale. If you want to set a different-size sheet, use the Windows Printer settings to change the properties of the print driver. You can access these settings from Printers and Faxes on the Control Panel.

To create a DWF file from a view in Autodesk MapGuide Author

1 In Autodesk MapGuide Author, zoom to the view that you want to save.



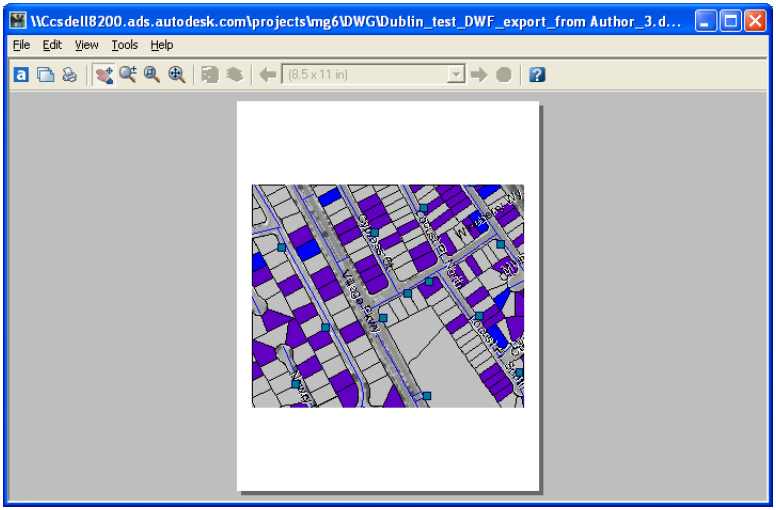
2 From the File menu, choose Save As DWF.

3 In the Save DWF File As dialog box, navigate to the folder where you want to save the DWF file.

4 Enter a name for the file.

5 Click Save.

6 Optionally, test the output DWF file by opening it in Autodesk DWF Viewer.



In this example, you see the standard page size used by default.

You can also save views to DWF files from the Autodesk MapGuide Viewer ActiveX Control or Plug-In, if the DWF Writer has been downloaded and installed on that PC.

To create a DWF file from a view in Autodesk MapGuide Viewer

1 In Autodesk MapGuide Viewer, zoom to the view that you want to save.
2 Right-click anywhere in the view, and then click Print on the shortcut menu.
3 In the Print dialog box, select Autodesk DWFWriter in the drop-down list and then click OK.
4 In the Save DWF File As dialog box, navigate to the folder where you want to save the DWF file.
5 Enter a name for the file.
6 Click Save.
7 Optionally, test the output DWF file by opening it in Autodesk DWF Viewer.

Enhanced Layer Functionality

4

This chapter describes the new enhanced layer functionality for layers that use spatial data provider (SDP) and OLE database data sources. You can use these new features to apply geometry functions to map features, apply filters to spatial queries, define custom spatial queries, and apply pre- and post-query SQL statements to the spatial queries.

To use the new enhanced layer features, you need to be an advanced Autodesk MapGuide user with a solid understanding of Oracle Structured Query Language (SQL).

In this chapter

Summary of Enhanced Layer Functionality

This release of Autodesk MapGuide provides powerful new tools that you can use to enhance the results of your work with layers. Specific new features included in this release are as follows:

- Access to API for enhanced layer functionality. Provide access to the enhanced layer functionality API for a layer. See "Providing Access to the Enhanced Layer Functionality API" on page 37.
- Enhanced geometry functionality. Apply geometry functions to selected features on maps. See "Using Geometry Functions," on page 39.
- Enhanced spatial query function. Add filters to queries to limit the data returned by a query. See"Applying Filters to Spatial Queries," on page 42.
- Custom spatial query functionality. Execute custom queries instead of the default Autodesk MapGuide spatial query. See "Using a Custom Spatial Query," on page 46.
- Pre- and post SQL statement functionality. Execute SQL statements before and after Autodesk MapGuide performs a spatial query against a layer. See "Using SQL Pass-Through Statements," on page 55.
- Ability to track new enhanced layer features using trace log parameters. See "Tracking Enhanced Layer Features," on page 58.

Providing Access to the Enhanced Layer Functionality API

The API for the new enhanced layer functionality provides access to your spatial data provider (SDP) and OLE database data sources. To secure your data against unwanted changes, access to the enhanced layer functionality API for a layer is blocked by default.

If you are adding enhanced layer functionality to your maps, you need to decide whether or not you want to provide developers with access to the API for enhanced layer functionality.

You can use Autodesk MapGuide's existing security functionality to control access to the existing API by setting a passkey using the options under the Security tab in the Map Layer Properties dialog box. In addition to the existing security options, a new option has been added that you can use to provide access to the API for the new enhanced layer functionality. If you check the new Allow Access To The Layer's Geometry Function And Advanced Settings check box, developers can access the enhanced layer functionality API by setting a passkey. If you do not check this new option, developers will not have access to the enhanced layer functionality API for the layer.

Note To provide access to the enhanced layer functionality API for maps created in earlier versions of Autodesk MapGuide, you need to open your existing maps in Autodesk MapGuide Author 6.5, check this new option for the layers to which you have applied the enhanced layer features, and then assign a passkey.

For more information about the existing security functionality, see "Specifying Security for Layers" in the *Autodesk MapGuide User's Guide*, which you can access by clicking Programs ➤ Autodesk MapGuide6.5 ➤ Documentation ➤ Autodesk MapGuide User's Guide on the Start menu.

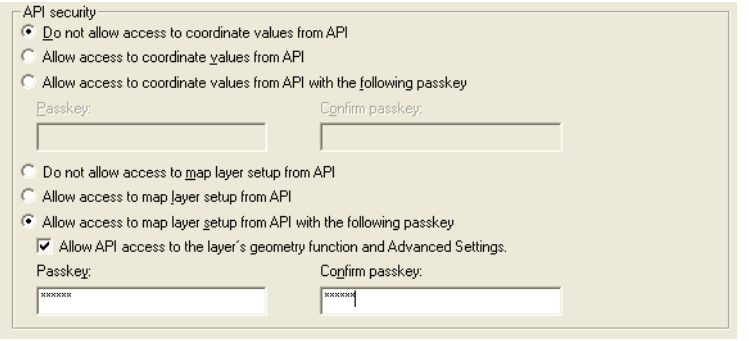
For more information about setting a passkey to access the enhanced layer functionality API, see "Accessing the Enhanced Layer Functionality API" on page 152 of Chapter 9, "Enhanced Layer Functionality API Additions."

To provide access to the enhanced layer functionality from a layer

1. Double-click the name of the layer in the list.

   The Map Layer Properties dialog box is displayed.
2. Click the Security tab to display the security options.
3. On the Security tab, select the Allow Access To Map Layer Setup From API With Following Passkey.
4. Check the Allow API Access To The Layer's Geometry Function and Advanced Settings check box.

   Note This option is only displayed if the layer uses a SDP or OLE Database data source.
5. Specify a passkey by typing it in the Passkey edit box.
6. Confirm the passkey by typing it the Confirm Passkey edit box.



7. Click OK to close the dialog box.

Using Geometry Functions

Using new enhanced layer functionality, you can apply geometry functions to selected features on maps drawn from spatial data provider (SDP) data sources. For example, you can apply a geometry function to a point layer in a map that shows the locations of cell phone towers. This function could direct Oracle to apply a buffer of a size equivalent to the range of each tower. By analyzing the buffered towers on the map, you could see where there are gaps between towers' ranges, indicating places where cell phone coverage may not be available.

Note You can only apply geometry functions to maps drawn from SDP data. Layers created from OLE database data do not support geometry functions.

New geometry functions apply to all layer types and GIS functions that Oracle supports. Advanced calculations required for the addition of buffers, centroid locations, and other geometry functions are performed by Oracle on the server side where the data resides, and are passed back to the Autodesk MapGuide client application. You can take advantage of Oracle's powerful geometry capabilities and quickly bring the results into your projects.

Note Currently, only Oracle supports geometry functions, and geometry functions must return geometry objects, such as points, polygons, polylines, or some combination of these objects. Also, the Autodesk MapGuide layer type must match the geometry type returned by the geometry function used.

To apply the Oracle geometry function

1. Double-click the name of the layer in the list. The Map Layer Properties dialog box is displayed.
2. To display its contents, click the Data Sources tab.
3. On the Data Sources tab, in the Geometry Function text box, enter the name of the geometry function you want to apply.
4. Click OK. The Map Layer Dialog box is closed, and the Oracle layer is buffered using Oracle's geometry function. The results are displayed on the layer.

Example: Suppose you are applying the buffer geometry function, for which you would enter

```
SDO_GEOM.SDO_BUFFER(%GEOMCOL,2,1)
```

The selected geometry column name for the layer is substituted for the %GEOMCOL parameter if it is used.

Note Using parameters when applying a geometry function is optional.

Supported Geometry Functions

The following is a list of some of the supported geometry functions that are available with Oracle:

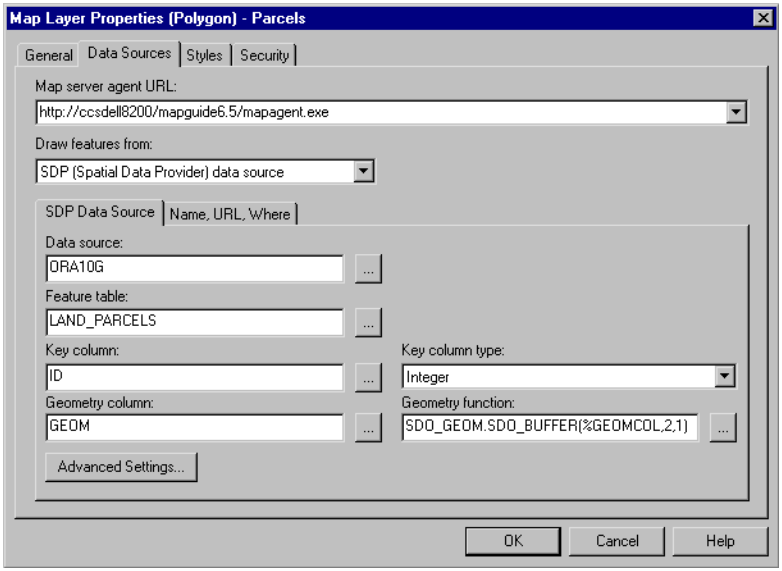
- SDO_ARC_DENSIFY
- SDO_BUFFER
- SDO_CENTROID
- SDO_CONVEXHULL
- SDO_MBR

Note Consult your Oracle documentation for a complete list of Oracle functions available in Oracle and for instructions on how to use them.

Sample Applied Geometry Functions

In this example, Oracle is creating a buffer around land parcels. It is simply one of several ways that you can use geometry functions to analyze and manipulate data without changing the original data in the database.

In the following sample, the parameter %GEOMCOL is used to substitute the geometry column name for the parameter. This parameter is optional, and you can use it to specify a geometry function without having to write a specific geometry column in the request.



Sample 1: Applied geometry functions

In this sample, when the Autodesk MapGuide Server processes the layer,

```
SDO_GEOM.SDO_BUFFER(%GEOMCOL,2,1)
```

becomes

```
SDO_GEOM.SDO_BUFFER(GEOM,2,1)
```

after the parameter has been updated.

Assuming no spatial filtering is being done, the actual Autodesk MapGuide default spatial query used by the server would look like the following:

```
SELECT ID, SDO_GEOM.SDO_BUFFER(GEOM,2,1) FROM LAND_PARCELS
```

Applying Filters to Spatial Queries

A spatial query returns potentially large volumes of geographic information pertaining to map features. By limiting the data returned by a query, a spatial filter reduces the query result set to a particular geography. Using spatial filters can protect against the unwanted return of much larger batches of data than you intended.

For more information about the default Autodesk MapGuide spatial queries, you should refer to your MapGuide user documentation, which you can access by clicking Programs ➤ Autodesk MapGuide6.5 ➤ Documentation ➤ Autodesk MapGuide User's Guide on the Start menu.

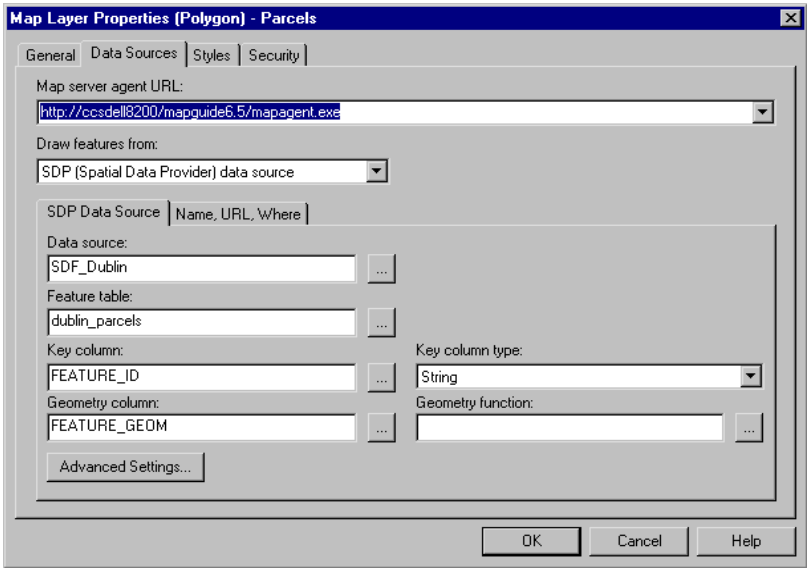
Turning Spatial Filtering On and Off

When you create a layer, spatial filtering is turned on by default. In this version of Autodesk MapGuide, you can choose to turn spatial filtering on or off.

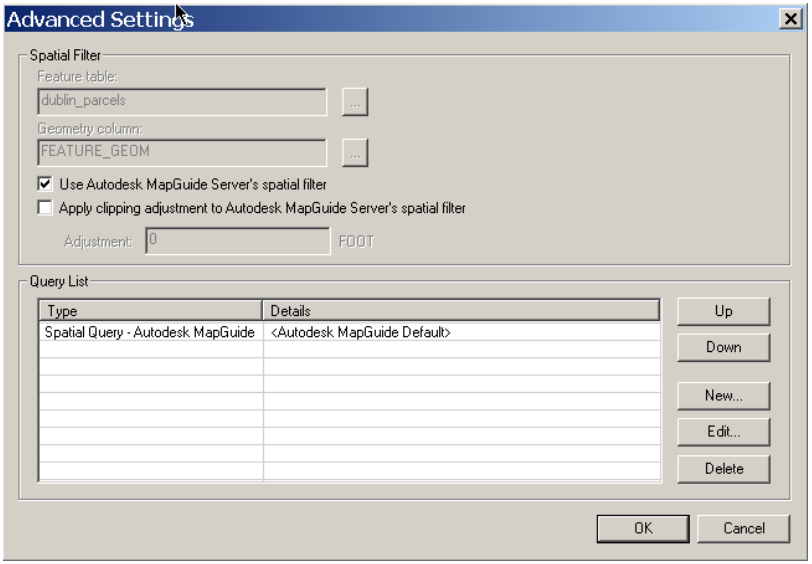
Note You should exercise caution if you decide to turn off spatial filtering. Removing filtering can drastically increase both the record volumes returned and the amount of time required for data retrieval. For example, you could be observing a layer displaying San Francisco primary streets, and the grid displayed could be extracted from a database containing the entire U.S. network of major city streets. Turning off the spatial filter could result in an attempt to return a database in excess of one terabyte (1,024 gigabytes!) in size. Established time limits in retrieval functions would probably shut down the query before it completes, and if not, you could wait a long time to retrieve the data — even if you have room to store it!

To turn spatial filtering on and off

1. Open the map containing the layer to which you want to apply or remove spatial filtering.
2. Double-click the name of the layer in the list to the left of the map. The Map Layer Properties dialog box is displayed.
3. In the Map Layer Properties dialog box, click the Data Sources tab, as shown in the following illustration.



4 Click the Advanced Settings button. The Advanced Settings dialog box is displayed, as shown in the following illustration.



Note that the Use Autodesk MapGuide Server Spatial Filter Setting option is selected by default.

5 To turn the spatial query filter off, clear the Use MapGuide Server's Spatial Filter checkbox.

Changing the Dimensions of the Autodesk MapGuide Spatial Filter

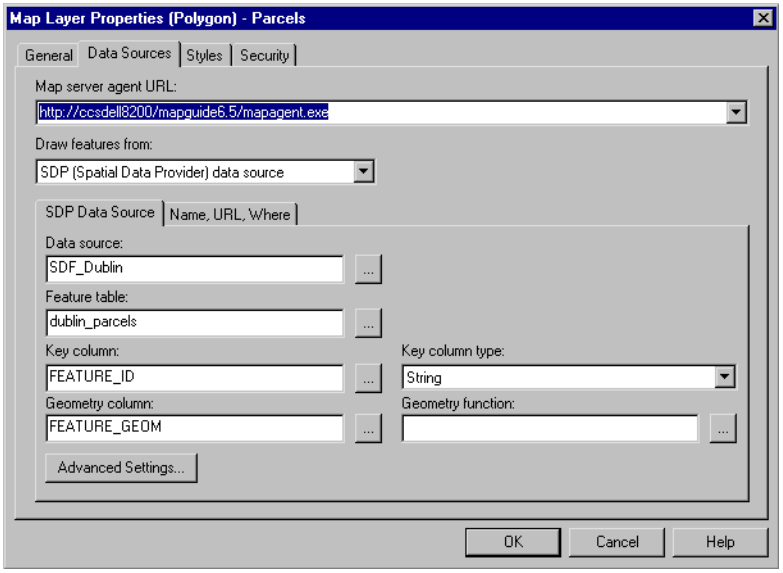
The default client window size determines the size of the spatial filter you use to limit data returned by a spatial query. Autodesk MapGuide allows you to change the spatial filter's extents by applying clipping adjustments that either increase or decrease the dimensions of the bounding box so that it is bigger or smaller than the default client size window. The clipping adjustments you apply can be set to either positive or negative values, depending on whether you want to make the spatial filter larger or smaller than the default client window size. The units for the clipping adjustment are the same as the units established for the map coordinate system.

