ArcIMS 9

Creating and Using Metadata Services

GIS by ESRI

Copyright © 2002, 2004 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 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 Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

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

U.S. GOVERNMENT RESTRICTED/LIMITED RIGHTS

Any software, documentation, and/or data delivered hereunder is subject to the terms of the License Agreement. In no event shall the U.S. Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR §52.227-14 Alternates I, II, and III (JUN 1987); FAR §52.227-19 (JUN 1987) and/or FAR §12.211/12.212 (Commercial Technical Data/Computer Software); and DFARS §252.227-7015 (NOV 1995) (Technical Data) and/or DFARS §227.7202 (Computer Software), as applicable. Contractor/Manufacturer is ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

ESRI, ArcCatalog, ArcExplorer, ArcObjects, ArcGIS, ArcIMS, ArcMap, GIS by ESRI, ArcReader, Spatial Database Engine, SDE, ArcSDE, Geography Network, the ArcGIS logo, the Geography Network logo, www.esri.com, and www.geographynetwork.com are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Microsoft and the Windows logo are registered trademarks and the Microsoft Internet Explorer logo is a trademark of Microsoft Corporation.

Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners.

Contents

1 Introduction 1

What is a Metadata Service? 2
Putting it all together 3
Tips on learning about Metadata Services 4

2 Finding data in Metadata Explorer 7

Starting Metadata Explorer 8
Finding data using Metadata Explorer 9
Browsing metadata 12
Searching for data 13
Viewing the results of your search 14
Viewing metadata 15
Viewing a map of the data 16
Printing search results, metadata, and maps 17

3 Creating Metadata Services 19

Setting up your Metadata Service 20
Creating a Metadata Service 26
Creating usernames and passwords 28
Troubleshooting publishing problems 31
Metadata for ArcIMS services 32
Defining a service's index 33
Manually indexing documents 37
Adding an administrative table to a service 39

4 Publishing metadata in ArcCatalog 41

Requirements for publishing metadata 42
Preparing metadata for publication 46
Publishing to a Metadata Service 53
Managing published documents 54
Finding data 58

5 Connecting Z39.50 clients 61

Incorporating Z39.50 into your metadata system 62
What is Z39.50? 63
Getting started 64
Changing the service's port 66
Setting the default Metadata Service 67
Setting log options 68
Protocols and restricting access to your metadata 71
Setting a communication protocol 74
Limiting access to your metadata 76
Customizing the connector 78
Controlling the format of responses 80
Available Z39.50 resources 81

A Configuring Metadata Explorer 83

Which ArcIMS services does Metadata Explorer need? 84
Starting the services 84
Configuring Metadata Explorer 85
Do you need a login? 86
Loading the gazetteer data 87
Customizing the gazetteer 89
An unattended Metadata Explorer session will expire. 94
Integrating Metadata Explorer into ArcExplorer 94

B ArcSDE considerations 95

Specifying ArcSDE connection information 96
Metadata Service database schema 96
Removing Metadata Service objects from the database 101
Configuring an Oracle database 102
Configuring a DB2 database 103
Configuring a SQL Server database 104
Configuring ArcSDE 105

Increasing the ArcIMS time-out for ArcSDE 110
Getting the best performance from the database 110

C Harvesting documents 113

Collecting documents to create a community GIS portal 114
Exporting and importing documents between Metadata Services 116
Adding the OAI-PMH Connector to a Metadata Service 122
Harvesting documents using the OAI-PMH Harvester 125

D Publishing non-English metadata 131

What languages will a Metadata Service support? 132
Configuring your Metadata Service 132
Indexing and harvesting considerations 132

E What's new with ArcIMS 9? 133

Changes for Metadata Services with ArcIMS 9 134
Migrating a Metadata Service to ArcIMS 9 135

Index 139

Introduction

1

IN THIS CHAPTER

- **What is a Metadata Service?**
- **Putting it all together**
- **Tips on learning about Metadata Services**

Whether you realize it or not, you've seen metadata before. A wide variety of everyday items has metadata, with varying levels of detail. The nutritional information on the label of a can of soup is metadata. A department store catalog is metadata. The price sticker on a new car is metadata. Anything that describes something else can be thought of as metadata.

Metadata is useful because it tells you about something without you actually having possession of that something. For example, you might browse through a store catalog before deciding to make a purchase. The metadata about the items in the catalog gives you valuable information, such as price, size, and material, to help you determine whether the item is right for you.

Metadata for geographic data is often described as data about data. Metadata for geographic data typically describes the content, quality, type, collection date, and spatial location of the actual data. Looking at metadata can help you determine whether a particular dataset is appropriate for your use. In fact, it's often easier to judge the suitability of the data from its metadata than by examining the actual data itself. This is because metadata is generally much smaller and easier to share than the actual data. And it's the ability to share metadata—through an ArcIMS® Metadata Service—that makes it so valuable. Sharing metadata lets you see what geographic information system (GIS) resources, such as data, others have and also lets others see what you have.

What is a Metadata Service?

A *Metadata Service* provides a forum for sharing metadata. If you're looking for data, you might search or browse a Metadata Service to find what you need. Similarly, if you have data that you want to share with others, you can do so by publishing it to a Metadata Service where others can see it.

Finding data

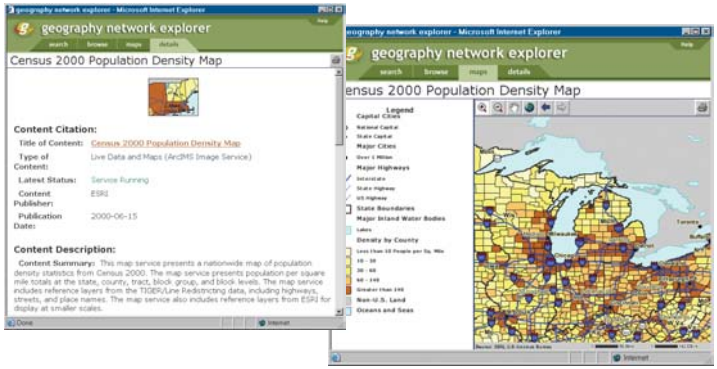
Suppose you're creating a map for a presentation and you need some basemap data to display behind the data you've collected about your study area. Many organizations collect geographic information and build datasets. Some organizations freely distribute their data while others may charge a small fee. The question is, how do you find that data?

The first place you might look is ESRI's Geography Network℠ (*www.geographynetwork.com*). The Geography Network is a global network of geographic information users and providers. Through the Geography Network, you can access many types of geographic content, including live maps, downloadable data, and more advanced services. One component of the Geography Network, the Geography Network Explorer, is a Metadata Service. With the Geography Network Explorer, you can search for data by geographic location; theme, such as environmental, business, or health data; scale; date collected; and so on.

Think of a Metadata Service as a geographic search engine that you use when you want to find data and other GIS resources. A Metadata Service works over the Internet, allowing anyone who has access to the Web to use it. The Geography Network is just one place you can go to look for data. Just as you can use different Internet search engines such as Yahoo!®, you can also search different Metadata Services hosted by other organizations. If you know the Web address of those Metadata Services, you can easily search their contents for geographic data.

When you search for data on a Metadata Service, your search will yield a list of resources that match your search criteria. You can view the complete metadata for each one to see if it's what you're looking for and can sometimes view an interactive map. Once you find a suitable dataset, you can download it or, in some instances, use it right from the Web.

The Geography Network Explorer helps you find the data you need.



By examining metadata, you can see if the data is right for you. Sometimes you can view an interactive map of the data.

Sharing data

If your organization is like most, you've invested a great deal of time and money collecting the data from which you've built your geographic database. Often, the people who build databases aren't the same people who need to use them. So, how do you share your data with others who need it? Depending on who needs access to your data, there are several approaches to utilizing the capabilities of a Metadata Service.

Share your data on the Geography Network—The Geography Network is a good place to start. If the data you collect is not restricted to internal use and can be shared—either freely or for a fee—you can publish it to the Geography Network. Simply browse to *www.geographynetwork.com* and follow the link that allows you to become a publisher on the Geography Network. Once you've published your metadata, anyone who searches the Geography Network will find your data if it meets the search criteria.

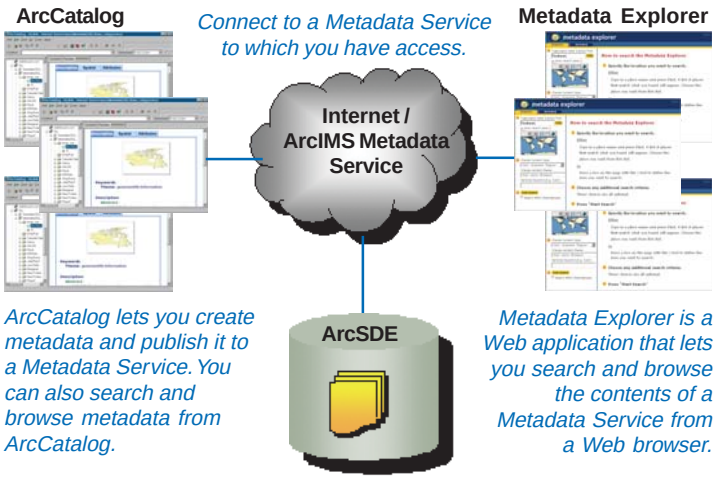
Share your data on another organization's Metadata Service—Any organization that has ESRI® ArcIMS can host a Metadata Service. If you have access to another organization's Metadata Service, you can publish your metadata to it. Anyone who can use that service will then be able to see the data you've shared. Like sharing your data on the Geography Network, this option requires that you have access to the Internet and have been granted permission to publish to the Metadata Service.

Share your data on your own Metadata Service—By building your own Metadata Service, you have full control over it. You can control who can search and browse its contents as well as who can publish metadata to it. For example, you might create a Metadata Service to facilitate data sharing between departments within your organization. You could configure your system so that only people within your organization could access it. Later, if you choose to give people outside your organization access to your service, you can easily do so.

Putting it all together

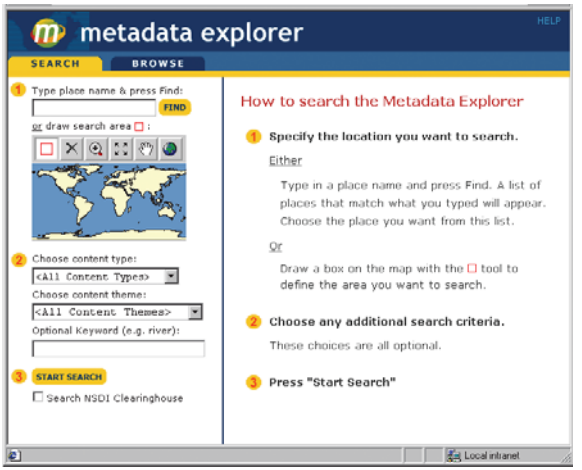
Once you've decided that you want to create your own Metadata Service, where do you begin? With metadata, of course. The fundamental components of a Metadata Service are the actual metadata documents that get published to the service. Most likely, if you're the administrator of the Metadata Service, you're not the person who will be authoring metadata—the departments or people who manage the data will be. You're simply providing a way to share metadata.

ArcIMS is just one part of the metadata system. ArcCatalog™ and ArcSDE® also play vital roles. The following diagram and descriptions show how these components work together.



ArcIMS—ArcIMS provides the mechanism for hosting a Metadata Service, allowing clients to publish to the service as well as search its contents.

Metadata Explorer—Metadata Explorer is a Web application included with ArcIMS that lets people search and browse the contents of a Metadata Service from a Web browser. Metadata Explorer makes accessing your Metadata Service look just like accessing the Geography Network.



Metadata Explorer gives people immediate access to your Metadata Service from a Web browser. Metadata Explorer requires no customization and takes only a few minutes to deploy on your Web server.

ArcCatalog—Create metadata in ArcCatalog, then publish it to a Metadata Service. ArcCatalog provides built-in editors to help you create metadata. When you're ready to publish, connect to the Metadata Service with a username and password provided by the host of the Metadata Service, then copy your metadata onto the service. Now you and anyone else who accesses the service will be able to see your published metadata.

ArcSDE—ArcIMS uses ArcSDE and the relational database with which it's configured to store and index the published metadata documents. Each time a client publishes a metadata document to the service, new records are added to database tables. Metadata documents are indexed to optimize searching and retrieving.

Tips on learning about Metadata Services

You don't need to be an ArcIMS and metadata expert if all you want to do is find some data to put on your map. If you already know of an organization that's running a Metadata Service and hosts Metadata Explorer, just point your Web browser to their site to begin searching for data. Metadata Explorer's interface is intuitive and easy-to-use. However, if you do have a question, look in Chapter 2, 'Finding data in Metadata Explorer', for the answer.

The remaining chapters of this book assume that you have a basic understanding of how ArcIMS works. For instance, you should already know what a service is and how to start and manage one. If you don't, you might want to start by reading *Using ArcIMS*, which will provide you with the necessary background.

This book will help you begin creating Metadata Services. Chapter 3, 'Creating Metadata Services', shows you how to create your own Metadata Service, and Chapter 4, 'Publishing metadata in ArcCatalog', describes how to publish metadata to a service. For more general information about metadata, metadata standards, and metadata tools, see *Using ArcCatalog*.

ArcIMS relies on the capabilities of ArcSDE and a relational database to store and index metadata. Appendix B, 'ArcSDE considerations', gives you some insight as to what's happening behind the scenes in the database. Before you create your first Metadata Service, meet with your database administrator to discuss the impact of a Metadata Service on the database; the administrator should review the information in Appendix B to ensure the database and ArcSDE are properly configured for a Metadata Service.

Finding answers to questions

Like most people, your goal is probably to complete your tasks while investing a minimum amount of time and effort on learning how to use software. You want intuitive, easy-to-use software that gives you immediate results without having to read pages of documentation. However, when you do have a question, you want the answer quickly so you can complete your task. That's what this book is all about—getting you the answers you need when you need them.

This book describes how you manage Metadata Services with ArcIMS. Although you can read this book from start to finish, you'll likely use it more as a reference. When you want to know how to perform a particular task, such as publishing metadata to a service, just look it up in the table of contents or index. What you'll find is a concise, step-by-step description of how to complete the task. Some chapters also include detailed information that you can read if you want to learn more about the concepts behind the tasks.

Getting help on your computer

In addition to this book, the ArcIMS and ArcCatalog online Help systems are valuable resources for learning how to use the software.

Contacting ESRI

If you need to contact ESRI for technical support, see the support card you received with ArcIMS or refer to 'Getting help' in the online Help system's 'Introduction' section.

You can also visit ESRI on the Web at *www.esri.com* and *support.esri.com* for more information on ArcIMS and ArcGIS®.

ESRI education solutions

ESRI provides educational opportunities related to geographic information science, GIS applications, and technology. Instructor-led courses, Web courses, and self-study workbooks are available to fit your learning style. For more information, go to *www.esri.com/education*.

Finding data in Metadata Explorer

2

IN THIS CHAPTER

- **Starting Metadata Explorer**
- **Finding data using Metadata Explorer**
- **Browsing metadata**
- **Searching for data**
- **Viewing the results of your search**
- **Viewing metadata**
- **Viewing a map of the data**
- **Printing search results, metadata, and maps**

Metadata Explorer is a Web-based application that allows you to search and browse the contents of a Metadata Service—a central repository for metadata. To get a general idea of what's available, you can browse through the contents of a Metadata Service. Alternatively, you can use Metadata Explorer to find data based on geographic extent, content type, content theme, or keyword. Then use the metadata returned by your search to determine whether the data it describes fits your needs.

Once you have access to a machine with Metadata Explorer, you are ready to search for data. This chapter introduces you to Metadata Explorer and shows you how to search for and find data. For Metadata Explorer installation and configuration instructions, see *ArcIMS Installation Guide* and Appendix A, 'Configuring Metadata Explorer'.

Starting Metadata Explorer

To start Metadata Explorer, you must know the name and domain of the computer on which it is installed. This might be your own computer or another computer to which you have access. For example, if the host computer is “mymachine” and the domain is “alpha”, start Metadata Explorer using the Web address http://mymachine.alpha.com/metadataexplorer. Metadata Explorer may not appear exactly as illustrated in this chapter. For example, if the gazetteer service is disabled, you cannot search by place name.

Tip

Why do I have to log in?

If authentication is enabled, you may have to log in to Metadata Explorer with a username and password. Authentication restricts access to Metadata Services and private metadata documents.

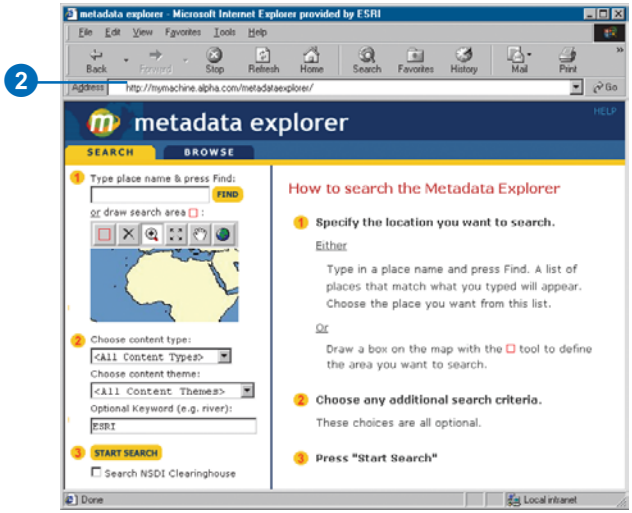
Tip

Starting Metadata Explorer from ArcMap

To start Metadata Explorer from ArcMap™, see the topic ‘Adding data from the Internet’ in the ArcGIS Desktop Help system.

1. Open a Web browser.
2. Type http://<localhost>.<domain>.<com>/metadataexplorer in the Location or Address text box.

   Metadata Explorer opens.

