



ArcGIS® Server in Practice Series

High-Capacity Map Services: A Use Case with CORINE Land-Cover Data

Copyright © 2009 ESRI
All rights reserved.
Printed in the United States of America.

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts and Legal Services Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100 USA.

The information contained in this document is subject to change without notice.

ESRI, the ESRI globe logo, ArcGIS, ArcMap, ArcInfo, ADF, Maplex, ArcSDE, ArcIMS, ModelBuilder, *ArcUser*, www.esri.com, and @esri.com are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Other companies and products mentioned herein may be trademarks or registered trademarks of their respective trademark owners.

ArcGIS Server in Practice Series

High-Capacity Map Services: A Use Case with CORINE Land- Cover Data

An ESRI White Paper

Contents	Page
About the ArcGIS Server in Practice Series	1
Introduction.....	1
GIS Dataset	2
Map Service Requirements	3
Map Data Dissemination Options in ArcGIS Server.....	4
Cached Map Service	4
Query Map Service to Return Features.....	5
Dynamic Map Service.....	5
Best Practices for Map Services	6
Tip #1	6
Tip #2	8
Tip #3	10
Tip #4	11
Tip #5	12
Tip #6	14
ArcGIS Server Enterprise Deployment Examples.....	18
Single SOM Load Balance.....	18
High-Availability SOM Load Balance	21
Web/SOM/SOC Network Load Balancing.....	23
Scalability Deployment Strategies Assessment	24
Conclusion	26

Contents	Page
Appendixes	
Appendix A: Hardware Configuration of the Test Servers	27
Appendix B: ArcGIS Server and Business Continuity	30

ArcGIS Server in Practice Series

High-Capacity Map Services: A Use Case with CORINE Land-Cover Data

About the ArcGIS Server in Practice Series

The ArcGIS® Server in Practice series provides practical information for the configuration and implementation of ArcGIS Server solutions. Documents in this series explore common and well-defined user workflows and system configurations. They provide example use cases for describing ArcGIS Server best practices. The emphasis of this series is to examine use cases from a holistic point of view and provide useful information enabling users to understand how to optimally configure and implement ArcGIS Server for real-world applications.

Introduction

Scenario: A data provider organization wants to publish an efficient and effective map service using ArcGIS Server. The map service will contain a large geographic information system (GIS) dataset (e.g., one with over a million features) that enables different types of advanced visualization and analysis to be performed on it from many diverse client applications. Client applications consist of a mix of Web browser applications and GIS desktop programs such as ArcGIS Desktop, AutoCAD, and Web Map Service (WMS)-capable applications. The map service needs to support heavy loads of up to 250,000 map requests per hour and allow client applications to perform operations on the data. Some example operations include on-the-fly projection, feature filtering via SQL queries, layer filtering (i.e., turning layer visibility on and off), and attribute/spatial queries. Since the map service will support many concurrent users, it is referred to as a high-capacity map service.

This document contains relevant information that helps guide users in implementing an ArcGIS Server solution that addresses the scenario described above. It discusses some best practices for creating and publishing map services. It also presents some example ArcGIS Server system deployments and their performance metrics, with respect to the use case.¹ The objective is to provide users with some practical information on how to implement high-capacity map services with ArcGIS Server successfully. Note that this document is a collection of tips and deployment considerations for high-capacity map services; it is not a detailed step-by-step implementation guide. An ArcGIS Server Standard Enterprise edition deployment is used for this use case.

This document contains the following sections:

- GIS Dataset
- Map Data Dissemination Options in ArcGIS Server

¹ A comprehensive discussion of all possible ArcGIS Server deployment configurations that can address the scenario's business requirements is beyond the scope of this document. Only the most common implementation patterns are discussed.

- Best Practices for Map Services
- ArcGIS Server Standard Enterprise Deployment Examples
- Summary
- Appendixes

The GIS Dataset section gives an overview of the data used for the ArcGIS Server Enterprise deployment examples that are discussed later in the document. It also outlines some requirements of this data when used in a map service, in the context of this use case. The Map Data Dissemination Options in ArcGIS Server section describes the possible approaches for distributing geographic data with ArcGIS Server. It explains why dynamic map services were selected as the means to publish the data and the different application programming interfaces (APIs) that clients can use to interact with them. The Best Practices for Map Services section presents a collection of tips and strategies that users should consider when developing map services, especially when supporting heavy loads. In the final section, ArcGIS Server Standard Enterprise Deployment Examples, three different deployment scenarios of ArcGIS Server Standard Enterprise are discussed. Some performance metrics are also presented and compared with respect to the scenario.

The target audience for this document consists of intermediate to advanced users with a good understanding of ArcGIS Server concepts. Some knowledge of enterprise system configurations and system performance tuning would also be beneficial.

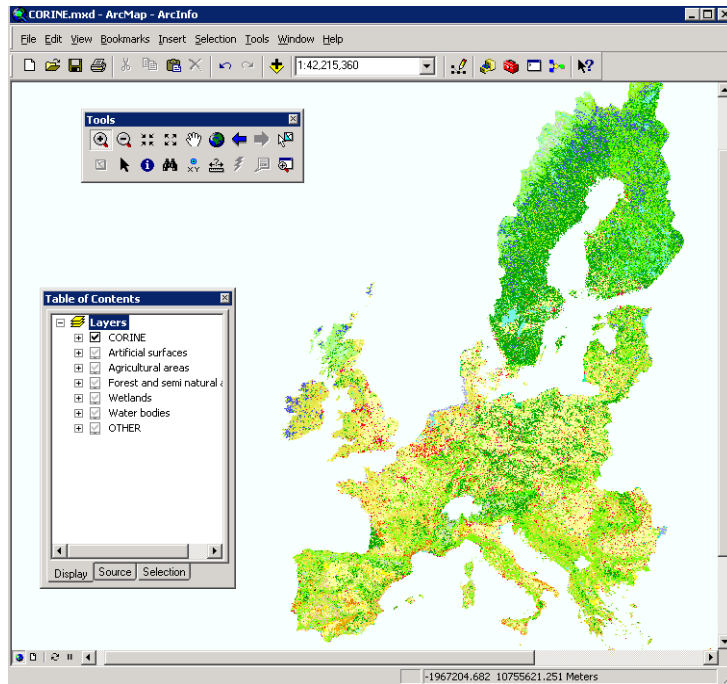
GIS Dataset

Land-use information from the European Coordination of Information on the Environment (CORINE) Program was used as the representative large GIS dataset in this scenario. The CORINE Land Cover (CLC)² 2000 database is provided by the European Environmental Agency and available in the public domain. It seamlessly covers all countries in the European Union (EU) and contains information on land cover, water resources, air quality, and so forth. It is a large and comprehensive environmental GIS dataset beneficial for both government agencies and private organizations.

The CLC GIS dataset contains over 1,800,000 land base polygon features that vary in size from approximately 25 to several thousand hectares (see figure 1). Large features in this dataset often have very complex shape boundaries that are composed of thousands of vertices. For example, a single polygon feature's boundary could be composed of over 60,000 vertices.

² A report on the CORINE land-cover database can be found at <http://www.eea.europa.eu/publications/CORO-landcover>.

Figure 1
CORINE Land-Cover Data in ArcMap



Map Service Requirements

The objective of the data provider organization is to publish the CLC GIS dataset as a map service that can be used as both a basemap to support other data and operational layers for analysis and queries. Therefore, this requires that the map service provide mashup capabilities. The term *mashup* in Web development refers to the concept of combining data and functionality from two or more external sources to create a new service in a Web page or application.³ This capability enables clients to not only access the map service information but also manipulate and combine it with other sources of information (e.g., other GIS data layers) to transparently build and support more complex applications.

Given the nature of the CLC GIS dataset—its coverage of a large geographic area, polygon features composed of many vertices, and the broad range of potential users (e.g., from casual users to GIS professionals to decision makers) that will be using it—publishing this data online as an effective map service presents some special requirements.

Based on the business requirements of the scenario, a map service of the CLC GIS dataset should be

- Performance: It should yield subsecond response times for map retrieval.
- Scalable: It should be able to support hundreds of concurrent users while keeping performance at desired levels (e.g., subsecond response times).

³ More information on Web development mashups can be found at [http://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)).

- Consistently available: There should be no down time during which the data is unavailable; it should always be readily available for use.
- Able to support multiple projections: The map service should be able to support many different projections on the fly such as Lambert, Universal Transverse Mercator (UTM), Stereographic, and so forth.
- Multiscale: An end user should be able to define a working map scale that can vary between scales of 1:25,000 (e.g., local management) and 1:1,000,000 (e.g., national and multinational management).
- Able to handle layer and feature filtering: An end user should be able to control the visibility at both the layer and feature levels, for example, only displaying "permanently irrigated land" or hiding "mineral extraction sites."
- Interoperable: It should support and conform to Open Geospatial Consortium, Inc. (OGC), standards.⁴ This is a very common business requirement for many European countries. This also refers to the idea that the map service should be accessible to a variety of clients including Web browsers and desktop applications of different types.

Map Data Dissemination Options in ArcGIS Server

An important general concept for developing ArcGIS Server solutions is that the business workflow/requirements of the use case should drive the solution design and implementation. This statement may seem obvious, but it can be challenging to put this concept into practice. The next section demonstrates applying this concept to the development of the ArcGIS Server map service solution for the scenario.