To adjust the spatial filter

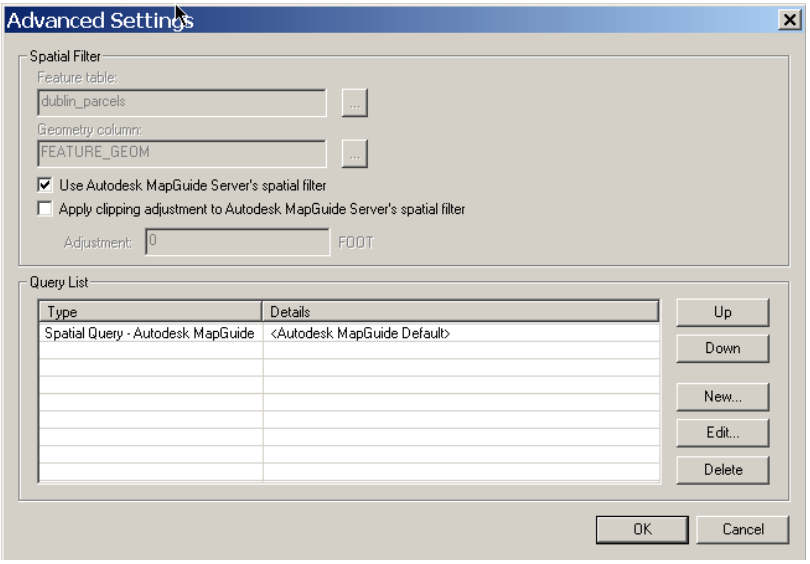
1 Open the map containing the layer to which you want to apply clipping adjustments.

2 Double-click the name of the layer in the list to the left of the map. The Map Layer Properties dialog box is displayed.

3 In the Map Layer Properties dialog box, click the Data Sources tab, as shown in the following illustration.



4 On the Data Sources tab, click the Advanced Settings button. The Advanced Settings dialog box is displayed, as shown in the following illustration.

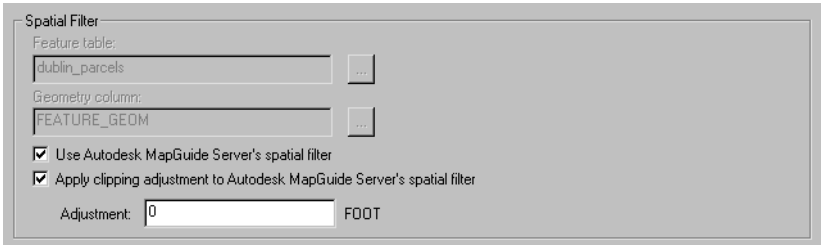


5 Select the Apply Clipping Adjustment To The Autodesk MapGuide Spatial Filter check box. When you select this option, Adjustment is available.

Note The unit of measure (FOOT, MILE, and so on) of the spatial filter extents is shown next to the Adjustment text box, where you enter the number of units you are adjusting. The Adjustment value defaults to zero (0).

6 In Adjustment, enter a positive number to replace the 0 value (1 or 2, for example) if you want to increase the size of the spatial filter's extents, or enter a negative number (-1 or -2, for example) if you want to decrease the size of the filter's extents.

The following illustration shows an adjusted spatial filter entry:



Using a Custom Spatial Query

Custom queries consist of a set of user-defined SQL statements that you can use instead of the default Autodesk MapGuide spatial queries. Default spatial queries compile a server request statement from user-defined parameters. These parameters appear on the Data Sources tab of the Layer Properties dialog box in the Autodesk MapGuide Author. By replacing the default spatial query with a custom spatial query, the client passes a user-defined request statement to the server.

For more information about spatial queries, see the Autodesk MapGuide Users Guide by clicking Programs ➤ Autodesk MapGuide6.5 ➤ Documentation ➤ Autodesk MapGuide User's Guide on the Start menu.

If you want to specify a custom spatial query for a layer, you must ensure that the order of selected columns in the custom query matches the column order expected by the Autodesk MapGuide clients.

The following tables show the expected column order to be returned to the client for a custom spatial query:

SDP Column Order:

Order	Data Field Name
1	KEY
2	GEOMETRY
3	NAME
4	URL

OLE DB Column Order (Text):

Order	Data Field Name
1	KEY
2	LAT
3	LON
4	NAME
5	URL
6	HEIGHT
7	ROTATION
8	HORIZONTAL ALIGNMENT
9	VERTICAL ALIGNMENT

OLE DB Column Order (Point):

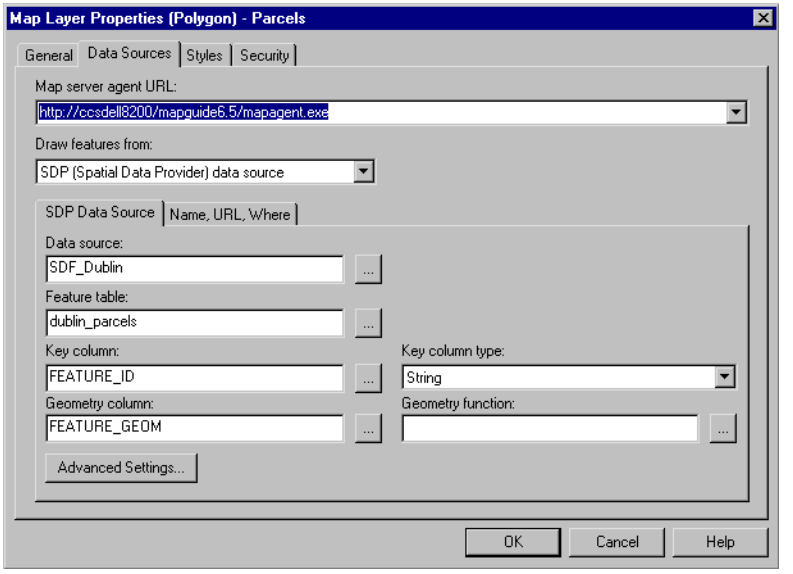
Order	Data Field Name
1	KEY
2	LAT
3	LON
4	NAME
5	URL

Order	Data Field Name
6	WIDTH
7	HEIGHT
8	ROTATION

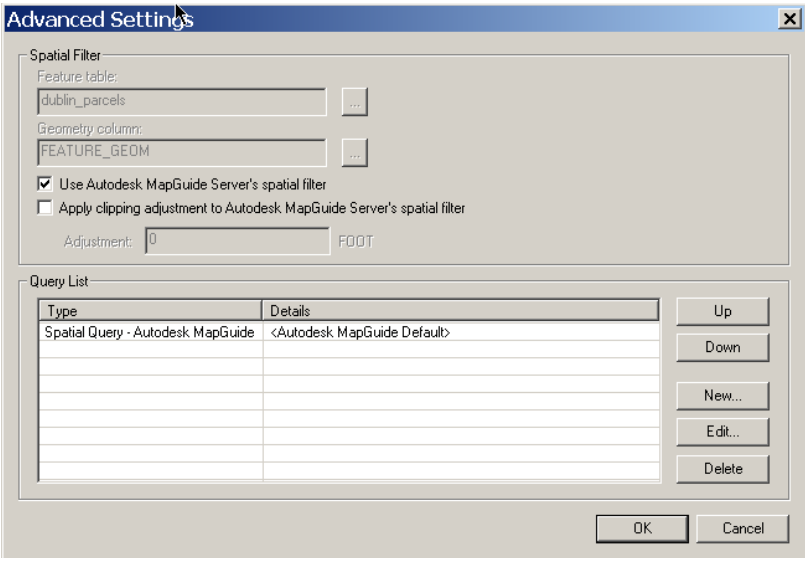
Before you create a custom spatial query, you should verify the content of the default Autodesk MapGuide spatial query. This will save you time if you discover that the default query may have returned the data you need.

To verify the default spatial query

1. Open the map containing the layer to which you want to apply a spatial query.
2. Double-click the name of the layer in the list to the left of the map. The Map Layer Properties dialog box is displayed.
3. In the Map Layer Properties dialog box, click the Data Sources tab, as shown in the following illustration.



4 On the Data Sources tab, click the Advanced Settings button. The Advanced Settings dialog box is displayed, as shown in the following illustration.

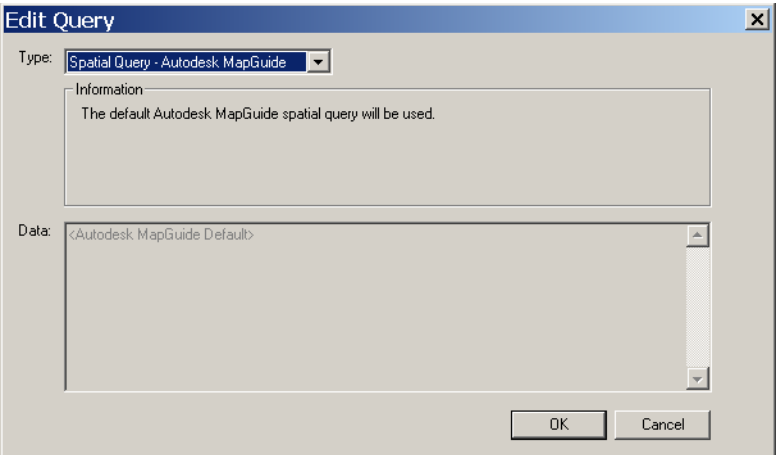


The spatial query that Autodesk MapGuide uses appears under Query List. Note that the Use MapGuide Server's Spatial Filter option is selected and the default Autodesk MapGuide spatial query is specified in the Details column. You can accept this to run the standard query attached to the layer, or you can specify a custom query.

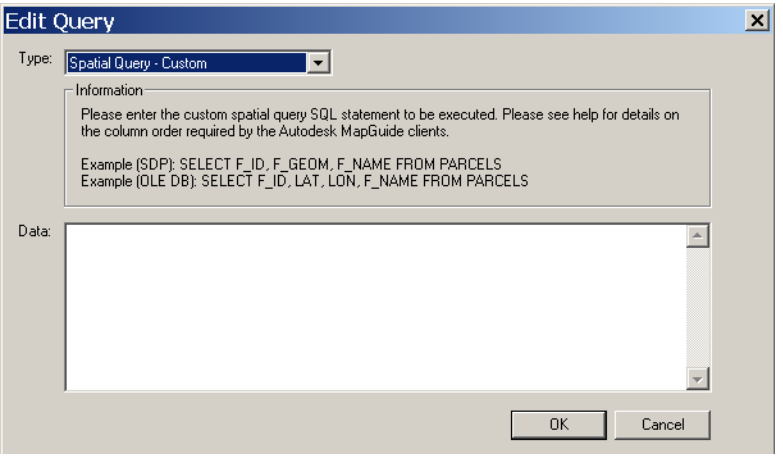
To specify a custom spatial query

1 In the Advanced Settings dialog box, click the Edit button. The Edit Query dialog box is displayed, as shown in the following illustration.

Note The Information portion of the dialog box under Type tells you that the Autodesk MapGuide default spatial query will be used, unless you change it.



2 In the Type list, select Spatial Query - Custom, as shown in the following illustration. Autodesk MapGuide prompts you to enter the name of the custom spatial query in the Data area of the Edit Query dialog box.

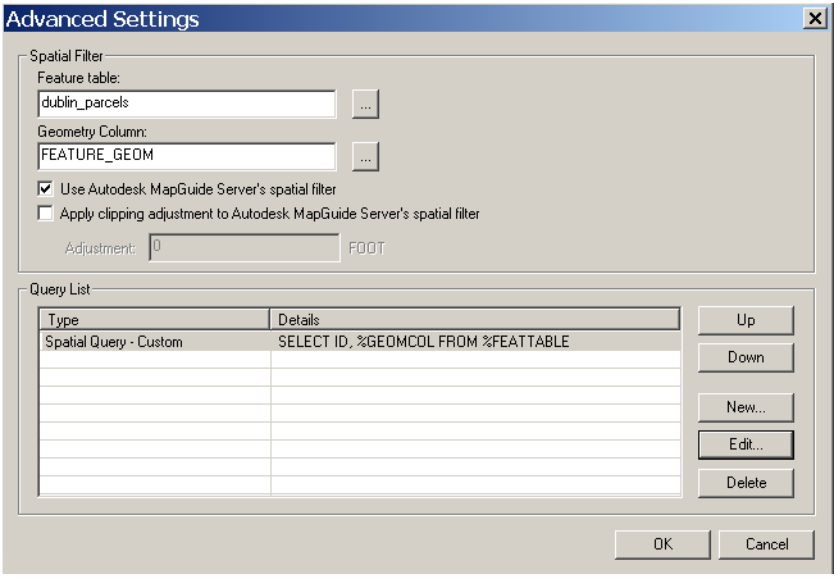


3 In the Data area, enter the custom spatial query SQL statement to be executed. You can use the sample statements under Information to help you with the statement's syntax, as shown in the preceding illustration.

Note When using a custom spatial query, you must specify the correct column order. Click the Help button, or see "Using a Custom Spatial Query," on page 46.

Sample 1: Running a Custom Query

In this sample custom query, the user is supplying parameters to customize a standard spatial query. The parameters simply provide the user with the convenience of using parameters instead of having to type in, in this case, geometry column and feature table names. This can be particularly helpful if such names appear multiple times within a spatial query. If the values change, the user can run the query multiple times without having to edit the query details each time by allowing the parameters to pick up the correct values.



Sample 1: Custom spatial query

In this sample, the following parameters are used:

- **%GEOMCOL:** Comes from the Geometry Column control.
- **%FEATTABLE:** Comes from the Feature Table control

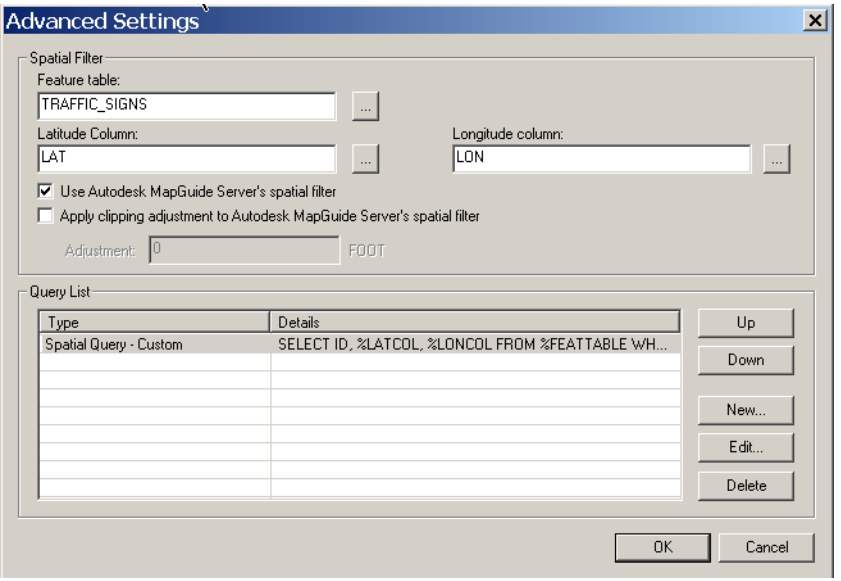
This custom spatial query becomes the following after the Autodesk MapGuide Server has updated the parameters with the appropriate values:

```
SELECT ID, FEATURE_GEOM FROM DUBLIN_PARCELS
```

Note Using parameters within a custom spatial query is optional.

Sample 2: Running a Custom Query Against an OLE Database

This sample again illustrates the convenience of adding user-supplied parameters when running the a custom spatial query, this time against an OLE database. The parameters shown in the following Advanced Settings dialog box are entered instead of the user's having to type literal values contained within columns and feature tables.



Sample 2: OLE/DB custom spatial query

The actual custom query is:

```
SELECT ID, %LATCOL, %LONCOL FROM %FEATTABLE WHERE (%LATCOL > %MINY
AND %LATCOL < %MAXY AND %LONCOL > %MINX AND %LONCOL < %MAXX)
```

In the preceding sample custom spatial query, the following parameters are used:

- **%LATCOL:** Comes from the Latitude Column control
- **%LONCOL:** Comes from the Longitude Column control
- **%FEATTABLE:** Comes from the Feature Table control
- **%MINX:** Comes from the minimum *X* extents of the client window
- **%MINY:** Comes from the minimum *Y* extents of the client window
- **%MAXX:** Comes from the maximum *X* extents of the client window
- **%MAXY:** Comes from the maximum *Y* extents of the client window

This custom spatial query becomes the following after the Autodesk MapGuide server has updated the parameters with the appropriate values:

```
SELECT ID, LAT, LON FROM TRAFFIC_SIGNS WHERE (LAT > -90 AND LAT <
90 AND LON > -180 AND LON <180)
```

Note Using parameters within a custom spatial query is optional.

Accessing the Oracle Linear Referencing System (LRS)

Autodesk MapGuide now provides access via the custom spatial query to Oracle's Linear Referencing System (LRS). You can use linear referencing to locate attributes along a linear map feature. For example, you can locate attributes along a road, using a measure parameter rather than specifying latitude and longitude coordinates. Further, you can reference sections of a linear map feature or create them dynamically by indicating the start and end locations along the feature without explicitly storing these location. LRS functions can be supported in one of two ways. You can do either of the following:

- Create an Oracle VIEW and apply linear referencing to any Oracle VIEW as long as it contains a geometry column.
- Use the new Autodesk MapGuide settings that are available to modify spatial queries (either custom or geometry functions).

Sample Code: Oracle VIEW using the Oracle LRS Function

This is an example of Autodesk MapGuide's new capability providing access to the linear referencing system function embedded in an Oracle VIEW. This example illustrates the syntax of the required code.

The Oracle VIEW description would be the following:

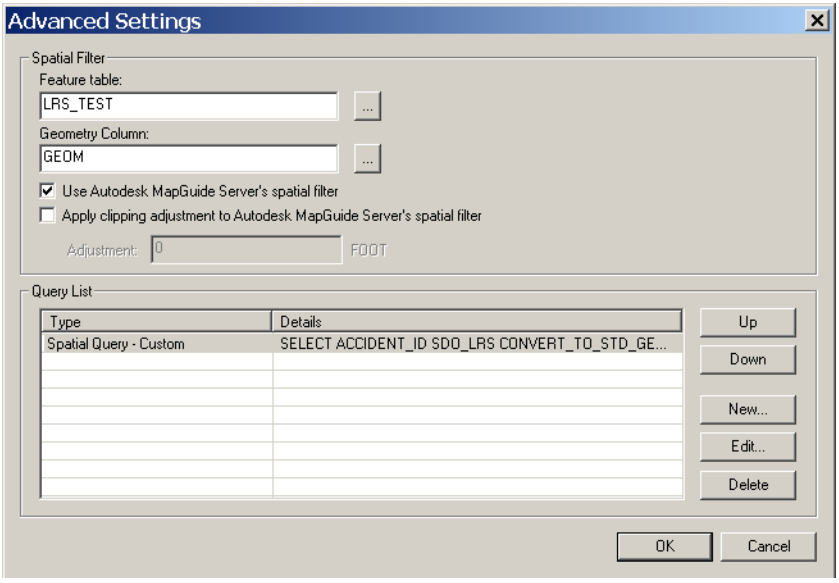
```
SELECT
P.ACCIDENT_ID,
SDO_LRS.CONVERT_TO_STD_GEOM(SDO_LRS.LOCATE_PT(A.GEOM, P.SM, 0),
M.DIMINFO) GEOM,
A.ID,
P.SM
FROM USER_SDO_GEOM_METADATA M, LRS_TEST A, ACCIDENTS P WHERE
A.ID = P.ID AND M.TABLE_NAME = 'NT_SF_POI'
```

Oracle VIEW is accessed as a Feature Table within Autodesk MapGuide, assuming that the Oracle VIEW has been added to the Oracle USER_SDO_GEOM_METADATA table.