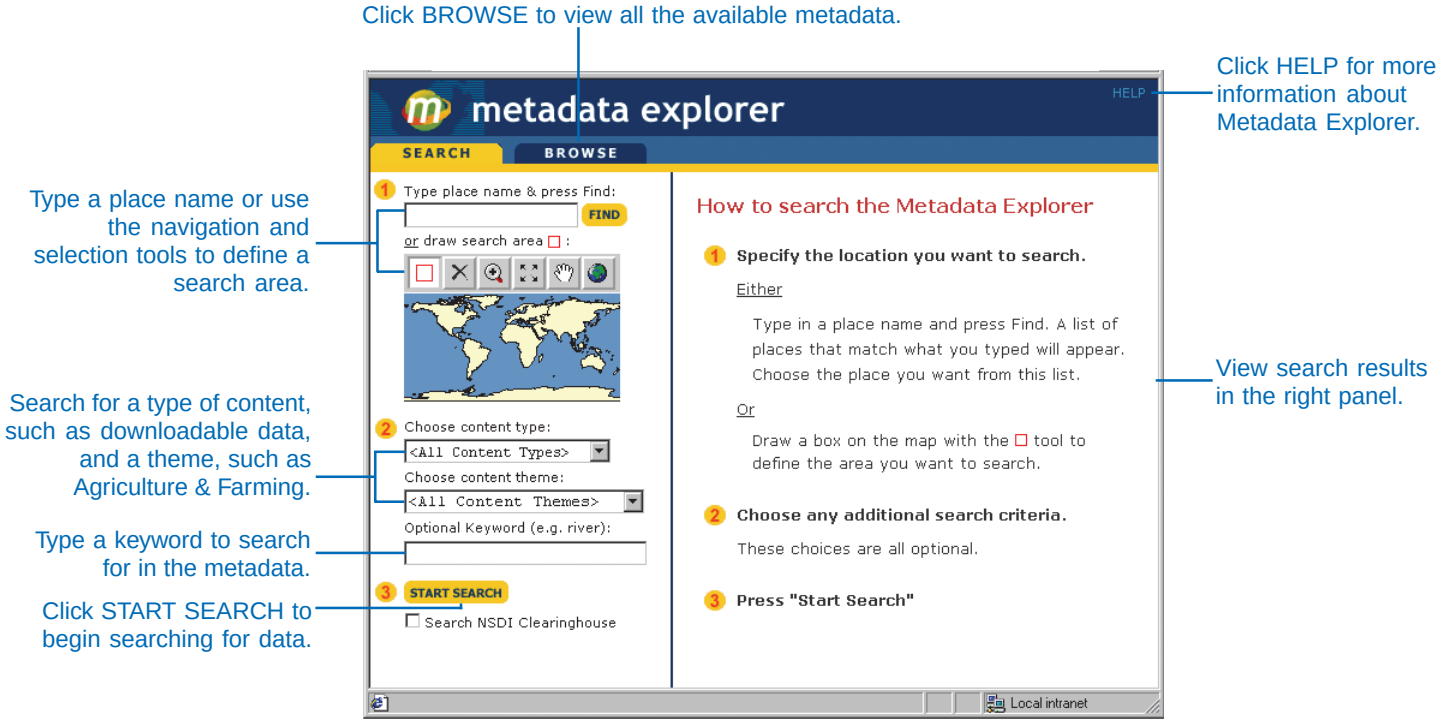


Finding data using Metadata Explorer

There are two ways to find data using Metadata Explorer: by searching or by browsing.

Published metadata documents are hierarchically organized into directories and subdirectories. Browsing involves drilling down through these directories to find the data you want. When you arrive at a directory that interests you, scroll through the list of its contents in the right panel.

Searching involves specifying search criteria, such as geographic extent, content type, content theme, and keyword. Your search can include one or more of these search criteria. Specify the geographic extent by typing a place name or by zooming, panning, and selecting a location on a map of the world. Once you have executed your search, view the search results in the right panel.



To find the best data to fit your needs, you must understand the search methods available to you. You can search by place name, geographic extent, content type, or content theme. Each method has its advantages. Use a combination of search criteria to narrow your search results.

For example, if you choose the United States as the geographic extent but do not specify a content type or content theme, Metadata Explorer returns metadata for all datasets covering the United States. If you are only interested in transportation networks or only want to find downloadable data, specifying a content theme or content type will narrow your search results considerably.

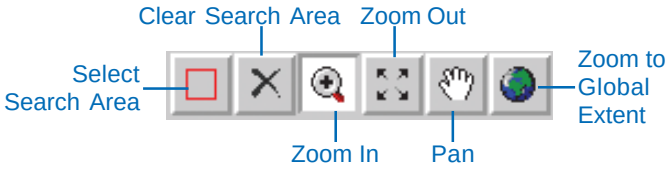
Each of the search methods is described below. Use this information to search more effectively.

Searching by geographic extent

If you know the name of the area for which you want data, it may be easiest to type it in the text box and click Find, for example, type “Orange County”. If you don’t know the correct place name but you do know how to find it on a map, you can zoom in and select it, for example, the area around Los Angeles.

When you type a place name and click Find, Metadata Explorer returns a list of matching places from the gazetteer. You might have been thinking of the state of Georgia in the United States, but your search results also include the country of Georgia in Asia, the city of Georgia Woods in Maryland, and so on. Select the place you are looking for from the list of place names that appears in the right panel. The map of the world zooms to the place that you selected.

As an alternative to searching by place name, use the Zoom In, Zoom Out, and Pan tools to find an area on the map. Once you are zoomed in enough to see the area, use the Select Search Area tool to select it.



Searching by content type

You can search for three types of content: data, documents, and resources. Each category has a number of subcategories. Search for data if you want to download, order, or add data directly to your map. Search for documents if you are interested in map files and geographic information. Search for resources if you are looking for links to external Web sites, data clearinghouses, GIS-based Web applications, and geographic services.

Data

This category is broken down into the subcategories of downloadable data, offline data, and data that can be added directly to your map.

Live Data and Maps—Dynamic geographic services that allow direct interaction with map content. This type of content is delivered in one of two ways: as a cartographic image or “snapshot” of a map or as compressed vector features that are streamed to you. Streamed features allow for greater client-side interaction, including dynamic labeling, feature symbolization, and MapTip creation. You do not need to download anything to use live data—just add it to your map and begin exploring.

Downloadable Data—Geographic data that can be downloaded to your computer. After the data has been downloaded, it can be added to your map.

Offline Data—Geographic data that can’t be directly downloaded to your computer but can be ordered online or offline from the publisher.

Documents

This category includes static map images and text or tabular information that is used in conjunction with geographic data.

Map Files—Any digital maps that can be viewed in a mapping application. Maps created for ArcReader™ and ArcMap are examples of digital maps that fit into this category.

Static Map Images—Image files that show a map. You can't directly interact with static map images as you do with live data and maps, but you can view them and download them to your computer.

Other Documents—Geographic information stored in text files, spreadsheets, or other formats. These documents are used in conjunction with geographic data. In many cases, they can be viewed and downloaded.

Resources

This category includes links to external geographic data sources, GIS applications, and geographic services.

Applications—Custom GIS tools or software. Applications usually have a complete user interface.

Geographic Services—Internet services that perform basic geoprocessing tasks, such as address matching and buffering.

Clearinghouses—Web sites that either provide links to available geographic data or allow you to directly download it.

Geographic Activities—An event or set of events that are geographic in nature, for example, land surveys, geographic explorations, and GIS conferences. The information about these events can be presented in many forms, such as Web sites.

When choosing the type of content, you should consider the level of interaction you want to have with the data and whether you want to download it to your own machine.

Searching by content theme

A content theme is a categorical description of the features in a dataset. Agriculture & Farming, Human Health & Disease, Oceans & Estuaries, and Imagery & Base Maps are all examples of content themes that you can choose. Sometimes a content theme precisely describes what you are looking for, making your choice simple. Other times, you are not sure how to classify the data. If you are not sure whether the soil data you are looking for is classified as Agriculture & Farming or Environmental, you could search twice (once by each category) or you could search by All Data Themes.

Searching by keyword

Searching by keyword is an excellent way to narrow your results. If you are searching for something specific, such as satellite data, typing the keyword "satellite" limits search results to datasets whose metadata includes the word "satellite". With keyword searches in Metadata Explorer, a full text search is used to check the published metadata documents.

Sometimes a keyword limits your search too much, eliminating datasets that might have been useful to you. For example, even though a RADARSAT image is, in fact, a satellite image, its metadata might not include the word "satellite". If you use a keyword in your search criteria and you don't find what you're looking for, try repeating your search using other words with a similar meaning. For example, if you are interested in rivers, you might try "rivers", "streams", "hydrology", "drainage", and so on.

Browsing metadata

All metadata documents are hierarchically organized into directories and subdirectories. When you click a directory, a list of subdirectories appears. Click each directory to see its subdirectories, and so on, drilling down as far as you want to go.

The right panel lists the documents in the current directory. The title and publisher of each dataset is provided. Often the coverage area and map scale are provided as well. You might also see a thumbnail image of the data. You can view the detailed metadata for each record and, in some cases, view a map of the data as well.

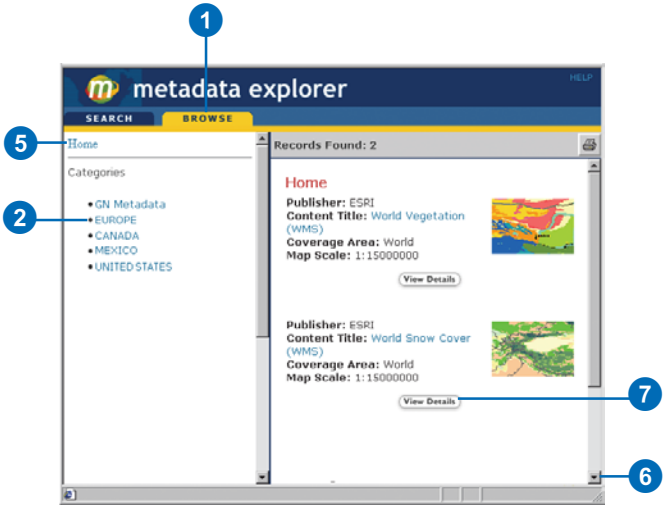
See Also

See 'Viewing the results of your search', 'Viewing metadata', and 'Viewing a map of the data' in this chapter for more information about metadata and maps.

1. Click BROWSE.
2. Click a directory in the left panel.

   A list of subdirectories appears in the left panel.
3. Click a subdirectory.

   A new list of subdirectories appears in the left panel, if any subdirectories exist.
4. Repeat step 3 to continue drilling down.
5. Your current position in the Metadata Service appears at the top of the left panel. Click one of these directories to list its contents.
6. Once you have arrived at the directory you want, scroll through the list of the contents in the right panel.
7. Click View Details to view a dataset's detailed metadata or click View Map to view a map of the data.



Searching for data

Many searches begin by specifying the geographic extent. The geographic extent is the geographic area for which you want to find data. You can specify the geographic extent by typing a place name or by selecting an area on the map. If you type a place name, places in the gazetteer matching that name are listed; pick the appropriate place in the list to set the geographic extent on the map. You don't have to specify a geographic extent for your search, but if you have a particular area in mind, doing so will narrow your search considerably.

To narrow your search further, include additional search criteria. For example, a forester interested in forest fires in Montana might search by the keyword "fire". He might also choose downloadable data because he wants to work with the data offline.

Tip

Using the zoom, pan, and selection tools

For more information about the navigation and selection tools, click Help.

1. Type a place name, click Find, and click a place name in the right panel or use the zoom, pan, and selection tools to select a location on the map.
2. Specify any combination of the following: content type, content theme, and keyword.

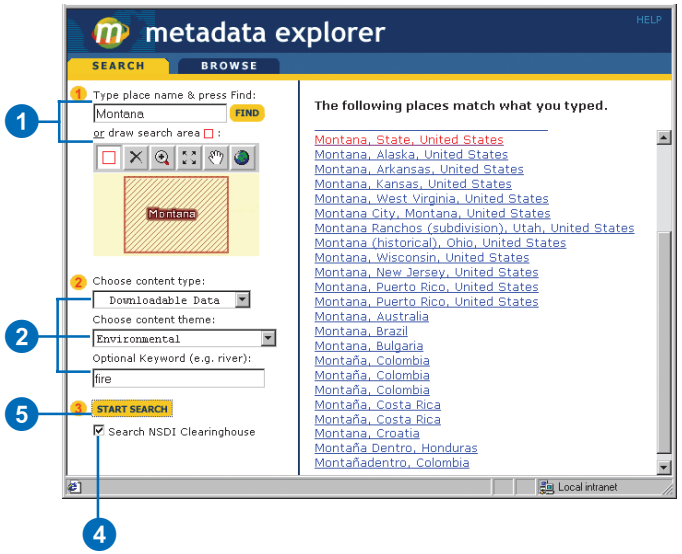
   Click the content type dropdown arrow and click Data, Documents, or Resources or one of their subcategories.

   Click the content theme dropdown arrow and click one of the available themes.

   Type a keyword.
3. Review your search criteria. If you typed a keyword, check it for typographical errors.
4. To extend your search to the Geospatial Data Clearinghouse, check Search NSDI Clearinghouse.
5. Click START SEARCH.

   Metadata Explorer search results are displayed in the right panel.

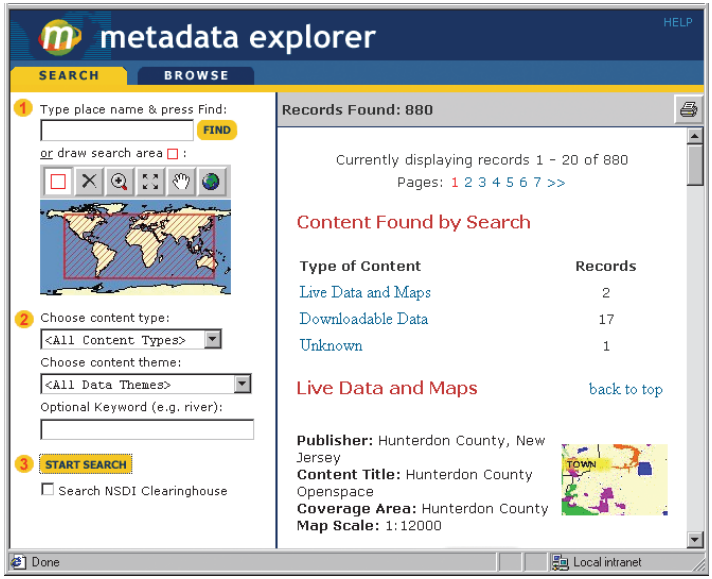
   If you checked Search NSDI Clearinghouse, the Geography Network NSDI search application opens. Search additional servers and view additional search results. Click CLOSE when you are finished.



Metadata Explorer may not appear exactly as illustrated above. For example, if the gazetteer is disabled, you won't be able to search by place name in step 1.

Viewing the results of your search

If you searched for data, the results of your search are summarized in the right panel. If you browsed to a directory, the contents of the directory are listed in the right panel. If searching or browsing returned more than one type of content, the records are listed by content type.



Use the scroll bar to move through the list of datasets. The description of each dataset includes the name of the publisher, the content title, and often the coverage area and map scale.

Viewing metadata details

The content summary panel gives you an indication as to whether or not a geographic service or file is suited to your needs. To know for sure, you must look at the detailed metadata.

The View Details button lets you view a summary of that dataset's metadata.

The publisher of the metadata must provide the title, type, spatial extent, and publisher of the content. You should also see the following information about the dataset: the publication place and date; a description of the content and its purpose; the state of the data; the frequency with which it is updated; the coverage area; and access or use constraints, which are restrictions or legal requirements affecting use of the data. The metadata might also include the data type, format, scale, and several keywords. If you want more information, click the View Full Metadata button at the bottom of the Details page—the entire metadata document will appear in a new window.

The metadata panel gives you the opportunity to view the coverage area. When you select this option, a map of the world opens in a new window. The region of the world covered by the dataset is outlined by a green box.

Viewing and using live data and maps

When you find live maps or data with your search, the View Map button appears beneath the record summary. You can view, zoom in, zoom out, and pan to explore the maps and data.

From the search results, you can also add these live services directly to your current ArcMap or ArcExplorer™ map, provided that:

- The selected dataset or map references a live service.
- Metadata Explorer was launched from within ArcMap or ArcExplorer.

When these criteria are met, the Add to ArcMap or Add to ArcExplorer button appears below the appropriate record in the search results.

Viewing metadata

Viewing the metadata for a dataset will generally either confirm your initial impression about its suitability or eliminate it as an option. Pay particular attention to the access and use constraints because these also affect the suitability of the data. If you want the data for a use prohibited by the publisher, you will have to find another dataset. The DETAILS tab shows you a summary of the dataset's metadata. If you want more information, you can see the complete metadata document in a new window.

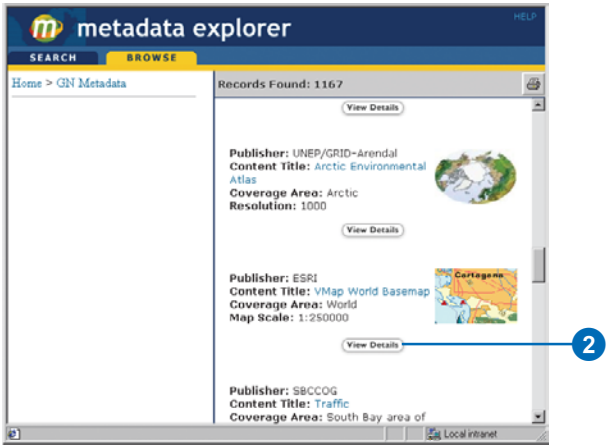
The Metadata tab also allows you to view the coverage area. This is particularly useful when you cannot view a map of the data. The coverage area is indicated by a green box.

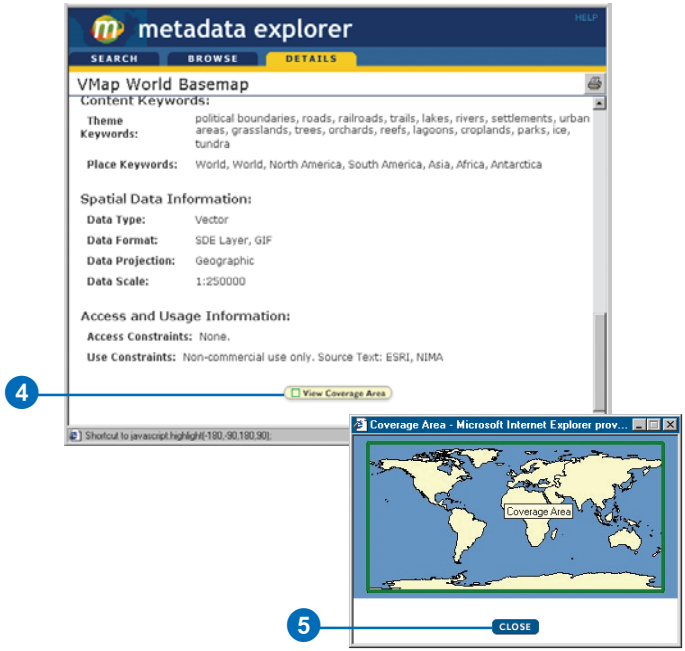
1. Find a dataset that interests you by browsing or executing a search.
2. Click the View Details button beneath the record in the right panel.