There are many different ways to disseminate map data over the Web with ArcGIS Server. Three different approaches for publishing map data that can potentially address the business requirements of the scenario are (1) cached map service, (2) query map service to return features, and (3) dynamic map service. The three different approaches for publishing a map service are briefly discussed and contrasted in the context of their applicability to satisfying the use case requirements.

Cached Map Service

This service allows clients to render map content quickly by exposing a collection of pregenerated map tiles. A cached map service⁵ provides the most highly performing and scalable way to deliver map content on the Web, although it does not address some key business requirements of the use case. For example, it does not support on-the-fly projection across all types of clients, nor does it allow feature filtering on a per-request basis. For these reasons, a cached map service is not considered in this document to be a solution to the scenario. Note that it would yield significantly better map throughput results compared with a dynamic map service.

⁴ More information on OGC can be found at <http://www.opengeospatial.org/>.

⁵ More information on cached map services can be found in the ArcGIS Server Help topic *What is map caching?*

Query Map Service to Return Features

In this approach, the map service is configured to return a set of features to the client application.⁶ The query capabilities of ArcGIS Server map services are exploited by allowing end clients to download geographic features (i.e., geometry plus attributes). The client renders the features with its own graphic capabilities. Given the nature of the CLC GIS dataset, which includes very large features composed of thousands of vertices, this approach is not practical for the use case. While the query capabilities of the CLC map service were available during the testing of the ArcGIS Server Standard Enterprise deployment examples, considerations for this option were outside the scope of the document.

Dynamic Map Service

This service is rendered at the time it is requested by a client (i.e., it does not use caches). The client invokes the map service and sets the size, projection, extent, scale, layers, and features to be displayed in the final map image. The image is generated on the server and downloaded to the client application for display. It is not as fast as a cached map service, but it satisfies all business requirements of the use case. It is strongly recommended that users tune a dynamic map service as much as possible.⁷ Configuring and scaling dynamic map services to support heavy loads will be the main focus of upcoming sections of this document.

Table 1 summarizes the three different approaches for publishing map data using ArcGIS Server and their ability to satisfy the requirements of the scenario.

Table 1
Types of Map Services and How They Satisfy the Use Case Requirements

Requirement	Cached Map Service	Query Map Service to Return Features	Dynamic Map Service
Performance	Very High	High , for small amounts of data to be displayed on the map	High
Scalable	Yes	Yes	Yes
Consistently available	Yes	Yes	Yes
Able to support multiple projections	No , except when supported by the client (e.g., ArcGIS Desktop)	Yes	Yes
Multiscale	Yes . Fixed scales as defined in the map cache tiling scheme	Yes, any scale	Yes, any scale
Able to handle layer and feature filtering	No	Yes	Yes

⁶ More information on enabling a map service to return features can be found in the ArcGIS Server Help topic *Tuning and configuring services*.

⁷ More information on tuning a dynamic map service can be found in the ArcGIS Server Help topic *Map authoring considerations for ArcGIS Server*.

ArcGIS Server map services support several different development environments for Web and systems development/integration.⁸ The REST⁹ API is used by client applications built on top of the ArcGIS Web mapping APIs (e.g., [JavaScript™](#), [Flex™](#), and [Silverlight™](#)). For this document, the REST interface was used to access the information in the dynamic map service during the stress testing of the ArcGIS Server Standard Enterprise deployment examples.

Other clients, such as ArcGIS Desktop, ArcGIS Explorer, and client applications built with the ArcGIS Server Web Application Development Frameworks (ADF™)—specifically [.NET](#) and [Java™](#)—use the SOAP¹⁰ API. Finally, other third-party client applications can also access the dynamic map service data with the WMS protocol¹¹ and Keyhole Markup Language (KML).¹²

Best Practices for Map Services

In the previous section, a dynamic map service was determined to be the best approach for publishing the CLC GIS dataset in ArcGIS Server. The performance of map services in ArcGIS Server can vary greatly depending on many different factors. Next, six tips and best practices for ArcGIS Server map services are discussed, with an emphasis on optimizing performance.

Tip #1

Use the Map Service Publishing toolbar to optimize map documents that will be published in ArcGIS Server.

ArcGIS Desktop 9.3.1 introduces a new Map Service Publishing toolbar in ArcMap™ (see figure 2). It is designed to help properly tune and optimize map documents for publication in ArcGIS Server.¹³

Figure 2
Map Service Publishing Toolbar



The icons on this toolbar provide the following capabilities (from left to right):

- Analyze map document.
- Preview map service.
- Save map service definition.
- Publish map service.
- Set display options.

⁸ More information on developing with ArcGIS Server can be found in the ArcGIS Server Help topic *Developing with ArcGIS Server: An overview*.

⁹ More information on REST can be found at <http://en.wikipedia.org/wiki/REST>.

¹⁰ More information on SOAP can be found at <http://en.wikipedia.org/wiki/SOAP>.

¹¹ More information on WMS can be found at http://en.wikipedia.org/wiki/Web_Map_Service.

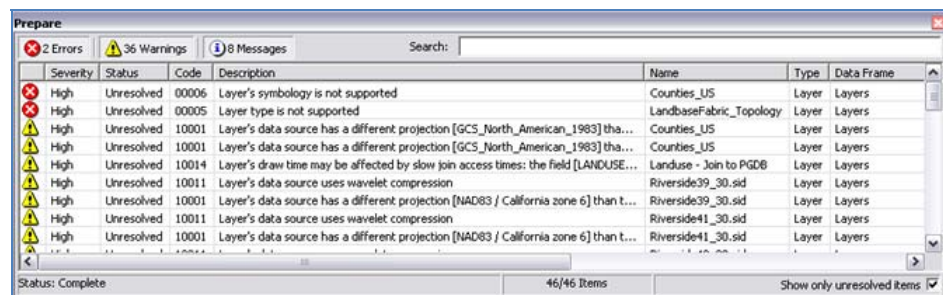
¹² More information on KML can be found at <http://en.wikipedia.org/wiki/KML>.

¹³ More comprehensive information on the new Map Service Publishing toolbar can be found in the ArcGIS Desktop Help topic *Publishing optimized map services*.

The Analyze map button (shown in the red square in figure 2) provides the most significant functionality. It cycles through all the layers in a map document and generates a report that identifies potential performance bottlenecks and/or errors that should be addressed before publishing a map service from the map. Some example issues that it might identify include layers with data sources that do not have spatial indexes; layers with complicated symbology rules (e.g., use of the Maplex[®] labeling engine); unsupported layer types; and two or more layers with data sources stored in different projections, which requires on-the-fly reprojection for map rendering, degrading performance.

The issue list is presented in a report table in ArcMap (see figure 3). The issues are categorized into three groups: (1) errors, or issues that must be fixed; (2) warnings, or issues that may affect drawing performance; and (3) information, or identification of potential display differences between the map and the map service.¹⁴

Figure 3
Analyze Map Report Window



Severity	Status	Code	Description	Name	Type	Data Frame
High	Unresolved	00006	Layer's symbology is not supported	Counties_US	Layer	Layers
High	Unresolved	00005	Layer type is not supported	LandbaseFabric_Topology	Layer	Layers
High	Unresolved	10001	Layer's data source has a different projection [GCS_North_American_1983] tha...	Counties_US	Layer	Layers
High	Unresolved	10001	Layer's data source has a different projection [GCS_North_American_1983] tha...	Counties_US	Layer	Layers
High	Unresolved	10014	Layer's draw time may be affected by slow join access times: the field [LANDUSE...	Landuse - Join to PGDB	Layer	Layers
High	Unresolved	10011	Layer's data source uses wavelet compression	Riverside39_30.sid	Layer	Layers
High	Unresolved	10001	Layer's data source has a different projection [NAD83 / California zone 6] than t...	Riverside41_30.sid	Layer	Layers
High	Unresolved	10011	Layer's data source uses wavelet compression	Riverside41_30.sid	Layer	Layers
High	Unresolved	10001	Layer's data source has a different projection [NAD83 / California zone 6] than t...	Riverside41_30.sid	Layer	Layers

Addressing the issues identified in the report will ensure optimal performance from the map service published from the map. ArcMap provides suggestions on how to fix the issues by right-clicking the issue(s) in the report table. Once potential issues are resolved, the map document can be exported to a Map Service Definition file (.msd). ArcGIS Server dynamic map services created from .msd files are known as optimized map services, which are discussed in more detail in the next tip.

As a side note to this tip, if the map document data sources are stored in an ArcSDE[®] geodatabase, proper and efficient geodatabase tuning following geodatabase management best practices is another factor that may indirectly help optimize map service performance. A discussion on ArcSDE geodatabase performance tuning and configuration is beyond the scope of this document, but there are many documentation resources available in the ArcGIS Server Help documentation.¹⁵ Note that geodatabase tuning strategies may vary depending on the database management system (DBMS) platform (e.g., DB2[®], Informix[®], Oracle[®], PostgreSQL, or SQL Server[®]) on which the ArcSDE geodatabase is implemented. Finally, there is the Geodatabase Toolset

¹⁴ More information on display differences can be found in the ArcGIS Desktop Help topic *Drawing differences between ArcMap and optimized map services*.

¹⁵ Information on ArcSDE geodatabase tuning can be found in the ArcGIS Server Help topic *An overview of tuning an ArcSDE geodatabase*. Also see the transcript of the ESRI Instructional Series podcast [Five Best Practices for Maintaining an Enterprise Geodatabase](#).

(GDBT),¹⁶ a collection of system analysis tools and functions that can help investigate an ArcSDE geodatabase and diagnose some performance issues.