Sample Advanced Settings: Custom Spatial Query Including the LRS Function

In this example, the Advanced Settings dialog box illustrates part of the custom query that uses an LRS function. In this case you are simply using the custom spatial query directly instead using Oracle VIEW, as in the preceding example.

You are in effect disabling the Autodesk MapGuide default spatial filter and using a custom spatial filter with the SQL statements listed in the preceding example. For more information, see ""Using Geometry Functions" on page 39 and ""Using SQL Pass-Through Statements" on page 55.



Example: Custom spatial query with LRS function

Using SQL Pass-Through Statements

The Autodesk MapGuide enhanced layer functionality supports the use of pre- and post-SQL statements with spatial queries, allowing you to execute these statements either before or after you perform a spatial query against a layer.

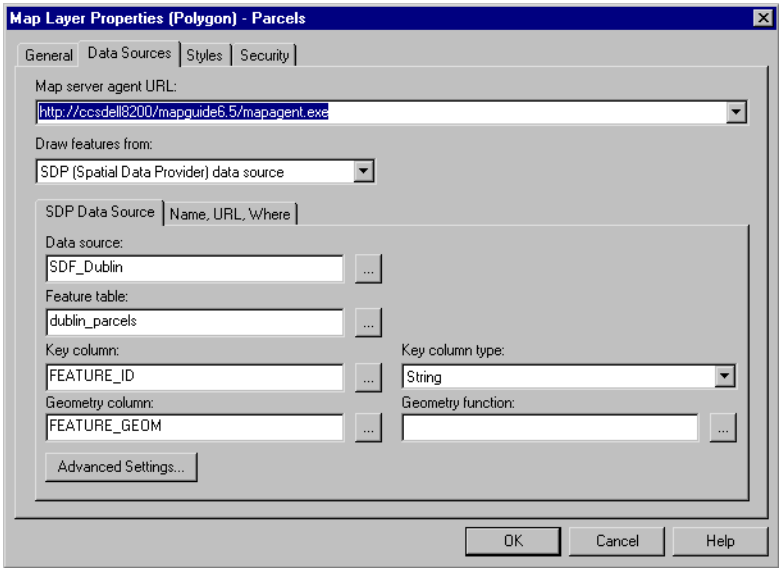
Note To use pre- and post-SQL statements with spatial queries, you must access the database management system from within Autodesk MapGuide.

Pre- and post-SQL statements provide access to the database management system from within Autodesk MapGuide. Although you can execute only one spatial query statement per layer, you can apply multiple SQL statements before and after the spatial query to further customize the results. The statements you enter are processed in the order in which you list them.

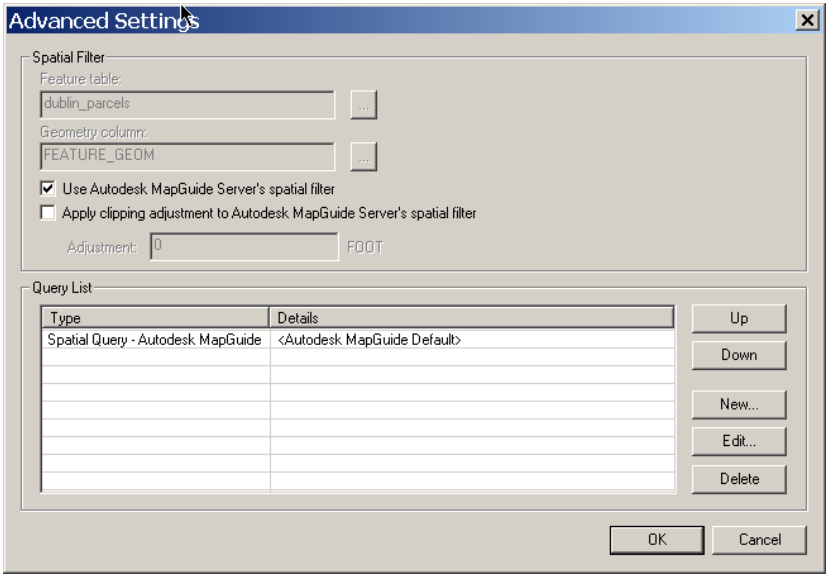
For example, you may want to access different versions of a version-enabled Oracle data set. In this case, you would add a pre-spatial query SQL statement to call the desired workspace in Oracle.

To execute pre- and post-spatial query SQL statements

1. Open the map containing the layer to which you want to apply SQL statements.
2. Double-click the name of the layer in the list to the left of the map. The Map Layer Properties dialog box is displayed.In the Map Layer Properties dialog box, click the Data Sources tab, as shown in the following illustration.

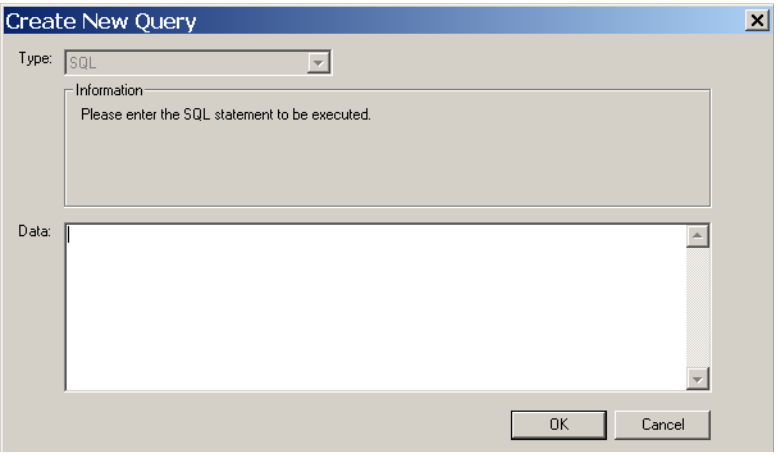


3 Click the Advanced Settings button. The Advanced Settings dialog box is displayed, as shown in the following illustration.



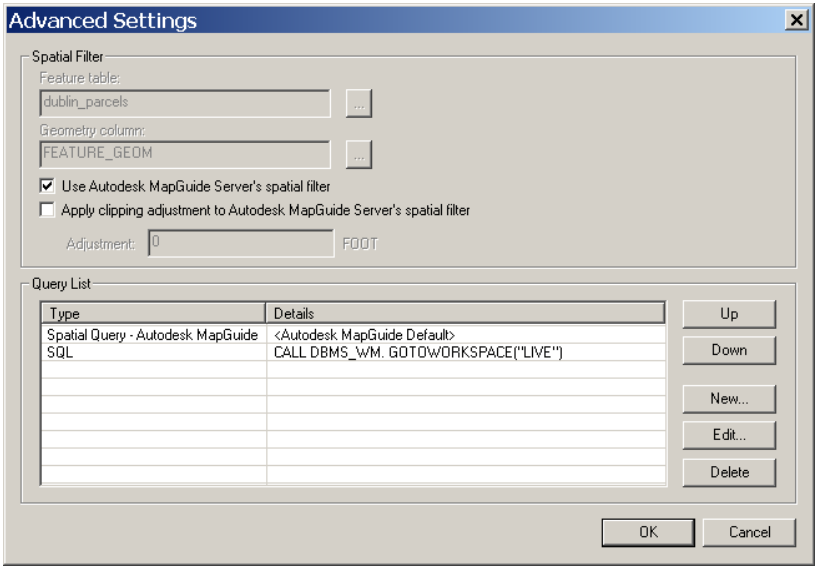
Note that the Query List area includes the standard Autodesk MapGuide default query.

4 Click the New button. The Create New Query dialog box is displayed, as shown in the following illustration.



5 In the Data area, enter a SQL statement, and click OK.

The new statement is displayed by default beneath the spatial query in the Advanced Settings dialog box, as show in the following illustration.



At this point, the added SQL statement is a post-spatial query statement, and would be executed after the query is executed.

6 To make the added SQL statement a pre-spatial query statement that will run before the spatial query is executed, select the statement, and click the Up button to reposition the statement above the spatial query.

7 Optionally, continue to add statements, selecting them individually and clicking the Up or Down button to position a statement above or below the spatial query as desired.

Note Only one spatial query statement — either the default or customized — is allowed per layer. Also, regardless of their position, the results of pre- and post-query SQL statements are ignored by the Autodesk MapGuide Server and are not used during its internal processing of the spatial query.

Tracking Enhanced Layer Features

The ITEM trace log parameter now has additional information that you can use to track the enhanced layer features, including geometry functions, custom spatial queries, filter adjustments, and both pre- or post- SQL statements.

Database access trace log parameters are available in this version of Autodesk MapGuide, as they were in the previous version. For more information about these parameters, see the Autodesk MapGuide Help topic, "Customizing the Access Log." Click the Help Contents tab to find this topic, or search for "Customizing" in the index.

Oracle Spatial Data Provider Enhancements

5

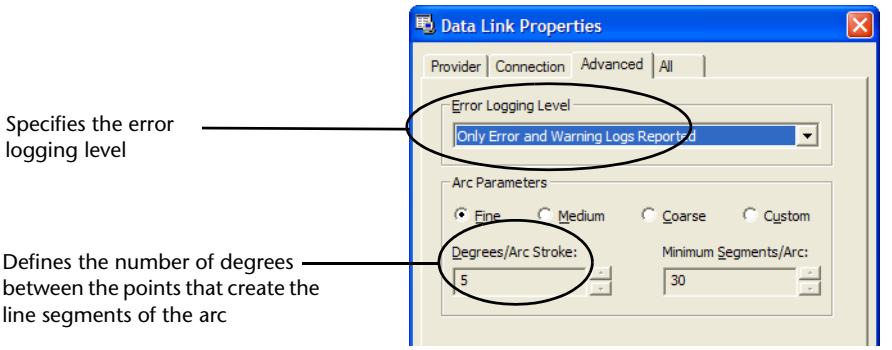
This chapter describes minor changes to the Provider for Oracle® Spatial (sometimes abbreviated to Oracle SDP). You can find a complete description of the provider in the *Provider for Oracle Spatial Guide* on your Autodesk® MapGuide Release 6.5 CD.

In this chapter

- Dialog box changes
- Registry structure changes
- Manually editing a UDL file

Dialog Box Changes

New arc parameters have been added to the Autodesk Data Link Properties dialog box. You access this dialog box by editing the properties of an Oracle data source in the Autodesk MapGuide Server Admin program, or by double-clicking an Oracle UDL file. Using the parameters, you can now specify how finely arcs are rendered as polylines when they are brought into Autodesk MapGuide. Keep in mind that finer arcs slow performance.



New Advanced tab for Autodesk Data Link Properties dialog box

Registry Structure Changes

In previous releases of Autodesk MapGuide, there were four settings for the Provider for Oracle Spatial, which applied to any Oracle data source accessed by the server. All of these settings have been removed. Two of these (SegmentsPerArc and RadiansPerArcStroke) have been replaced by the arc parameters described in "Dialog Box Changes" on page 60, which means that you can now set these parameters indvidually for each data source. The other two settings (EnableDimensionalityCheck and EnableGeodeticCheck) are no longer needed because Autodesk MapGuide performs these checks automatically.

Manually Editing a UDL File

The *Provider for Oracle Spatial Guide* contains instructions about how to manually edit a UDL file. However, manual editing is no longer recommended.

If you use a text editor to enter values into a UDL file that you cannot enter using the Data Source Properties dialog box, Autodesk MapGuide, or any other application that uses UDL files, may not be able to read the file. Also, if you have changed the parameters to illegal values, you cannot open the UDL file in the Data Link Properties dialog box by double-clicking.

To avoid these problems, we strongly recommend against manually editing your UDL files.

Using Buzzsaw with Autodesk MapGuide Viewer

6

This chapter explains how the Microsoft ActiveX Control Viewer works with files that are stored in Buzzsaw®. Autodesk MapGuide Viewer and Buzzsaw interface in three ways. You can use Buzzsaw as a Web server where you can publish your maps and provide others access to them. You can access MWFs stored in Buzzsaw from the Viewer. Finally, you can link features on a map to documents stored in Buzzsaw.

This chapter also provides information about managing the interface between Autodesk MapGuide Viewer and Buzzsaw.

Note: You should be familiar with both Buzzsaw and Autodesk MapGuide before you begin working with the Buzzsaw–MapGuide interface.

In this chapter

Accessing Maps and MWF Files from Buzzsaw

You can use Buzzsaw as a Web server where you can publish your Autodesk MapGuide applications so that others can access them. For more information about publishing maps to a Web server, see "Publishing a Map" in the *Autodesk MapGuide User's Guide*. You can access the *Autodesk MapGuide User's Guide* by clicking Programs ➤ Autodesk MapGuide Release 6.5 ➤ Documentation ➤ Autodesk MapGuide User's Guide from the Start menu.

You can also use Buzzsaw to store MWFs that you want to make available to Autodesk MapGuide Viewer. This program can display MWF files directly from Buzzsaw, or it can display a MWF file referenced in an Autodesk MapGuide application.

Note Only the Microsoft ActiveX Control version of Autodesk MapGuide Viewer can display maps and MWF files stored in Buzzsaw. You can use MWF files with other viewers, but only the ActiveX Control can be used to open the MWFs within Buzzsaw.

Referencing MWF Files from Autodesk MapGuide Applications

If your Autodesk MapGuide application references a MWF stored in Buzzsaw, a reference to it cannot contain URL parameters. Buzzsaw generates an error if you add parameters to the HTML code PARAM statement that references a MWF. You can avoid this problem by adding all PARAM statements separately, after the statement that references the MWF. For example, the following HTML would result in a Buzzsaw error:

```
<PARAM NAME="URL" VALUE="https://projectpoint.buzzsaw.com/project/
myMWF.mwf?LayersViewWidth=120"
```

This entry will not execute within Buzzsaw because of the **...LayersViewWidth=120** setting added to the PARAM statement that references the URL, `myMWF.mwf`. Instead, the LayersViewWidth parameter (and any others) must be included using a separate PARAM statement.

Sample URL Parameter Entry

If the HTML that references an Autodesk MapGuide MWF looks like this:

```
<HTML>

<BODY>

<OBJECT ID="map" WIDTH=100%, HEIGHT=80%
CLASSID="CLSID:62789780-B744-11D0-986B-00609731A21D"
CODEBASE="ftp://ftp.autodesk.com/pub/mapguide/viewer/
mgaxctrl.cab#Version=6,5,0,0:>
     <PARAM NAME="URL" VALUE="http://calpc161/mapguide/
demo.mwf?Lat=0&Lon=0&MapWidth=5000&Units=M">
</OBJECT>

</BODY>
</HTML>
```

where the Autodesk MapGuide URL parameters are specified as part of the URL, an error would result, and the above would have to be re-written as shown in the following code sample:

```
<HTML>

<BODY>

<OBJECT ID="map" WIDTH=100%, HEIGHT=80%
CLASSID="CLSID:62789780-B744-11D0-986B-00609731A21D"
CODEBASE="ftp://ftp.autodesk.com/pub/mapguide/viewer/
mgaxctrl.cab#Version=6,5,0,0:>
     <PARAM NAME="URL" VALUE="http://calpc161/mapguide/demo.mwf>
     <PARAM NAME="Lat" VALUE=0>
     <PARAM NAME="Lon" VALUE=0>
     <PARAM NAME="MapWidth" VALUE=5000>
     <PARAM NAME="Units" VALUE="M">
</OBJECT>

</BODY>
</HTML>
```

Note For HTML coding as required for referencing a MWF, see the sample HTML code in the following section, "Avoiding Buzzsaw Interface Authentication." Also, note that the version number following the ActiveX control reference in the preceding code sample is for illustration purposes only.

Associating a Buzzsaw Document with a Map Feature

After you have created a map in Autodesk MapGuide Author and published it to a Web site, you can add URLs that connect features in the map to documents stored in Buzzsaw. Autodesk MapGuide provides quick access to support documentation about a particular map feature. Any Buzzsaw document file, such as a Word DOC, an Excel XLS, a Project MPP, or an HTML file can be associated with a feature in a map in Autodesk MapGuide, as long as the file you want to associate has been assigned a URL. This section explains how to associate a Buzzsaw document with a map feature.

Note Before you associate Buzzsaw documents with a map feature, you should familiarize yourself with the ways you can customize your interactions with maps using the Autodesk MapGuide Viewer API. For information about the Autodesk MapGuide Viewer API, see the *Autodesk Viewer API Help*, which you can access by clicking Programs ➤ Autodesk MapGuide Release 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help from the Start menu. You can get more information regarding the viewer API by going to the Autodesk MapGuide Web site at http://www.autodesk.com/mapguide.

To associate a Buzzsaw document with a map feature

1. Insert the Buzzsaw URL information into the data (SDF or SHP file, OLE database table, XLS spreadsheet, and so on) that you want displayed in Autodesk MapGuide.

   Note The method you use to insert URL information depends on the type of data you are using. See Steps 3 and 4 for more information about obtaining Buzzsaw URL information.

2. Open Buzzsaw and navigate to the document to which you want to link.
3. To obtain the URL information you want to add to your data, right-click the document in the Project Files list, and then click Copy URL.