   The DETAILS tab appears.
3. Examine the metadata summary for the selected dataset.
4. Scroll down and click View Coverage Area.

   A new window appears showing a map of the world, zoomed in to the coverage area of the selected dataset. A green box outlines the coverage area.
5. Click the CLOSE button when you are finished looking at the coverage area.
6. Click View Full Metadata.

   A new window appears showing the complete metadata document.





Viewing a map of the data

Metadata Explorer lets you view maps and datasets that are available as live services, such as a Web Mapping Service (WMS) service or an ArcIMS Feature Service. Several navigation tools let you pan and zoom to explore the service. Use these tools to take a closer look at different features of the map or data to discover if it is suitable for your needs.

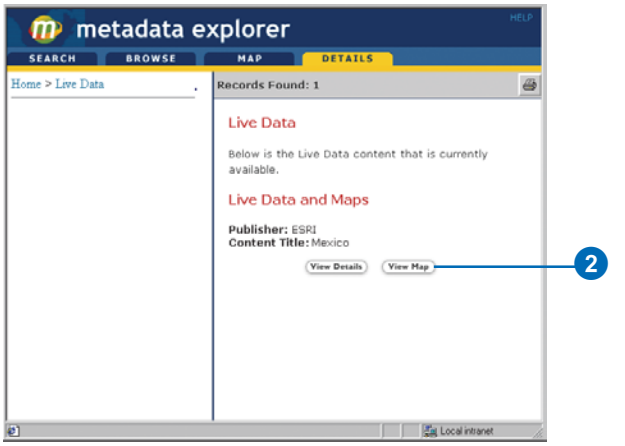
Tip

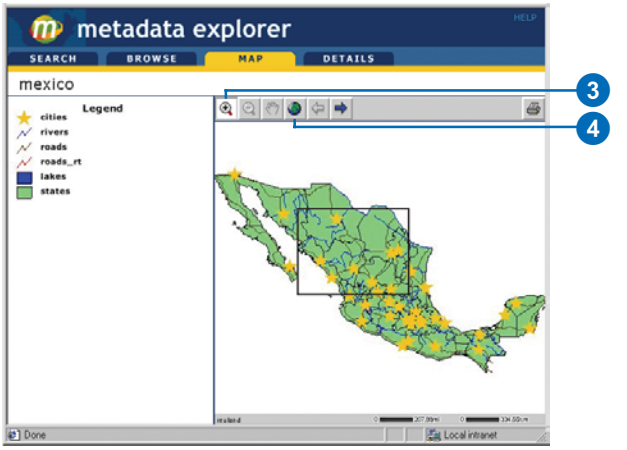
Why can't I view a map?

You'll only see the View Map button when the metadata references live services.

1. Find a dataset that interests you by browsing or executing a search.
2. Click the View Map button beneath the record in the right panel, if one exists.

   A map of the data appears in the right panel.
3. Click Zoom In, Zoom Out, or Pan on the toolbar to change the map extent.
4. Click Zoom to Full Extent to return to the full extent of the map.





Printing search results, metadata, and maps

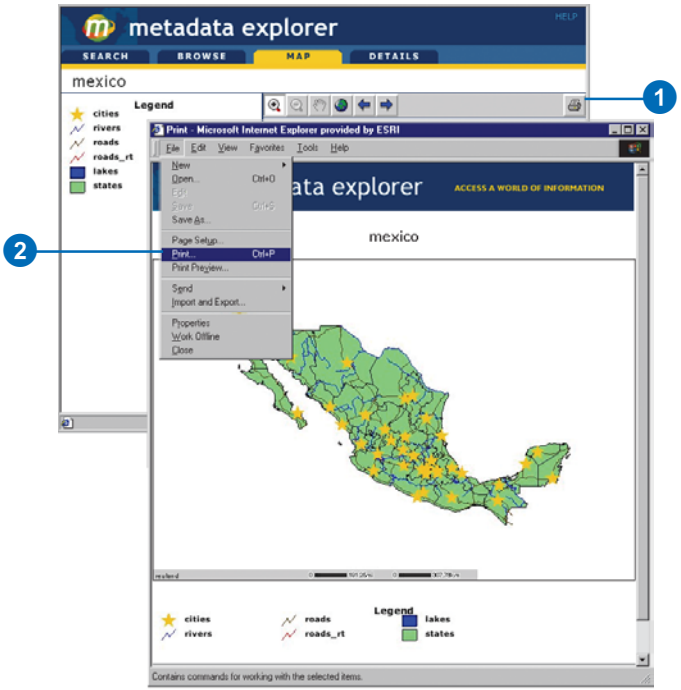
Anything that appears in the right panel of Metadata Explorer can be printed. This includes maps, metadata, and search results.

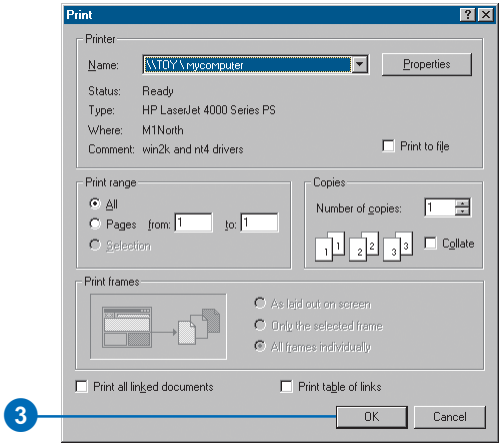
The printable version of search and browse results excludes the View Details and View Map buttons. The printable version of the map excludes the navigation tools.

1. Click the Create Printable Version button in the upper right corner of the right panel.

   A new browser window appears containing a printable version of the map, metadata, or search results.

2. Click File and click Print to open the Print dialog box.
3. Click OK on the Print dialog box.
4. Click CLOSE at the bottom of the browser window to close the printable version.





Creating Metadata Services

3

IN THIS CHAPTER

- **Setting up your Metadata Service**
- **Creating a Metadata Service**
- **Creating usernames and passwords**
- **Troubleshooting publishing problems**
- **Metadata for ArcIMS services**
- **Defining a service's index**
- **Manually indexing documents**
- **Adding an administrative table to a service**

Once you've decided that the best way to share your organization's data is to create your own Metadata Service, creating one is straightforward. Before you do so, there are a few things you should think about, such as:

- Who will use your Metadata Service?
- Will you allow people outside your organization to publish to it or view its contents?
- How many metadata documents will be published to your service? This will determine the amount of storage space required in your database.
- In which languages are metadata documents authored?
- How many people will be using the service at the same time?
- How many documents will be published at a time and when will they be indexed? Indexing many documents during publishing takes longer and puts a greater load on the system.
- How will documents be indexed? You can determine what information is indexed and whether it's indexed as text or as numbers.
- Do you need to create more than one Metadata Service? If you want to keep metadata documents from different groups completely separate, you may choose to create more than one Metadata Service.

Your answers to these kinds of questions will help determine how you set up your Metadata Service.

Setting up your Metadata Service

A Metadata Service provides two fundamental operations. It allows people to:

- Publish a metadata document to a Metadata Service.
- Retrieve metadata documents from a Metadata Service.

Before you start your Metadata Service, you should:

- Identify who will use it.
- Configure your relational database to store the metadata documents published to the service.
- Decide how you want to configure your Metadata Service to meet your needs, such as determining how to index documents.

Following is an overview of the above topics. The specific steps required to complete these tasks are described later in this chapter.

Providing access to your Metadata Service

When you create your Metadata Service, you should know who needs to access it. Some people will be providing the content of the Metadata Service when they publish metadata documents to it. Others will only search for and view metadata documents to see if the data referenced by the documents is suitable for their use.

You control access to your Metadata Service by requiring people to verify who they are by providing a valid username and password before they are granted access to view or publish metadata. The process of verifying the usernames and passwords is called *authentication*. By default, authentication is disabled in ArcIMS; thus, you must enable it to effectively work with Metadata Services. For more information on authentication, see the topic ‘Restricting access to services’ in *ArcIMS Help*.

You will want to create usernames for people to publish documents to your Metadata Service because you don’t want just anyone to be able to do it. However, you may choose not to require a username for simply viewing and searching your Metadata Service. Ultimately, your decision depends on whether or not your Metadata Service contains sensitive information not intended for public consumption.

How many different usernames you need to set up to access your Metadata Service will depend on how strictly you want to control access. It may be sufficient to have one username that everyone uses to access the service. Anyone who needs to publish or view metadata documents would log in as this user. Alternatively, you might want to create a unique username for every individual who will access the service. Or, you may find that creating a few different usernames will suffice—for instance, one for each department, organization, or group of people that needs to publish metadata.

Setting up the database

A Metadata Service relies on ArcSDE and a database management system (DBMS) to store, search for, index, and retrieve published documents. Whenever you create a Metadata Service, ArcIMS creates several objects in the database to store and manage the published documents. Before you can create a Metadata Service, the database and ArcSDE must be properly configured to support ArcSDE eXtensible Markup Language (XML) columns. You should meet with your database administrator to discuss the impact of a Metadata Service on the database, including how to index documents. Appendix B, ‘ArcSDE considerations’, provides the information you and your database administrator should know.

Publishing and indexing documents

By default, whenever someone publishes a document to a Metadata Service, all information in the document is indexed as text. This lets clients, such as ArcCatalog and the Metadata Explorer, quickly search for and retrieve documents that contain user-specified text. For example, if you use ArcCatalog to search for documents that contain the word “population”, the index is checked to determine which documents contain that word.

As you might imagine, both publishing and indexing documents can impact the performance of your database. If someone publishes a large number of documents, both the publishing and indexing processes can slow the response time for other users who are searching and browsing the Metadata Service. This is because all processes are ultimately handled by the database.

The amount of time it takes to publish and index a single document depends on how much information it contains and how indexing is configured. You can define which pieces of information will be indexed and whether they are indexed as text or as numbers. Configuring the index to handle numbers is important because date and numeric searches can only be performed on information that has been indexed numerically. Following the principle “less is more”, the less information indexed, the faster both publishing and indexing will be.

When to index documents

When you start your Metadata Service, you can control when indexing occurs. Indexing can occur automatically whenever anyone publishes a metadata document to your server, or you can control indexing manually. The following list highlights some of the advantages and disadvantages of the different indexing methods.

Automatic indexing

- When you start a Metadata Service, ArcIMS can automatically index documents as they are published.
- Metadata documents can be searched using text immediately.
- When a large number of metadata documents are being published (by one or more publishers), automatic indexing will slow down the response time for searching and browsing the Metadata Service while those documents are being indexed.
- People publishing metadata documents in ArcCatalog must wait for indexing to complete. During publishing, ArcCatalog will be busy while the documents are stored and indexed on the server.

Manual indexing

- Requires that you set up a program to run the aimsmetaindx command at specific intervals—for example, a cron job or at command. Or you can work with the database administrator to schedule index updates using the database’s tools.
- Documents published to your Metadata Service will not be found with text-based searches until the documents are indexed.
- Because you control when indexing occurs, you may choose to index documents at off hours when there is less load on your server.
- Allows for faster publishing of documents, especially when publishing many documents at once. On average, publishing with manual indexing takes less than half the time it does with automatic indexing.

Automatic indexing is not currently supported with DB2®. With SQL Server™, additional options in the database provide different levels of automatic indexing. See Appendix B, ‘ArcSDE considerations’, for details.

In general, you'll probably want to start your Metadata Service with automatic indexing enabled. Depending on the number of people publishing to and searching your service, you may choose to switch to manual indexing to improve performance. See the section 'Manually indexing documents' later in this chapter for information on how to implement manual indexing.

How to index documents

The information in a metadata document serves a variety of purposes. Some information helps you search for resources and filter out the candidates. Other information provides details about data you've obtained and helps you track changes to it over time.

A Metadata Service needs an index that lets users quickly find the resources they need. The index should only include information used to discover resources. For example, the document's title, abstract, and any keywords should be indexed; if the search interface lets you search by dates indicating how recent the resource is, that date must be indexed also. Typically, you would not index information about data attributes, coordinate systems, or data quality unless you have a requirement to do so, for example, to support predefined searches that are internal to your organization.

Two settings control how the Metadata Service's index is created: the index type and the definition file. Set these properties for a Metadata Service using the MDAlterIndex command.

There are three index type options. The "auto_string" option automatically indexes everything in the document as text; this is the default setting. The "auto" option automatically indexes everything in the document; its default behavior is to determine whether to index a piece of information as either text or number based on the first value encountered. The "definition" option only indexes the information specified in a definition file.

Index definition options

	AutoString	Auto	Definition
Indexes entire document	Yes	Yes	No
Definition files include and exclude information	Yes	Yes	Yes
Indexes text automatically	Yes	Yes	No
Indexes text with definition file	Yes	Yes	Yes
Indexes numbers automatically	No	Yes	No
Indexes numbers with definition file	No	Yes	Yes
Numeric index conflicts reported	No	Yes	Yes

This table summarizes the relationship between your choice of index type and how you can use a definition file to configure the index.

A definition file lets you determine what information to include and what information to exclude from the index. For example, the definition file can be used with the auto_string and auto index types to index everything except the excluded information. Or the definition file can be used with the definition index type to index only the included information.

A definition file also lets you determine whether to index each piece of information as text or numerically. To support date and numeric searches, the appropriate information must be indexed

numerically; therefore, the auto or definition index types must be used. With a definition file, the auto index type will index information as specified in the file rather than devising automatically how to index information.

During the indexing process, if a piece of information is supposed to be indexed as a number but the document actually contains text, the value won't indexed and a conflict will be recorded in the database. For example, if the definition file specifies that the map's scale should be indexed numerically, the value "24000" will index successfully but the value "1:24000" or "1/24000" will not. Indexing conflicts can be listed using the MDTagConflictList command.

In contrast, if a piece of information is supposed to be indexed as text but the document actually contains a number, the value will be indexed as text. A conflict won't be recorded in the database.

For the information indexed as text, indexing is accomplished using the database's full text indexing capabilities. The words that are indexed will depend on the database's indexing rules. Typically, words that have meaning, such as "river", "pollution", and "population", will be indexed, but words such as "and", "the", and "in" won't be indexed. Different rules may be used to index text when the database is set to use a language other than English.

For text-based searches, the Metadata Service queries the database's text index; those queries will be faster if the text index is smaller. Also, the amount of time it takes to publish and index a single document depends on how much text must be indexed. A succinct index definition file will improve the performance of publishing and indexing as well as querying.

Using validation to support searches

Metadata Services don't require metadata to follow a metadata standard, such as the Federal Geographic Data Committee (FGDC) *Content Standard for Digital Geospatial Metadata* or the International Organization for Standardization (ISO) standard 19115, *Geographic information—Metadata*. In fact, you may have your own organizational standard for documenting your data. As long as you have a properly formatted XML file, you can publish it to a Metadata Service.

However, to optimize searches in Metadata Explorer and ArcCatalog, you can choose to validate metadata documents published to your server. By enabling validation, you'll improve the likelihood that documents will be found by searches. Metadata Explorer and ArcCatalog check both FGDC and ISO tags when performing searches; in some cases, they expect particular values in those tags. For example, when searching by Content Theme, the theme code defined in the search is compared to values in the XML tags that are supposed to store the Content Theme codes. If an appropriate code doesn't exist in one of those tags it will never be found by a Content Theme search.

By turning validation on, you are requiring that any metadata document published to your Metadata Service will contain the appropriate tags set with the appropriate values. Enabling validation only verifies that the five XML tags ArcCatalog and Metadata Explorer use for searching have been set. It does not validate the complete contents of the metadata document nor ensure that the document conforms to a metadata standard.

Extending Metadata Services with administrative tables

A Metadata Service stores a record in the database for every document published to it. For each record, columns in the database contain information about the document, such as its name, the geographic extent of the resource it represents, an online link to the data or information about the data, and whether or not it has any related documents. Some of this information is

derived from the document during publishing, and some is maintained by the Metadata Service to manage the published documents. See Appendix B, ‘ArcSDE considerations’, for details about the database schema for Metadata Services.

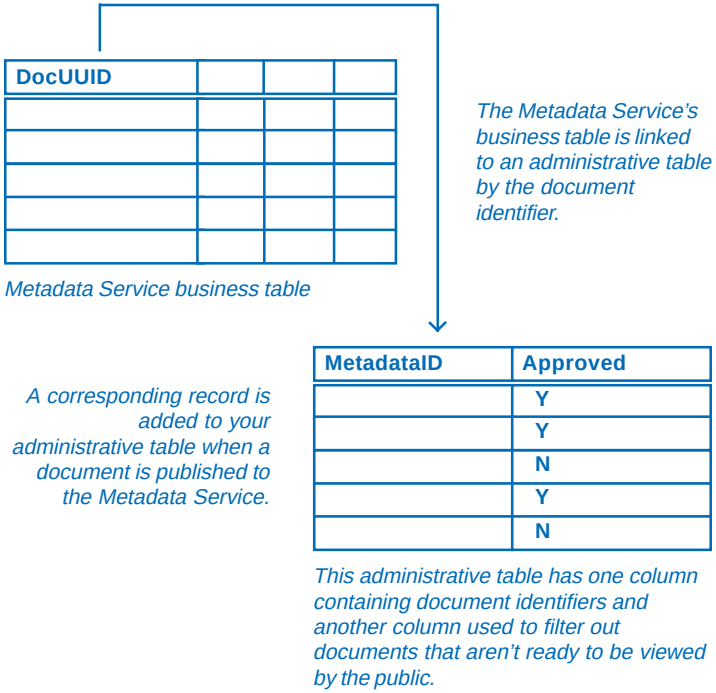
This database schema was designed and developed to work in conjunction with ArcCatalog and Metadata Explorer. For example, Metadata Explorer lets you preview an ArcIMS service associated with a published document. The information about the ArcIMS service is stored in columns in the database and is included directly in the ArcIMS response when the document is found by a search. Therefore, Metadata Explorer can get the information directly from the response and not have to open and look through the contents of the published XML document to find the online link. The database schema allows Metadata Explorer to quickly retrieve the information it needs.