Tip #2 When possible, use optimized map services (.msd files) instead of standard map services (.mxd files).

As discussed in the previous tip, the Map Service Publishing toolbar in ArcMap helps tune map documents for optimal performance. After the map has been tuned, ESRI recommends saving the map document (.mxd file) as an .msd file. A map service definition is a configuration file for an optimized map service, which yields significant performance gains over map services based on standard map documents. They are the recommended format for the publication of dynamic maps in ArcGIS Server.

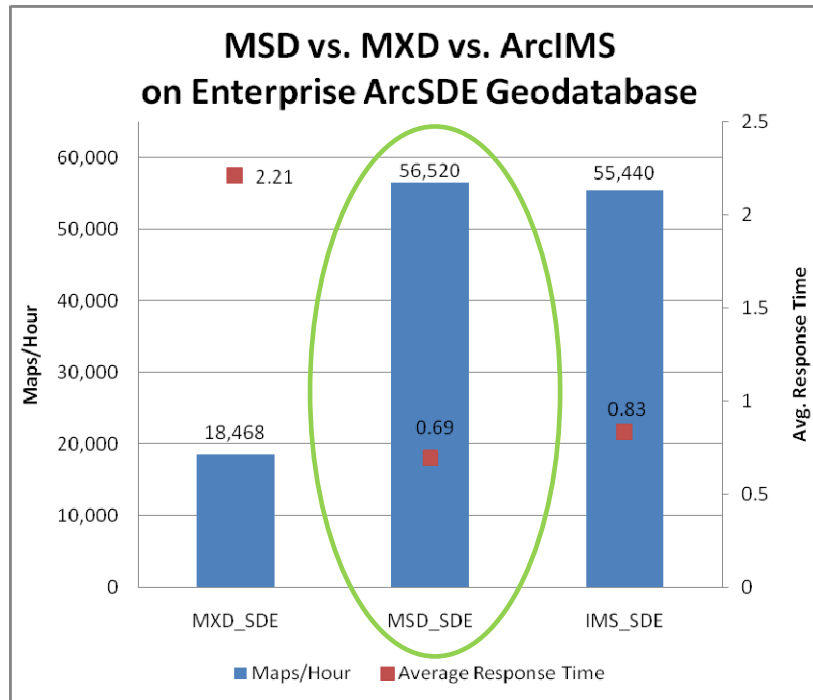
A performance comparison between an ArcIMS[®] map service file (.axl),¹⁷ an ArcGIS Server standard map service, and an ArcGIS Server optimized map service was completed to support this tip. The test measured the number of maps generated per hour (i.e., maximum throughput) using the CLC GIS dataset as the source data. Map requests were made at different scales and with different spatial extents, and all requests forced the map service to generate a new map output at the proper local projection for the particular area of interest.

For this test, ArcGIS Server Standard Enterprise was configured as follows: the Web server, server object manager (SOM), and server object container (SOC) components were installed on one eight-core Windows[®] server, while an enterprise ArcSDE geodatabase was installed on a second eight-core Windows server. For more details on the test hardware configuration and the characteristics of the requests made, see appendix A. The results of the comparison test are shown in figure 4.

¹⁶ The GDBT is available from ESRI at <http://www.esri.com/software/arcgis/extensions/gdbt/index.html>.

¹⁷ An .axl file is the functional equivalent of a map document for ArcIMS written in XML.

Figure 4
MXD/MSD/AXL Performance Comparison Test Results



The comparison test results between the three map services show that overall, the optimized map service (MSD_SDE) performed slightly better than the ArcIMS map service (IMS_SDE) and significantly better than the standard map service (MXD_SDE). Figure 4 shows the average response times¹⁸ for map requests at the point where the server was working at its maximum throughput:

- Optimized map service—56,520 maps/per hour; average response time: 0.69 seconds
- ArcIMS map service—55,440 maps/per hour; average response time: 0.83 seconds
- Standard map service—18,468 maps/per hour; average response time: 2.21 seconds

Clearly, the optimized map service outperformed the standard map service.

Note: The comparison test results show that the optimized map service only performed slightly better than the ArcIMS map service. However, it is important to highlight that performance differences between the three types of map services can vary depending on the complexity of the map contents. In general, optimized map services tend to outperform equivalent ArcIMS map services and standard map services as the complexity of the map increases. The more layers and complex symbology contained in the map, the more differences in map throughput will occur in favor of optimized map services.

¹⁸ In this document, *response time* is defined as the measure of elapsed time between the initial request and the last byte of data received by the client. This response time does not account for rendering time on the client or network latencies.

Tip #3 Adjust the information displayed in the map document according to the map scale.

The amount of time needed to query and render features is in part related to the number and complexity of the features being displayed. In other words, the more vertices and line segments being requested by a client, the more time it will take for both the server's database tier to retrieve the information and the drawing engine to physically render the map image.

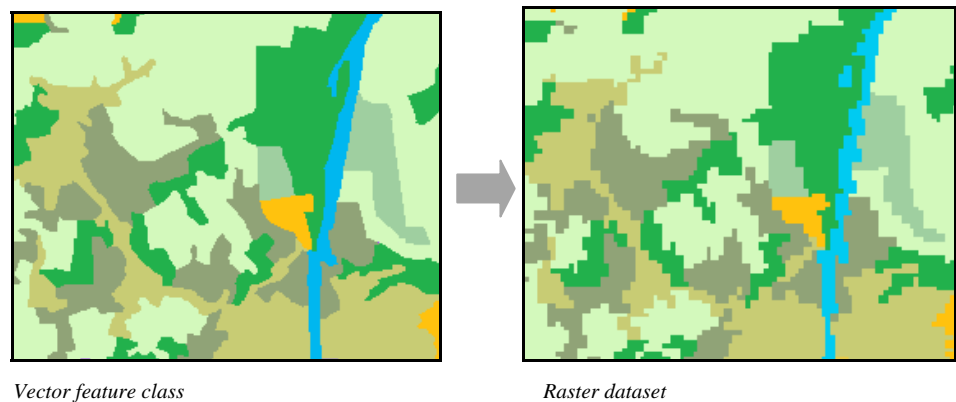
Scale dependencies on map service layers should be set to limit the number of features being displayed to those that make sense cartographically for a given map scale. For example, a street map typically uses the following scale dependency pattern: only display major roads (e.g., highways) at small scales (e.g., 1:250,000), and display major and minor roads (e.g., highways and city streets) at larger scales (e.g., 1:50,000). Rendering all the street features at small scales would not only overload the map with information (i.e., too many features in the display) but also stress the server unnecessarily. Filtering the features to be displayed at specific map scale intervals is a simple cartographic generalization method. Using appropriate scale dependencies is a technique that can potentially yield tremendous performance improvements with minimal effort.

The CLC GIS dataset in this scenario is provided as a single layer of information (i.e., as a single feature class) requiring hundreds of thousands of complex features to be rendered at small scales (e.g., at the continental level). Since the information is geographically contiguous, filtering the features as described in the street map example is not an applicable strategy. For instance, arbitrarily selecting one land cover type to be hidden at small scales would yield obvious gaps in the map display. An alternative approach might be to filter the land-cover types based on their geographic area—land covers with small areas would not display at small scales and only appear at larger scales. However, it would still be difficult to avoid gaps in the map display.

For this use case, a different data generalization approach was taken. The CLC GIS dataset was rasterized to derive a new layer of information for optimal display at small map scales. The CLC vector polygon feature class was converted to a raster data layer with the Polygon to Raster conversion tool.¹⁹ Small scales are defined as those ranging from 1:1,000,000 to 1:45,000,000. At these scales, the map could display millions of features in one map view. Figure 5 illustrates a section of the CLC GIS dataset converted from a vector polygon feature class to a raster dataset.

¹⁹ More information on the Polygon to Raster conversion tool is available in the ArcGIS Desktop Help topic *Polygon to Raster (conversion)*. The tool is available as part of the ArcGIS Desktop Spatial Analyst extension.

Figure 5
CLC GIS Dataset: Vector Feature Class Rasterized into a Raster Dataset



The raster dataset representation of the CLC GIS dataset has a resolution of 100 x 100 meter cell size. It eliminated land-cover polygon features that were smaller than the cell resolution size and simplified the shapes of the other areas. When displayed at small scales, there is no real difference to the eye between the vector and raster versions of the CLC GIS dataset. In figure 5, the differences are visible because the data is shown at a map scale that is beyond the threshold 1:1,000,000 to conceptually illustrate the effects of rasterization on the vector data. In the final Web application, the data would not be displayed at map scales where the pixilation is visible. Since the CLC has fewer than 256 categories of land cover, an eight-bit raster was created and assigned pixel values according to the land-cover type representative for each pixel. A color map that matched the original CLC GIS dataset was then applied to the raster dataset.

The CLC raster dataset is an optimal cartographic representation of the land-cover data for scales down to 1:1,000,000 and yielded a significant increase in performance for map requests—close to 50 times faster for small map scales. Scale dependencies were used to control the visibility of the CLC raster dataset at scales of 1:1,000,000 and above. For map scales beyond 1:1,000,000, the vector CLC GIS dataset was used.

There are multiple techniques for cartographic generalization, including filtering of features, simplification of geometries, aggregation of geometries, and so forth. In this document, rasterization was only briefly described, because it was the technique that made sense for this particular dataset and scenario.²⁰

Tip #4 Review the map service log to further investigate response times on the map service.