   Note Selected URLs are copied to the Windows clipboard in the following form:

   ```
   https://folders.buzzsaw.com/client/Project/document_path/
   document_name.typ
   ```

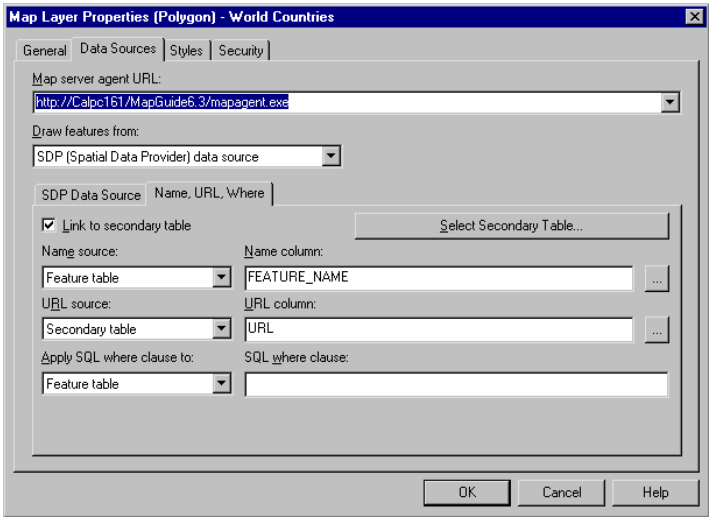
4. After adding the URL, remove **/client** from the path shown in the Note following Step 3. This URL now opens the document directly, bypassing the Buzzsaw client.

5 After the URL information has been added to your data, open the map containing the feature with which you want the document stored in Buzzsaw to be associated.

6 Using both the Data Sources tab and the Name, URL, Where tab in the Map Layer Properties dialog box, assign a data source column that contains the desired URL for the document that you want to associate with a map feature. Use the data source column from the data that contains the URL information specified in Step 1.

Note For instructions about how to assign URLs to Buzzsaw documents that you want to access, see the *Autodesk MapGuide Author Help*, which you can access by clicking Programs ➤ Autodesk MapGuide Release 6.5 ➤ Documentation ➤ Autodesk MapGuide Author Help from the Start menu. In the Table of Contents, click Reference ➤ Autodesk MapGuide Author Dialog Boxes ➤ Data Sources tab ➤ Name, URL, Where tab for the Map Layer Properties dialog box.

The following illustrates a typical approach for an SDP data source The dialog box appearance and the exact workflow may vary slightly for other types of data.



Managing the Autodesk MapGuide and Buzzsaw Interface

This section explains how to manage the integration of the Autodesk MapGuide and Buzzsaw interfaces. You can manage three aspects of this integration:

- **Avoiding Buzzsaw interface authentication.** Eliminate the requirement to log on through a dialog box when opening a document in Buzzsaw.
- **Map state retention.** Prevent the loss of the current map state when you use the Back button to return to the Autodesk MapGuide application from a Buzzsaw document.
- **Non-secure site warning.** Prevent the Buzzsaw warning that a page you are opening contains unsecure information.

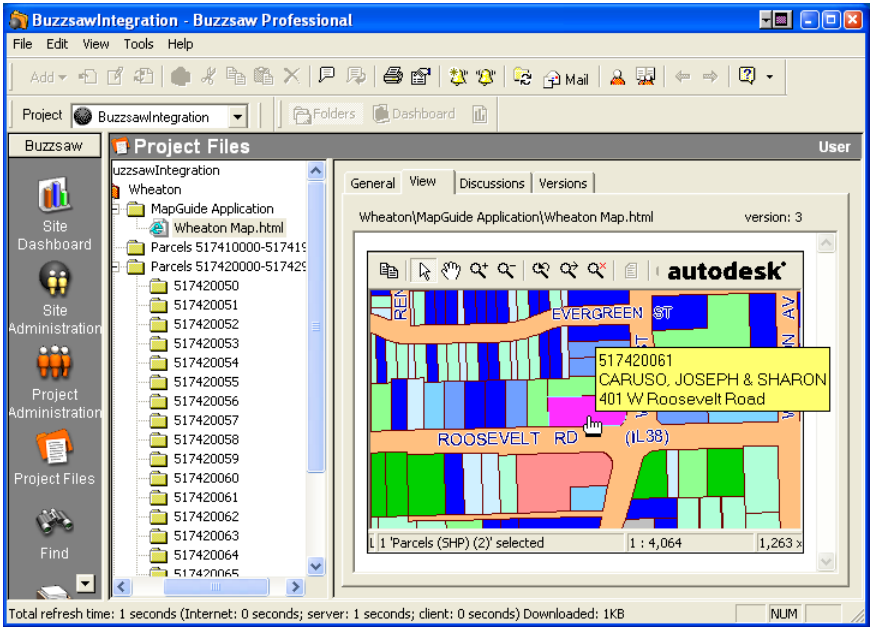
Each of these issues is discussed in the following sections.

Avoiding Buzzsaw Interface Authentication

You can open documents associated with Autodesk MapGuide map features without having to enter a logon ID and password to access the appropriate Buzzsaw folder.

To avoid Buzzsaw Interface Authentication

1 Display the Autodesk Mapguide project document in Buzzsaw, showing the list of HTML documents, as shown in the following illustration.



2 Right-click the HTML document in the Project Files list in Buzzsaw and click Edit to display the Save File To Edit As dialog box.

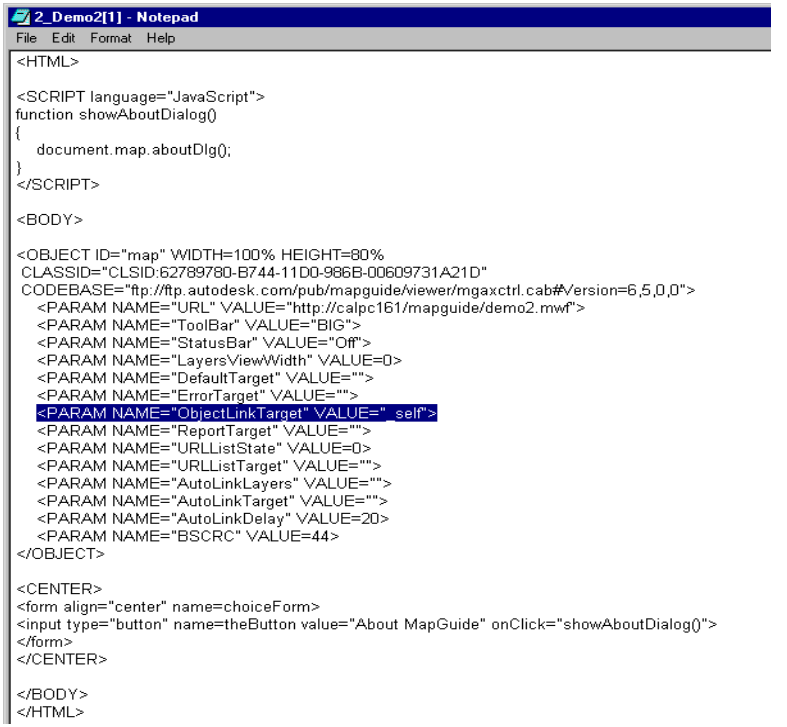
3 Add the saved document to the desired local drive, and click Save. The default HTML editor on your local system opens.

4 Add the following parameter to the HTML, anywhere between the beginning and end object tags:

```
<PARAM NAME="ObjectLinkTarget" VALUE="_self">
```

Note If the ObjectLinkTarget parameter is already present, make sure that it is set to the value **_self**.

The HTML appears as shown in the following sample code, with the ObjectLinkTarget highlighted:



2_Demo2[1] - Notepad

File Edit Format Help

```
<HTML>

<SCRIPT language="JavaScript">
function showAboutDialog()
{
    document.map.aboutDlg();
}
</SCRIPT>

<BODY>

<OBJECT ID="map" WIDTH=100% HEIGHT=80%
 CLASSID="CLSID:62789780-B744-11D0-986B-00609731A21D"
 CODEBASE="ftp://ftp.autodesk.com/pub/mapguide/viewer/mgaxctrl.cab#Version=6,5,0,0">
    <PARAM NAME="URL" VALUE="http://calpc161/mapguide/demo2.mwf">
    <PARAM NAME="ToolBar" VALUE="BIG">
    <PARAM NAME="StatusBar" VALUE="Off">
    <PARAM NAME="LayersViewWidth" VALUE=0>
    <PARAM NAME="DefaultTarget" VALUE="">
    <PARAM NAME="ErrorTarget" VALUE="">
    <PARAM NAME="ObjectLinkTarget" VALUE="_self">
    <PARAM NAME="ReportTarget" VALUE="">
    <PARAM NAME="URLListState" VALUE=0>
    <PARAM NAME="URLListTarget" VALUE="">
    <PARAM NAME="AutoLinkLayers" VALUE="">
    <PARAM NAME="AutoLinkTarget" VALUE="">
    <PARAM NAME="AutoLinkDelay" VALUE=20>
    <PARAM NAME="BSCRC" VALUE=44>
</OBJECT>

<CENTER>
<form align="center" name=choiceForm>
<input type="button" name=theButton value="About MapGuide" onClick="showAboutDialog()">
</form>
</CENTER>

</BODY>
</HTML>
```

Note The version number following the ActiveX control reference in the preceding code sample is for illustration purposes only.

5 Close the editor and save your changes.

6 In Buzzsaw, right-click the HTML document again in the Project File list, and click Update to display the Update Project Document dialog box. You will see the local drive path, ending with the HTML document you have updated. Verify that both the path and the document shown are correct.

7 Click the Next button to add a comment or send an email notification in the next dialog box, or click the Finish button to close the Update Project Document dialog box.

When you have finished updating the HTML, you can click any feature on the corresponding Autodesk MapGuide map to display an attached document without being required to supply a logon ID and password.

Retaining the Current Map State

Autodesk MapGuide and Buzzsaw Viewer integration is designed to ensure that your maps retain their most recent state when you leave them to access a document in Buzzsaw.

Map state retention would help, for example, if you have spent a significant amount of time navigating the map, turning layers on and off, and zooming to display a particular feature on the map, such as a building or road. When you return to the map view from a Buzzsaw document, you want to see the same map view you left to access the document. Otherwise, you must reconstruct the view you created before you accessed the Buzzsaw document. Autodesk MapGuide remembers both your map's selection state and the last mode that you were in for a particular session.

To ensure map state retention, three key categories of information must be saved during the process:

- Area of the map that you are currently viewing
- Current layer and group visibility
- Currently selected map feature objects

For example, you may have displayed a series of layers in a drawing order that exposes a point layer containing hydrants on top of a land parcel layer. Then, you may have zoomed to fill your screen with a particular parcel and its hydrants. Now, you may want to access a Buzzsaw document containing location data about a selected hydrant. You need to be able to save the above three categories of information and return to the map in the state you left it in before accessing the Buzzsaw document.

To prevent map states from being lost, cookies that normally expire at the end of a session must be enabled, which permits the preceding three categories of information to be saved to the cookies. To save these three categories of map information, you must perform two tasks:

- Serialize the map state and save the serialized map state in cookies.
- Use the cookies to restore the map state the next time you open the page.

To save the map states in cookies

1. Use the **onUnload** event handler, which is the onUnload HTML BODY tag, to save the state of the map to cookies when you leave the Web page. The event handler calls a JavaScript function that creates the state cookies and allows the Autodesk MapGuide Viewer API to get the map state information.
2. Save the current map view by using the following APIs:

- MGMap.getLat
- MGMap.getLon
- MGMap.getWidth
- MGMap.getUnits

3 Save the list of layers, groups, and their corresponding visibility by using the following APIs:

- MgMap.getMapLayersEx
- MgMap.getMapLayerGroups
- MgMapLayer.getVisibility
- MgMapLayerGroup.getVisibility

4 Save the feature selection state by using the following API:

- MgMap.getSelection

To restore the map state

1 When the HTML page loads, restore the current view and selection state using the PARAMs to the HTML OBJECT tag that specifies the MapGuide map.

2 Use `Document.write` to specify the PARAMs dynamically when the page loads, setting the following PARAMS:

- Selobjs
- Lat
- Lon
- MapWidth
- Units

Note All parameters are described in Autodesk MapGuide Viewer Help, Advanced Topics ➤URL Parameters

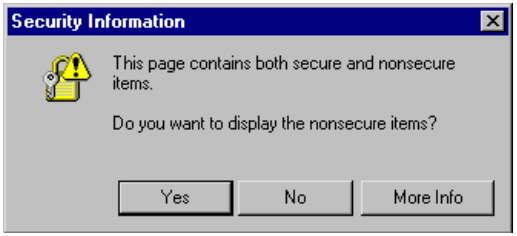
3 Set the layer and group visibility using the following MapGuide Viewer APIs:

- MGMap.getMapLayerEx
- MGMapLayer.setVisibility
- MGMap.getMapLayerGroups
- MGMapLayerGroup.setVisibility

Note Use the **onMapLoaded** event to get the list of layers and groups and change their visibility.

Preventing the Non-Secure Site Warning

Buzzsaw is a secure site, requiring any HTML pages to contain references to secure information. If a referenced document is considered potentially non-secure, Buzzsaw displays the following message:



You can prevent the display of this warning.

To prevent the display of the non-secure site warning

1. Add the Autodesk MapGuide ActiveX Control cab file *mgaxctrl.cab*, located on the Autodesk Mapguide installation CD, in the ActiveXCab sub-folder, to Buzzsaw. You can use the Buzzsaw Add Document interface or simply drag the document onto the Project Files list
2. In the HTML page, reference the cab file added in Step 1 by altering CODEBASE in the HTML OBJECT tag of the MapGuide application to appear as shown in the following code sample:

   ```
   CODEBASE="https://projectpoint.buzzsaw.com/project/
   mgaxctrl.cab#Version=6,5,0,0"
   ```

   where **/project** is the location in path where the cab file is stored.

   Note The version number following the ActiveX control reference in the preceding code sample is for illustration purposes only.

Appendix: Buzzsaw Code Integration

The following code sample illustrates how to integrate Autodesk MapGuide and Buzzsaw so that you can avoid the need to authenticate the Buzzsaw interface, retain the current map state, and prevent the non-secure site warning message from appearing. See the preceding section for more information about managing these three elements of the interface.

Note The URL parameters to the OBJECT tag have to be specified before the sample will work. Also, the CODEBASE parameter to the OBJECT tag should be either specified or removed. The cookieNameXXX variables can be changed to give unique cookie names for multiple MapGuide applications stored in Buzzsaw.

```
<HTML>
<HEAD>
<TITLE>MapGuide Buzzsaw Integration</TITLE>
</HEAD>

<SCRIPT LANGUAGE="VBScript">
Sub map_onMapLoaded(map)
    onMapLoaded map
End Sub
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
// The following variables identify the names of the cookies used.
var cookieNameMapStateLayers = "mapStateLayers";
var cookieNameMapStateGroups = "mapStateGroups";
var cookieNameMapStateURLParam = "mapStateURLParam";

// When the map is initially loaded this event handler is called.
// This event is used to turn on any layers that were on when the map was ///
// previously loaded.
function onMapLoaded(map) {
    if (map != null) {
        var layerVisibility = getCookie(cookieNameMapStateLayers);
        if (layerVisibility != null) {
            var layers = map.getMapLayersEx();
            for (var i = 0; i < layers.size(); i++) {
            if (isNameIn(layerVisibility,escape(layers.item(i).getName()))) {
                    layers.item(i).setVisibility(true);
                } else {
                    layers.item(i).setVisibility(false);
                }
            }
        }
        var groupVisibility = getCookie(cookieNameMapStateGroups);
        if (groupVisibility != null) {
            var layerGroups = map.getMapLayerGroups();
            for (var i = 0; i < layerGroups.size(); i++) {
              if (isNameIn(groupVisibility, escape(layerGroups.item(i).getName()))){
```

```
                    layerGroups.item(i).setVisibility(true);
                } else {
                    layerGroups.item(i).setVisibility(false);
                }
            }
        }
    }
}

// Utility function to check to see if the string in variable name occurs in the
// semicolon delimited string in variable names.
// E.g. isNameIn("A;Bb;Cde", "Bb") => true
//      isNameIn("A;Bb;Cde", "B") => false
function isNameIn(names, name) {
    var cname = name + ";";
  var clen = names.length;

        var cbegin = 0;
        while (cbegin < clen) {
            var cend = cbegin + cname.length;
            if (names.substring(cbegin, cend) == cname) {
                return true;
            }
            cbegin = names.indexOf(";", cbegin) + 1;
            if (cbegin == 0) break;
        }
        return false;
    }

    // This method is called when this page is unloaded (see onUnload ///////
    // procedure).
    // Its purpose is to save the state of the map into cookies.
    // The following information is saved: the selected object (if there is
    // only one selected object); the map center and width; and the
    // layers that are turned on.
    function saveMapState() {
        var map = window.map;

        // selected object
        if (map != null) {
            // selected object
            var selStringPARAM = "";
            var selection = map.getSelection();
            if (selection.getNumObjects() == 1) {
                var selObject = selection.getMapObjectsEx(null).item(0);
                selStringPARAM = "<PARAM NAME=\"Selobjs\" VALUE=\""
                    + selObject.getmapLayer().getName() + ","
                     + selObject.getKey() + "\">";
            }
              // map center and width
            document.cookie = cookieNameMapStateURLParam + "="
                + escape("<PARAM NAME=\"Lat\" VALUE=\"" + map.getLat() + "\">"
                + "<PARAM NAME=\"Lon\" VALUE=\"" + map.getLon() + "\">"
                + "<PARAM NAME=\"MapWidth\" VALUE=\""
                 + map.getWidth(map.getUnits()) + "\">"
                + "<PARAM NAME=\"Units\" VALUE=\"" + map.getUnits() + "\"> "
```

```
                + selStringPARAM);

            // visible layers
            var layers = map.getMapLayersEx();
            var visibleLayersCookie = "";
            for (var i = 0; i < layers.size(); i++) {
                if (layers.item(i).getVisibility()) {
                    visibleLayersCookie = visibleLayersCookie
                        + escape(layers.item(i).getName()) + ";";
                }
            }
            document.cookie = cookieNameMapStateLayers + "="
                + escape(visibleLayersCookie);

            // visible groups
            var groups = map.getMapLayerGroups();
            var visibleGroupsCookie = "";
        for (var i = 0; i < groups.size(); i++) {
            if (groups.item(i).getVisibility()) {
                visibleGroupsCookie = visibleGroupsCookie
                    + escape(groups.item(i).getName()) + ";";
            }
        }
        document.cookie = cookieNameMapStateGroups + "="
            + escape(visibleGroupsCookie);
    }
}

// This function makes sure that when we get the parameters that we
// get an empty string instead of null.
function getMapStatePARAM() {
    var v = getCookie(cookieNameMapStateURLParam);
    if (v == null) return "";
    else return v;
}
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
<!--
// Functions for handling cookies.

/* This code is Copyright (c) 1996 Nick Heinle and Athenia Associates,
 * all rights reserved. In order to receive the right to license this
 * code for use on your site the original code must be copied from the
 * Web site webreference.com/javascript/. License is granted to user to
 * reuse this code on their own Web site if and only if this entire copyright
 * notice is included. Code written by Nick Heinle of webreference.com.
 */

function getCookie (name) {
    var dcookie = document.cookie;
    var cname = name + "=";
    var clen = dcookie.length;
    var cbegin = 0;
    while (cbegin < clen) {
        var vbegin = cbegin + cname.length;
```

```
        if (dcookie.substring(cbegin, vbegin) == cname) {
            var vend = dcookie.indexOf (";", vbegin);
            if (vend == -1) vend = clen;
                return unescape(dcookie.substring(vbegin, vend));
        }
        cbegin = dcookie.indexOf(" ", cbegin) + 1;
        if (cbegin == 0) break;
    }
    return null;
}

function setCookie (name, value, expires) {
    if (!expires) expires = new Date();
    document.cookie = name + "=" + escape (value) +
        "; expires=" + expires.toGMTString() +  "; path=/";
}



    function delCookie (name) {
        var expireNow = new Date();
        document.cookie = name + "=" +
            "; expires=Thu, 01-Jan-70 00:00:01 GMT" +  "; path=/";
    }
    // -->
    </SCRIPT>

    <!-- By using the onUnload event, the map state will be saved when the user
          leaves the page -->
    <BODY onUnload="saveMapState()">
    <SCRIPT LANGUAGE="JavaScript">
    document.write('<OBJECT ID="map" WIDTH=100% HEIGHT=100%');
    document.write(' CLASSID="CLSID:62789780-B744-11D0-986B-00609731A21D"');
    // if you put the CODEBASE parameter here, it must point to a secure
    // site, otherwise you will get the "This page contains both secure
    // and nonsecure items." dialog box.
    document.write(' CODEBASE="https://folders.buzzsaw.com/project/);
    document.write(mgaxctrl.cab#Version=6,0,2,2">');
    document.write(' <PARAM NAME="URL" VALUE="https://folders.buzzsaw.com/');
    document.write(Project/pathtomwf">');
    document.write(' <PARAM NAME="ObjectLinkTarget" VALUE="_self">');
    // Put any other parameters that you want here, but don't use Lat, Lon,
    // MapWidth, Units or SelObjs.
    document.write(getMapStatePARAM());  // restores the view and selection
    document.write('</OBJECT>');
    </SCRIPT>
    </BODY>
    </HTML>
```

Part II

Changes and Additions to the MapGuide Viewer API

The chapters in Part II document the changes and additions made to Autodesk MapGuide Viewer 6.5 API in order to support the new DWG Theming functionality, DWF Support, and Enhanced Layer functionality.