Suppose you want to write your own application that utilizes a Metadata Service. You might want to store your own information about a document in the database so it can be quickly retrieved by your application. While you can’t add columns to the Metadata Service’s business table, you can extend it by creating your own table in the database to hold the required information. This table is referred to as the administrative table.

The administrative table is joined to the business table by the document identifier. In the business table the document identifier is stored in the DocUUID column. One of the columns in your table must be able to contain strings 38 characters long so it can contain the identifiers matching those stored in the DocUUID column.

When a document is published to your Metadata Service, a Structured Query Language (SQL) insert statement is executed to add a record to your administrative table that corresponds to the published document. The insert statement is specified in the ArcXML (AXL) configuration file that you use to start your Metadata Service. This stores the document identifier in the appropriate column along with any other default values you wish to add.

When a document is deleted from a Metadata Service, the corresponding record in the administrative table is *not* deleted. Your application must manage deletion of records from your administrative table. ArcIMS maintains a table in the database that keeps track of deleted metadata documents. Your application could use the information in this table to delete records from your administrative table.



When a document is retrieved from the Metadata Service, a SQL where clause specified in the configuration file is appended to the query sent by ArcIMS to the database; this filters the documents returned based on information in your table. Suppose your organization has strict requirements regarding the content of metadata published to your Metadata Service. The service administrator or librarian will probably want to review the published documents to verify they meet your requirements before letting them be viewed by others. You might validate documents against a metadata standard or simply verify that a value is present to support a predefined search that is available in a customized version of Metadata Explorer. You can use this mechanism to filter out documents that haven't passed your inspection.

At a minimum, your administrative table would need two columns: one for the document ID, which is set by the Metadata Service when the document is published, and the other for your approval code, such as "yes" or "no". When a document is published, a record is added to the service's primary table. The insert statement adds a record to the administrative table and sets the value in the approval column to "no". The Metadata Service's filter only retrieves documents with the value "yes" in the approval column. With a custom application, you could review unapproved documents and, once their content has been verified, change the value in the approval column to "yes". The published document would appear in search results after being approved.

This example illustrates a simple administrative table with only one custom column. You could easily create a more sophisticated administrative table or set of tables containing many additional columns that are required to support your custom application.

Administrative tables must be created directly in the database, for example, by executing a SQL create table statement. They must not be created using ArcCatalog or ArcSDE commands.

Administrative tables must exist before a Metadata Service has been created; if they are created afterward, they won't contain records corresponding to any documents that have already been published. To add an administrative table to an existing Metadata Service, you must create the table in the database, create a new configuration file that references it, and use it to start a new Metadata Service, then republish your documents to the new service.

Creating a Metadata Service

Just like Image and Feature Services, a Metadata Service starts with a configuration file. To help you create your Metadata Service, a sample file is installed with ArcIMS that you can edit. Before starting your first Metadata Service, read Appendix B, ‘ArcSDE considerations’, and discuss its contents with your database administrator. Then, edit the configuration file in a text editor.

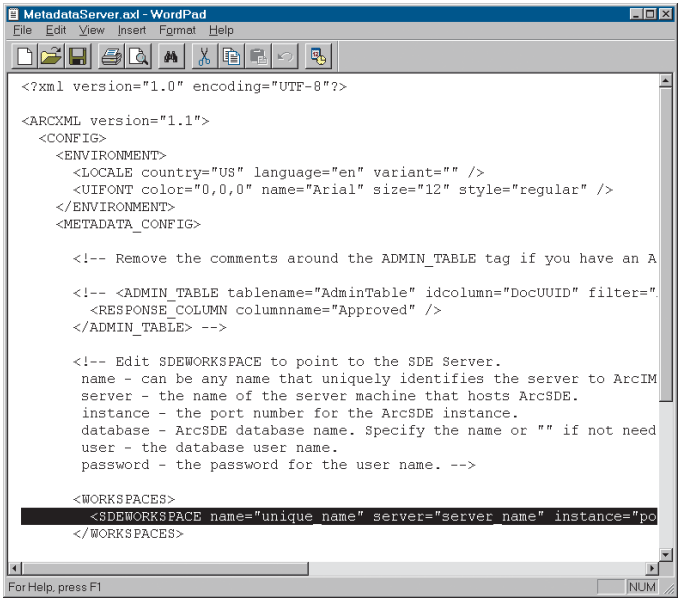
You must provide the appropriate ArcSDE connection information. For a basic configuration, use default values for the other options, which are:

Table name prefix—The prefix used to name the objects created in the database by ArcIMS. To create a second Metadata Service that contains a different set of documents, you must use a different prefix.

Validation—Whether or not documents will be validated to see if they contain the information required by Metadata Services to support searches in ArcCatalog and Metadata Explorer; documents lacking the required information won’t be published. This ▸

Editing the MetadataServer.axl file

1. In a text editor, edit the MetadataServer.axl file in the AXL directory created on installation. On Windows, this file is located by default at C:\ArcIMS\AXL\Metadata.
2. Edit the SDEWORKSPACE element and type in the appropriate information to connect to ArcSDE.
3. Set TABLE_NAME prefix to a unique name for the Metadata Service objects in the database. The default value is “imsmetadata”. This name should not contain spaces or be more than 14 characters in length.
4. Set METADATA_CONTENT validate to “true” or “false”. When set to true, documents will be validated before they are published to ensure they contain the five pieces of information required for Metadata Explorer and ArcCatalog.
5. Set METADATA_CONTENT index_words to “automatic” or “manual”. When set to manual, you must run a command or use database tools to index the published documents. For SQL Server and DB2 databases, see Appendix B, ‘ArcSDE considerations’.



Edit MetadataServer.axl in a text editor. You only need to specify the connection information to your ArcSDE database. The other elements in this file have default values you can use when you first start your Metadata Service. Later, you can modify those elements to suit your needs.

option does not ensure the content conforms to a metadata standard, such as FGDC or ISO.

Indexing—Whether or not metadata documents are indexed immediately after they are published. In general, when you first start your service, leave indexing set to automatic. Change it to manual if you notice performance issues or if index updates are scheduled in the database. For SQL Server and DB2, see the indexing information in Appendix B, 'ArcSDE considerations'.

After editing the configuration file, you can start the Metadata Service using Administrator. However, before publishing documents, you must enable authentication and define usernames and passwords for accessing the service. For information about using an administrative table to extend a Metadata Service, see 'Adding an administrative table to a service'.

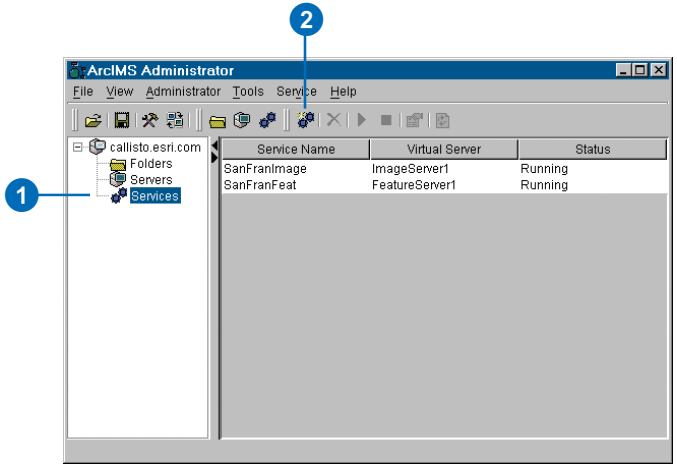
See Also

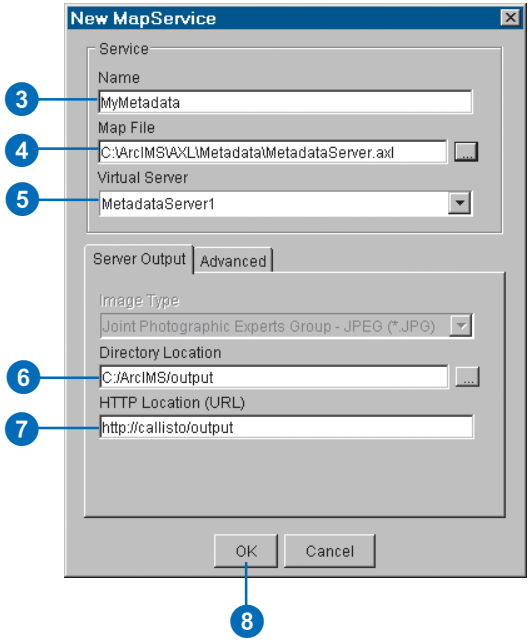
For more information, see 'Setting up your Metadata Service' in this chapter. Before starting your Metadata Service, meet with your database administrator and discuss the ArcSDE and database configuration topics in Appendix B, 'ArcSDE considerations'.

Starting a Metadata Service

1. Start Administrator and click Services in the left panel to display the currently running services.
2. Click the New Service button.
3. Type in the Name for your Metadata Service. This is the name people will see when they access your Metadata Service.

   Before starting your first Metadata Service, read Appendix B, 'ArcSDE considerations'.
4. Locate the Metadata Service configuration file that you've created. The sample file is named MetadataServer.axl.
5. Click the Virtual Server dropdown arrow and click MetadataServer1.
6. Type the Directory Location where ArcIMS will store the output XML files returned when a user searches or browses your Metadata Service.
7. Type the HTTP Location associated with the directory location specified in the step above.
8. Click OK.





Creating usernames and passwords

You can control access to your published documents by creating usernames and passwords that people must use to access the Metadata Service. The list of usernames is stored in an access control list (ACL). You can implement an ACL using a text file or Java™ Database Connectivity (JDBC). This section describes a file-based implementation.

The process of verifying the usernames and passwords is called *authentication*. Both ArcCatalog and Metadata Explorer utilize authentication; however, each application communicates with ArcIMS through different connectors—ArcCatalog uses the Servlet Connector and Metadata Explorer the Java Connector. Thus, you enable authentication differently for each application, but they can both use the same ACL if you choose.

See Also

For more information about enabling authentication, setting up usernames, see the topic 'Restricting access to services' in the ArcIMS Help.

Creating usernames and passwords to access the Metadata Service

1. Create an ACL file with a text editor and name it aimsacl.xml.
2. Define username and password pairs that provide access to your ArcIMS services.
3. List the services that can be accessed with the username and specify a role.

   The sample ACL file to the right illustrates a typical Metadata Service configuration. The Metadata Service is called MyMetadata. The sample also contains four map services (SanFrancisco, California, USA, and SearchMap) and another Metadata Service to support Metadata Explorer (Gazetteer).
4. Save and close the file. You can save this file in any location; however, you'll probably want to save it somewhere in the ArcIMS Installation Directory.

   You must stop and start the servlet engine for changes to an ACL file to take effect. If the servlet engine is part of the Web server, start and stop the Web server.

```
<?xml version="1.0" ?>
<AIMSACL>
<USER name="*" services="SanFrancisco,
   California, USA, SearchMap, Gazetteer,
   MyMetadata" roles="metadata_browser"/>
<USER name="publish" password="publish"
   services="MyMetadata"
   roles="metadata_publisher" active="1" />
<USER name="author" password="author"
   services="*" roles="metadata_service_author"
   active="1" />
<USER name="admin" password="admin"
   services="MyMetadata"
   roles="metadata_administrator" active="1" />
</AIMSACL>
```

Access to Metadata Services is defined by the roles parameter in the ACL. The following roles are available:

metadata_browser—Allows access to public documents in a Metadata Service; the default if a user's role isn't specified.

metadata_browser_all—Same as metadata_browser, but allows access to public and private documents.

metadata_publisher—Allows publishing to a Metadata Service.

metadata_service_author—Same as metadata_browser, but allows creating metadata for ArcIMS Image and Feature Services. The service metadata must be published to a Metadata Service to be available for searching.

metadata_administrator—Allows full access to documents, including the ability to delete documents. Also, a Metadata Service can be emptied and reset using an ArcXML request.

In the sample ACL file above, the username "*" indicates that users don't need to log in to access the specified services; thus, anyone can browse the MyMetadata service. However, to publish documents or create metadata for a service, users must log in. Specifying services "*" indicates that the username, password, and role apply to all services.

Tip

Where is the Esrimap_prop file?

During installation, you specify the location of your servlet engine. The Esrimap_prop file is copied to that location. Common Microsoft Windows® locations include:

- ***Sun Java System:*** *\iPlanet\servers\docs\servlet*
- ***Tomcat:*** *<Tomcat Install Directory>\webapps\ROOT\WEB-INF\classes*
- ***JRun:*** *<JRun Install Directory>\servers\default\defaultear\default-war\WEB-INF\classes*
- ***ServletExec:*** *<ServletExec Install Directory>\ServletExec AS\se-<instance-name>\Servlets*

See the ArcIMS Installation Guide *for more information.*

Tip

Basic or Digest authentication?

Digest authentication is more secure; however, not all browsers support it.

Tip

Can I use the same ACL file for Metadata Explorer and ArcCatalog?

Yes, although you may choose not to. The gazetteer and search map must be free services for Metadata Explorer; ArcCatalog users will see them if you use the same ACL.

Turning on authentication for the Servlet Connector (for ArcCatalog)

1. Edit the Esrimap_prop file in a text editor; it's installed with your servlet engine.

   ArcIMS installs a backup copy of all the files that comprise the Servlet Connector in the ArcIMS installation directory. Do not edit the Esrimap_prop file at this location; doing so will have no impact on the operation of the Servlet Connector.

2. Enable authentication.

   `authenticate=True`

3. Set the authentication method.

   `authMethods=Digest`

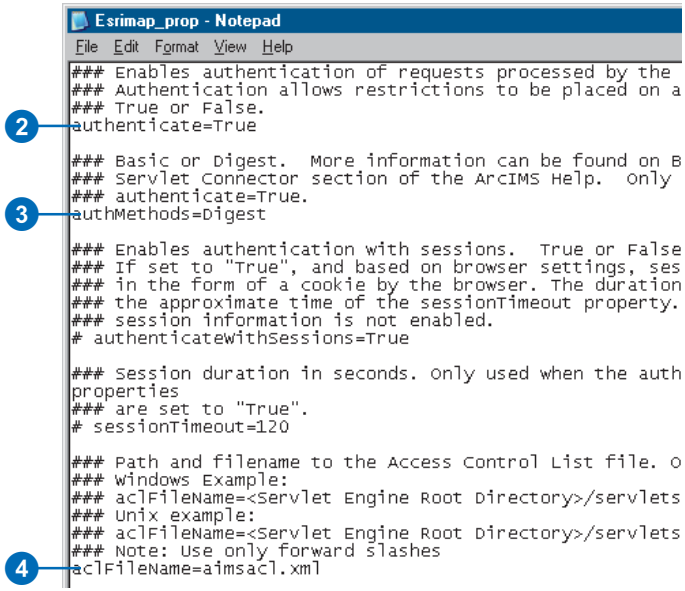
4. Set the pathname to the ACL file. Slashes in the pathname must be forward slashes (/).

   `aclFileName=<full pathname to the ACL file>/aimsacl.xml`

5. Save and close the file.

6. Stop and start the servlet engine. If the servlet engine is part of the Web server, start and stop the Web server.

   This is required for your edits to the Esrimap_prop file to take effect.



Tip

Where is the Authenticate.properties file?

The Authenticate.properties file is located in the install directory at:

- ***Windows:*** *<ArcIMS install directory>\Metadata\metadataexplorer\WEB-INF\classes\Authenticate.properties*
- ***UNIX:*** *<ArcIMS install directory>/arcims/Metadata/metadataexplorer/WEB-INF/classes/Authenticate.properties*

If you are using JRun, Tomcat, or WebSphere, it will be located where you installed the Web software in the directory shown below:

- ***JRun 4:*** *<JRun home>\servers\default\SERVER-INF\temp\metadataexplorer.war-0123456789\WEB-INF\classes*

  The 10-digit number is generated when the servlet is deployed.
- ***Tomcat:*** *<Tomcat install directory>\webapps\metadataexplorer\WEB-INF\classes*
- ***Websphere 5.0:*** *<WebSphere install directory>\AppServer\installedApps\<machineName>\metadataexplorer_war.ear\metadataexplorer.war\WEB-INF\classes*

See Also

For more about setting up Metadata Explorer, see Appendix A, 'Configuring Metadata Explorer'.

Turning on authentication for the Java Connector (for Metadata Explorer)

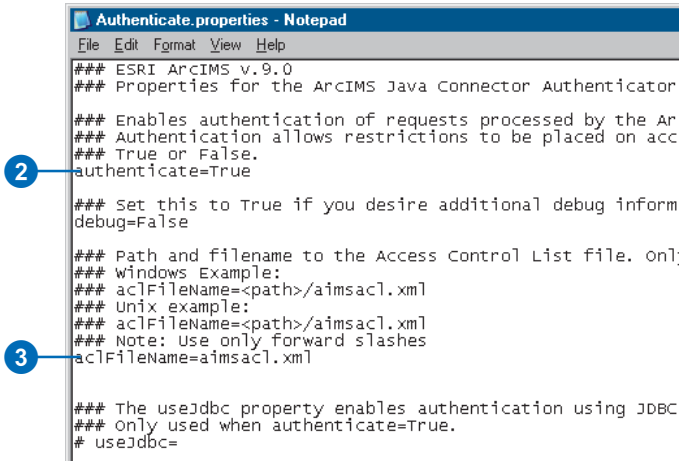
1. Edit the authenticate.properties file in a text editor. This file is usually located in the ArcIMS installation directory. See the Tip to the left for other locations.
2. Enable authentication.