ArcGIS Server records events that occur in the server (including errors) to log files. Log files are an important tool for monitoring and troubleshooting problems with ArcGIS Server. The SOM is the centralized logging mechanism for ArcGIS Server and records all events that occur in the SOM, SOCs, and the services they contain. The logs consist of two files: an XML file and a .dat file. The XML file contains the <log> tags, and the .dat file contains the messages that are constantly being appended.²¹

²⁰ This tip only briefly discusses some data generalization strategies. A more comprehensive discussion is available in the ArcGIS Desktop Help topic *An overview of the Generalization toolset*.

²¹ More information on log files is available in the ArcGIS Server Help topic *How log files work*.

All loggable messages have six explicit properties:

- Time—The time at which the logged event occurred
- Level—The level of detail of the message
- Code—The result code associated with the message
- Target—Either the server or the name of a service configuration associated with the message
- Thread—The SOM process thread that generated the message
- Message—The human-readable description of the logged event²²

Messages in the map service log can be categorized into two groups: MapServer messages and Carto messages. The former group is a set of errors, warnings, and detailed messages written by the map service, while the latter group is a set of detailed messages about the map drawing. It is the Carto message group that is the focus of this tip.

The Carto messages show detailed information on all the rendering phases of a map service that occur when the *exportMapImage* method is called by ArcGIS Server.²³ For example, this category of log messaging will record the time needed to query, render, and label every layer in a map document. To enable Carto messages to be logged, the Info:Detailed logging level must be enabled in ArcGIS Server. Examining the Carto messages in the map service log may provide additional guidance on how to improve map service performance. For example, perhaps one layer in the map renders unusually slowly compared with other layers; this suggests that the display properties of the layer might not be optimized compared with the other layers. If the layer's source data was from an ArcSDE geodatabase, additional geodatabase tuning may be needed.

Tip #5 Implement a production–publication geodatabase replication workflow to support the ArcGIS Server system.

The geodatabase is the native data model in ArcGIS. It is the foundation for data storage and management in many ArcGIS systems. Enterprise ArcSDE geodatabases are designed to support medium to large GIS systems. They leverage the underlying DBMS architecture to provide better security, distributed file management, backup/recovery capabilities, and data integrity. ArcSDE technology helps provide additional geodatabase functionality such as versioning, geodatabase replication, and historical archiving.²⁴

In most cases, organizations work with an enterprise ArcSDE geodatabase in a production scenario. In other words, it is usually accessed by multiple ArcGIS Desktop clients concurrently working with its GIS datasets (e.g., editing data) within an internal

²² The message component also includes the process and thread IDs of the container processes where the service that generated the message is running and the SOC machine on which that process is running (if applicable).

²³ More information on the different types of Carto log messages is available in the ArcGIS Server Help topic *Map service log codes*.

²⁴ More information on enterprise ArcSDE geodatabases can be found in the *ArcUser* article [Enterprise Geodatabase 101](#).

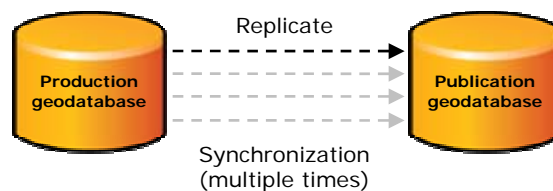
network. Additional ArcGIS clients may also be accessing the enterprise ArcSDE geodatabase data to perform GIS analyses or other tasks. In these scenarios, the enterprise ArcSDE geodatabase is commonly called a transactional geodatabase or production geodatabase.

Given the unique requirements of the Internet and Web usage, it might not be ideal to directly reference and/or access a production geodatabase to support ArcGIS Server map services for external Web use. This is particularly true for the use case, since the objective is to create a map service of the CLC GIS dataset that supports up to 250,000 requests per hour over the Web. To avoid causing any potential issues with the production geodatabase, it would be better to have the ArcGIS Server map service reference and/or access a *copy* of the production geodatabase.

Geodatabase replication²⁵ enables GIS data to be distributed across two or more geodatabases in a manner that allows them to synchronize any data changes that are made. This functionality supports the full geodatabase data model and uses replicas. A replica is data in the geodatabase that is copied from a source geodatabase to a target geodatabase during the replication process. It describes what data has been replicated and includes information needed to synchronize changes. Geodatabase replication can be used in connected and disconnected environments. It works with local geodatabase connections over a network, as well as over the Internet using ArcGIS Server geodata services.²⁶

One-way replication satisfies the requirements of the workflow described previously. This strategy is ideal for production–publication scenarios where there are two geodatabases, one for data creation and editing (i.e., the production/source geodatabase) and the other only containing finished data that is shared and distributed to other users (i.e., the publication/target geodatabase) (see figure 6).

Figure 6
One-Way Replication in a Production–Publication Workflow



GIS data in the production geodatabase is actively edited, but GIS data in the publication geodatabase is considered to be read-only. In this workflow, data changes can be sent multiple times in one direction from the production geodatabase to the publication geodatabase, possibly at regular intervals (e.g., weekly, monthly, quarterly, or annually). Changes can be sent as often as required by the business workflow. At ArcGIS 9.2, this workflow required that both participating geodatabases be ArcSDE geodatabases. At ArcGIS 9.3 and later releases, this workflow supports file geodatabases to participate as publication geodatabases.

²⁵ More information on geodatabase replication can be found in the white paper [An Overview of Distributing Data with Geodatabases](#).

²⁶ More information on geodata services is available in the ArcGIS Server Help topic *Geodata services*.

ArcGIS Server map services would reference and access data from the publication geodatabase and would not have a direct relationship to the production geodatabase. There are several advantages to this approach for modeling the database tier of the ArcGIS Server system:

- **Impact on existing users:** For the use case, Web usage is substantial (e.g., supporting up to 250,000 requests per hour) and to some extent unpredictable by nature. By creating a publication geodatabase and using it as the map service data source, Web requests will not impact users on the production geodatabase.
- **Management and tuning of the geodatabase:** Map services access geodatabase data in a read-only mode. Since geodatabase management and tuning strategies are different for read-only versus read/write usage, the publication geodatabase can be custom-tuned to better support read-only Web use—for example, only have a Default version in the publication geodatabase—while the production geodatabase is maintained with a strategy that best supports its existing users.
- **Enhanced security:** If the map services are accessed over the open Web, it is common practice that the publication geodatabase would reside in a demilitarized zone (DMZ). Conversely, the production geodatabase would reside behind a firewall for enhanced security. In addition, the publication geodatabase would only contain data that is needed to support the map services. Data unrelated to the map services, but still part of the GIS business needs of the organization, would only reside in the production geodatabase.

One minor drawback to the one-way replication workflow implementation is that the GIS data replicated to the publication geodatabase can become outdated over time. The replica is conceptually like a snapshot of the production geodatabase at the time it is created. However, regularly scheduled synchronizations between the two geodatabases can keep the publication geodatabase up-to-date with the production geodatabase. The synchronization process can also be modeled in ModelBuilder™, exported to a Python® script, then automated as a scheduled task executed by the server operating system.

Tip #6 Consider using a file geodatabase as the data source for map services.

As discussed in the previous tip, ESRI recommends using an enterprise ArcSDE geodatabase as the database tier for supporting medium to large ArcGIS Server system implementations. However, using a file geodatabase as the data source for map services can in some instances be a good alternative strategy. For example, in the one-way replication production–publication workflow described previously, should the publication geodatabase be an enterprise ArcSDE geodatabase or a file geodatabase?

It is generally recommended that an enterprise ArcSDE geodatabase be used as the publication geodatabase because of its many benefits: high-availability support, backup and recovery, concurrency, scalability, and a tendency to provide superior throughput. However, this recommendation is provided with the assumption that an organization will have a dedicated database administrator (DBA) to optimize, tune, and maintain the enterprise ArcSDE geodatabase. If an organization does not have a DBA on staff, then using a file geodatabase is a good alternative. File geodatabases generally provide great performance with minimal extra configuration or tuning needed. Depending on the characteristics of the GIS data, in some cases, extra optimization and tuning of an

enterprise ArcSDE geodatabase might be required to surpass the performance of a file geodatabase.

A second consideration is general ease of maintenance. A file geodatabase is much easier to maintain over an enterprise ArcSDE geodatabase. Given that most end users of ArcGIS Server map services generally do not use versioning and other features of an enterprise ArcSDE geodatabase, some ArcGIS Server implementation architects prefer the simplicity of a file geodatabase.