Please note only the ActiveX Control and the Java Edition of the Viewer support the new DWG, DWF, and Enhanced Layer API Functionality.

DWG API Additions

7

This chapter documents the new properties and methods that have been added to the MGDwgDataSources object in order to support the new DWG theming and query functionality in Autodesk MapGuide 6.5. For more information about new DWG support, see Chapter 2, "Improved DWG Support" on page 5.

In this chapter

MGDwgDataSources Object

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

For more information about the MGDwgDataSources object, see the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

getFilterType Method

Syntax

```
int getFilterType()
```

Description

Gets the current filter type that is applied to the DWG data layer. A value of 0 indicates a layer filter type; a value of 1 indicates a query filter type; a value of 2 indicates a theme filter type.

For information about the different filter types, see "How Queries Work in Autodesk Map" on page 9, "Creating a Layer from a Theme" on page 13, and "Creating a Layer By Geometry Type" on page 15.

Parameters

none

Return Values

integer – Value indicating the current filter type.

0 – Layer Filter

1 – Query Filter

2 – Theme Filter

Error Codes

-99 (null pointer)

See Also

FilterType

FilterType Property

Syntax

```
FilterType
```

Description

Read only propety that returns the current filter type applied to the DWG data layer. A value of 0 indicates the layer filter type; a value of 1 indicates the query filter type; a value of 2 indicates a theme filter type.

For information about the different filter types, see "How Queries Work in Autodesk Map" on page 9, "Creating a Layer from a Theme" on page 13, and "Creating a Layer By Geometry Type" on page 15.

Parameters

none

Return Values

integer (read only) – Value indicating the current filter type.

0 – Layer Filter

1 – Query Filter

2 – Theme Filter

Error Codes

-1 (Busy) – This error code is returned for the write operation.

-3 (illegal argument)) – This error code is returned for the write operation.

-99 (null pointer) – This error code is returned for the read operation.

See Also

getFilterType

getQueryName Method

Syntax

```
String getQueryName()
```

Description

Gets the name of the query that is assigned as the current DWG filter.

In DWG layers, you can specify an Autodesk Map query as a filter for layer data. If FilterType is set as a query filter (value of 1), this method returns the name of the query used in the filter. If FilterType is not set as a query filter, this method returns NULL with the error code DoesNotApply. For more information about FilterType, see "FilterType Property," on page 83.

For information about the query filter type, see "How Queries Work in Autodesk Map" on page 9.

Parameters

none

Return Values

string – Represents the name of the query assigned as the DWG filter or an empty string.

Error Codes

-15 (DoesNotApply)

See Also

setQueryName, QueryName, getQueryCategory, setQueryCategory, QueryCategory

setQueryName Method

Syntax

```
boolean setQueryName(String queryName)
```

Description

Sets the name of the query you want to assign as the DWG filter. It also automatically sets FilterType to a Query Filter (value of 1). For infomation about FilterType, see "FilterType Property" on page 83.

In DWG layers, you can specify an AutoCAD Map query as a filter for layer data. Use this method to specify the name of the query to use as the filter.

For information about the query filter type, see "How Queries Work in Autodesk Map" on page 9.

Parameters

queryName – String representing the name of the query you want to assign as the filter.

Return Values

boolean – Specifies whether or not the name of the query has been successfully set.

True – Indicates that the query name has been set successfully.

False – Indicates that the query name has not been set.

Error Codes

-1 (Busy)

See Also

getQueryName, QueryName, getQueryCategory, setQueryCategory, QueryCategory

QueryName Property

Syntax

```
QueryName
```

Description

Sets and gets the name of the query assigned as the DWG filter. It also automatically sets FilterType to a Query Filter (value of 1). For infomation about FilterType, see “FilterType Property” on page 83.

In DWG layers, you can specify an AutoCAD Map query as a filter for layer data. Use this property to specify the name of the query to use as the filter.

For information about the query filter type, see “How Queries Work in Autodesk Map” on page 9.

Parameters

none

Return Values

string (read/write) – Represents the name of the query assigned as the filter.

Error Codes

-1 (Busy) – This error code is returned for the write operation.

-15 (DoesNotApply) – This error code is returned for the read operation.

See Also

getQueryName, getQueryCategory, setQueryCategory, QueryCategory

getQueryCategory Method

Syntax

```
String getQueryCategory()
```

Description

Gets the category of the query that is assigned as the DWG filter.

In DWG layers, you can specify an AutoCAD Map query as a filter for layer data. If FilterType is set as a query filter (value of 1), this method returns the category of the query used in the filter. If FilterType is not set as a query filter, this method returns NULL with the error code DoesNotApply. For more information about FilterType, see "FilterType Property," on page 83.

For information about setting the category of a query filter, see Step 8 under "How Queries Work in Autodesk Map," on page 9.

Parameters

none

Return Values

String (read/write) – Represents the category of the query or an empty string.

Error Codes

-15 (DoesNotApply)

See Also

setQueryCategory, QueryCategory, getQueryName, setQueryName, QueryName

setQueryCategory Method

Syntax

```
boolean setQueryCategory(String queryCategory)
```

Description

Sets the category of the query that you want to assign as the DWG filter. It also automatically sets FilterType to a Query Filter (value of 1). For infomation about FilterType, see "FilterType Property" on page 83.

Use this method to specify the category of the query used as the filter.

For information about setting the category of a query filter, see Step 8 under "How Queries Work in Autodesk Map," on page 9.

Parameters

queryCategory – String representing the category of the query.

Return Values

boolean – Specifies whether or not the category of the query has been successfully set.

True – Indicates that the category of the query has been set successfully.

False – Indicates that the category of the query has not been set.

Error Codes

-1 (Busy)

See Also

getQueryCategory, QueryCategory, getQueryName, setQueryName, QueryName

QueryCategory Property

Syntax

```
QueryCategory
```

Description

Sets and gets the category of the query that is assigned as the DWG filter. It also automatically sets FilterType to a Query Filter (value of 1). For infomation about FilterType, see "FilterType Property" on page 83.

In DWG layers, you can specify an AutoCAD Map query as the filter for layer data. Use this property to specify the category of the query to use as the filter.

For information about setting the category of a query filter, see Step 8 under "How Queries Work in Autodesk Map," on page 9.

Parameters

none

Return Values

String (read/write) – Represents the category of the query that is assigned as the filter.

Error Codes

-1 (Busy) – This error code is returned for the write operation.

-15 (DoesNotApply) – This error code is returned for the read operation.

See Also

getQueryCategory, setQueryCategory, getQueryName, setQueryName, QueryName

getThemeName Method

Syntax

```
String getThemeName()
```

Description

Gets the name of the theme that is assigned as the DWG filter.

In DWG layers, you can specify an AutoCAD Map theme as the filter for layer data. If FilterType is set as a theme filter (value of 2), this method returns the name of the theme used in the filter. If FilterType is not set as a theme filter, this method returns NULL with the error code DoesNotApply. For more information about FilterType, see "FilterType Property," on page 83.

For information about setting the name of theme filters, see Step 8 under "Creating a Layer from a Theme" on page 13.

Parameters

none

Return Values

String – Represents the name of the theme that is assigned as the filter or an empty string.

Error Codes

-15 (DoesNotApply)

See Also

setThemeName, ThemeName, getThemeCategory, setThemeCategory, ThemeCategory

setThemeName Method

Syntax

```
boolean setThemeName(String themeName)
```

Description

Sets the name of the theme that is assigned as the DWG filter. It also automatically sets FilterType to a Theme Filter (value of 2). For infomation about FilterType, see “FilterType Property” on page 83.

In DWG layers, you can specify an AutoCAD Map theme as the filter for layer data. Use this method to set the name of the theme for the filter.

For information about setting the name of theme filters, see Step 8 under “Creating a Layer from a Theme” on page 13.

Parameters

themeName – String representing the name of the theme that is assigned as the filter.

Return Values

boolean – Specifies whether or not the name of the theme has been successfully set.

True – Indicates that the theme name has been set successfully.

False – Indicates that the theme name has not been set.

Error Codes

-1 (Busy)

See Also

getThemeName, ThemeName, getThemeCategory, setThemeCategory, ThemeCategory

ThemeName Property

Syntax

```
ThemeName
```

Description

Sets and gets the name of the theme that is assigned as the DWG filter. It also automatically sets FilterType to a Theme Filter (value of 2). For infomation about FilterType, see "FilterType Property" on page 83.

In DWG layers, you can specify an AutoCAD Map theme as the filter for layer data. Use this property to set and get the name of the theme for the filter.

For information about setting the name of theme filters, see Step 8 under "Creating a Layer from a Theme" on page 13.

Parameters

none

Return Values

String (read/write) – Represents the name of the theme that is assigned as the filter.

Error Codes

-1 (Busy) – This error code is for the write operation.

-15 (DoesNotApply) – This error code is returned for the read operation.

See Also

getThemeName, setThemeName, getThemeCategory, setThemeCategory, ThemeCategory

getThemeCategory Method

Syntax

```
String getThemeCategory()
```

Description

Gets the category of the theme that is assigned as the DWG filter.

In DWG layers, you can specify an AutoCAD Map theme as that the filter for layer data. If FilterType is set as a theme filter (value of 2), this method returns the category of the theme used in the filter. If FilterType is not set as a theme filter, this method returns NULL with the error code DoesNotApply. For more information about FilterType, see "FilterType Property," on page 83.

For information about theme categories, see Step 8 under "Creating a Layer from a Theme" on page 13.

Parameters

none

Return Values

String – Represents the category of the theme that is assigned as the filter or an empty string.

Error Codes

-15 (DoesNotApply)

See Also

setThemeCategory, ThemeCategory, getThemeName, setThemeName, ThemeName

setThemeCategory Method

Syntax

```
boolean setThemeCategory(String themeCategory)
```

Description

Sets the category of the theme that you want to assign as the DWG filter. It also automatically sets FilterType to a Theme Filter (value of 2). For infomation about FilterType, see "FilterType Property" on page 83.

In DWG layers, you can specify an AutoCAD Map theme as the filter for layer data. Use this method to set the category of the theme for the filter.

For information about theme categories, see Step 8 under "Creating a Layer from a Theme" on page 13.

Parameters

themeCategory – String representing the category of the theme that you want to assign as the filter.

Return Values

boolean – Specifies whether or not the category of the theme is successfully set.

True – Indicates that the theme category has been set successfully.

False – Indicates that the theme category has not been set.

Error Codes

-1 (Busy)

See Also

getThemeCategory, ThemeCategory, getThemeName, setThemeName, ThemeName

ThemeCategory Property

Syntax

```
ThemeCategory
```

Description

Gets and sets the category of the theme that is assigned as the DWG filter. It also automatically sets FilterType to a Theme Filter (value of 2). For infomation about FilterType, see "FilterType Property" on page 83.

In DWG layers, you can specify an AutoCAD Map theme as the filter for layer data. Use this property to get and set the category of the theme that is assigned as the filter.

For information about theme categories, see Step 8 under "Creating a Layer from a Theme" on page 13.

Parameters

none

Return Values

String (read/write) – Represents the category of the theme that is assigned as the filter.

Error Codes

-1 (Busy) – This error code is for the write operation.

-15 (DoesNotApply) – This error code is returned for the read operation.

See Also

getThemeCategory, setThemeCategory, getThemeName, setThemeName, ThemeName

getNameSource Method

Syntax

```
int getNameSource()
```

Description

Gets the source of the feature names for layers that are created from DWG data. Feature names can be taken either from an Autodesk DWG file or a secondary table. A value of 0 indicates the Autodesk DWG file as the feature names source. A value of 1 indicates the secondary table as the feature name source.

For more information about getting the source of feature names for a layer created from DWG data, see Step 9 under “Accessing Data Using a Link Template” on page 20.

Parameters

none

Return Values

Integer – Value indicating the source of the feature name.

0 – Autodesk DWG file

1 – Secondary table

Error Codes

none

See Also

setNameSource, NameSource

setNameSource Method

Syntax

```
boolean setNameSource(int nameSource)
```

Description

Sets the source of feature names for layers that are created from DWG data. Feature names can be taken either from an Autodesk DWG file or a secondary table. A value of 0 sets the Autodesk DWG file as the name source. A value of 1 sets the secondary table as the name source.

This method sets the name source only for polyline, polygon, and point layers created from DWG data. It does not set the name source for Autodesk DWG layers.

For more information about setting the name source of features on a layer, see Step 9 under "Accessing Data Using a Link Template" on page 20.

Parameters

nameSource – Value indicating the source of the feature names.

0 – Sets the Autodesk DWG file as the name source.

1 – Sets the secondary table as the name source.

Return Values

boolean – Specifies whether or not the source of the feature names has been successfully set.

True – Indicates that the name source has been set successfully.

False – Indicates that the name source has not been set.

Error Codes

-1 (Busy)

-15 (DoesNotApply)

See Also

getNameSource, NameSource

NameSource Property

Syntax

```
NameSource
```

Description

Gets and sets the source of feature names for layers that are created from DWG data. Feature names can be taken from an Autodesk DWG file or a secondary table. A value of 0 sets the Autodesk DWG file as the name source. A value of 1 sets the secondary table as the name source.

This property gets and sets the name source only for polyline, polygon, and point layers created from DWG data. It does not set the name source for Autodesk DWG layers.

For more information about getting and setting the source of feature names for a layer, see Step 9 under “Accessing Data Using a Link Template” on page 20.

Parameters

none

Return Values

integer (read/write) – Value indicating the source of feature names.

0 – Indicates the Autodesk DWG file as the name source.

1 – Indicates the secondary table as the name source.

Error Codes

-1 (Busy) – This error code is for the write operation.

-15 (DoesNotApply) – This error code is for the write operation.

See Also

getNameSource, setNameSource

getUrlSource Method

Syntax

```
int getUrlSource()
```

Description

Gets the source of the feature URLs for layers that are created from DWG data. These URLs enable users to go the a web page related to a map feature by double clicking the feature.

Feature URLs can be taken either from an Autodesk DWG file or a secondary table. A value of 0 indicates the DWG file as the source feature URLs. A value of 1 indicates the secondary table as the source for feature URLs.

For more information about getting feature URLs for a layer created from DWG data, see Step 9 under "Accessing Data Using a Link Template" on page 20.

Parameters

none

Return Values

integer – Value indicating the source of the feature URLs.

0 – Indicates Autodesk DWG files as the source.

1 – Indicates secondary tables as the source.

Error Codes

none

See Also

setUrlSource, UrlSource

setUrlSource Method

Syntax

```
boolean setUrlSource(int urlSource)
```

Description

Sets the source of feature URLs for layers that are created from DWG data. These URLs enable users to go the a web page related to a map feature by double clicking the feature.

Feature URL sources can be taken either from an Autodesk DWG file or a secondary table. A value of 0 sets the Autodesk DWG file as the URL source. A value of 1 sets the secondary table as the URL source.

This method sets the URL source only for polyline, polygon, and point layers created from DWG data. It does not set the URL sources for Autodesk DWG layers.

For more information about using secondary tables as the source of feature URLs for a layer created from DWG data, see Step 9 under “Accessing Data Using a Link Template” on page 20.

Parameters

urlSource – Value indicating the URL source.

0 – Sets the Autodesk DWG file as the URL source.

1 – Sets the secondary table as the URL source.

Return Values

boolean – Specifies whether or not the source of the feature URLs has been successfully set.

True – Indicates that the URL source has been set successfully.

False – Indicates that the URL source has not been set.

Error Codes

-1 (Busy)

-15 (DoesNotApply)

See Also

getUrlSource, UrlSource

UrlSource Property

Syntax

```
NameSource
```

Description

Gets and sets the source of feature URLs for layers that are created from DWG data. These URLs enable users to go the a web page related to a map feature by double clicking the feature.

Feature URLs can be taken either from an Autodesk DWG file or a secondary table. A value of 0 sets the Autodesk DWG file as the URL source. A value of 1 sets the secondary table as the URL source.