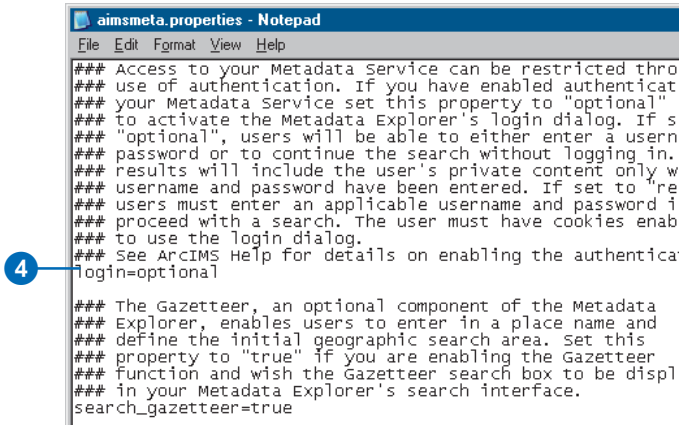
   `authenticate=True.`
3. Set the pathname to the ACL file. Slashes in the pathname must be forward slashes (/).

   `aclFileName=<full pathname to the ACL file>/aimsacl.xml`
4. To see the login page in Metadata Explorer, you also need to enable the login page by editing the aimsmeta.properties file in a text editor. This file is in the same directory as authenticate.properties.

   `login=optional`

   It can either be "required" or "optional" for users to log in. To disable the login page, set the login property to "off".
5. Stop and start the Web server for changes to the authenticate.properties and aimsmeta.properties files to take effect.





Troubleshooting publishing problems

If you're having trouble creating folders in or publishing metadata documents to a Metadata Service in ArcCatalog, the most common problem is that authentication is not working properly. Here are some things you can check.

Did you enable authentication in the Esrimap_prop file? This file is installed with your servlet engine. You should have set authenticate=True. Make sure you didn't edit the copy of the file in the ArcIMS installation directory.

Is the pathname to the ACL file correct? A common mistake is to use a backslash (\) to separate folder names when setting the aclFileName property in the Esrimap_prop file. You must use forward slashes (/).

Did you use the correct role? To publish metadata, you must connect with a username that has one of these roles: metadata_publisher, metadata_service_author, or metadata_administrator.

How did you name your service in Administrator? The service names in the ACL are case sensitive and must match the name you specified in Administrator.

Did you restart your servlet engine or Web server? Any changes you make to the Esrimap_prop file or a file-based ACL require you to restart your servlet engine, or Web server, when the servlet is native to the Web server.

Are you using Microsoft Internet Information Server? The Microsoft® Internet Information Server (IIS) must be configured for authentication. Whether you are using file- or JDBC-based authentication with either the Java or Servlet Connector, see the online Help topic 'Restricting access to services' and follow the links to the appropriate 'Enabling authentication' topic. In these topics, you'll find instructions for configuring IIS for authentication.

Is the Metadata Service taking a long time to publish documents? If you have automatic indexing enabled, you may experience long waits during publishing—especially if you have large metadata documents or many clients are publishing at the same time. To alleviate this problem, you can switch to manual indexing. See the section 'Manually indexing documents' later in this chapter for more information.

Are you publishing metadata documents with thumbnails greater than one megabyte? The thumbnail associated with a metadata document is stored as a binary large object (BLOB) in the database. The default ArcSDE BLOB size is approximately one megabyte. The thumbnails created by ArcCatalog are small. However, if your document doesn't contain a thumbnail created by ArcCatalog and it references an external graphic following the convention defined by the FGDC metadata standard, the first GIF or JPG file found will be loaded into the database as a thumbnail. If graphic files greater than one megabyte in size are published to your service, you will need to adjust ArcSDE parameters to accommodate them. See Appendix B, 'ArcSDE considerations', for more information.

Look for errors in the metadata log file. If none of the above suggestions work, you can look at the Spatial Server log files, specifically those for the Metadata Virtual Server. The log files are located in <ArcIMS Installation Directory\Server\Log on Windows and $AIMSHOME/Log on UNIX®. Look for log file names prefixed with "metadata" and examine the contents of the most recent one in a text editor.

Metadata for ArcIMS services

In ArcCatalog, you can create metadata for ArcIMS Image and Feature Services the same way that you can create metadata for shapefiles. A Metadata Service doesn't have to be running to create metadata for ArcIMS services. Once metadata has been created, it can be published to a Metadata Service run by another organization, such as the Geography Network. Or, you can publish the services' metadata to your own Metadata Service when you start one later.

As the ArcIMS administrator, you let someone create metadata for ArcIMS services by providing them with login information that grants them permission to do so. To author metadata for a service, they must connect to the ArcIMS Server in ArcCatalog as the appropriate user, select it in the Catalog tree, and click the Metadata tab. For more information about creating metadata, see Chapter 4, 'Publishing metadata in ArcCatalog'.

You create logins for metadata authors in the ACL. Create a user with the role metadata_service_author and explicitly list the ArcIMS services for which that user will be able to create metadata; include the Metadata Service to which the metadata will be published in the list of services. Alternatively, use an asterisk (*) to let the user create metadata for all Image and Feature Services and publish it to any Metadata Service. For example, a file-based ACL file might contain the following entries:

```
<USER name="author" password="author"
   services="MyMetadata, SanFrancisco, California"
   roles="metadata_service_author"  active="1"
   />
<USER name="author2" password="author2"
   services="*"  roles="metadata_service_author"
   active="1" />
```

In the above example, the "author" user can create metadata for the SanFrancisco and California services and publish that metadata to the MyMetadata service. The "author2" user can create metadata for any Image and Feature Services and publish it to any Metadata Service because an asterisk is specified for the services parameter.

If the same person will be creating metadata for all services, you'll probably want one user in your ACL with the metadata_service_author role with services="*". The same login should be used to create and publish metadata; otherwise, the metadata author may have trouble later updating and republishing the metadata. If you list specific services in the ACL for the user with the metadata_service_author role, be sure to include the Metadata Service in the services list.

In ArcCatalog, when metadata is created for file-based data, such as shapefiles, ArcCatalog creates an XML file and stores it on the hard disk with the data. Because ArcIMS services are accessed over the Web, metadata describing Image and Feature Services is stored on the ArcIMS server machine. The metadata XML files are stored in a Metadata directory on the machine hosting the Application Server. This directory is located at:

- **Windows:** <ArcIMS install directory>\AppServer
- **UNIX:** <ArcIMS install directory>/Middleware/Application_Server

The Metadata directory is created the first time metadata is created for a service using ArcCatalog. The Metadata directory will contain subdirectories—one for each service for which metadata has been created—with the same name as the service. This subdirectory contains the metadata for the service. For Image Services, there will only be one XML file in the subdirectory. For Feature Services, there may be several XML files; there may be one for the Feature Service itself and additional ones for each layer in the service that has metadata.

Once metadata for a service exists, it won't be deleted by ArcIMS. You must manually delete the folder for a service from the Metadata directory when the service is retired. Of course, if the service isn't running, its metadata can't be accessed in ArcCatalog even though its metadata exists on the server.

Defining a service's index

By default, Metadata Services are created with an auto_string index; the values of all elements in the published documents are indexed as text. However, to support date and numeric searches, values must be indexed numerically—either the auto or definition index types must be used. For ISO metadata documents, some codes must be indexed numerically to support searches. For the gazetteer to work in Metadata Explorer, place names must be indexed as text. Use the MDAlterIndex command to change a Metadata Service's index type.

With the auto index type, an element's data type is determined the first time it's encountered. If the first scale found is text, such as "1:10000", the element will always be indexed as text even if the scale is a number in other documents. Use a definition file to ensure that elements are indexed appropriately if documents aren't guaranteed to be authored correctly.

With auto_string and auto index types, you may want to exclude elements from the index to reduce its size and improve ►

Creating an index definition file

1. Create an index definition file with a text editor, for example, indexdef.txt.
2. Specify the XML elements and attributes handled by the definition file. Set the location path to an absolute XPath identifying an element or attribute, collectively referred to as tags, in the XML document. Relative paths, which include an asterisk, "*", can't be used, for example, to specify that all tags in a group should be indexed; tags are matched exactly using the XPaths specified in the definition file.
3. Define which tags to include in and exclude from the index. Set exclusion to "false" to include a tag's values or to "true" to exclude them. The default value is false, if exclusion isn't specified.

   For an auto_string index, only excluded tags must be defined. For a definition index, only included tags must be defined. For an auto index, the definition file should specify the tags to exclude and the tags that must be indexed numerically. ►

```
DESCRIPTION: Sample Index Definition

##TAG
LOCATION PATH: /metadata/idinfo/native
DESCRIPTION: FGDC Native Data Set Environment
EXCLUSION: TRUE
END

##TAG
LOCATION PATH: /gazetteer-entry
DESCRIPTION: Metadata Explorer Gazetteer Places
DATA TYPE: STRING
EXCLUSION: FALSE
END

##TAG
LOCATION PATH: /metadata/idinfo/descript/
abstract
DESCRIPTION: FGDC Abstract
DATA TYPE: STRING
EXCLUSION: FALSE
END

##TAG
LOCATION PATH: /metadata/dataIdInfo/descKeys/
keyword
DESCRIPTION: ISO Descriptive Keywords
DATA TYPE: STRING
EXCLUSION: FALSE
END

##TAG
LOCATION PATH: /metadata/dataIdInfo/descKeys/
keyTyp/KeyTypCd/@value
DESCRIPTION: ISO Descriptive Keyword Type
DATA TYPE: DOUBLE
EXCLUSION: FALSE
END

##TAG
LOCATION PATH: /metadata/idinfo/keywords/theme/
themekey
DESCRIPTION: FGDC Theme Keywords and Content
Theme
DATA TYPE: STRING
EXCLUSION: FALSE
END ►
```

performance for publishing, indexing, and searching—this is also accomplished with a definition file. You should exclude elements whose values aren't relevant for searching. For example, unless you have requirements to support searches based on feature, attribute, and spatial reference details, this information should be excluded from the index.

Using a definition file with the auto_string and auto index types would be the best option if you want some control over the index, but you still want the flexibility offered by those index types. They allow any XML document to be searched using full text, not just documents that contain the elements specified in a definition file. In addition, they let you provide a more complete body of text against which full text searches can be performed, and they give you the flexibility to make some changes to Metadata Explorer without rebuilding the index.

However, if performance is your primary concern, you should use the definition index type with a definition file that specifies only the few elements whose values must be indexed to support searching. For more information about index types and definition files, see 'Setting up your Metadata ►

4. For tags included in the index, specify how to index their values. Set the data type to "string" to index values as text, or "double" to index values numerically. The default value is string, if data type isn't specified.
5. Provide a description of the tag, if desired. This may help you edit the definition file later. Tag descriptions must not be longer than 64 characters.

   The sample definition file to the right represents a fragment of a typical definition file.
6. Save and close the file. You can save this file in any location; however, you'll probably want to save it somewhere in the ArcIMS Installation Directory.

```
##TAG
LOCATION PATH: /metadata/dataIdInfo/tpCat/
TopicCatCd/@value
DESCRIPTION: ISO Topic Category - Content Theme
DATA TYPE: DOUBLE
EXCLUSION: FALSE
END

##TAG
LOCATION PATH: /metadata/idinfo/timeperd/
timeinfo/sngdate/caldate
DESCRIPTION: FGDC Time Period of Content Single
Date
DATA TYPE: DOUBLE
EXCLUSION: FALSE
END
```

The sample definition file above represents a fragment of a typical definition file that handles both FGDC and ISO metadata documents that may be published to the Metadata Service. The Native Data Set Environment element is excluded because it isn't useful for searching. Title, abstract, and keywords are included in the text index. The date when the data was collected and the ISO codes specifying the topic category and keyword type, such as theme or place, are indexed numerically. For the gazetteer to work in Metadata Explorer, the place names must be included in the text index.

Changing how documents are indexed

1. From the operating system's command line, navigate to the <ArcIMS install directory>\Metadata\Commands directory.
2. Run MDAlterIndex.exe to change how published documents are indexed. ►

```
Usage: MDAlterIndex.exe <sde_server_machine>
<sde_instance> <database> <username> <password>
<xml_index_def> <index_type>
{table_name_prefix} {config_keyword} {verify}
{localcodepage}
```

The following example indexes documents published to a Metadata Service that uses the default prefix "imsmetadata":

```
mdalterindex sdeServer 5151 sde mdadmin
mdadmin.pass indexdef.txt
se_xml_index_definition # defaults
```

Service' at the beginning of this chapter.

After deciding which index type to use and creating a definition file, use the MDAlterIndex command to redefine and rebuild the Metadata Service's index. With this command you can provide an ArcSDE configuration keyword from the ArcSDE dbtune table. This command drops and rebuilds the ArcSDE XML side tables associated with the service, which are the foundation of its index. Use the keyword to set storage parameters for the side tables and properties of the text index in the database. For more information, see Appendix B, 'ArcSDE considerations'.

If you have more than one Metadata Service containing different sets of documents, problems may be encountered if you try to use different index types or definition files with each service. A tag's data type and description must be the same for all XML columns in an ArcSDE database. If service A uses the auto_string index type, you'll have trouble using a definition file with service B if tags are specified as "double", because it conflicts with the "string" data type already specified by service A. Service A's index must be altered to use the same definition file as ▸

Required arguments are identified by "<>", while optional arguments are identified by "{}". Provide all required arguments, and use "#" to skip optional arguments, if desired.

3. You may want to run MDTagConflictsList.exe to see if there were any problems indexing the documents after modifying the Metadata Service's index configuration.

sde_server_machine—The name of the ArcSDE server.

sde_instance—The ArcSDE instance or port number, for example, 5151. Or the instance name, for example, "esri_sde".

database—The ArcSDE database name. The database name is required for SQL Server and DB2 databases. For Oracle®, this information isn't appropriate; skip this parameter by specifying "#" instead.

username—The ArcSDE username.

password—The password associated with the username.

xml_index_def—The name and location of the index definition file. Either provide a full path to the location of the file or a path relative to the Commands directory. If there is a space in the path, enclose it within double quotes ("").

index_type—se_xml_index_auto_string, se_xml_index_auto, or se_xml_index_definition; se_xml_index_auto_string by default. Defines the method that will be used to index values in the published documents.

table_name_prefix—The table name prefix specified in the Metadata Service's configuration file. By default, "imsmetadata".

config_keyword—An ArcSDE configuration keyword. Configuration keywords let the database administrator define storage parameters for objects created in the database. For a Metadata Service, properties of its XML column and the text index may also be specified.

verify—"True" or "false"; true by default. With true, after the MDAlterIndex command is issued, you must type "y" or "n" on the command line to verify that you want to redefine and rebuild the Metadata Service's index. Set to false if you don't want to be prompted.

localcodepage—"True" or "false"; false by default. Set to true for Metadata Services where localcodepage is set to true in the configuration file. Localcodepage should be true for services that manage documents authored using 8-bit languages, such as Russian and Thai, or multibyte languages, such as Chinese and Japanese. See Appendix D, 'Publishing non-English metadata', for details.

service B. If two different files are used, tags that exist in both files must have the same data type. In contrast, if services A and B both have auto_string or auto indexes, their definition files can exclude different tags from the two indexes. The ArcSDE sdexml command with the describe and xi_stats options can help troubleshoot indexing issues.

After rebuilding a Metadata Service's index, you should check to see if there were any numeric indexing conflicts using the MDTagConflictList command. It produces a file that lists which elements in which documents couldn't be indexed numerically. For example, if a document has a date in the format "07/31/03" or "Spring 1985", those values can't be indexed numerically; a conflict will occur when incorrectly defined dates are encountered. Dates must be provided in the format YYYYMMDD to support searching. The Metadata Service administrator could contact the publisher and notify them that certain documents must be modified before they can be indexed and searched. A conflict won't exist when an element that is indexed as text contains a number; the number is merely indexed as text.

Checking indexing errors

1. From the operating system's command line, navigate to the <ArcIMS install directory>\Metadata\Commands directory.
2. Run MDTagConflictsList.exe to check if conflicts were encountered when indexing published documents. Required arguments are identified by "<>", while optional arguments are identified by "{}". Provide all required arguments, and use "#" to skip optional arguments, if desired.
3. You may want to set up an automated program—such as a cron job or at command—to run MDTagConflictsList.exe at regular intervals. After reviewing the output file, the owner of the documents can be notified of changes that should be made. The document owner would republish the documents after making the requested updates.

```
Usage: MDTagConflictList.exe
<sde_server_machine> <sde_instance> <database>
<username> <password> {table_name_prefix}
{output_file} {localcodepage}
```

The following example lists conflicts encountered when the documents published to a Metadata Service that uses the default prefix, "imsmetadata", are indexed:

```
mdtagconflictlist sdeServer esri_sde sde
mdadmin mdadmin.pass # logs\tagconflictlist.log
```

sde_server_machine—The name of the ArcSDE server.

sde_instance—The ArcSDE instance or port number, for example, 5151. Or the instance name, for example, "esri_sde".

database—The ArcSDE database name. The database name is required for SQL Server and DB2 databases. For Oracle, this information isn't appropriate; skip this parameter by specifying "#" instead.

username—The ArcSDE username.

password—The password associated with the username.

table_name_prefix—The table name prefix specified in the Metadata Service's configuration file. By default, "imsmetadata".

output_file—The name and location of the log file that records which values in which documents couldn't be indexed numerically. For example, a conflict occurs if scale is indexed numerically but the value is "1/24,000". A log file is always created if conflicts exist; by default, a "tagconflictlist.log" file is created in the Commands directory. Either provide a full path, or a path relative to the Commands directory; enclose it within double quotes if there are spaces.

localcodepage—"True" or "false"; false by default. Set to true for Metadata Services where localcodepage is set to true in the configuration file. Localcodepage should be true for services that manage documents authored using 8-bit languages, such as Russian and Thai, or multibyte languages, such as Chinese and Japanese. See Appendix D, 'Publishing non-English metadata', for details.