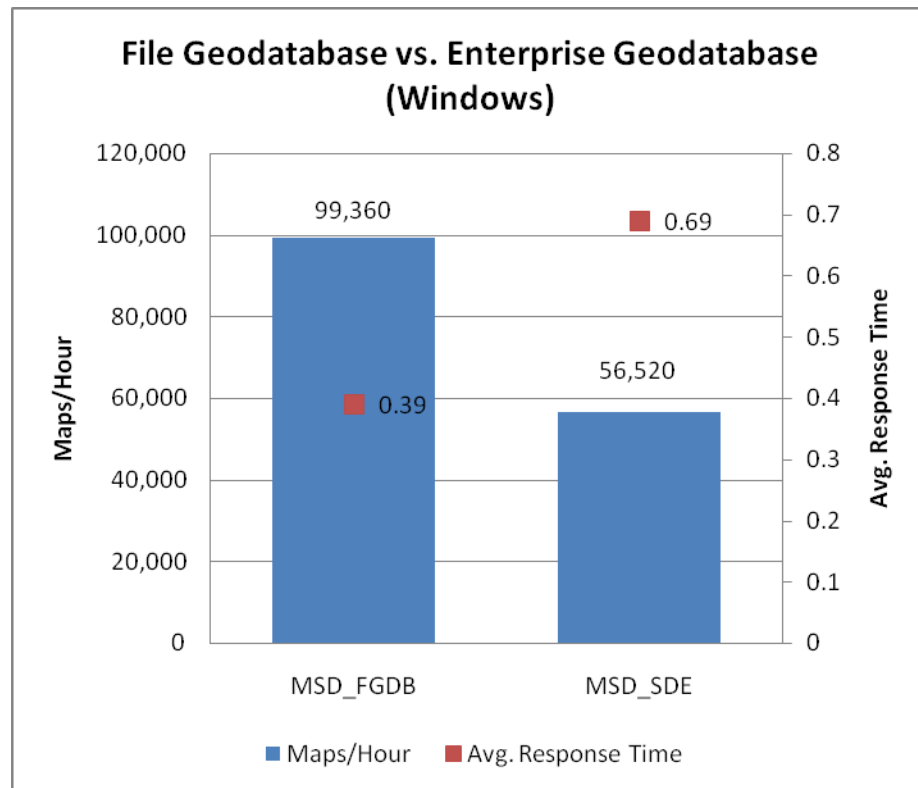
One last factor to consider is licensing costs. Using a file geodatabase does not require additional licensing, whereas an enterprise ArcSDE geodatabase might, depending on a variety of factors. Examples include hardware specifications, deployment configurations, and types of connections to the geodatabase. Additionally, an enterprise ArcSDE geodatabase implementation also requires third-party DBMS software.

A performance comparison between two map services configured with different data sources (enterprise ArcSDE geodatabase²⁷ versus file geodatabase²⁸) was completed. The test measured the number of maps per hour generated (i.e., maximum throughput) using the CLC GIS dataset as the source data. For more details on the test hardware configuration and the characteristics of the requests made, see appendix A. The results of the comparison test are shown in figure 7.

²⁷ The enterprise ArcSDE geodatabase was implemented on Oracle 11g and accessed via direct connections. The CLC GIS dataset was stored as ST_Geometry, and the geodatabase was optimized and tuned following best practices described in the ArcGIS Desktop Help topic *An overview of tuning an ArcSDE geodatabase*.

²⁸ Data in the file geodatabase was uncompressed.

Figure 7
File Geodatabase vs. Enterprise ArcSDE Geodatabase
Performance Comparison Test Results



The file geodatabase-based optimized map service (MSD_FGDB) had a maximum throughput of 99,360 maps/per hour with an average response time of 0.39 seconds. The enterprise ArcSDE geodatabase-based optimized map service (MSD_SDE) had a maximum throughput of 56,520 maps/per hour with an average response time of 0.69 seconds. The results of the comparison test indicate that the optimized map service that referenced a local file geodatabase (MSD_FGDB) outperformed the same optimized map service that referenced an enterprise ArcSDE geodatabase (MSD_SDE).

It is important to emphasize that the comparison test results shown in figure 7 are only representative of the CLC GIS dataset in this use case and for a moderately tuned enterprise ArcSDE geodatabase. Additional optimization and tuning of both the geodatabase and network may make the enterprise ArcSDE geodatabase map throughput match and/or exceed the file geodatabase throughput. The map throughput results for an enterprise ArcSDE geodatabase are influenced by the optimization and tuning practices applied to it. These practices also vary depending on the DBMS platform.²⁹ The intent of figure 7 is to provide a specific example that demonstrates the importance of geodatabase optimization and tuning when deploying an enterprise ArcSDE geodatabase.

²⁹ A comprehensive discussion on this topic is beyond the scope of this document. More information can be found in DBMS-specific help documentation.

Conversely, file geodatabases are less demanding from a configuration perspective, although there are some factors to consider. When using a file geodatabase as the map service data source, the map services' SOC processes should be accessing local copies of the file geodatabase. For example, for a distributed ArcGIS Server deployment with three SOCs on different servers, each SOC should be accessing a local copy of the file geodatabase on its server. The SOCs should not be accessing the same file geodatabase over the network. This configuration minimizes network communication traffic among the different ArcGIS Server components and reduces I/O contention when accessing the file geodatabase(s).

Limited testing did not reveal disk I/O contention on a shared file geodatabase that was accessed under heavy load of 23 concurrent map instances (i.e., disks configured on RAID-1³⁰). However, disk I/O contention is expected at some point if the same file geodatabase was shared across a large-enough number of map instances across all the SOCs. Factors that influence potential disk I/O contention for a shared file geodatabase include the number of layers in the map service, the nature of the data in the file geodatabase, and the file storage device.

In this use case, the file geodatabase yielded excellent results in terms of performance and did not require additional configuration. However, for other scenarios, the file geodatabase might not be the most appropriate map service data source.

- File geodatabases work well when accessed in read-only mode by concurrent processes but remain locked³¹ for editing while under heavy use. Because of this, in scenarios where the file geodatabase is a publication geodatabase (in one-way replication workflows), replica synchronization would have to occur during periods of inactivity in the map service or by releasing the file geodatabase from being used by the map service. This can be done by stopping the service or, for multiple SOC deployments, by temporarily disconnecting the SOC machine from the SOM, then reconnecting them after the file geodatabase has been updated.
- Some functionality of ArcSDE geodatabases, such as versioning, geodatabase replication, and historical archiving, are not available in file geodatabases.
- Standard DBMS functionality available in ArcSDE geodatabases, such as logging, backup and recovery, and failover configuration, are not available in file geodatabases.

In summary, file geodatabases can be a great alternative map service data source for some scenarios, if the published data is relatively static and the capabilities of ArcSDE geodatabases (and the DBMS platforms they are implemented on) are not needed.

³⁰ More information on RAID technology can be found at <http://en.wikipedia.org/wiki/RAID>.

³¹ More information on file geodatabase locking is available in the ArcGIS Server Help topic *File geodatabases and locking*.

ArcGIS Server Enterprise Deployment Examples

In this final section, several different scalability deployments of ArcGIS Server Enterprise are explored. The objective is to increase the capacity of the CLC GIS dataset optimized map service (henceforth referred to as the CORINE map service) to address the business requirements of the use case. There are many different scalability strategies for ArcGIS Server Enterprise to support dynamic map services, but a comprehensive discussion of all possible configurations is beyond the scope of this document.

For this use case, three common scalability configurations are examined and discussed. Table 2 lists the different strategies and the number of each ArcGIS Server component. To make performance testing easier, local file geodatabases (i.e., local to the SOC processes) were used instead of a centralized enterprise ArcSDE geodatabase. For additional details on the test hardware configurations, see appendix A.

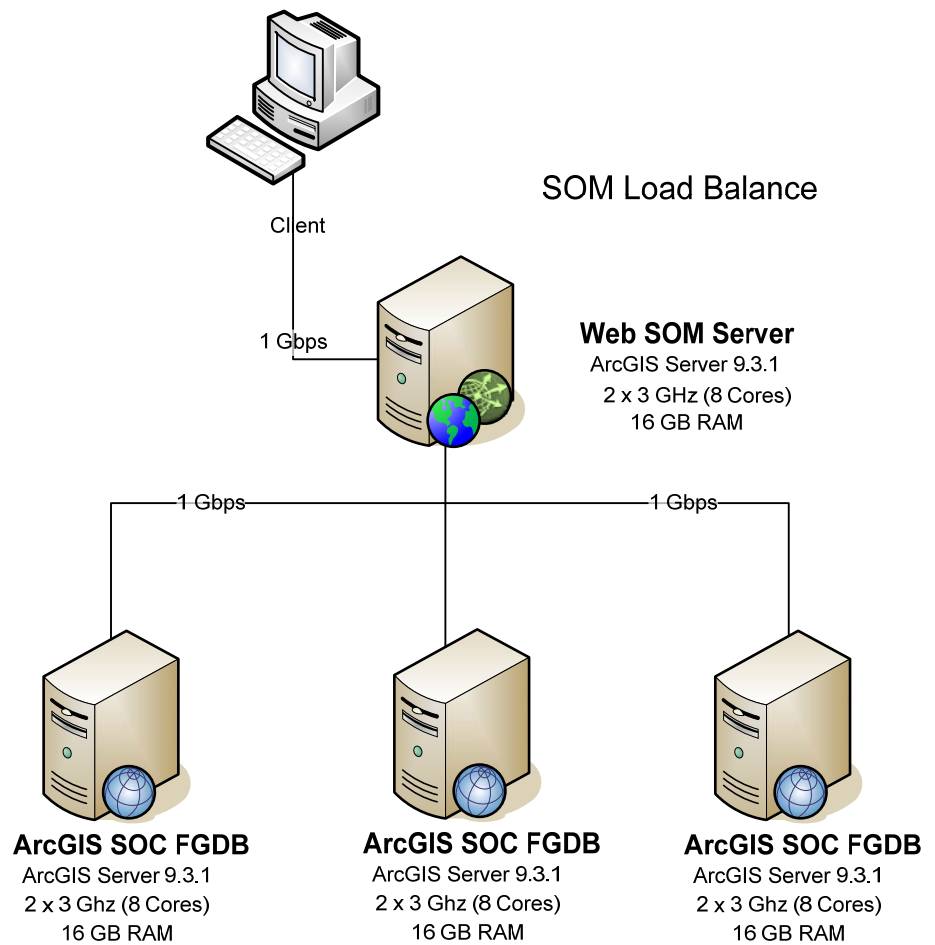
Table 2
ArcGIS Server Enterprise Deployment Configurations