This property sets the URL source only for polyline, polygon, and point layers created from DWG data. It does not set the URL source for Autodesk DWG layers.

For more information about information on getting and setting the source of feature URLs for a layer created from DWG data, see Step 9 under "Accessing Data Using a Link Template" on page 20.

Parameters

none

Return Values

integer (read/write) – Value indicating the URL source.

0 – Indicates Autodesk DWG files as the URL source.

1 – Indicates secondary tables as the URL source.

Error Codes

-1 (Busy) – This error code is for the write operation.

-15 (DoesNotApply) – This error code is for the write operation.

See Also

getUrlSource, setUrlSource

getWhereSource Method

Syntax

```
int getWhereSource()
```

Description

Gets the source of the SQL WHERE clause for layers that are created from DWG data. This method sets the SQL WHERE clause only for polyline, polygon, and point layers created from DWG data. It does not set the SQL WHERE clause for Autodesk DWG layers.

SQL WHERE clauses can be taken either from an Autodesk DWG file or a secondary table. A value of 0 indicates that the SQL WHERE clause is taken from a DWG file. A value of 1 indicates that the SQL WHERE clause is taken from a secondary table.

For more information about using secondary tables as the source for SQL WHERE clauses, see Step 9 under "Accessing Data Using a Link Template" on page 20.

Parameters

none

Return Values

integer – Value indicating the source for the SQL WHERE clause of the layer.

0 – Indicates an Autodesk DWG files as the SQL WHERE clause source.

1 – Indicates a secondary table as the SQL WHERE clause source.

Error Codes

-15 (DoesNotApply)

See Also

setWhereSource, WhereSource

setWhereSource Method

Syntax

```
boolean setWhereSource(int whereSource)
```

Description

Sets the source of the SQL WHERE clause for layers that are created from DWG data. This method sets the source of the SQL WHERE clause only for polyline, polygon, and point layers created from DWG data. It does not set the SQL WHERE clause source for Autodesk DWG layers.

SQL WHERE clause sources can be taken from an Autodesk DWG file or a secondary table. A value of 0 sets the Autodesk DWG file as the source for the SQL WHERE clause. A value of 1 sets the secondary table as the SQL WHERE clause source.

For more information about using secondary tables as the source for SQL WHERE clauses, see Step 9 under "Accessing Data Using a Link Template" on page 20.

Parameters

whereSource – Value indicating the source of the SQL WHERE clause.

0 – Sets the Autodesk DWG file as the source for the SQL WHERE clause.

1 – Sets the secondary table as the SQL WHERE clause source.

Return Values

boolean – Specifies whether or not the source of the SQL WHERE clause has been successfully set.

True – Indicates that the source has been set successfully.

False – Indicates that the source has not been set.

Error Codes

-1 (Busy)

-15 (DoesNotApply)

See Also

getWhereSource, WhereSource

WhereSource Property

Syntax

```
WhereSource
```

Description

Gets and sets the source of the SQL WHERE clause for layers that are created from DWG data. This property sets the source of the SQL WHERE clause only for polyline, polygon, and point layers created from DWG data. It does not set the SQL WHERE clause source for Autodesk DWG layers.

SQL WHERE clause sources can be taken from an Autodesk DWG file or a secondary table. A value of 0 sets the Autodesk DWG file as the source for the SQL WHERE clause. A value of 1 sets the secondary table as the SQL WHERE clause source.

For more information about using secondary tables as the source for SQL WHERE clauses, see Step 9 under "Accessing Data Using a Link Template" on page 20.

Parameters

none

Return Values

integer (read/write) – Value indicating the source for SQL WHERE clause.

0 – Indicates Autodesk DWG files as the source for the SQL WHERE clause.

1 – Indicates secondary tables as the source for the SQL WHERE clause.

Error Codes

-1 (Busy) – This error code is for the write operation.

-15 (DoesNotApply)

See Also

getWhereSource, setWhereSource

getSQLWhereClause Method

Syntax

```
String getSQLWhereClause()
```

Gets the SQL WHERE clause for layers that are created from DWG data. This property gets the SQL WHERE clause only for polyline, polygon, and point layers created from DWG data. It does not get the SQL WHERE clause for Autodesk DWG layers.

SQL WHERE clauses can be applied only to secondary tables.

For more information about using secondary tables as the source for SQL WHERE clauses, see Step 9 under "Accessing Data Using a Link Template" on page 20.

Parameters

none

Return Values

String – Represents the SQL WHERE clause.

Error Codes

-15 (DoesNotApply)

See Also

setSQLWhereClause, SQLWhereClause

setSQLWhereClause Method

Syntax

```
boolean setSQLWhereClause(String sqlWhereClause)
```

Description

Sets the SQL WHERE clause for layers that are created from DWG data. This method sets the SQL WHERE clause only for polyline, polygon, and point layers created from DWG data. It does not set the SQL WHERE clause for Autodesk DWG layers.

SQL WHERE clauses can only be applied to secondary tables.

For more information about using secondary tables as the source for SQL WHERE clauses, see Step 9 under "Accessing Data Using a Link Template" on page 20.

Parameters

sqlWhereClause – String representing the name of the SQL WHERE clause.

Return Values

boolean – Specifies whether or not the SQL WHERE clause is successfully set.

True – Indicates that the SQL WHERE clause has been set successfully.

False – Indicates that the SQL WHERE clause has not been set.

Error Codes

-1 (Busy)

-15 (DoesNotApply)

See Also

getSQLWhereClause, SQLWhereClause

SQLWhereClause Property

Syntax

```
SQLWhereClause
```

Description

Gets and sets the SQL WHERE clause for layers that are created from DWG data. This property gets and sets the SQL WHERE clause only for polyline, polygon, and point layers created from DWG data. It does not get or set the SQL WHERE clause for Autodesk DWG layers.

SQL Where clauses can only be applied to secondary tables.

For more information about using secondary tables as the source for SQL WHERE clauses, see Step 9 under "Accessing Data Using a Link Template" on page 20.

Parameters

none

Return Values

String (read/write) – Represents the SQL WHERE clause.

Error Codes

-1 (Busy) – This error code is for the write operation.

-15 (DoesNotApply)

See Also

getSQLWhereClause, setSQLWhereClause

isLinkedToSecondaryTable Method

Syntax

```
Boolean isLinkedToSecondaryTable()
```

Description

Specifies whether or not the current layer is linked to a secondary table. A value of True indicates that a link exists. A value of False indicates that a link does not exist.

In order to link to a secondary table, you need to call the setSecondaryDataSource, setSecondaryTable, and setSecondaryKeyColumn methods. After you set these, you can call the setNameSource, setUrlSource, or setWhereSource methods to link to a secondary table. If either setNameSource, setUrlSource, or setWhereSource is set to 1, this method returns a value of True. Otherwise, this method returns a value of False.

For more information about linking layers that are created from DWG data to secondary tables, see "Accessing Data Using a Link Template" on page 20.

Parameters

none

Return Values

boolean – Specifies whether or not the layer is linked to a secondary table.

True – Indicates that the layer is linked to a secondary table.

False – Indicates that the layer is not linked to a secondary table.

Error Codes

-15 (DoesNotApply)

See Also

LinkedToSecondaryTable

LinkedToSecondaryTable Property

Syntax

```
LinkedToSecondaryTable
```

Description

Specifies whether or not the current layer is linked to a secondary table. A value of True indicates that a link exists. A value of False indicates that a link does not exist.

In order to link to a secondary table, you need to call the setSecondaryDataSource, setSecondaryTable, and setSecondaryKeyColumn methods. After you set these, you can call the setNameSource, setUrlSource, or setWhereSource methods to link to a secondary table. If either setNameSource, setUrlSource, or setWhereSource is set to 1, this property returns a value of True. Otherwise, this property returns a value of False.

For more information about linking layers that are created from DWG data to secondary tables, see "Accessing Data Using a Link Template" on page 20.

Parameters

none

Return Values

boolean value (read) – Specifies whether or not the layer is linked to a secondary table.

True – Indicates that the layer is linked to a secondary table.

False – Indicates that the layer is not linked to a secondary table.

Error Codes

-15 (DoesNotApply)

See Also

getLinkToSecondaryTable

getSecondaryDataSource Method

Syntax

```
String getSecondaryDataSource()
```

Description

Gets the secondary data source linked to the layers that are created from DWG data. This method gets the secondary data source only for polyline, polygon, and point layers created from DWG data. It does not get the secondary data source linked to Autodesk DWG layers.

For more information about linking layers created from DWG data to secondary data sources, see "Accessing Data Using a Link Template" on page 20.

Parameters

none

Return Values

String – Represents the name of secondary data source.

Error Codes

-15 (DoesNotApply)

See Also

setSecondaryDataSource, SecondaryDataSource

setSecondaryDataSource Method

Syntax

```
boolean setSecondaryDataSource(String secondaryDataSource)
```

Description

Sets the secondary data source linked to layers that are created from DWG data. This method sets the secondary data source only for polyline, polygon,

and point layers created from DWG data. It does not set the secondary data source linked to Autodesk DWG layers.

For more information about linking layers that are created from DWG data to secondary data sources, see “Accessing Data Using a Link Template” on page 20.

Parameters

secondaryDataSource – String representing the name of the secondary data source.

Return Values

boolean – Specifies whether or not the secondary data source is successfully set.

True – Indicates that the secondary data source is successfully set.

False – Indicates that the secondary data source is not set.

Error Codes

-1 (Busy)

-15 (DoesNotApply)

See Also

getSecondaryDataSource, SecondaryDataSource

SecondaryDataSource Property

Syntax

```
SecondaryDataSource
```

Description

Gets and sets the secondary data source linked to layers that are created from DWG data. This property gets and sets the secondary data source only for polyline, polygon, and point layers created from DWG data. It does not get or set the secondary data source linked to Autodesk DWG layers.

For more information about linking layers that are created from DWG data to secondary data sources, see “Accessing Data Using a Link Template” on page 20.

Parameters

none

Return Values

String (read/write) – Represents the name of the secondary data source.

Error Codes

-1 (Busy) – This error code is for the write operation.

-15 (DoesNotApply)

See Also

getSecondaryDataSource, setSecondaryDataSource

getSecondaryTable Method

Syntax

```
String getSecondaryTable()
```

Description

Gets the secondary table linked to layers that are created from DWG data. This method gets the secondary table linked only to polyline, polygon, and point layers. It does not get the secondary table linked to Autodesk DWG layers.

For more information about linking layers that are created from DWG data to secondary tables, see “Accessing Data Using a Link Template” on page 20.

Parameters

none

Return Values

String – Represents the name of secondary table linked to the layer.

Error Codes

-15 (DoesNotApply)

See Also

setSecondaryTable, SecondaryTable

setSecondaryTable Method

Syntax

```
boolean setSecondaryDataSource(String secondaryDataSource)
```

Description

Sets the secondary table linked to layers that are created from DWG data. This method sets the secondary table linked only to polyline, polygon, and point layers. It does not set the secondary table linked to Autodesk DWG layers.

For more information about linking layers that are created from DWG data to secondary tables, see "Accessing Data Using a Link Template" on page 20.

Parameters

secondaryTable – String representing the name of the secondary table to link to the layer.

Return Values

boolean – Specifies whether or not the secondary table is successfully set.

True – Indicates that the secondary table is successfully set.

False – Indicates that the secondary table is not set.

Error Codes

-1 (Busy)

-15 (DoesNotApply)

See Also

getSecondaryTable, SecondaryTable

SecondaryTable Property

Syntax

```
SecondaryTable
```

Description

Gets and sets the secondary table linked to layers that are created from DWG data. This property gets and sets the secondary table linked only to polyline, polygon, and point layers. It does not get or set the secondary table linked to Autodesk DWG layers.

For more information about linking layers that are created from DWG data to secondary tables, see "Accessing Data Using a Link Template" on page 20.

Parameters

none

Return Values

String (read/write) – Represents the name of the secondary table that is linked to the layer.

Error Codes

-1 (Busy) – This error code is for the write operation.

-15 (DoesNotApply)

See Also

getSecondaryTable, setSecondaryTable

getSecondaryKeyColumn Method

Syntax

```
String getSecondaryKeyColumn()
```

Description

Gets the secondary key column linked to layers that are created from DWG data. This method gets the secondary key column linked only to polyline, polygon, and point layers. It does not get the secondary key column linked to Autodesk DWG layers.

For more information about linking layers that are created from DWG data to secondary key columns, see "Accessing Data Using a Link Template" on page 20.

Parameters

none

Return Values

String – Represents the name of the secondary key column that is linked to the layer.

Error Codes

-15 (DoesNotApply)

See Also

setSecondaryKeyColumn, SecondaryKeyColumn

setSecondaryKeyColumn Method

Syntax

```
boolean setSecondaryKeyColumn(String secondaryKeyColumn)
```

Description

Sets the secondary key column linked to layers that are created from DWG data. This method sets the secondary key column linked only to polyline,

polygon, and point layers. It does not set the secondary key column linked to Autodesk DWG layers.

For more information about linking layers that are created from DWG data to secondary key columns, see “Accessing Data Using a Link Template” on page 20.

Parameters

secondaryKeyColumn – String representing the name of the secondary key column that is linked to the layer.

Return Values

boolean – Specifies whether or not the secondary key column is successfully set.

True – Indicates that the secondary key column is successfully set.

False – Indicates that the secondary key column is not set.

Error Codes

-1 (Busy)

-15 (DoesNotApply)

See Also

getSecondaryKeyColumn, SecondaryKeyColumn

SecondaryKeyColumn Property

Syntax

```
SecondaryKeyColumn
```

Description

Gets and sets the secondary key column linked to layers that are created from DWG data. This property gets and set the secondary key column linked only to polyline, polygon, and point layers. It does not get or set the secondary key column linked to Autodesk DWG layers.

For more information about linking layers that are created from DWG data to secondary key columns, see “Accessing Data Using a Link Template” on page 20.

Parameters

none

Return Values

String (read/write) – Represents the name of the secondary key column that is linked to the layer.

Error Codes

-1 (Busy) – This error code is for the write operation.

-15 (DoesNotApply)

See Also

getSecondaryKeyColumn, setSecondaryKeyColumn

getTreatClosedPolylinesAsPolygons Method

Syntax

```
Boolean getTreatClosedPolylinesAsPolygons()
```

Description

Specifies whether or not polygon layers that are created from DWG data treat closed polylines as polygons. A value of True indicates that closed polylines are treated as polygons. A value of False indicates that closed polylines are treated as individual polylines, not as polygons.

Note If this method is set to False, and closed polylines are treated as polylines instead of polygons, the polylines are displayed on a polyline layer instead of the polygon layer.

For more information about how closed polylines are treated on polygon layers that are created from DWG sources, see Chapter ? on page ?.

Parameters

none

Return Values

boolean – Specifies whether or not closed polylines are treated as polygons.

True – Indicates that closed polylines are treated as polygons.

False – Indicates that closed polylines are not treated as polygons. They are treated as individual polylines and are displayed on a polyline layer.

Error Codes

Parameters

-15 (DoesNotApply)

See Also

setTreatClosedPolylinesAsPolygons, TreatClosedPolylinesAsPolygons

setTreatClosedPolylinesAsPolygons Method

Syntax

```
boolean setTreatClosedPolylinesAsPolygons(Boolean
treatClosedPolylinesAsPolygons)
```

Description

Sets how polygon layers that are created from DWG data should treat closed polylines. A value of True indicates that closed polylines are treated as polygons. A value of False indicates that closed polylines are treated as individual polylines, not as polygons.

Note If this method is set to False, and closed polylines are treated as polylines instead of polygons, the polylines appear on a polyline layer instead of the polygon layer.

For more information about how closed polylines are treated on polygon layers created from DWG sources, see Chapter ? on page ?.

Parameters

treatClosedPolylinesAsPolygons – Boolean value specifying whether or not the layer treats closed polylines as polygons.

True – Sets the layer to treat closed polylines as polygons.

False – Sets the layer not to treat closed polylines as polygons.

Return Values

boolean – Specifies whether or not the value that determines if the layer treats closed polylines as polygons is successfully set.

True – Indicates that the value that determines if the layer treats closed polylines as polygons is successfully set.

False – Indicates that the value that determines if the layer treats closed polylines as polygons is not set.

Error Codes

-1 (Busy)

-15 (DoesNotApply)

See Also

getTreatClosedPolylinesAsPolygons, TreatClosedPolylinesAsPolygons

TreatClosedPolylinesAsPolygons Property

Syntax

```
TreatClosedPolylinesAsPolygons
```

Description

Gets and sets how polygon layers that are created from DWG data treats closed polylines. A value of True indicates that closed polylines are treated as polygons. A value of False indicates that closed polylines are treated as individual polylines, not as polygons.

Note If this property is set to False, and closed polylines are treated as polylines instead of polygons, the polylines are displayed on a polyline layer, instead of the polygon layer.

For more information about how closed polylines are treated on polygon layers, see Chapter ? on page ?.

Parameters

none

Return Values

boolean (read/write) – Specifies whether or not to treat closed polylines as polygons.

True – Indicates that the layer is set to treat closed polylines as polygons.

False – Indicates that the layer is not set to treat closed polylines as polygons. They are treated as individual polylines and are displayed on a polyline layer.

Error Codes

-1 (Busy) – This error code is for the write operation.

-15 (DoesNotApply)

See Also

getTreatClosedPolylinesAsPolygons, setTreatClosedPolylinesAsPolygons

getTreatBlocksAsPoints Method

Syntax

```
Boolean getTreatBlocksAsPoints()
```

Description

Specifies whether or not point layers that are created from DWG data treat AutoCAD blocks as points. A value of True indicates that blocks are treated as points. A value of False indicates that blocks are treated as individual polylines and polygons, not as points.

Note If this method is set to False, and blocks are treated as polylines and polygons instead of points, the polylines and polygons appear on polyline and polygon layers, instead of the point layer.

For more information about how blocks are treated on point layers, see Chapter ? on page ?.

Parameters

none

Return Values

boolean – Specifies whether or not blocks are treated as points.

True – Indicates that blocks are treated as points.

False – Indicates that blocks are not treated as points. They are treated as individual polylines and polygons.

Error Codes

-15 (DoesNotApply)

See Also

setTreatBlocksAsPoints, TreatBlocksAsPoints

setTreatBlocksAsPoints Method

Syntax

```
boolean setTreatBlocksAsPoints(Boolean treatBlocksAsPoints)
```

Description

Sets whether or not point layers that are created from DWG data treat AutoCAD blocks as points. A value of True sets the layers so that they treat blocks as points. A value of False sets the layers so that they treat blocks as individual polylines and polygons, not as points.