Manually indexing documents

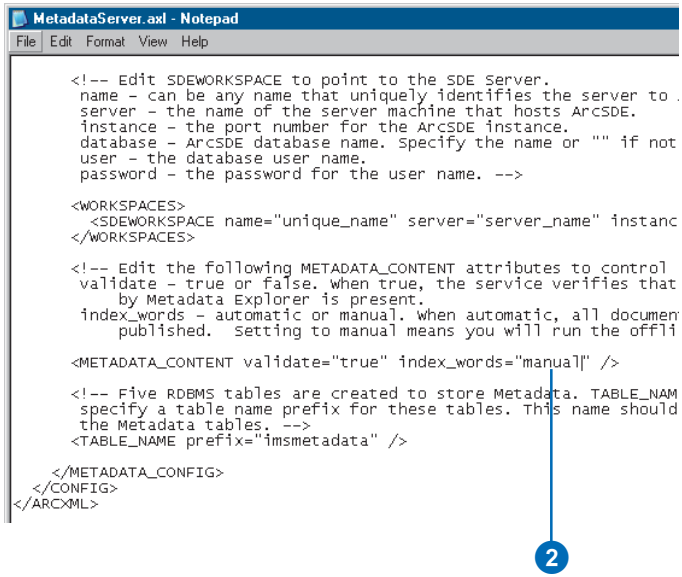
The amount of time it takes to publish and index a document depends on whether or not the Metadata Service uses automatic indexing, how the service is configured to index documents, and how much text the document contains. Indexing can affect search performance, especially if there are many documents to index or there is a heavy load on your database. To improve the performance of your service, switch from automatic to manual indexing in the service's configuration file. With manual indexing, you must use the aimsmetaindx command to index documents that have been published to the Metadata Service but haven't yet been indexed.

Metadata Services use the native capability of the DBMS to index text in the published documents. When you run the aimsmetaindx command, it will finish immediately and a log file will be written indicating if the indexing command was successfully sent to the database. However, the indexing process has just begun; the database may take some time to index the ▸

Setting a Metadata Service to index documents manually

1. In a text editor, edit the configuration file used to start your Metadata Service.
2. Set METADATA_CONTENT index_words to "manual".
3. Refresh your Metadata Service using Administrator or Manager.

   The Metadata Service will no longer index documents at the time they're published; now you must run the aimsmetaindx command to update the text index in the database.



published documents. Because indexing can impact the performance of your database, you may want to run this command during periods of low usage. For example, set up an automated program, such as a cron job or at command, to run it nightly. Alternatively, work with the database administrator to schedule updates to the text index using the database's tools; this may be particularly desirable with DB2 databases because there are some issues with using the aimsmetaindx command. See Appendix B, 'ArcSDE considerations', for more indexing information.

Indexing documents

1. From the operating system's command line, navigate to the <ArcIMS install directory>\Metadata\Commands directory.
2. Run aimsmetaindx.exe to index documents that have been published but not yet indexed. Required arguments are identified by "<>", while optional arguments are identified by "{}". Provide all required arguments, and use "#" to skip optional arguments, if desired.

   Running this command may impact the performance of your database. You may want to run it at times of low usage.

Usage: `aimsmetaindx.exe <sde_server_machine> <sde_instance> <database> <username> <password> {table_name_prefix} {logfile} {localcodepage}`

The following example indexes documents published to a Metadata Service that uses the default prefix "imsmetadata":

```
aimsmetaindx sdeServer esri_sde sde mdadmin
mdadmin.pass # logs\aimsmetaindx.log
```

sde_server_machine—The name of the ArcSDE server.

sde_instance—The ArcSDE instance or port number, for example, 5151. Or the instance name, for example, "esri_sde".

database—The ArcSDE database name. The database name is required for SQL Server and DB2 databases. For Oracle, this information isn't appropriate; skip this parameter by specifying "#" instead.

username—The ArcSDE username.

password—The password associated with the username.

table_name_prefix—The table name prefix specified in the Metadata Service's configuration file. By default, "imsmetadata".

logfile—The name and location of the log file, which indicates if the indexing command was successfully sent to the database. An error will occur if incorrect connection information was provided. A log file is always created; by default, an "aimsmetaindx.log" file is created in the Commands directory. Either provide a full path to another location for the file, or a path relative to the Commands directory. If there is a space in the path, enclose it within double quotes ("").

localcodepage—"True" or "false"; false by default. Set to true for Metadata Services where localcodepage is set to true in the configuration file. Localcodepage should be true for services that manage documents authored using 8-bit languages, such as Russian and Thai, or multibyte languages, such as Chinese and Japanese. See Appendix D, 'Publishing non-English metadata', for details.

Adding an administrative table to a service

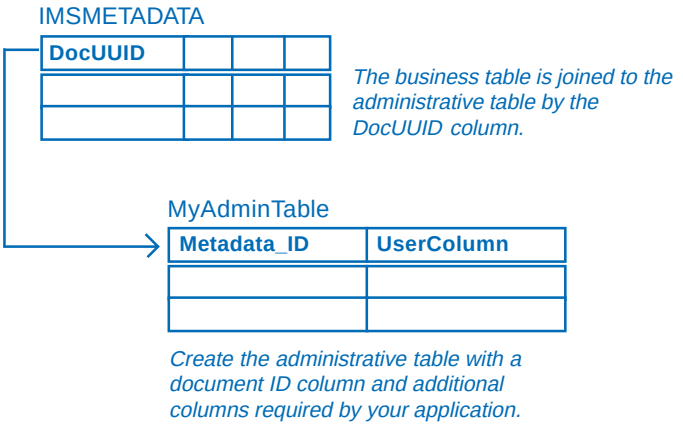
Administrative tables let you extend the information stored about published documents. They aren't required for general use, but they're useful to support custom applications for accessing and administering Metadata Services. An administrative table is simply an additional database table that is joined to the service's business table; see Appendix B, 'ArcSDE considerations', for details. You must create the administrative table before starting a Metadata Service. To add one to an existing service, create the table, modify a copy of the existing configuration file, start a new service, then republish existing documents to the new service. A record is inserted into the administrative table for each document when they are republished.

The administrative table must include a column, for example, Metadata_ID, that stores document identifiers matching the values in the Metadata Service's DocUUID column; it must be able to store strings 38 characters long. Add other columns as required by your application. This table must ▸

1. Create a table in the database using a SQL CREATE TABLE statement.
2. Define one column to contain strings 38 characters long. This column will contain the document identifiers.
3. Define any additional columns required by your application.
4. Create a copy of your Metadata Service's configuration file, then edit the copy in a text editor.

   You can use the MetadataServer.axl file distributed with ArcIMS. For example, on Windows, this file is located by default at C:\ArcIMS\AXL\Metadata.
5. Remove the comment characters enclosing the ADMIN_TABLE element: "<--" and "-->".
6. Specify the appropriate values for tablename, idcolumn, filter, insert, and columnname.
7. Create a new Metadata Service using the modified configuration file.

   When the service is created, a record is added to your administrative table representing the Metadata Service itself; without this record, the administrative table can't ▸




```
<ADMIN_TABLE
     tablename="MyAdminTable"
     idcolumn="Metadata_ID"
     filter="UserColumn='approved'"
     insert="INSERT INTO MyAdminTable
     (Metadata_ID, UserColumn) VALUES ('%s',
     'published')">
   <RESPONSE_COLUMN columnname="UserColumn" />
</ADMIN_TABLE>
```

tablename—The name of your administrative table. This name is case sensitive with some databases. Match the case used in the database for the table name and column names.

idcolumn—The name of the column in your administrative table that contains the document identifiers.

filter—The SQL where clause appended to all queries sent to the database by the Metadata Service. This lets you filter out documents that aren't ready for public consumption using values in your administrative table. For example, a search would find documents updated before April 1 and documents set to "approved". Use "filter='" if you don't want to filter documents, or remove the filter attribute entirely. ▸

be created in the database with a SQL create table statement; it must not be created using ArcCatalog or ArcSDE.

When a document is published, a record is added to the administrative table using the SQL insert statement defined in the configuration file. Specify the value "%s" to be stored in the Metadata_ID column; ArcIMS substitutes the document's identifier for this value. If default values weren't defined for other columns when the table was created, you can provide them in the insert statement as well. Values in other columns can be included in the ArcIMS response when documents are retrieved to support your application. Do this by defining RESPONSE_COLUMNs in the configuration file; be sure to provide default values for these columns—they must not contain null values.

ArcIMS won't delete records from an administrative table when documents are deleted from a Metadata Service. Your application must manage when records are removed.

See Also

For additional information, see the ArcXML Guide *and search for the ADMIN_TABLE element.*

be properly joined to the service's business table. The record representing the Metadata Service will only be added to the administrative table the first time a Metadata Service is created; this is why administrative tables can't be added to existing services. If problems are encountered starting the Metadata Service, delete any objects that were created from the database and check the service's log file to determine the cause of the problem. Modify the configuration file or recreate the administrative table as appropriate and try to start the service again.

8. Publish documents to the new Metadata Service.

   For every document published, a corresponding record is added to your administrative table using the insert statement specified in the configuration file.

insert—The SQL INSERT statement used to add records to your administrative table when documents are published. The name of the administrative table specified in tablename must appear in the INSERT statement between "INTO" and the first set of parentheses, "()". List columns in the administrative table into which default values will be inserted in the first set of parentheses, and list the default values in the second set of parentheses. The value "%s" must be inserted into the column specified in idcolumn.

columnname—The name of the column in the administrative table whose values will be returned in the response when documents are found in a Metadata Service. The column's name is added as an attribute to the METADATA_DATASET element in the response. You can include more than one RESPONSE_COLUMN element if you want to return the values from more than one column. Only columns containing strings can be response columns, and they must not contain NULL values.

Publishing metadata in ArcCatalog

4

IN THIS CHAPTER

- **Requirements for publishing metadata**
- **Preparing metadata for publication**
- **Publishing to a Metadata Service**
- **Managing published documents**
- **Finding data**

ArcCatalog is part of the ArcGIS Desktop suite of applications. It's where you go to create and update metadata, publish metadata to a Metadata Service, and organize and manage the published metadata documents. The ArcCatalog Search tool can also be used to browse and search the contents of a Metadata Service.

Because this chapter is focused on publishing, it doesn't cover some of the general concepts and tasks for creating and editing metadata. You can find that information in the book *Using ArcCatalog* and in the ArcGIS Desktop Help system. Among the ArcGIS Developer samples, which are available for download from *support.esri.com*, you may also find some helpful utilities, such as a spelling checker for metadata.

Requirements for publishing metadata

Before you can publish metadata to a service, you must have metadata. By default, when you attempt to view an item's metadata, ArcCatalog will automatically create it if it doesn't already exist and will update many of the metadata properties for you. You can use the metadata editors in ArcCatalog to add documentation, such as why the item was created.

You should take a step back and review the metadata's content before it is published. The usefulness of a Metadata Service depends entirely on the quality of the metadata available on that service. The same care should be taken with your metadata that you take when constructing a written document or the data that the metadata describes.

Ask yourself if you have provided enough information for people to find your document with a search. Once someone has found your document, will they be able to find the resource it describes and use it successfully? For example, make sure there is a link to an online order form and that you've described the values that each attribute column contains. Have any legal or security issues regarding your resource been documented? You should have your legal department review this information.

The review process will take time initially, but it can save you time in the long run. Use the experience to develop a good template that will become the foundation of all your future metadata efforts. When developing a series of metadata templates, think about what applies to all your datasets and what applies just to items developed for a specific project.

What can be published to a metadata service

ArcIMS Metadata Services don't require metadata to conform to any content standard. However, by default, a Metadata Service requires five pieces of information to be provided: the title, publisher, spatial extent, content theme, and content type of the resource. This information supports basic searches with the Metadata Explorer and ArcCatalog. You can add this information using either the FGDC or ISO metadata editor that comes with ArcCatalog.

When validation is enabled for a Metadata Service, the service checks a submitted document to see if it contains the required pieces of information. If any are missing, the document won't be published. ArcCatalog will show a report indicating what information is missing.

If validation has been disabled, you can publish any XML document with any content to a Metadata Service, but you may not be able to find the document with a search.

It is important to be clear that validation according to an ArcIMS Metadata Service has nothing to do with validation according to a metadata content standard. Metadata that contains information that is required according to a content standard, such as FGDC or ISO, may not contain the information that is required by the Metadata Service.

Defining theme and content type codes

To search for resources by theme and content type, the metadata must contain one of the theme or content type codes that the Metadata Service understands. Because the same codes must be used in all metadata documents, you will be able to publish metadata in any language to any Metadata Service, and its Metadata Explorer, which may be translated for any language, will be able to find your document.

The theme categories used by the Metadata Service are described below. More than one theme can be defined for a resource. For example, if your dataset describes soil tests, you might specify the theme codes for agriculture, geophysical, and environmental categories. Searches on any of those categories will find your dataset.

The descriptive name of each category appears below with bold text. These categories actually come from the ISO metadata standard. The ISO metadata editor presents you with a list of descriptive names for each category, but the editor actually saves the appropriate code into the metadata.

Because the FGDC metadata standard doesn't directly support theme codes, the FGDC editor isn't designed to handle them. To work around this situation, the Metadata Service looks for theme codes among the document's theme keywords. The code that must be typed into the FGDC metadata editor appears in parentheses below. These codes must be typed exactly as they are shown regardless of the language in which the rest of the metadata has been defined. For example, for a shapefile of cell phone antennas, you would type the theme code "utilitiesCommunication" into the FGDC metadata editor.

Administrative and political boundaries (boundaries)—Legal land descriptions.

Agriculture and farming (farming)—The rearing of animals or cultivation of plants. For example, resources describing irrigation, aquaculture, herding, and pests and diseases affecting crops and livestock.

Atmosphere, climatology, and meteorology (climatologyMeteorologyAtmosphere)—Atmospheric processes and phenomena. For example, resources describing cloud cover, weather, atmospheric conditions, climate change, and precipitation.

Biologic and ecologic (biota)—Naturally occurring flora and fauna. For example, resources describing wildlife, biological sciences, ecology, wilderness, sea life, wetlands, and habitats.

Business and economic (economy)—Economic activities or employment. For example, resources describing labor; revenue; commerce; industry; tourism and ecotourism; forestry; fisheries; commercial or subsistence hunting; and exploration and exploitation of resources, such as minerals, oil, and gas.

Cadastral and land planning (planningCadastre)—Land use. For example, resources describing zoning maps, cadastral surveys, and land ownership.

Cultural, society, and demographic (society)—The characteristics of societies and cultures. For example, resources describing natural settlements, anthropology, archaeology, education, traditional beliefs, manners and customs, demographic data, crime and justice, recreational areas and activities, social impact assessments, and census information.

Elevation and derived products (elevation)—Height above or below sea level. For example, resources describing altitude, bathymetry, digital elevation models, slope, and products derived from this information.

Environment and conservation (environment)—Environmental resources, protection, and conservation. For example, resources describing pollution, waste storage and treatment, environmental impact assessment, environmental risk, and nature reserves.

Facilities and structures (structure)—Man-made construction. For example, resources describing buildings, museums, churches, schools, hospitals, factories, housing, monuments, and towers.

Geological and geophysical (geoscientificInformation)—The earth sciences. For example, resources describing geophysical features and processes; minerals; the composition, structure, and origin of the earth's rocks; earthquakes; volcanic activity; landslides; gravity information; soils; permafrost; hydrogeology; and erosion.

Human health and disease (health)—Health services, human ecology, and safety. For example, resources describing human disease and illness, factors affecting health, hygiene, mental and physical health, substance abuse, and health services.

Imagery, basemaps, and land cover (imageryBaseMapsEarthCover)—Basemaps. For example, resources describing land cover, topographic maps, and classified and unclassified images.

Inland water resources (inlandWaters)—Inland water features, drainage systems, and their characteristics. For example, resources describing rivers and glaciers, salt lakes, water use plans, dams, currents, floods, water quality, and hydrographic charts.

Locations and geodetic networks (location)—Positional information and services. For example, resources describing addresses, geodetic networks, postal zones and services, control points, and place names.

Military and intelligence (intelligenceMilitary)—Military bases, structures, and activities. For example, resources describing barracks, training grounds, military transportation, and information collection.

Oceans and estuaries (oceans)—The features and characteristics of salt water bodies excluding inland waters. For example, resources describing tides, tidal waves, coastal information, and reefs.

Transportation networks (transportation)—The means and aids for conveying people and goods. For example, resources describing roads, airports and airstrips, shipping routes, tunnels, nautical charts, vehicle or vessel location, aeronautical charts, and railways.

Utility and communication networks (utilitiesCommunication)—Energy, water and waste systems, and communications infrastructure and services. For example, resources describing hydroelectricity; geothermal, solar, and nuclear sources of energy; water purification and distribution; sewage collection and disposal; electricity and gas distribution; data communication; telecommunication; radio; and communication networks.

The content type codes used by the Metadata Service are listed below; they are described in Chapter 2 in the section 'Finding data using Metadata Explorer'. Like the theme codes, they must be typed in exactly as they are shown. But unlike the theme codes, only one content type code can be defined for a resource.

- Live Data and Maps
- Downloadable Data
- Offline Data
- Map Files (supported in ArcCatalog 8.3 and later)
- Static Map Images
- Other Documents
- Applications
- Geographic Services
- Clearinghouses
- Geographic Activities (supported in ArcCatalog 8.3 and later)

When ArcCatalog automatically creates and updates metadata, the Geography Network and ISO synchronizers will add the same default content type code. For ArcIMS Image and Feature Services, the code "Live Data and Maps" is added to the metadata. For ArcMap map documents (.mxd or .pmf) and layer files, the code "Map Files" is added to the metadata. For all other items, the code "Downloadable Data" is added. Before you publish metadata, you should consider whether the default content type code is appropriate for your resource and change the value if necessary.

Document identifiers and republishing metadata

When metadata is published, a copy of your original document is added to the Metadata Service's database and indexed for fast searching. You can later republish the original document, for example, after the dataset and its metadata have been updated. Because only one copy of the document is stored with the service, all references to that document in all folders on the service will access the new version; each reference does not have to be updated.

This is possible because ArcCatalog and the Metadata Service use a unique identifier to refer to the metadata document. The unique identifier is generated within the metadata during the publishing process. The second time a document is published, the identifier already exists; the Metadata Service locates and replaces the original document. When a document is republished, the indexes are updated with the latest information in the metadata.

While it is possible to directly edit the published document using the ArcCatalog metadata editors, those changes are made only on the Metadata Service, not to your local version. If you later republish the original document, those changes will be lost.