ArcGIS Server Deployment Type	Web Server and Server Object Manager Count	Server Object Container Count	Total Number of Server Machines
Single SOM load balance	1	3	4
High-availability SOM load balance	2	3	5
Web/SOM/SOC network load balancing	3	3	3

Single SOM Load Balance

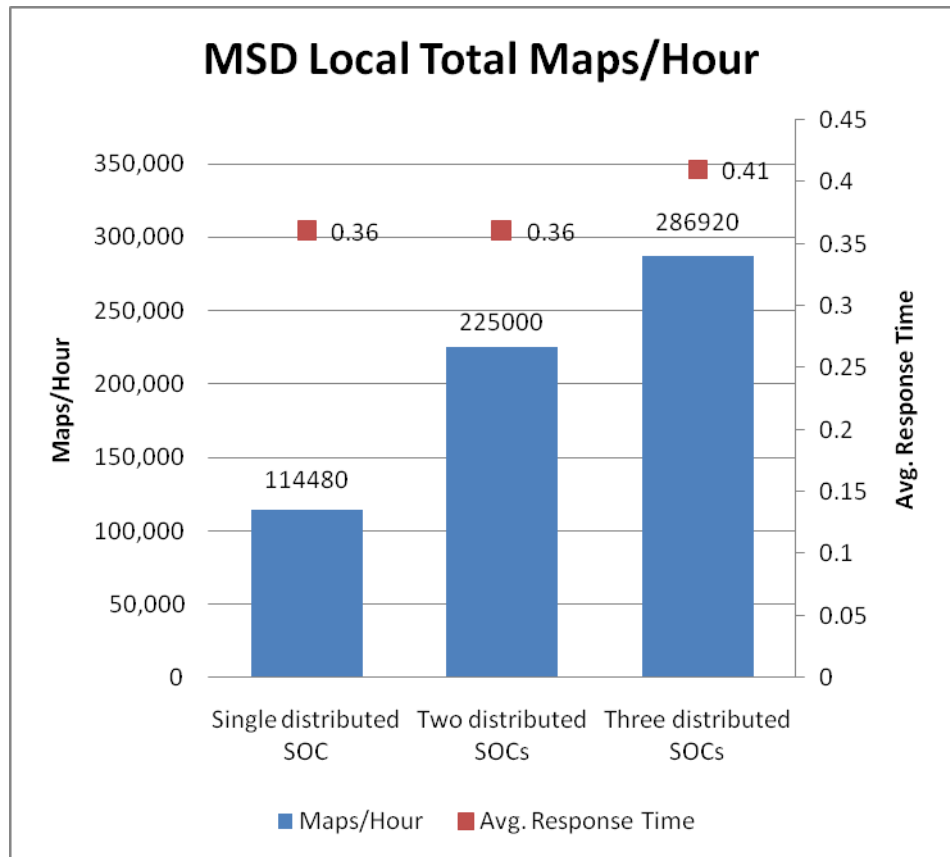
This strategy maximizes the use of the hardware to support the SOC tier. SOC processes are scaled out across several different servers. In this example, one machine contained the Web server and SOM, while the SOCs had three fully dedicated servers (see figure 8).

Figure 8
Single SOM Load Balance Strategy—Architectural Diagram



A performance test was applied to this deployment to measure the number of maps per hour generated (i.e., maximum throughput) and the average response time. The CORINE map service was used as the test data. First, the configuration was measured with only one SOC available. Next, a second SOC was added, and the test was reapplied. The test was repeated a third time with a third SOC added to the deployment. Performance testing results are shown in figure 9.

Figure 9
Single SOM Load Balance Performance Test Results



In general, the maximum throughput and average response time increased linearly as more SOC were added to the deployment. More precisely, the system scaled linearly between one (8 cores total at the SOC level) and two SOC (16 cores), but a degradation in performance was observed when a third SOC (24 cores) was added.

A bottleneck was observed at the Web tier that prevents the SOC from working at full capacity. It was determined that the REST interface endpoint could not handle more than approximately 70 map requests per second before some performance degradation occurred. The capacity of a REST Web service to handle simultaneous map requests depends on two factors: the resources that exist on the Web server and the size of the map images being requested.

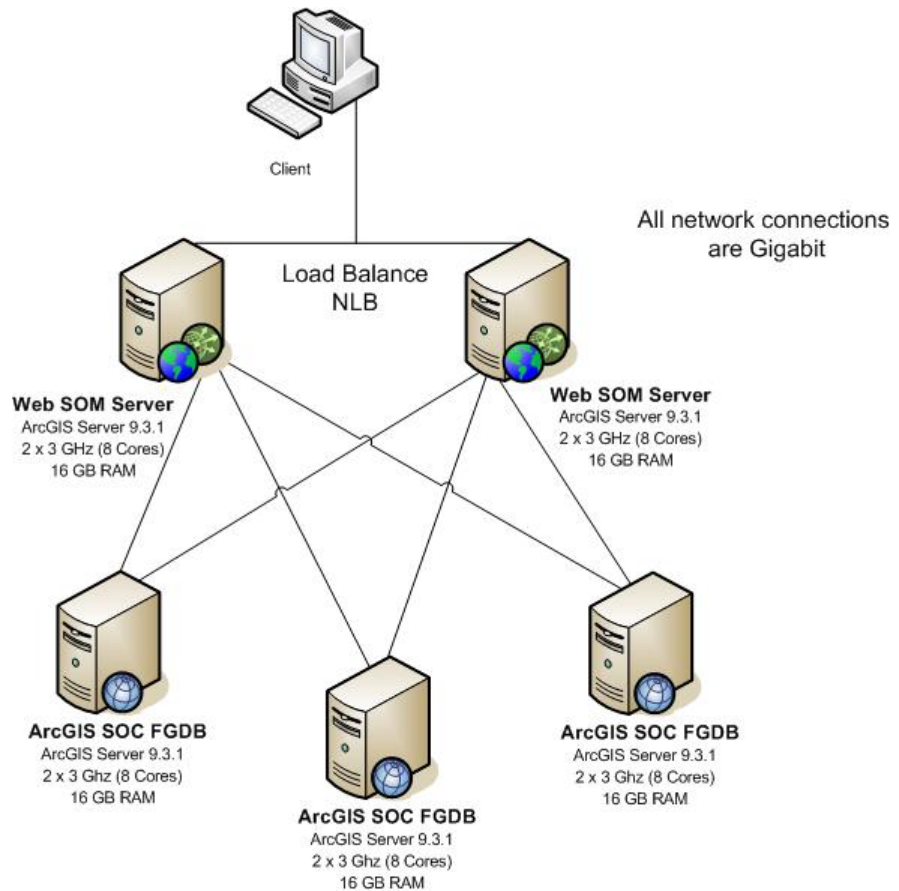
Essentially, there is a tipping point beyond which adding more capacity at the SOC tier will not yield more overall capacity. The system needs to be balanced by adding additional Web servers. For the record, similar load tests on a SOAP interface endpoint of the same map service scaled linearly up to at least nine SOC (78 cores). So this issue only applies to the REST Web services on Windows. From a high-availability perspective, this deployment strategy example has a single point of failure at the Web/SOM component.

J-9808

High-Availability SOM Load Balance

To eliminate the single point of failure in the previous architecture, a second deployment strategy was implemented and tested. This deployment configuration is a variation of the first strategy. There is no single point of failure at the Web tier because a second Web/SOM component and a network load balancer were added (see figure 10).

Figure 10
High-Availability SOM Load Balance Strategy—Architectural Diagram



A network load balancer was used to spread the load across the two separate Web/SOM servers following an active/active configuration. This means that requests from Web clients were distributed evenly between the two Web/SOM servers. In this case, a load balancer (e.g., Network Load Balancer [NLB] for Windows platforms³²) was used to front multiple instances of ArcGIS Server (e.g., Web/SOM components). If one of the machines were to become unavailable for any reason, the load balancer would still be able to utilize the remaining ArcGIS Server instances.

Sensitivity testing of the SOC servers revealed that the number of map instances that yields the highest map throughput at the SOC tier is 2.5 instances per CPU core. Therefore, for the eight-core SOC servers used in these tests, $2.5 \times 8 = 20$ map instances

³²The NLB is implemented as a service and included with the Windows 2003 OS. It is not a physical hardware NLB component. More information on NLB service technology can be found at http://en.wikipedia.org/wiki/Network_Load_Balancing_Services.

