
# TerraGo Publisher for ArcGIS 6.0.3

# Software Developer's Guide

# Addendum for ArcGIS Server

# Introduction

This document is an addendum to *TerraGo Publisher for ArcGIS Software Developer's Guide* (referred to as *Software Developer's Guide* in the remainder of this document). It contains information unique to software development in the ArcGIS Server execution environment. Memory management differs in this environment and requires special techniques detailed in ESRI's ArcGIS Server documentation.

Please note that GeoMark Export, GeoMark Import and GeoPDF View are not supported in TerraGo Publisher for ArcGIS Server.

# Document Conventions

The given code sample is for C#. Note that references to TerraGo.MAP2PDF.ESRI.Export.dll, TerraGo.MAP2PDF.ESRI.View.dll and TerraGo.MAP2PDF.ESRI.GeoMark.dll may need to be added to your project before they will compile and run.

# Differences and Limitations

TerraGo COM interfaces and their usage (other than memory management) in the ArcGIS Server environment are identical to those documented in *Software Developer's Guide*. However, there are some limitations due to some ArcObjects interfaces that are only available in the ArcGIS Desktop environment. As noted in *Software Developer's Guide*, the ArcMap hyperlink base property is not available to the ExportGeoPDF coclass, so hyperlinks using a relative path will probably not work correctly.

With this release ExportGeoPDF no longer stores configuration information as a graphic element in the PageLayout object. Instead configuration information is persisted in a GeoPDFExportConfig object stored in the PageLayout's IPageLayoutExtensions interface. When a GeoPDFExportConfig object is not accessible, ExportGeoPDF creates one and initializes it with the user's stored preferences.

# GeoPDF Export Code Example

This code snippet shows how an exporter is created, configured and an export performed. The logic is similar to the export example in *Software Developer's Guide*.

```
// Create the export object on the server
// Predefined:
//   IServerContext ags_context;
//   IActiveView layoutView;
//   string exportFileName;
//   double resolution;

IExport ex_geopdf =
  ags_context.CreateObject("map2pdf.ExportGeoPDF") as IExport;
IExportGeoPDF2 export2 =
  ex_geopdf as IExportGeoPDF2;
export2.ActiveView = layoutView;

// Set the export parameters
ex_geopdf.ExportFileName = exportFileName;
ex_geopdf.Resolution = resolution;

IGeoPDFConfig4 config4 =
  ex_geopdf as IGeoPDFConfig4;
config4.ToolbarCompatibility =
```

```
  geoPDFToolbarCompatType.geoPDFToolbar50;
config4.OptimizeFile = true;

// Get the screen resolution
int screenResolution = 96;

// Set the export bounds
double scaleFactor = ex_geopdf.Resolution / screenResolution;
tagRECT exportRect;
exportRect.left = Convert.ToInt32(layoutView.ExportFrame.left * scaleFactor);
exportRect.top = Convert.ToInt32(layoutView.ExportFrame.top * scaleFactor);
exportRect.right = Convert.ToInt32(layoutView.ExportFrame.right * scaleFactor);
exportRect.bottom = Convert.ToInt32(layoutView.ExportFrame.bottom *
scaleFactor);
IEnvelope pixelBoundsEnv =
  ags_context.CreateObject("esriGeometry.Envelope") as IEnvelope;
pixelBoundsEnv.PutCoords(exportRect.left,
                         exportRect.top,
                         exportRect.right,
                         exportRect.bottom);
ex_geopdf.PixelBounds = pixelBoundsEnv;

// Start the export
int hdc = ex_geopdf.StartExporting();

ex_geopdf.FinishExporting();
ex_geopdf.CleanUp();
```

## GeoPDF View Code Example

This code snippet shows how to add a GeoPDFView layer to a map in the ArcGIS Server environment.

```
// Create the layer object on the server
// Predefined:
//   IServerContext ags_context;
//   string geoPDFName;
//   IMap map;

IGeoPDFLayer geopdflayer =
  ags_context.CreateObject("map2pdf.GeoPDFViewLayer") as IGeoPDFViewLayer;

// Initialize the layer
geopdflayer.LoadMapFrame(geoPDFName,
                         0,
                         0,
                         null);

// Add the layer
map.AddLayer(geopdflayer as ILayer);
```

# GeoMark Import Code Example for Shapefile

This code snippet shows how to perform a GeoMark import for Shapefile in the ArcGIS Server environment.

```
// Create the helper object on the server
// Predefined:
//   IServerContext ags_context;
//   string importTwzName;
//   string importFolder;
//   string importBase;
//   IMap map;

IGeoMarkImport import =
  ags_context.CreateObject("map2pdf.GeoMarkHelper") as IGeoMarkImport;

// Perform the export
ILayer layer = null;
layer = import.CreateLayer(importTwzName,
                           importFolder,
                           importBase);
map.AddLayer(layer);
```

# GeoMark Import Code Example for File Geodatabase

This code snippet shows how to perform a GeoMark import for File Geodatabase in the ArcGIS Server environment.

```
// Create the helper object on the server
// Predefined:
//   IServerContext ags_context;
//   string importTwzName;
//   string importFgdb;
//   IMap map;

IGeoMarkImport import =
  ags_context.CreateObject("map2pdf.GeoMarkHelper") as IGeoMarkImport2;

// Perform the import
import.CreateFgdbFromTwx(importTwzName, importFgdb);
import.AddFGDBGeoMarks(importFgdb, map);
```

# GeoMark Export Code Example

This code snippet shows how to perform a GeoMark export of a map layer in the ArcGIS Server environment.

```
// Create the export object on the server
// Predefined:
//   IServerContext ags_context;
//   string exportTwzName;
//   IMap map;
//   ILayer layer;

IGeoMarkExport ex_geomark =
  ags_context.CreateObject("map2pdf.GeoMarkExporter") as IGeoMarkExport;
IActiveView activeView = map as IActiveView;
```

```
// Perform the export
ex_geomark.ExportLayer(layer as IFeatureLayer,
                       activeView.Extent,
                       esriSpatialRelEnum.esriSpatialRelIntersects,
                       false,
                       false,
                       false,
                       exportTwzName);
```