If metadata was created with ArcCatalog 8.1 or earlier, it will contain an identifier that is not guaranteed to be unique. For example, if you imported a metadata template, the template and every metadata document based on that template would have the same identifier. If you publish those documents, the Metadata Service can't distinguish between them; the first document will be replaced by the second document, and so on. To solve this problem, use the MetaIDChecker utility provided with ArcCatalog. It can be found in the Utilities folder in the location where ArcGIS was installed.

Preparing metadata for publication

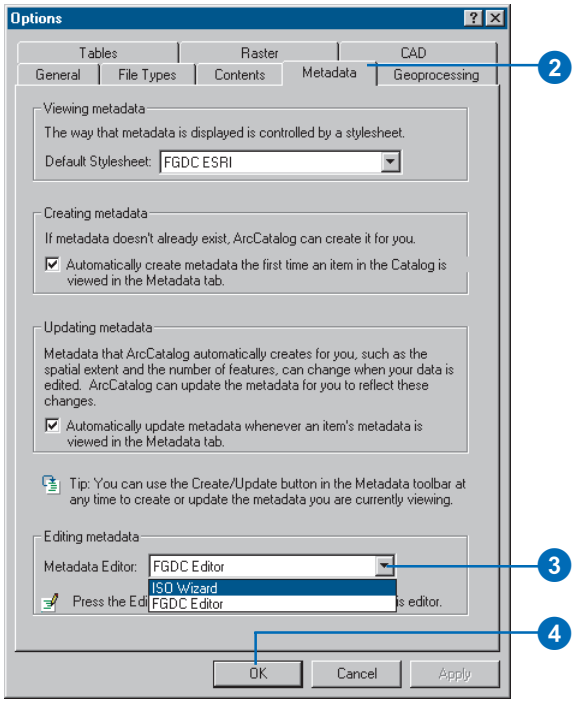
Before you can publish metadata, you need to define the pieces of information that are required by the Metadata Service: title, publisher, spatial extent, theme, and content type. Either the FGDC or ISO metadata editor, or both, may be used to do so.

The drawback of the FGDC editor is that you must type in the theme and content type codes; the ISO editor lets you pick them from a list. With the FGDC editor you can define any number of theme keywords—the requirement is for one of those keywords to come from the Metadata Service's content theme code list. For example, metadata for a soils dataset might have existing keywords: soil, clay, and corn. To publish the metadata, add the theme code "farming" to that list.

If you want to keep the content theme code separate from the existing set of keywords, add a new group of keywords and define that group's thesaurus as "Metadata Service Theme Categories". Alternatively, leave your FGDC theme keywords list the way it is and use the ISO ▸

Choosing a metadata editor

1. Click the Tools menu and click Options.
2. Click the Metadata tab.
3. Click the Metadata Editor dropdown arrow and click the editor you want to use.
4. Click OK.



editor to define the content theme code.

Similarly, when defining the content type code, the FGDC editor's Resource Description text box must contain one of the Metadata Service's codes. That text box can't contain additional information. If your metadata contains an existing resource description, you have a few alternatives: put the existing information in another metadata element; add a second distribution information section by clicking the plus sign under the General tab, then add the content type code; or use the ISO editor to define the content type. ►

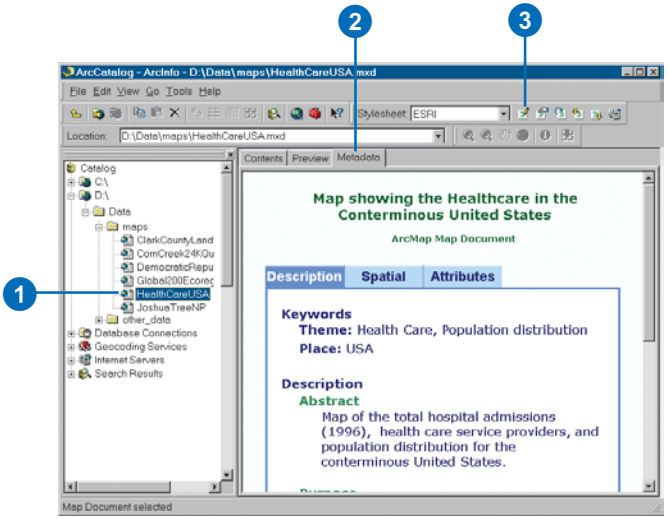
Tip

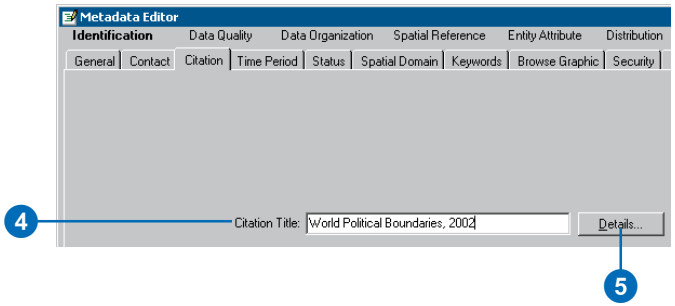
Creating metadata for an ArcIMS Image or Feature Service

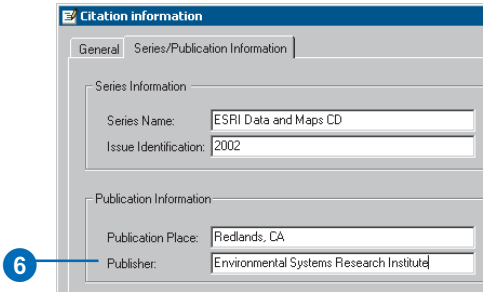
To create metadata for an ArcIMS Image or Feature Service, you must have permission to author metadata for that service. Contact the ArcIMS administrator for the appropriate username and password. Metadata that you create for an Image or Feature Service isn't automatically published to a Metadata Service that exists on the same ArcIMS server.

Adding required information with the FGDC metadata editor

1. In ArcCatalog, click the item in the Catalog tree whose metadata you want to edit. For example, click a shapefile.
2. Click the Metadata tab.
3. Click the Edit Metadata button on the Metadata toolbar.
4. To add a title, click Identification at the top of the editor, click the Citation tab, then type a descriptive title that clearly identifies the item.
5. Click Details to show the Citation information dialog box.
6. To add the publisher, click the Series/Publication Information tab, then type in the publisher. ►





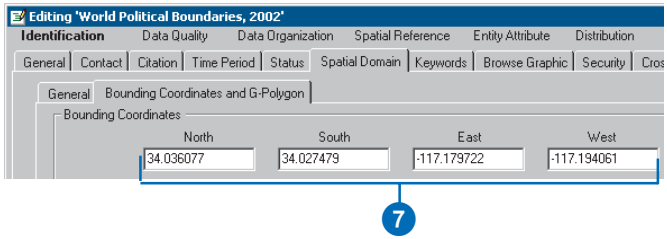


When the metadata is published, the Metadata Service looks for the required information in the appropriate FGDC and ISO metadata elements. If a title has been defined in both the FGDC element and the ISO element, the value in the FGDC element will be used.

The FGDC and ISO synchronizers both add a default title to the metadata—the FGDC synchronizer adds it to the appropriate FGDC element, the ISO synchronizer to the appropriate ISO element. After modifying the default title with the ISO editor, the default title will still exist in the FGDC element. Because of the Metadata Service's rules, the default FGDC title takes precedence and will be indexed as the title. A full text search based on words in the ISO title will still find the document, but the FGDC indexed title will be the one that appears in the Metadata Explorer's search results.

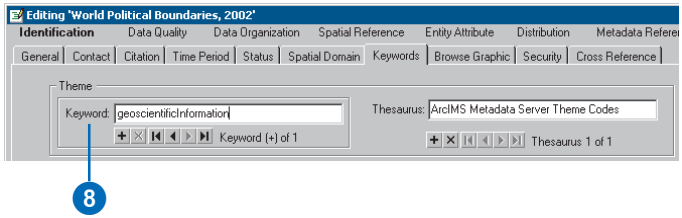
To work around this situation, you may want to define the Metadata Service's required information using both the FGDC and ISO editors. Or, if you only want to use the ISO metadata editor, you may want to turn off the FGDC metadata synchronizer. For information ►

7. To add the spatial extent, click the Spatial Domain tab and click the Bounding Coordinates and G-Polygon tab. The North, South, East, and West coordinates will be added automatically by ArcCatalog if the data's coordinate system has been defined. If these boxes are empty or if the item is nonspatial, such as a table, you will have to type in an appropriate extent in decimal degrees.



8. To define the item's theme, click the Keywords tab, click the plus sign (+) under Theme Keyword, then type one of the following codes into the Keyword text box:

biota
boundaries
climatologyMeteorologyAtmosphere
economy
environment
farming
geoscientificInformation
health
imageryBaseMapsEarthCover
intelligenceMilitary
inlandWaters
location
oceans
planningCadastre
society
structure
transportation
utilitiesCommunication ►



about how to do this, refer to metadata synchronizer information in the ArcGIS Desktop Help system.

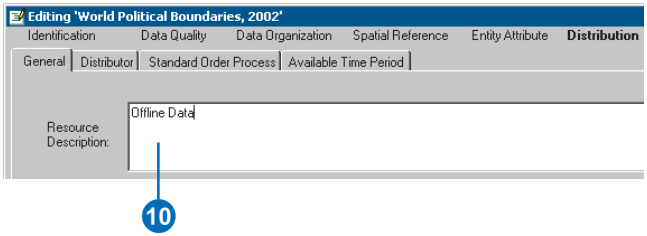
Before publishing metadata, you should remove the documentation hints that are added by the FGDC synchronizer to the elements that are mandatory according to the FGDC standard. Because these hints are stored as values of those elements, they will be in the metadata that is published to the Metadata Service. You can delete the hints manually with the FGDC metadata editor; the Help button in the editor shows where to find the mandatory elements. Alternatively, the ArcObjects™ developer samples include a Visual Basic® for Applications (VBA) script that can be used to delete the FGDC hints. Documentation hints are not added to ISO metadata elements.

While thumbnails aren't required, they can be useful. The Metadata Explorer will show thumbnails in the search results if they exist. Existing FGDC standard metadata that is published on the National Spatial Data Infrastructure (NSDI) Clearinghouse may have "browse graphics" rather than thumbnails. These are separate graphic files that are ▸

9. If the item falls into more than one theme category, click the plus sign again and type in another code.
10. To define the item's content type, click Distribution at the top of the editor, then type one of the following codes into the Resource Description text box:

    Live Data and Maps
    Downloadable Data
    Offline Data
    Map Files
    Static Map Images
    Other Documents
    Applications
    Geographic Services
    Clearinghouses
    Geographic Activities

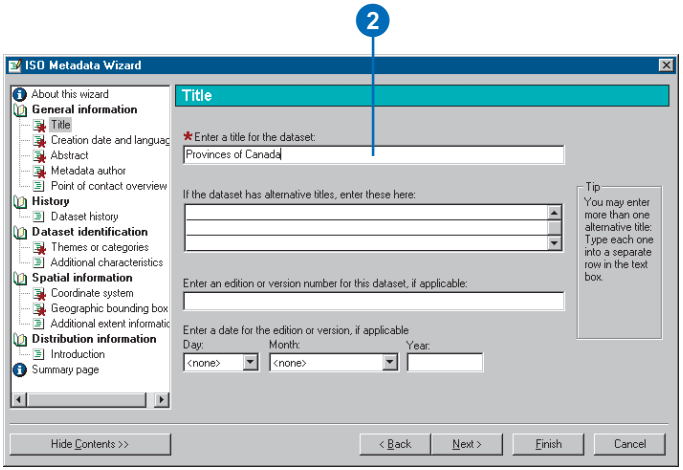
11. Click Save.



NOTE: The Map Files and Geographic Activities content type codes are only supported in ArcGIS 8.3 and later. These values are not supported in earlier versions.

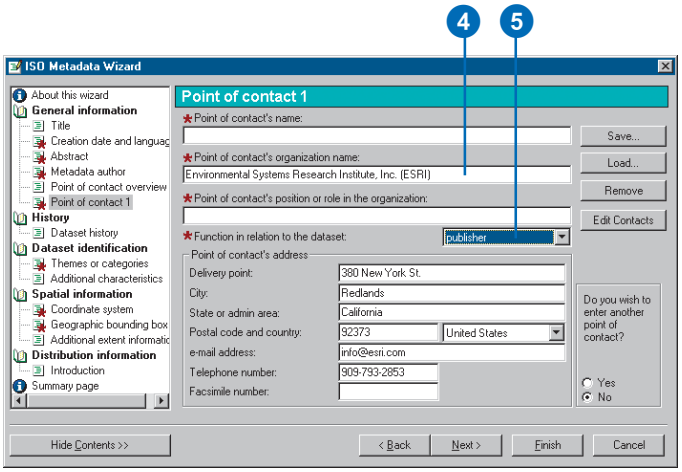
Adding required information with the ISO metadata editor

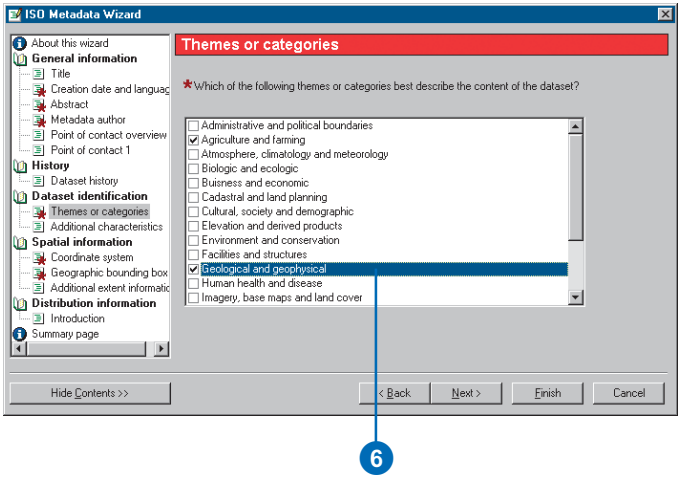
1. Click the Edit Metadata button on the Metadata toolbar.
2. To add a title, click the Title page in the table of contents and type a descriptive title that clearly identifies the item. ▸



referenced, rather than enclosed, in the metadata. Like thumbnails, a browse graphic may illustrate the dataset symbolized in various ways. If a thumbnail does not exist, the Metadata Service will see if a browse graphic exists; the first browse graphic in GIF or JPEG format that can be accessed will be loaded into the database as the document's thumbnail. Additional browse graphics will not be loaded.

3. Click the Point of contact overview page in the table of contents, click Yes in response to the second question to add a new point of contact, then click Next.
4. To add the publisher, click the last Point of contact page in the table of contents; it will be empty. Load a contact into the page from the contacts database, or type the publisher's name into the organization name text box.
5. To define the contact as the publisher, click the Function in relation to the dataset dropdown list and click publisher.
6. To define the item's theme, click the Themes or categories page in the table of contents and check all the categories that apply for this item. ▸



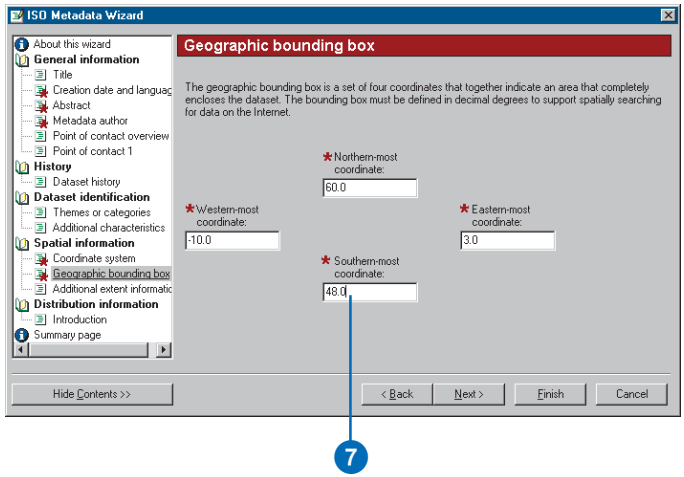


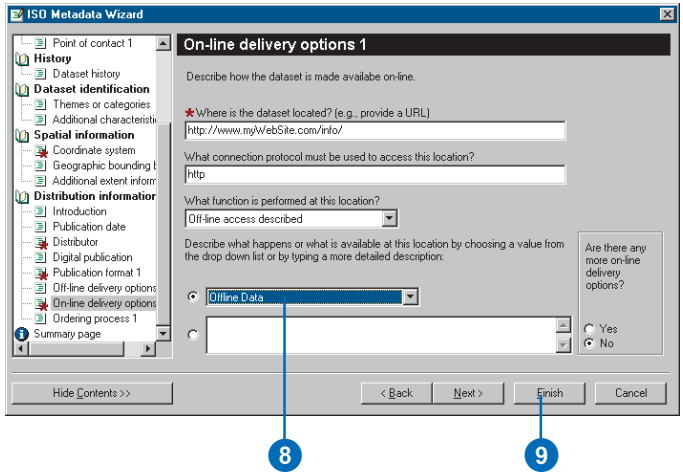
7. If the Geographic bounding box page appears in the table of contents, click that page and type in an appropriate spatial extent in decimal degrees. If the page doesn't appear, skip to step 8.

   The page doesn't appear in the table of contents if ArcCatalog automatically added the item's extent in decimal degrees, which happens when the item contains spatial data and its coordinate system has been defined.

8. Define the item's content type by clicking the On-line delivery options 1 page in the table of contents and click the appropriate content type from the resource description dropdown list.

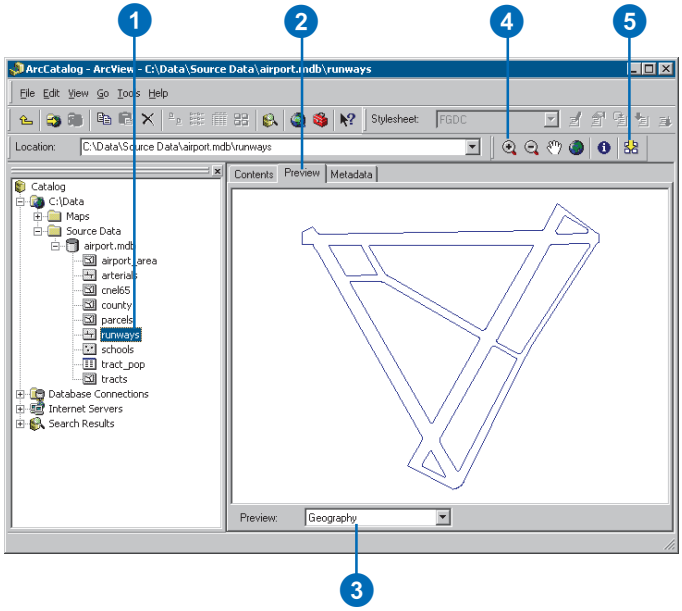
9. Click Finish.