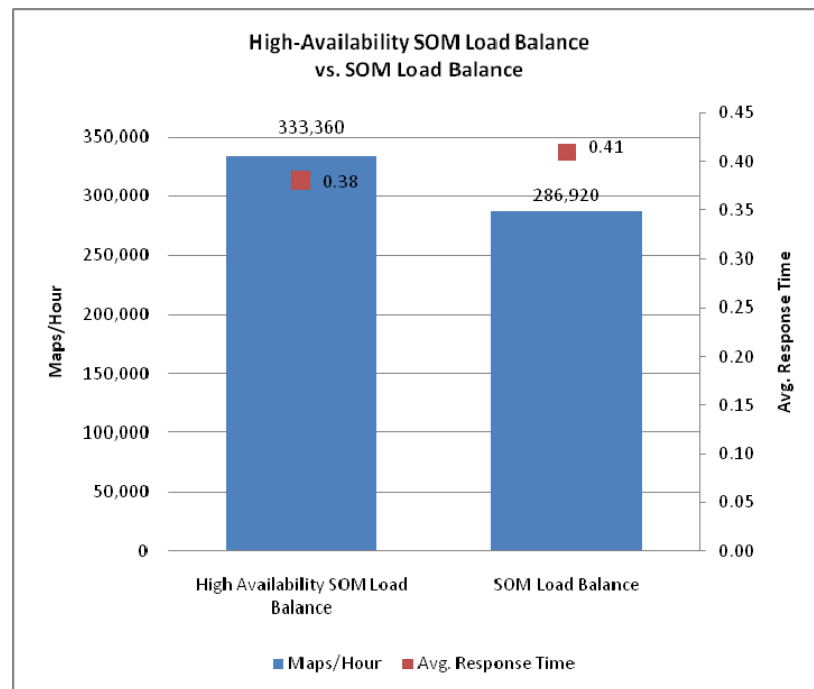
per SOC machine,³³ totaling 60 for the entire deployment. Since the SOC machines were shared between the two Web/SOM servers in this deployment strategy, the CORINE map service in the SOMs was configured to have a maximum number of map instances set to 30. This enabled each SOC server to hold a total of 20 map instances—10 from one SOM and the remaining 10 from the other SOM. This means a total of 60 map instances across the SOC tier.

If one of the Web/SOM servers were to become unavailable, a decline in the maximum throughput is expected, sometimes reduced by half. However, only a 25 percent decline was observed in testing this circumstance. This is due to the fact that while the number of map instances is reduced by half (given that half the map instances in the SOC tier become orphaned and are eventually removed), the overall CPU capacity of each SOC machine remains constant, and there are more than enough map instances active to maximize these resources.

A variation of this configuration sets the number of minimum instances to 30 and the maximum number of instances to 60. In this configuration, in the event of a Web/SOM failure, the remaining SOM is permitted to increase the number of map instances in the SOC tier to accommodate increasing load, up to the number of map instances (60) that optimizes the SOC tier.

A performance comparison between the SOM load balance configuration and the high-availability SOM load balance configuration is shown in figure 11.

Figure 11
Performance Comparison between Deployment Strategies 1 and 2



³³ This total number of map instances per SOC machine is valid for all the configurations tested in this document.

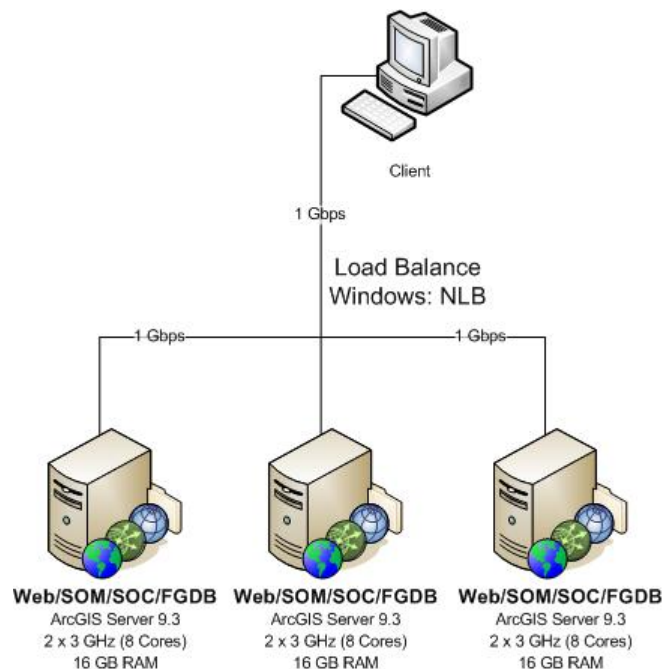
The maximum throughput observed for the high-availability SOM load balance deployment is 333,360 maps/per hour with an average response time of 0.38 seconds. This result is an approximate increase of 14 percent compared with the single SOM load balance configuration. By adding a second Web/SOM server to the system, the Web tier single point of failure was removed, and the system was able to scale linearly to no less than 24 cores at the SOC tier.

Note: In this ArcGIS Server deployment, the administrator is responsible for ensuring that the CORINE map service is duplicated across all the SOM servers. In other words, the CORINE map service needs to be configured the same in all the SOM instances.

Web/SOM/SOC Network Load Balancing

In the final deployment strategy example, all the ArcGIS Server tiers (i.e., Web server, SOM, and SOC) were scaled out on every machine in the system with network load balancing across all the system nodes (see figure 12).

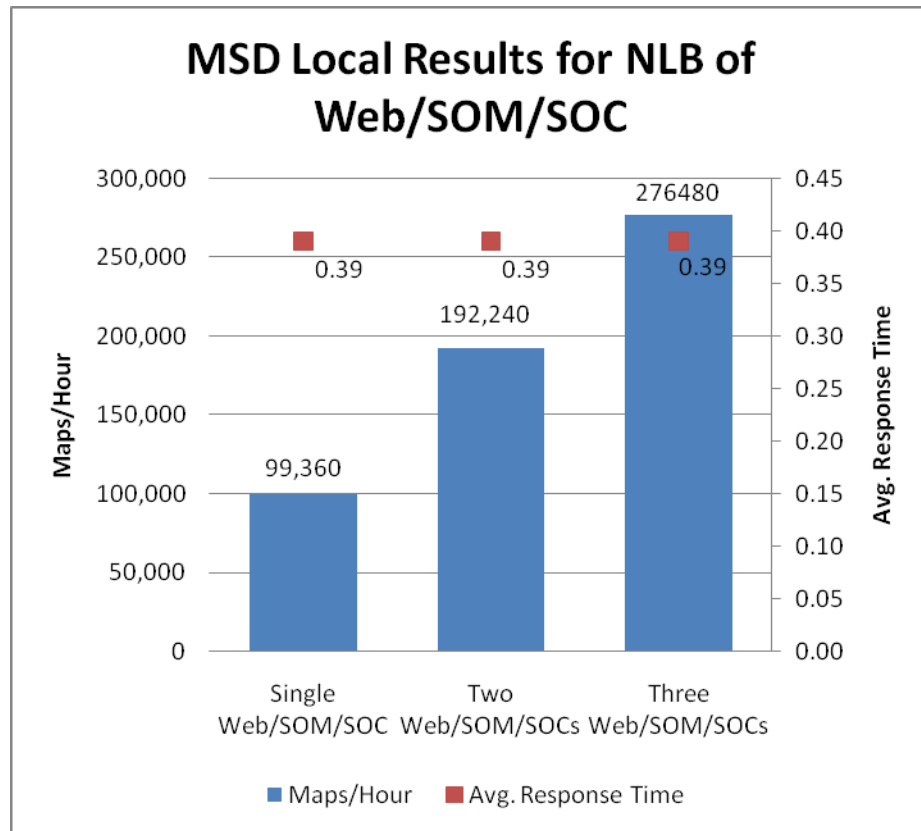
Figure 12
Web/SOM/SOC Network Load Balancing Strategy—Architectural Diagram



In this deployment, there is the extra overhead of maintaining a copy of the CORINE map service for each SOM machine participating in the deployment. The CORINE map service must be configured as many times as there are nodes in the system. In this case, it was configured three times.

A performance test was applied to this deployment to measure the number of maps per hour generated (i.e., maximum throughput) and the average response time. First, the configuration was measured with only one machine, then a second machine, and finally a third machine. Performance testing results are shown in figure 13.

Figure 13
Performance Comparison between All Three Deployment Strategies



The Web/SOM/SOC network load balancing deployment achieved linear scalability. As more machines were added, the throughput increased, and the average response time remained the same at 0.39 seconds.

When compared with the two previous deployments, there is approximately a 10 percent penalty for sharing the Web server and SOM processes on the same machine as the SOC process. This value is expected to be lower if a SOAP interface endpoint of the Web service is used because the SOAP Web service handler is less CPU intensive than the REST interface endpoint tested in these deployment examples.

***Scalability
Deployment
Strategies
Assessment***

Table 3 summarizes the performance test results of the three different ArcGIS Server Standard Enterprise scalability deployments that were discussed in this document.

Table 3
Summary of Deployment Configuration Performance Testing

ArcGIS Server Deployment Type	Total Number of Server Machines	Total Number of Cores Overall	Maps per Hour (maximum throughput)	Average Response Time (seconds)	Maps Generated per Core, per Hour ³⁴
Single SOM load balance	4	32	286,920	0.41	8,966
High-availability SOM load balance	5	40	333,360	0.38	8,334
Web/SOM/SOC network load balancing	3	24	276,480	0.39	11,520

The three configurations tested provided some valuable information for the deployment of the CORINE map service. The single SOM load balance deployment offered a strategy with the easiest configuration to implement. However, this configuration needs additional Web servers to overcome bottlenecks in the REST interface endpoint. It also does not provide redundancy at the Web/SOM tier.

The high-availability SOM load balance deployment adds an extra Web/SOM node to the system through an active/active configuration of a network load balancer. This provides redundancy at the Web/SOM tier, which ensures the CORINE map service will continue to operate even if one of the Web/SOM machines was to become unavailable. This deployment yields the maximum overall map throughput of all three strategies tested. From an administrative perspective, this deployment requires the configuration of the CORINE map service in the two SOMs, adding some extra overhead.

Finally, the Web/SOM/SOC network load balancing deployment provided a redundant configuration at all levels in the system. In terms of map throughput, this strategy generated the most maps per core, per hour, maximizing the use of the hardware employed in the test. If an equivalent number of machines had been used for this deployment as were for the high-availability SOM load balance test (five server machines), the expected map throughput would be close to half a million maps per core, per hour. This would greatly surpass the 333,360 maps per hour (i.e., maximum throughput) for the second deployment strategy tested. From an administrative perspective, this deployment requires the configuration of the CORINE map service in all the SOMs, resulting in additional maintenance overhead.