Note If this method is set to False, and blocks are treated as polylines and polygons instead of points, the polylines and polygons are displayed on polyline and polygon layers, instead of the point layer.

For more information about how blocks are treated on point layers, see Chapter ? on page ?.

Parameters

treatBlocksAsPoints – Boolean value specifying whether or not the layer treats blocks as points.

True – Sets the layer to treat blocks as points.

False – Sets the layer not to treat blocks as points.

Return Values

boolean – Specifies whether or not the value that determines if the layer treats blocks as points is successfully set.

True – Indicates that the value specifying whether or not the layer treats blocks as points has been set.

False – Indicates that the value specifying whether or not the layer treats blocks as points has not been set.

Error Codes

-1 (Busy)

-15 (DoesNotApply)

See Also

getTreatBlocksAsPoints, TreatBlocksAsPoints

TreatBlocksAsPoints Property

Syntax

```
TreatBlocksAsPoints
```

Description

Gets and sets how polygon layers that are created from DWG data treats AutoCAD blocks. A value of True indicates that blocks are treated as points. A value of False indicates that blocks are treated as individual polylines and polygons, not as points.

Note If this property is set to False, and blocks are treated as polylines and polygons instead of points, the polylines and polygons are displayed on polyline and polygon layers, instead of the point layer.

For more information about how blocks are treated on point layers, see Chapter ? on page ?.

Parameters

none

Return Values

boolean (read/write) – Specifies whether or not the layer treats blocks as points.

True – Indicates that the layer is set to treat blocks as points.

False – Indicates that the layer is not set to treat blocks as points.

Error Codes

-1 (Busy) – This error code is for the write operation.

-15 (DoesNotApply)

See Also

getTreatBlocksAsPoints, setTreatBlocksAsPoints

DWF API Additions

8

This section documents the API changes that you can use to access the new MWF layer properties and that support the new DWF functionality in Autodesk MapGuide 6.5.

The DWF API changes include changes and additions to the methods and properties for the existing MGMap, MGMapLayer, and MGMapLayerSetup objects and the addition of a new MGDwfDataSource object.

For more information about the new DWF functionality, see Chapter 3, “DWF Support” on page 23.

In this chapter

MGMap Object

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

This object has one new method that you can use to save the current view of the MWF file in the Autodesk MapGuide Viewer to a DWF file.

For more information about the MGMap object, see the Autodesk MapGuide Viewer API Help. You can access the Autodesk MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

saveAsDwf Method

Syntax

```
boolean saveAsDWF(String pathFileName)
```

Description

Saves the current view of the map to a DWF file.

Note This method supports both local and UNC absolute paths. It does not support relative paths or the creation of multiple nested folders. In addition, you cannot save DWF files to a read-only folder, but you can create a new folder within a read-only folder, and save the DWF file to that new folder.

For more information about saving maps as DWF files, see "Saving a View As a DWF File" on page 32.

Parameters

pathFileName – String representing the absolute path and file name for the saved DWF.

Returns

boolean – Specifies whether or not the map has been successfully saved as a DWF file.

True – Indicates it has been successfully saved.

False – Indicates it has not been saved.

Error Codes

-1 (busy)

-2 (not ready)

-3 (illegal argument)

-16 (write permission denied)

-19 (driver not detected)

-20 (API disabled)

-22 (write error)

MGMapLayer

The MGMapLayer object represents a layer on the map. Its getLayerType method and LayerType property return strings indicating the layer type. This method and property can now also return DWF as the layer type.

For more information about the MGMapLayer object and its getLayerType method and LayerType property, see the Autodesk MapGuide Viewer API Help. You can access the Autodesk MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

getLayerType Method

Returns

String – Represents the layer type. The following are valid layer types: Unknown, Point, Text, Polyline, Polygon, Raster, Buffer, GIS Design Server Theme, Autodesk DWG, Autodesk DWF, and Redline.

For more information about the getLayerType method, see the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

LayerType Property

Returns

String (read/write) – Represents the layer type. The following are valid layer types: Unknown, Point, Text, Polyline, Polygon, Raster, Buffer, GIS Design Server Theme, Autodesk DWG, Autodesk DWF, and Redline.

For more information about the LayerType property, see the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

MGMapLayerSetup Object

The MGMapLayerSetup object provides run-time access to map layer setup through methods that you can use to 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 the failure in the MGError object.

The following methods and properties have been added or updated for MGMapLayerSetup:

- getDwfDataSources – New method (see page 126)
- MGDwfDataSources – New property (see page 127)
- getSourceType – Existing method that can now return DWFs as a data source (see page 128)
- SourceType – Existing property that can now return DWFs as a data source (see page 128)

getDwfDataSources Method

Syntax

```
MGDwfDataSources getDwfDataSources()
```

Description

Gets the MGDwfDataSources object for this layer.

For more information about the MGDataSources object, see “MGDwfDataSources Object” on page 128.

Parameters

none

Returns

MGDwfDataSources – The MGDwfDataSources object for this layer.

See Also

MGDwfDataSources

MGDwfDataSources Property

Syntax

```
MGDwfDataSources
```

Description

Gets the MGDwfDataSources object for this layer.

For more information about the MGDataSources object, see “MGDwfDataSources Object” on page 128.

Parameters

none

Returns

MGDwfDataSources (read only) – The MGDwfDataSources object for this layer.

See Also

getDwfDataSources

getSourceType Method

This is an existing method that returns the type of data source for the layer. In this release, this method now can return DWF as a data source type.

For more information about the getSourceType method, see the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

Returns

String – Represents the source type for the layer data. The following are valid source types for map data: Database, DWG, DWF, RasterImageFile, SpatialDataFile, or VisionTheme.

SourceType Property

This is an existing property that returns the type of data source for the layer. In this release, this property can now set and return DWF as a data source type.

For more information about the SourceType property, see the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

Returns

String (read only) – Represents the source type for the layer data. The following are valid source types for map data: Database, DWG, DWF, RasterImageFile, SpatialDataFile, or VisionTheme.

MGDwfDataSources Object

MGDwfDataSources is a new object that implements methods and properties to support Autodesk DWF data sources for the new DWF layer functionality.

The MGDwfDataSources object includes the following new methods and properties:

- getDataSource (see page 129)
- setDataSource (see page 130)

- DataSource (see page 131)
- getDwf (see page 132)
- setDwf (see page 132)
- Dwf (see page 133)
- getKeyColumn (see page 134)
- setKeyColumn (see page 135)
- KeyColumn (see page 136)
- getKeyColumnType (see page 136)
- setKeyColumnType (see page 138)
- KeyColumnType (see page 139)
- getNameColumn (see page 140)
- setNameColumn (see page 141)
- NameColumn (see page 142)
- getUrlColumn (see page 143)
- setUrlColumn (see page 143)
- UrlColumn (see page 144)
- getLayerFilter (see page 145)
- setLayerFilter (see page 146)
- LayerFilter (see page 147)
- getSheetName (see page 148)
- setSheetName (see page 149)
- SheetName (see page 150)

For more information about the new DWF functionality in this release of Autodesk MapGuide, see Chapter 3, "DWF Support" on page 23.

getDataSource Method

Syntax

```
String getDataSource()
```

Description

Gets the name of the DWF data source that establishes a connection between Autodesk MapGuide Server and the Autodesk Data Link (ADL) data provider.

For more information about DWF data sources, see "Setting Up a Data Source for DWF Files" on page 25.

Parameters

none

Returns

String – Represents the name of the DWF data source.

See Also

setDataSource, DataSource

setDataSource Method

Syntax

```
boolean setDataSource(String dataSource)
```

Description

Sets the name of the DWF data source that establishes a connection between Autodesk MapGuide Server and the Autodesk Data Link (ADL) data provider.

If this method is successful, it sets the rebuild flag of the layer to True. For more informaion about the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

When you change the DWF data source, you may also need to update several other properties. These properties include: Dwf, SheetName, KeyColumn, KeyColumnType, NameColumn, UrlColumn, and LayerFilter.

For more information about DWF data sources, see "Setting Up a Data Source for DWF Files" on page 25.

Parameters

dataSource – String representing the name of the data source.

Returns

boolean – Specifies whether of not the data source is successfully set.

True – Indicates that the data source is successfully set.

False – Indicates that the data source is not set.

Error Codes

-1 Busy

See Also

getDataSource, DataSource

DataSource Property

Syntax

```
DataSource
```

Description

Gets and sets the name of the DWF data source that establishes a connection between Autodesk MapGuide Server and the Autodesk Data Link (ADL) data provider. If this property is modified successfully, it automatically sets the rebuild flag of the layer to True. For more information about the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help . You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

When you change the DWF data source, you may also need to update several other properties. These properties include: Dwf, SheetName, KeyColumn, KeyColumnType, NameColumn, UrlColumn, and LayerFilter.

For more information about DWF data sources, see “Setting Up a Data Source for DWF Files” on page 25.

Parameters

none

Returns

String (read/write) – Represents the name of the DWF data source.

Error Codes

-1 Busy

See Also

getDataSource, setDataSource

getDwf Method

Syntax

```
String getDwf()
```

Description

Gets the name of the Autodesk DWF file for the Autodesk DWF layer.

For more information about using DWF files as data source for layers, see “Setting Up a Data Source for DWF Files” on page 25.

Parameters

none

Returns

String – Represents the name of the DWF file.

See Also

setDwf, Dwf

setDwf Method

Syntax

```
boolean setDwf(String DWF)
```

Description

Sets the name of the Autodesk DWF file for the Autodesk DWF layer. If this method is successful, it sets the rebuild flag of the layer to True. For more information about the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

When you change the Autodesk DWF file, you may also need to update several other properties. These properties include: SheetName, KeyColumn, KeyColumnType, NameColumn, UrlColumn, and LayerFilter.

Note If you do not include a path to the DWF file, this method assigns the first one it finds in the search path.

For more information about using DWF files as data source for layers, see "Setting Up a Data Source for DWF Files" on page 25.

Parameters

DWF – String representing the name of the Autodesk DWF file for the layer.

Returns

boolean – Specifies whether or not the DWF file is successfully set.

True – Indicates that the DWF file is successfully set.

False – Indicates that the DWF file is not set.

Error Codes

-1 Busy

See Also

getDwf, Dwf

Dwf Property

Syntax

```
Dwf
```

Description

Gets and sets the name of the Autodesk DWF file for the Autodesk DWF layer. If this property is modified successfully, it automatically sets the rebuild flag of the layer to True. For more information about the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

When you change the Autodesk DWF file, you may need to update several other properties. These properties include: SheetName, KeyTable, KeyColumn, KeyColumnType, NameTable, NameColumn, UrlTable, UrlColumn, and LayerFilter.

Note If you do not include a path to the DWF file, this method assigns the first one it finds in the search path.

For more information about using DWF files as data source for layers, see "Setting Up a Data Source for DWF Files" on page 25.

Parameters

none

Returns

String (read/write) – Represents the name of the DWF file for the DWF layer.

Error Codes

-1 Busy

See Also

getDwf, setDwf

getKeyColumn Method

Syntax

```
String getKeyColumn()
```

Description

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

For information about using key columns with DWF layers, see "Displaying Friendly Names and URLs" on page 31.

Parameters

none

Returns

String – Represents the name of the key column that contains the primary key for the map features.

See Also

setKeyColumn, KeyColumn

setKeyColumn Method

Syntax

```
boolean setKeyColumn(String column)
```

Description

Sets the name of the column that contains the primary key for each map feature on the DWF layer. If this method is successful, it automatically sets the rebuild flag of the layer to True. For more information on the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

For information about using key columns with DWF layers, see "Displaying Friendly Names and URLs" on page 31.

Parameters

column – String representing the name to assign to the key column.

Returns

boolean – Specifies whether or not the name of the key column is successfully set.

True – Indicates that the name of the key column is successfully set.

False – Indicates that the name of the key column is not set.

Error Codes

-1 Busy

See Also

getKeyColumn, KeyColumn

KeyColumn Property

Syntax

```
String getKeyColumn()
```

Description

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

For information about using key columns with DWF layers, see "Displaying Friendly Names and URLs" on page 31.

Parameters

none

Returns

String – Represents the name of the key column.

Error Codes

-1 Busy

See Also

getKeyColumn, setKeyColumn

getKeyColumnType Method

Syntax

```
String getKeyColumnType()
```

Description

Gets the data type of the key column. You can set the data type to one of the following:

- String
- Numeric
- Decimal
- Integer
- SmallInt
- Float
- Double
- Date
- Time
- TimeStamp
- Boolean
- UnsignedInteger
- UnsignedSmallInt
- Byte
- UnsignedByte

The API uses string values. The MWX data type (XML files) uses enumerated constants in the form of string expressions.

It is recommended that you do not use Float or Double data types because of the inherent inaccuracies in comparing them.

For information about using key columns with DWF layers, see "Displaying Friendly Names and URLs" on page 31.

Parameters

none

Returns

String – Represents the data type of the key column.

See Also

setKeyColumnType, KeyColumnType

setKeyColumnType Method

Syntax

```
boolean setKeyColumnType(String keyColumnType)
```

Description

Sets the data type of the key column. You can set the data type to one of the following:

- String
- Numeric
- Decimal
- Integer
- SmallInt
- Float
- Double
- Date
- Time
- TimeStamp
- Boolean
- UnsignedInteger
- UnsignedSmallInt
- Byte
- UnsignedByte

It is recommended that you do not use Float or Double data types because of the inherent inaccuracies in comparing them.

If this property is modified successfully, it automatically sets the rebuild flag of the layer to True. For more information on the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

Note This method is not case-sensitive.

For information about using key columns with DWF layers, see "Displaying Friendly Names and URLs" on page 31.

Parameters

keyColumnType – String representing the data type to assign to the key column.

Returns

boolean – Specifies whether or not the data type of the key column is successfully set.

True – Indicates that the data type of the key column is successfully set.

False – Indicates that the data type of the key column is not set.

Error Codes

-1 Busy

-3 Illegal Argument

See Also

getKeyColumnType, KeyColumnType

KeyColumnType Property

Syntax

```
KeyColumnType
```

Description

Gets and sets the data type of the key column.

You can set the key column data type to one of the following:

- String
- Numeric
- Decimal
- Integer
- SmallInt
- Float
- Double
- Date
- Time
- TimeStamp
- Boolean
- UnsignedInteger
- UnsignedSmallInt
- Byte
- UnsignedByte

It is recommended that you do not use Float or Double types because of the inherent inaccuracies in comparing them.

If this property is modified successfully, it automatically sets the rebuild flag of the layer to True. For more information on the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

For information about using key columns with DWF layers, see "Displaying Friendly Names and URLs" on page 31.

Parameters

none

Returns

String (read/write) – Represents the data type of the key column.

Error Codes

-1 Busy

-3 IllegalArgument

See Also

getKeyColumnType, setKeyColumnType

getNameColumn Method

Syntax

```
String getNameColumn()
```

Description

Gets the name of the column that contains the name of each map feature on the Autodesk DWF layer.

For information about using name columns with DWF layers, see "Displaying Friendly Names and URLs" on page 31.

Parameters

none

Returns

String – Represents the name of the column containing the names of the map features on the DWF layer.

See Also

setNameColumn, NameColumn

setNameColumn Method

Syntax

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

Description

Sets the name of the column that contains the name of each map feature on the Autodesk DWF layer. If this method is successful, it sets the rebuild flag of this layer to True. For more information on the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

Properties include NameColumn.

For information about using name columns with DWF layers, see "Displaying Friendly Names and URLs" on page 31.

Parameters

column – String representing the name to assign to the column with the names for the map features.

Returns

boolean – Specifies whether or not the name of the column is successfully set.

True – Indicates that the column name is successfully set.

False – Indicates that the column name is not set.

Error Codes

-1 Busy

See Also

getNameColumn, NameColumn

NameColumn Property

Syntax

```
NameColumn
```

Description

Gets and sets the name of the column that contains the name for each map feature on the Autodesk DWF layer. Returns an empty string if this property does not apply. If this property is modified successfully, it automatically sets the rebuild flag of the layer to True. For more information on the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

For information about using name columns with DWF layers, see "Displaying Friendly Names and URLs" on page 31.

Parameters

none

Returns

String (read/write) – Represents the name of the column containing the names of the map features on the Autodesk DWF layer.

Error Codes

-1 Busy

See Also

getNameColumn, setNameColumn

getUrlColumn Method

Syntax

```
String getUrlColumn()
```

Description

Gets the name of the column that contains the URL associated with each map feature on the Autodesk DWF layer.

These URLs enable users 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 JavaScript commands. Use caution, however, with non-standard protocols, such as Javascript, as not all browsers support them.

For information about using URL columns with DWF layers, see "Displaying Friendly Names and URLs" on page 31.

Parameters

none

Returns

String – Represents the name of the column containing the URLs for the map features.

See Also

setUrlColumn, UrlColumn

setUrlColumn Method

Syntax

```
boolean setUrlColumn(String column)
```

Description

Sets the name of the column that contains the URL associated with each map feature on the Autodesk DWF layer. If this method is successful, it sets the rebuild flag of the layer to True. For more information on the rebuild flag, see

setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

These URLs enable 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.

For information about using URL columns with DWF layers, see "Displaying Friendly Names and URLs" on page 31.

Parameters

column – String representing the name to assign to the column containing the URLs for the map features.

Returns

boolean – Specifies whether or not the name of the column for the URLs is successfully set.

True – Indicates that the name of the URL column is successfully set.

False – Indicates that the name of the URL column is not set.

Error Codes

-1 Busy

See Also

getUrlColumn, UrlColumn

UrlColumn Property

Syntax

```
UrlColumn
```

Description

Gets and sets the name of the column that contains the URLs associated with each map feature on the Autodesk DWF layer. If this method is successful, it sets the rebuild flag of the layer to True. Returns an empty string if this property does not apply. For more information on the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

These URLs enable 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 JavaScript commands. Use caution, however, with non-standard protocols, such as Javascript, as not all browsers support them.

For information about using URL columns containing DWF layers, see "Displaying Friendly Names and URLs" on page 31.

Parameters

none

Returns

String (read/write) – Represents the name of the column containing the URLs for the map features.

Error Codes

-1 Busy

See Also

getUrlColumn, setUrlColumn

getLayerFilter Method

Syntax

```
String getLayerFilter()
```

Description

Gets the layer filter that specifies which layers to extract from the specified Autodesk DWF file.

Note The layer filter is a comma-delimited string.

For more information about using DWF files as data sources, see "Setting Up a Data Source for DWF Files" on page 25.