There are numerous factors to consider when selecting the appropriate strategy for deploying high-capacity map services, including software and hardware costs, how the ArcGIS Server system will be integrated into an existing enterprise IT system, system performance, and business continuity (see appendix B).

³⁴ This value was determined by dividing the maximum throughput by the number of cores overall.

Conclusion To conclude, this document discussed high-capacity map services in the context of a use case with the CORINE land-cover data. It presented three different approaches for publishing map data with ArcGIS Server: (1) cached map service, (2) query map service to return features, and (3) dynamic map service. Their advantages, disadvantages, and suitability for the CORINE data were examined.


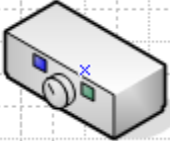

Several best practices for ArcGIS Server map services were highlighted:

- Use the Map Service Publishing toolbar to optimize map documents that will be published in ArcGIS Server.
- When possible, use optimized map services instead of standard map services.
- Adjust the information displayed in the map document according to the map scale.
- Review the map service log to further investigate response times on the map service.
- Implement a production–publication geodatabase replication workflow to support the ArcGIS Server system.
- Consider using a file geodatabase as the data source for map services.

Finally, three different scalability ArcGIS Server Standard Enterprise deployment configurations were explored. The advantages and disadvantages for each deployment strategy were discussed including performance, scalability, maintenance, and high-availability considerations.

Appendix A: Hardware Configuration of the Test Servers

General Information on the Physical Hardware

	Hardware	Software
 Client	CPU: 8 x 3.0 GHz Memory: 16 GB RAM Disk: 135 GB Disk	Microsoft Windows Server 2003 R2 64 bit
 Network	1 gigabyte	Not applicable
 Server	Several machines were used. They are identified in the network with a single virtual IP for the entire cluster. Each machine had the following: CPU: 8 x 3.0 GHz Memory: 16 GB RAM Disk: 135 GB Disk (Disks are mirrored: RAID-1)	Microsoft Windows Server 2003 R2 64 bit ArcGIS Server Standard Enterprise Oracle Release 11g

Performance Testing

An HTTP debugger program was used to record network traffic between a client and the server, while user activity (e.g., simple map display requests) was performed on the CORINE map service. The captured traffic was subsequently modified into a custom test, then replayed as a performance test in Microsoft® Visual Studio® 2008 Test Team.

In the throughput tests, requests for images of five countries—England, Finland, Germany, Italy, and Spain—were made.

The tests performed repeated random pan and zoom queries on the CORINE map service at two different map scales: 1:100,000 and 1:30,000. Each request required the generation of a new map image with a different local map projection. The type of image requested was PNG, and the image size was set at 1024 x 768. The maximum throughput (i.e., maps/hour) was measured along with the average response time (think time was not considered). All test configurations tested used the same load test except that more load was applied for the load-balanced configurations.

More detailed information on the stress test load procedure can be found at the Enterprise GIS Resource Center at [ArcGIS Server Performance and Scalability—Testing Methodologies](#).

Example Request Code

Export Map REST Request

```
http://<WebServerName>/ArcGIS/rest/services/msd_Corine6_Local/MapServer/export?transparent=true&format=png8&dpi=96&size=1024,768&f=image&bbox={%22xmin%22:-251271.837890625,%22ymin%22:6751904.83203125,%22xmax%22:-223331.78125,%22ymax%22:6773415.5,%22spatialReference%22:{%22wkid%22:102113}}&bboxSR=102113&imageSR=32630
```

ArcIMS Request

```
http://<WebServerName>/servlet/com.esri.esrimap.Esrimap?ServiceName=corine6&ClientVersion=4.0&Form=True&Encode=False
```

Request Body

```
ArcXMLRequest=%3c%3fxml+version%3d%221.0%22+encoding%3d%22UTF-8%22+%3f%3e%3cARCXML+version%3d%221.1%22%3e%0d%0a%3cREQUEST%3e%0d%0a%3cGET_IMAGE%3e%0d%0a%3cPROPERTIES%3e%0d%0a%3cENVELOPE+minx%3d%22-5436069.342653396%22+miny%3d%223203377.2038%22+maxx%3d%227264799.7023533955%22+maxy%3d%2211101806.322%22+%2f%3e%0d%0a%3cIMAGESIZE+height%3d%22648%22+width%3d%221042%22+%2f%3e%0d%0a%3cLAYERLIST+%3e%0d%0a%3cLAYERDEF+id%3d%226%22+visible%3d%22true%22+%2f%3e%0d%0a%3cLAYERDEF+id%3d%225%22+visible%3d%22true%22+%2f%3e%0d%0a%3cLAYERDEF+id%3d%223%22+visible%3d%22true%22+%2f%3e%0d%0a%3cLAYERDEF+id%3d%222%22+visible%3d%22true%22+%2f%3e%0d%0a%3cLAYERDEF+id%3d%220%22+visible%3d%22true%22+%2f%3e%0d%0a%3c%2fLAYERLIST%3e%0d%0a%3c%2fPROPERTIES%3e%0d%0a%3cLAYER+type%3d%22acetate%22+name%3d%22theCopyright%22%3e%0d%0a%3cOBJECT+units%3d%22pixel%22%3e%0d%0a%3cTEXT+coords%3d%223%22+label%3d%22Map+created+with+ArcIMS+-+Copyright+(C)+1992-2009+ESRI+Inc.%22%3e%0d%0a%3cTEXTMARKERSYMBOL+fontstyle%3d%22Regular%22+fontsize%3d%228%22+fontcolor%3d%220%2c0%22+antialiasing%3d%22True%22+blockout%3d%225%2c5%2c5%22+overlap%3d%22false%22+%2f%3e%0d%0a%3c%2fTEXT%3e%0d%0a%3c%2fOBJECT%3e%0d%0a%3c%2fLAYER%3e%0d%0a%3cLAYER+type%3d%22acetate%22+name%3d%22theNorthArrow%22%3e%0d%0a%3cOBJECT+units%3d%22pixel%22%3e%0d%0a%3cNORTHARROW+type%3d%224%22+size%3d%2215%22+coords%3d%2220+35%22+shadow%3d%2232%2c32%22+angle%3d%220%22+antialiasing%3d%22True%22+overlap%3d%22False%22+%2f%3e%0d%0a%3c%2fOBJECT%3e%0d%0a%3c%2fLAYER%3e%0d%0a%3cLAYER+type%3d%22acetate%22+name%3d%22theScaleBar%22%3e%0d%0a%3cOBJECT+units%3d%22pixel%22%3e%0d%0a%3cSCALEBAR+screenlength%3d%22208%22+coords%3d%22625+3%22+fontcolor%3d%220%2c0%2c0%22+fontstyle%3d%22Regular%22+barcolor%3d%22128%2c128%2c128%22+mapunits%3d%22meters%22+scaleunits%3d%22meters%22+ant
```


J-9808

```
ialiasing%3d%22True%22+fontsize%3d%229%22+barwidth%3d%225%22+overlap%3d%22False%22++%2f%3e%0d%0a%3c%2fOBJECT%3e%0d%0a%3c%2fLAYER%3e%0d%0a%3c%2fGET_IMAGE%3e%0d%0a%3c%2fREQUEST%3e%0d%0a%3c%2fARCXML%3e&JavaScriptFunction=parent.MapFrame.processXML&BgColor=%23000000&FormCharset=ISO-8859-1&RedirectURL=&HeaderFile=&FooterFile=
```

Appendix B: ArcGIS Server and Business Continuity

Solution architecture surrounding ArcGIS Server should promote best practices for deployment that align with common information technology strategies. Business continuity planning is one strategy that involves architectural support to a logistical plan. For example, at a minimum, an ArcGIS software-based solution should be delivered in a highly available and redundant environment when supporting workflows that are business critical, affecting public safety and critical infrastructure, and/or participating in revenue stream-related transactions. Details regarding a solution supporting business continuity were not in the scope of this document. For a more in-depth discussion and an ESRI approach to supporting business continuity, please see the paper titled "ArcGIS Server and Business Continuity" located in the ESRI Enterprise GIS Resource Center accessible at resources.esri.com.



About ESRI

Since 1969, ESRI has been helping organizations map and model our world. ESRI's GIS software tools and methodologies enable these organizations to effectively analyze and manage their geographic information and make better decisions. They are supported by our experienced and knowledgeable staff and extensive network of business partners and international distributors.

A full-service GIS company, ESRI supports the implementation of GIS technology on desktops, servers, online services, and mobile devices. These GIS solutions are flexible, customizable, and easy to use.

Our Focus

ESRI software is used by hundreds of thousands of organizations that apply GIS to solve problems and make our world a better place to live. We pay close attention to our users to ensure they have the best tools possible to accomplish their missions. A comprehensive suite of training options offered worldwide helps our users fully leverage their GIS applications.

ESRI is a socially conscious business, actively supporting organizations involved in education, conservation, sustainable development, and humanitarian affairs.

Contact ESRI

1-800-GIS-XPRT (1-800-447-9778)

Phone: 909-793-2853

Fax: 909-793-5953

info@esri.com

www.esri.com

Offices worldwide

www.esri.com/locations



ESRI
380 New York Street
Redlands, California
92373-8100 USA