Parameters

none

Returns

String – Represents the layer filter.

See Also

setLayerFilter, LayerFilter

setLayerFilter Method

Syntax

```
boolean setLayerFilter(String layerFilter)
```

Description

Sets the DWF layer filter that specifies which layers to extract from the specified Autodesk DWF file. If this method is successful, it sets the rebuild flag of the layer to True. For more information on the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

Note The layer filter is a comma-delimited string.

For more information about using DWF files as data sources, see "Setting Up a Data Source for DWF Files" on page 25.

Parameters

layerFilter – String (comma-delimited) representing the names of the layers to extract from the DWF file. For example, if a DWF sheet conists of 4 layers (blue, red, green, and yellow), and you only want to extract the red layer, you pass the "red" as the parameter to setLayerFilter.

An empty layer filter string ("") applies all filter layers.

Returns

boolean – Specifies whether or not the layer filter is successfully set.

True – Indicates that the filter is successfully set.

False – Indicates that the filter is not set.

Error Codes

-1 Busy

See Also

getLayerFilter, LayerFilter

LayerFilter Property

Syntax

```
LayerFilter
```

Description

Gets or sets the layer filter that specifies which layers to extract from the specified Autodesk DWF file. If this property is modified successfully, it automatically sets the rebuild flag of the layer to True. For more information on the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

Note The layer filter is a comma-delimited string.

For more information about using DWF files as data sources, see "Setting Up a Data Source for DWF Files" on page 25.

Parameters

none

Returns

String (read/write) – Represents the layer filter.

Error Codes

-1 Busy

See Also

getLayerFilter, setLayerFilter

getSheetName Method

Syntax

```
String getSheetName()
```

Description

Gets the name of the DWF sheet that contains the map features on this Autodesk DWF layer.

DWF files consist of a number of *sheets*, each of which can contain a view of a different DWG file or a different view of the same DWG file, for example a layout with different layers turned on or off.

For more information about using DWF sheets, see "How DWFs are Published in Autodesk Map" on page 28 and "Creating a Layer from a DWF File Sheet" on page 29.

Parameters

none

Returns

String – Represents the name of the sheet with the map features.

See Also

setSheetName, getSheetName

setSheetName Method

Syntax

```
boolean setSheetName(String sheetName)
```

Description

Sets the name of the DWF sheet that contains the map features on this Autodesk DWF layer.

DWF files consist of a number of *sheets*, each of which can contain a view of a different DWG file or a different view of the same DWG file, for example a layout with different layers turned on or off.

If this method is successful, it sets the rebuild flag of the layer to True. Form more information on the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

For more information about using DWF sheets, see "How DWFs are Published in Autodesk Map" on page 28 and "Creating a Layer from a DWF File Sheet" on page 29.

Parameters

sheetName – String (comma-delimited) representing the name of the sheet that contains the map features for this DWF layer.

Returns

boolean – Specifies whether or not the sheet name is successfully set.

True – Indicates that the name is successfully set.

False – Indicates that the name is not set.

Error Codes

-1 Busy

See Also

getSheetName, SheetName

SheetName Property

Syntax

```
SheetName
```

Description

Sets the name of the DWF sheet that contains the map features on this Autodesk DWF layer.

DWF files consist of a number of *sheets*, each of which can contain a view of a different DWG file or a different view of the same DWG file, for example a layout with different layers turned on or off.

If this property is modified successfully, it automatically sets the rebuild flag of the layer to True. For more information on the rebuild flag, see setRebuild in the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

For more information about using DWF sheets, see "How DWFs are Published in Autodesk Map" on page 28 and "Creating a Layer from a DWF File Sheet" on page 29.

Parameters

none

Returns

String – Represents the name of the sheet with the map features.

Error Codes

-1 Busy

See Also

getSheetName, setSheetName

Enhanced Layer Functionality API Additions

9

This chapter documents the API changes that support the new enhanced layer functionality in Autodesk MapGuide 6.5.

The enhanced layer features provide you with the ability to apply geometry functions to map features, apply filters to spatial queries, define custom spatial queries, and apply pre- and post-query SQL statements to the spatial queries.

The enhanced layer functionality API changes include additions to the methods and properties for the existing MGMapLayerSetup object. A new error code has also been added to the existing methods of the MGMapLayerSetup and MGDatabaseSetup objects.

For more information about the new Enhanced Layer Functionality, see Chapter 4, "Enhanced Layer Functionality" on page 35.

In this chapter

Accessing the Enhanced Layer Functionality API

The enhanced layer functionality API is only available to layers that use a spatial data provider (SDP) or OLE database data source. To protect data sources against unwanted changes, map authors must provide passkey access for a layer before you can access the API for enhanced layer functionality. Map authors set up access to the API via the Map Layer Properties dialog box in Autodesk MapGuide Author.

For more information about providing access to the enhanced layer functionality API from a layer, see "Providing Access to the Enhanced Layer Functionality API" on page 37 of Chapter 4, "Enhanced Layer Functionality."

If the map author has provided passkey access, you must set the passkey via the existing Autodesk MapGuide API. If you do not set the passkey, you will get a -5 (Security Violation) error.

For more information about setting a passkey using the API, see "Accessing Secure Data" in the *Autodesk MapGuide Developer's Guide*, which you can open by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Developer's Guide on the Program menu.

MGMapLayerSetup Object

The MGMapLayerSetup object provides run-time access to map layer setup through methods that you can use to change data source names and alter the way map features are linked to databases.

For more information about the MGMapLayerSetup object, see the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

This object has the following new methods:

- getGeometryFunction (see page 153)
- setGeometryFunction (see page 154)
- getClipAdjust (see page 155)
- setClipAdjust (see page 156)
- getClipEnabled (see page 157)
- setClipEnabled (see page 158)
- getPreSQLStatements (see page 159)
- addPreSQLStatement (see page 159)
- clearPreSQLStatements (see page 160)
- getPostSQLStatements (see page 161)
- addPostSQLStatement (see page 162)
- clearPostSQLStatements (see page 163)
- getSpatialQuery (see page 164)
- setSpatialQuery (see page 164)

getGeometryFunction Method

Syntax

```
String getGeometryFunction()
```

Description

Gets the geometry function text that is applied to the geometry column of the SDP layer.

You can apply geometry functions directly to selected features on a layer. For more information about using geometry functions, see “Using Geometry Functions” on page 39.

Parameters

none

Returns

String – Represents the geometry function that is applied to layer features. Returns an empty string if there is no geometry function.

Error Codes

-5 (Security Violation)

-15 (Does not apply)

-18 (Custom Spatial Query)

See Also

setGeometryFunction

setGeometryFunction Method

Syntax

```
boolean setGeometryFunction(String function)
```

Description

Sets the geometry function text to apply to the geometry column of the SDP layer.

You can apply geometry functions directly to selected features on a layer. For more information about using geometry functions, see "Using Geometry Functions" on page 39.

Parameters

function – String representing the geometry function that you want to use.

Returns

boolean – Specifies whether or not the geometry function was successfully set.

True – Indicates that the geometry function was successfully set.

False – Indicates that the geometry function has not been set.

Error Codes

-5 (Security Violation)

-15 (Does not apply)

-18 (Custom Spatial Query)

See Also

getGeometryFunction

getClipAdjust Method

Syntax

```
double getClipAdjust()
```

Description

Gets the clipping adjustment that is applied to the spatial filter of the layer.

You use the clipping adjustment to change the extents of the filter by increasing or decreasing the dimensions of its bounding box so that it is bigger or smaller than the default client window size. For more information about setting the clipping adjustment for filters, see "Changing the Dimensions of the Autodesk MapGuide Spatial Filter" on page 44.

Parameters

none

Returns

double – Value indicating the clipping adjustment for the filter.

0 – Indicates the default filter size which is the size of the client window.

Positive values indicate progressively larger dimensions of the default filter size. Negative values indicate progressively smaller dimensions of the default filter size.

Error Codes

-5 (Security Violation)

-15 (Does not apply)

See Also

setClipAdjust

setClipAdjust Method

Syntax

```
boolean setClipAdjust(double adjust)
```

Description

Sets the clipping adjustment to apply to the spatial filter of the layer.

You us the clipping adjustment to change the extents of the filter by increasing or decreasing the dimensions of the bounding box so that it is bigger or smaller than the default client window size. For more information about setting the clipping adjustment for filters, see "Changing the Dimensions of the Autodesk MapGuide Spatial Filter" on page 44.

Parameters

adjust – Value indicating the clipping adjustment to apply to the filter.

0 – Sets the filter to its default size (size of client window).

Positive values progressively increase the size of the filter. Negative values progressively decrease the size of the filter.

Returns

boolean – Specifies whether or not the clipping adjustment was successfully set.

True – Indicates that the clipping adjustment was successfully set.

False – Indicates that the clipping adjustment was not set.

Error Codes

-1 (Busy)

-5 (Security Violation)

-15 (Does not apply)

See Also

getClipAdjust

getClipEnabled Method

Syntax

```
boolean getClipEnabled()
```

Description

Specifies whether or not the Autodesk MapGuide spatial filtering is being applied to the layer. A value of True indicates that spatial filtering is enabled. A value of False indicates that spatial filtering is not enabled.

A spatial filter reduces the geographic information returned by a spatial query to a particular geography. For more information about applying filters to spatial queries, see "Applying Filters to Spatial Queries" on page 42.

Parameters

none

Returns

boolean – Specifies whether or not the MapGuide spatial filter is enabled.

True – Indicates that spatial filtering is enabled.

False – Indicates that spatial filtering is disabled.

Error Codes

-5 (Security Violation)

-15 (Does not apply)

See Also

setClipEnabled

setClipEnabled Method

Syntax

```
boolean setClipEnabled(boolean enabled)
```

Description

Sets a value that specifies whether or not to apply the MapGuide spatial filtering to the layer. A value of True applies spatial filtering to the layer. A value of False does not apply spatial filtering to the layer.

A spatial filter reduces the geographic information, which is returned by a spatial query, to a particular geography. For more information about applying filters to spatial queries, see "Applying Filters to Spatial Queries" on page 42.

Parameters

enabled – Boolean value indicating whether or not to apply spatial filtering to the MapGuide spatial query.

True – Applies the filter to the spatial query.

False – Does not apply the filter to the spatial query.

Returns

boolean – Specifies whether or not the MapGuide spatial filter was successfully set.

True – Indicates that spatial filtering has been successfully set.

False – Indicates that spatial filtering has not been set.

Error Codes

-1 (Busy)

-5 (Security Violation)

-15 (Does not apply)

See Also

getClipEnabled

getPreSQLStatements Method

Syntax

```
MGCollection getPreSQLStatements()
```

Description

Gets the pre-SQL statements that are applied to the layer.

Pre-SQL statements are executed before Autodesk MapGuide performs a spatial query. You can use them to further customize your query results. For more information about applying pre-SQL statements to queries, see "Using SQL Pass-Through Statements" on page 55.

Parameters

none

Returns

MGCollection – A list of strings representing the pre-SQL statements that are applied to the query.

Error Codes

-5 (Security Violation)

-15 (Does not apply)

See Also

addPreSQLStatement, clearPreSQLStatements

addPreSQLStatement Method

Syntax

```
boolean addPreSQLStatement(String sql)
```

Description

Adds the specified SQL statement to the end of the list of pre-SQL statements that are applied to the layer.

Pre-SQL statements are executed before Autodesk MapGuide performs a spatial query. You can use them to further customize your query results. For more information about applying pre-SQL statements to queries, see "Using SQL Pass-Through Statements" on page 55.

Parameters

sql – String representing the SQL statement to add to the end of the list.

Returns

boolean – Specifies whether or not the specified SQL statement was successfully added to the end of the pre-SQL statement list.

True – Indicates that the SQL statement was successfully added to the end of the list.

False – Indicates that the SQL statement was not added to the end of the list.

Error Codes

-1 (Busy)

-5 (Security Violation)

-15 (Does not apply)

See Also

getPreSQLStatements, clearPreSQLStatements

clearPreSQLStatements Method

Syntax

```
boolean clearPreSQLStatements()
```

Description

Clears the list of pre-SQL statements that is applied to the layer.

Pre-SQL statements are executed before Autodesk MapGuide performs a spatial query. You can use them to further customize your query results. For more information about applying pre-SQL statements to queries, see "Using SQL Pass-Through Statements" on page 55.

Parameters

None

Returns

boolean – Specifies whether or not the list of pre-SQL statements has been successfully cleared.

True – Indicates that the list of pre-SQL statement was successfully cleared.

False – Indicates that the list of SQL statement was not cleared.

Error Codes

-1 (Busy)

-5 (Security Violation)

-15 (Does not apply)

See Also

getPreSQLStatements, addPreSQLStatement

getPostSQLStatements Method

Syntax

```
MGCollection getPostSQLStatements()
```

Description

Gets the list of post-SQL statements that is applied to the layer.

Post-SQL statements are executed after Autodesk MapGuide performs a spatial query. You can use them to further customize your query results. For more information about applying post-SQL statements to queries, see "Using SQL Pass-Through Statements" on page 55.

Parameters

none

Returns

MGCollection – A list of strings representing the post-SQL statements that are applied to the query.

Error Codes

-5 (Security Violation)

-15 (Does not apply)

See Also

addPostSQLStatement, clearPostSQLStatements

addPostSQLStatement Method

Syntax

```
boolean addPostSQLStatement(String sql)
```

Description

Adds the specified SQL statement to the end of the list of post-SQL statements that is applied to the layer.

Post-SQL statements are executed after Autodesk MapGuide performs a spatial query. You can use them to further customize your query results. For more information about applying post-SQL statements to queries, see "Using SQL Pass-Through Statements" on page 55.

Parameters

sql – String representing the SQL statement to add to the end of the list.

Returns

boolean – Specifies whether or not the specified SQL statement was successfully added to the end of the post-SQL statement list.

True – Indicates that the SQL statement was successfully added to the end of the list.

False – Indicates that the SQL statement was not added to the end of the list.

Error Codes

-1 (Busy)

clearPostSQLStatements Method

Syntax

```
boolean clearPostSQLStatements()
```

Description

Clears the list of post-SQL statements so that are not applied to the layer.

Post-SQL statements are executed after Autodesk MapGuide performs a spatial query. You can use them to further customize your query results. For more information about applying post-SQL statements to queries, see "Using SQL Pass-Through Statements" on page 55.

Parameters

none

Returns

boolean – Specifies whether or not the list of post-SQL statements has been successfully cleared.

True – Indicates that the list of post-SQL statement was successfully cleared.

False – Indicates that the list of SQL statement was not cleared.

Error Codes

-1 (Busy)

-5 (Security Violation)

-15 (Does not apply)

See Also

getPostSQLStatements, addPostSQLStatement

getSpatialQuery Method

Syntax

```
String getSpatialQuery()
```

Description

Gets the custom spatial query that is applied to the layer.

A custom spatial query consists of user-defined SQL statements that you can use instead of the default Autodesk MapGuide spatial queries. You can use custom spatial queries if the parameters of the default spatial queries don't include the restraints you need for your data. For more information about custom spatial queries, see "Using a Custom Spatial Query" on page 46.

Parameters

none

Returns

String – Represents the custom spatial query that is applied to the layer. Returns an empty string if no custom spatial query is defined.

Error Codes

-5 (Security Violation)

-15 (Does not apply)

See Also

setSpatialQuery

setSpatialQuery Method

Syntax

```
boolean setSpatialQuery(String query)
```

Description

Sets the custom spatial query to apply to the layer.

Custom spatial queries consist of a set of user-defined SQL statements that you can use instead of the default Autodesk MapGuide spatial queries. You can use custom spatial queries if the parameters of the default spatial queries don't include the restraints you need for your data. For more information about custom spatial queries, see "Using a Custom Spatial Query" on page 46.

Parameters

query – String representing the custom spatial query that you want to apply the layer.

Returns

boolean – Specifies whether or not the custom query was successfully applied.

True – Indicates that the custom query was successfully applied.

False – Indicates that the custom query has not been set.

Error Codes

-1 (Busy)

-5 (Security Violation)

-15 (Does not apply)

See Also

getSpatialQuery

New -18 Custom Spatial Query Error Code

Some of the existing methods for the MGMapLayerSetup and the MGDatabaseSetup objects have a new -18 Custom Spatial Query error code.

The -18 Custom Spatial Query error code indicates that these methods cannot return or set the columns for the specified attributes (key, URLLink, SQL WHERE clause, symbol size, and text alignment) for custom spatial

queries. Custom spatial queries consist of complex and nested SQL statements, making it impossible to determine which column contains the specified attribute.

For example, the following SQL statement is confusing.

SELECT (SELECT B.F_ID FROM ALLTYPES B WHERE B.F_ID=A.F_ID), A.F_ID || ' - ' || A.F_NAME, A.F_ID, A.F_GEOM, A.F_NAME FROM ALLTYPES A

The key column is (SELECT B.F_ID FROM ALLTYPES B WHERE B.F_ID=A.F_ID). However, this is difficult to determine and confusing because users are accustomed to seeing column names rather than an entire SQL statement.

In comparison, it is much easier to determine the key column for the following query:

SELECT F_ID, F_GEOM, F_NAME FROM ALLTYPES

It is clear that the key column is F_ID.

MGMapLayerSetup Object

The following MGMapLayerSetup methods include the new -18 (Custom Spatial Query) error code:

- getSDPKeyColumn
- setSDPKeyColumn
- getSDPKeyColumnType
- setSDPKeyColumnType

For more information about theses methods, see the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

MGDatabaseSetup Object

The following MGDatabaseSetup methods include the new -18 (Custom Spatial Query) error code:

- getNameColumn
- setNameColumn
- getUrlLinkColumn
- setUrlLinkColumn
- getWhereClause
- setWhereClause
- getSymbolAngleColumn

- setSymbolAngleColumn
- getSymbolWidthColumn
- setSymbolWidthColumn
- getSymbolHeightColumn
- setSymbolHeightColumn
- getTextHeightColumn
- setTextHeightColumn
- getTextAngleColumn
- setTextAngleColumn
- getTextHorizAlignColumn
- setTextHorizAlignColumn
- getTextVertAlignColumn
- setTextVertAlignColumn

For more information about these methods, see the Autodesk MapGuide Viewer API Help. You can access the MapGuide Viewer API Help by clicking Autodesk MapGuide 6.5 ➤ Documentation ➤ Autodesk MapGuide Viewer API Help on the Program menu.

Index

F

G

I

K

L

M

N

O

P

Q

R

S

T

U

V

W