Creating a thumbnail

1. In the Catalog tree, click the item for which you want to create a thumbnail.
2. Click the Preview tab.
3. Click the Preview dropdown arrow on the Preview tab and click Geography.
4. Click the Zoom In button on the Geography toolbar and zoom to the area that best represents the layer's contents.
5. Click the Create Thumbnail button.



Publishing to a Metadata Service

You can publish an item's metadata or any XML document to an ArcIMS Metadata Service using ArcCatalog. To do so, you must have permission to publish; the ArcIMS administrator will give you a username that has the role metadata_publisher. After connecting to the ArcIMS server with that login information, the icon of the Metadata Service to which you can publish will show a hand holding a pencil.

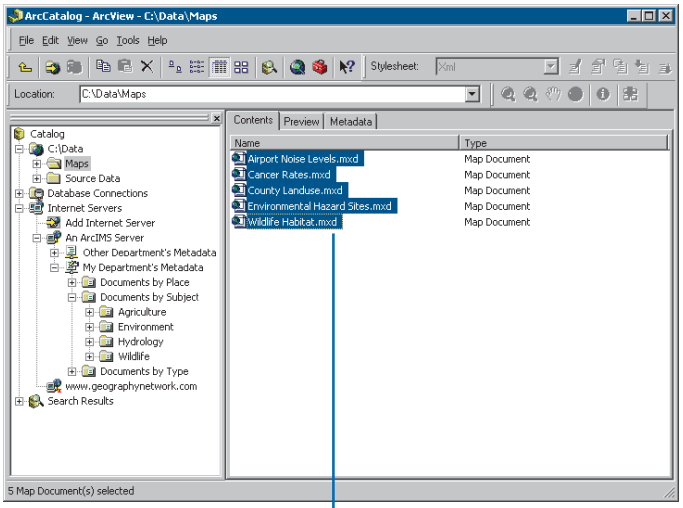
To publish an item's metadata, either copy and paste or drag and drop the item onto the Metadata Service. Metadata created for ArcIMS Image and Feature Services is not automatically published to a Metadata Service that exists on the same ArcIMS server; you must publish their metadata by dragging and dropping them onto the Metadata Service. When an item is pasted or dropped, its metadata is retrieved and a copy of it is added to the Metadata Service's database. The original metadata document remains in its initial location. If the original metadata changes, you must republish it to the Metadata Service.

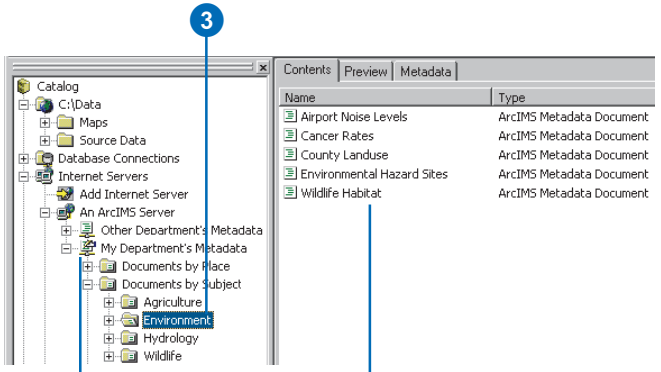
1. In ArcCatalog, click the item whose metadata you want to publish. To select more than one item in the Contents tab, hold down the Ctrl key while clicking the items.

   These datasets should already have metadata.

2. Click the Copy button.
3. Click the folder in the ArcIMS Metadata Service to which you want to publish the metadata.
4. Click the Paste button.

   The published documents will appear in the ArcIMS metadata folder. Publishing can take a few moments. If any problems were encountered, a report will appear indicating why a document could not be published. If any documents were previously published, the report will also show which documents were updated.





If you have permission to publish, the Metadata Service's icon shows a hand holding a pencil.

The published documents will appear in the ArcIMS metadata folder.

Managing published documents

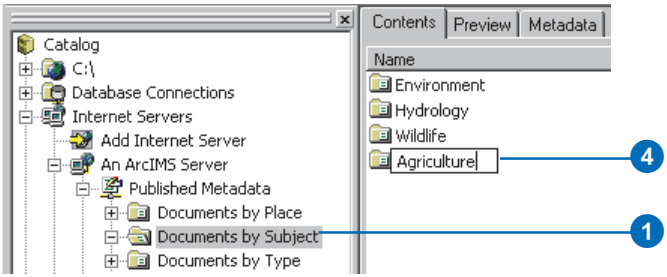
A Metadata Service is only useful if the published documents are easy to find. Like building an index for a book, creating a true catalog of documents requires you to think about the various ways that people will look for data.

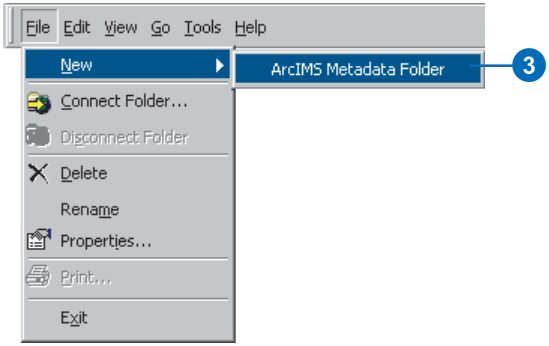
You can use ArcCatalog to construct a series of folders that organize the published documents in different ways. One document would appear in a different place in each hierarchy. A Metadata Service might contain one folder that organizes the documents by geographic location like a gazetteer, another folder that organizes documents by publisher, another by subject, and so on. When browsing the contents of the service in ArcCatalog or the Metadata Explorer, people will use each of those folders as a different entry point to drill down through the catalog and find what they need.

For example, a Metadata Service that publishes information about the western United States might have one folder that organizes documents by ▸

Creating a folder in a Metadata Service

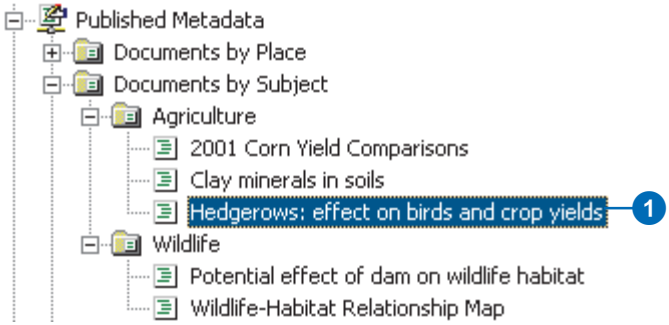
1. Click the Metadata Service or the folder within that service in which you want to create a new folder.
2. Click the File menu.
3. Point to New and click ArcIMS Metadata Folder.
4. Type a new name for the folder.
5. Press Enter.





Copying a published document

1. Click the document you want to copy. ▸



publisher and another that organizes by geographic location. Each organization that contributes information could publish its metadata into a folder that is just for them.

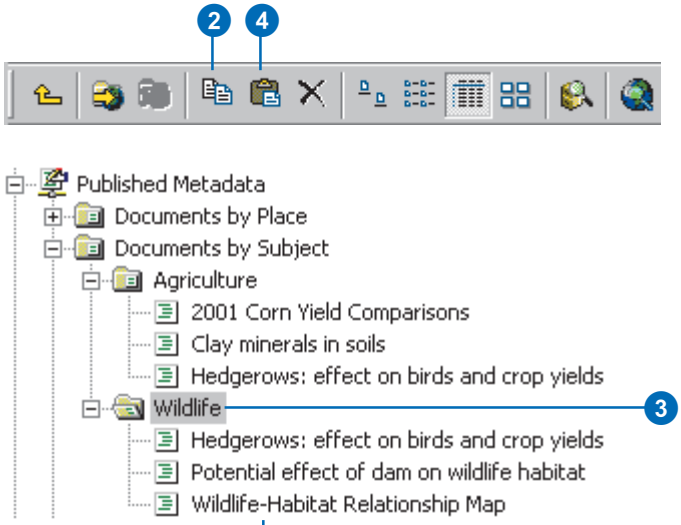
The librarian who oversees the Metadata Service could then copy the published document into the appropriate places in the other hierarchies. The folder that organizes documents by geographic location could contain one folder for each state, which in turn would contain one folder for each county, and so on. Metadata describing an image that covers a large area of California would be copied to the California folder. Metadata describing a report about election results in Riverside County would be copied to the Riverside County folder.

While a published document can be referenced by many folders, only one physical copy of the document is stored in the database. And no matter how many folders the published document appears in, the document will only appear once in a list of search results.

Similarly, when you delete a document on the Metadata Service you are only deleting the selected reference to that document. All references to a ▸

2. Click the Copy button.
3. Click the Metadata Service or folder within that service to which you want to copy the document.
4. Click the Paste button.

   The metadata document now appears in both folders on the Metadata Service.

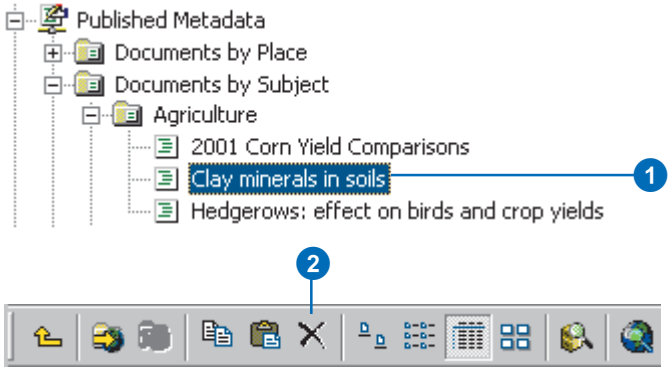


The metadata document appears in both folders.

Deleting a published document

1. Click the metadata document you want to delete.
2. Click the Delete button.
3. Click OK to confirm that you want to delete the document.

   If this is the last reference to the document on the Metadata Service, you must confirm that you want to remove the document from the service.
4. Click Yes to remove the document from the service.



document must be deleted to remove that document from the service. When you delete the last reference to a document, a message will appear confirming that you want to remove the document from the Metadata Service.

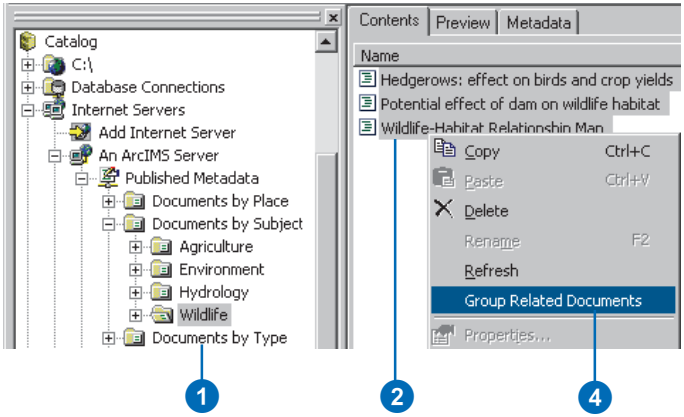
Another way to make it easier to browse through the contents of a Metadata Service is by establishing relationships between different documents. For example, you might establish a relationship between metadata for a coverage describing electoral regions and metadata describing a paper that analyzes election results. Or you might establish a relationship between a paper map and the aerial photographs that cover the area on that map. When you look at the contents of a document in the Details tab of the Metadata Explorer, links to any related documents are listed at the bottom. In ArcCatalog you can view and manage the list of related documents with the Properties dialog box.

Anyone who has permission to publish to a Metadata Service can create folders on that service, copy documents, and establish relationships between documents on the service. Publishers also have permission to rename, edit, and delete the ▸

Establishing a relationship between documents in the same folder

1. Click the folder containing the related documents in the Catalog tree.
2. Press the Ctrl key while clicking the related documents in the Contents tab.
3. Right-click the selected documents.
4. Click Group Related Documents.

   When viewing the contents of a document in the Metadata Explorer, its related documents are listed at the bottom.



When looking at a document's Details in the Metadata Explorer, its related documents are listed at the bottom.

documents that they own and to make their documents private. To view a private document, a person must log in to the Metadata Service with the appropriate username and password.

The librarian who manages the catalog of documents on a Metadata Service should be granted administration privileges by the ArcIMS administrator. The metadata administrator can view and delete all documents that are available on the service.

Tip

Renaming published documents

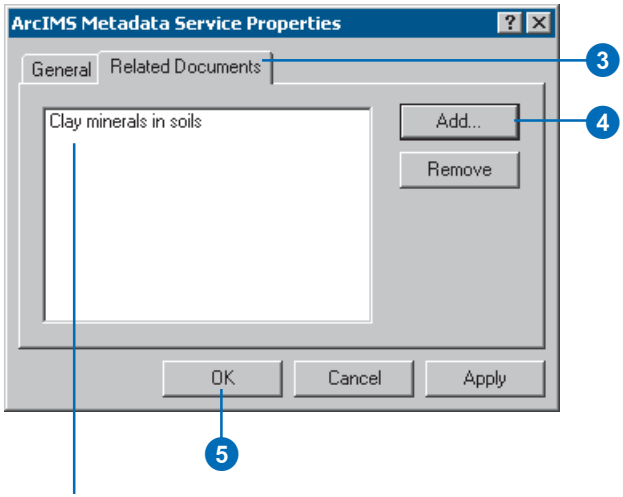
When metadata is published, the title within the metadata is used as the name of the document. You can rename a document in ArcCatalog, but doing so doesn't change the title stored within the metadata.

Establishing a relationship between documents in different folders

1. Click the document with which you want to establish a relationship.
2. Click File and click Properties.

   The ArcIMS Metadata Service Properties dialog box appears.
3. Click the Related Documents tab.
4. Click Add, navigate to and select the related document, then click Add.
5. Click OK.

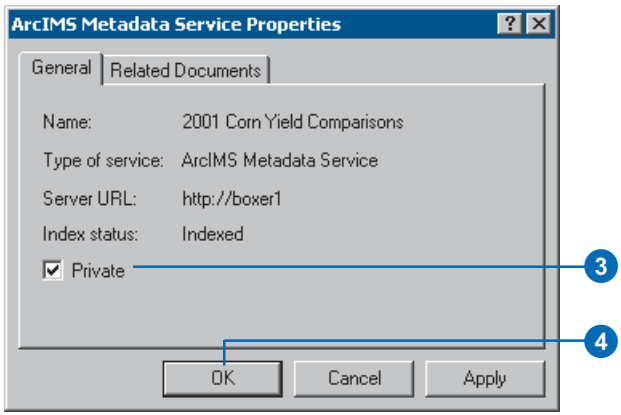
   The document appears in the related documents list.



The related document appears in the list.

Making a published document private

1. Click the document that you want to be private.
2. Click File and click Properties.
3. Check Private on the General tab.
4. Click OK.



Finding data

When you need to find data to use in your maps, you can use ArcCatalog or the Metadata Explorer to browse through or search the contents of a Metadata Service. The Metadata Explorer runs in a Web browser; it is described in Chapter 2, 'Finding data in Metadata Explorer'.

Before you can browse or search a Metadata Service's contents in ArcCatalog, you need to connect to the ArcIMS server. Once you establish a connection, you can browse the server's contents just like you would browse through folders on your local computer. You can look at published documents in the Metadata tab. If a published document describes an ArcIMS Image or Feature Service, its icon has a little globe; in addition to looking at the document, you can explore the ArcIMS service's data in the Preview tab.

The ArcCatalog Search tool lets you create more complex searches than the Metadata Explorer. You can search for datasets that are available in a specific format, such as shapefiles or AutoCAD® drawings. With geographic searches, you can specify that ▸

Browsing published metadata documents in ArcCatalog

1. In ArcCatalog, double-click GIS Servers and double-click Add ArcIMS Server.

   The Add ArcIMS Server dialog box appears.

2. Type the URL of the ArcIMS server to which you want to connect.

3. If appropriate, check Show secure services and type your username and password.

   Usernames and passwords for accessing secure Metadata Services are defined in the ACL file.

4. Click OK.

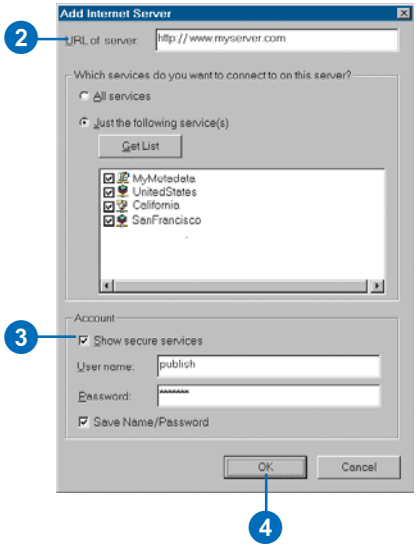
   The ArcIMS server appears in ArcCatalog.

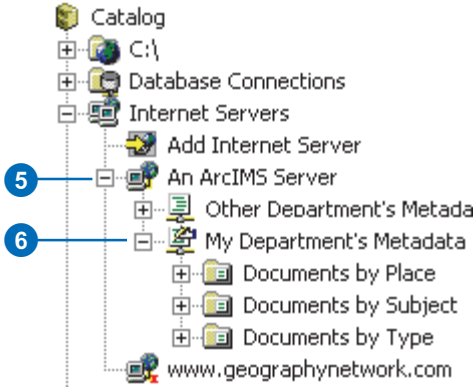
5. Double-click the ArcIMS server to show its contents.

   If you have permission to publish to a Metadata Service, its icon has a hand with a pencil.

6. Double-click a Metadata Service to show its contents.

   The folders and documents contained by the Metadata Service appear in ArcCatalog.





you want to find datasets that are either completely within or that overlap the specified area. You can search using dates, such as when the document was modified, and using keywords, such as the name of the person who created the data. Unfortunately, date searches won't find information defined in ISO metadata elements in the published documents, but other types of searches will work.

Tip

Searching in ArcCatalog

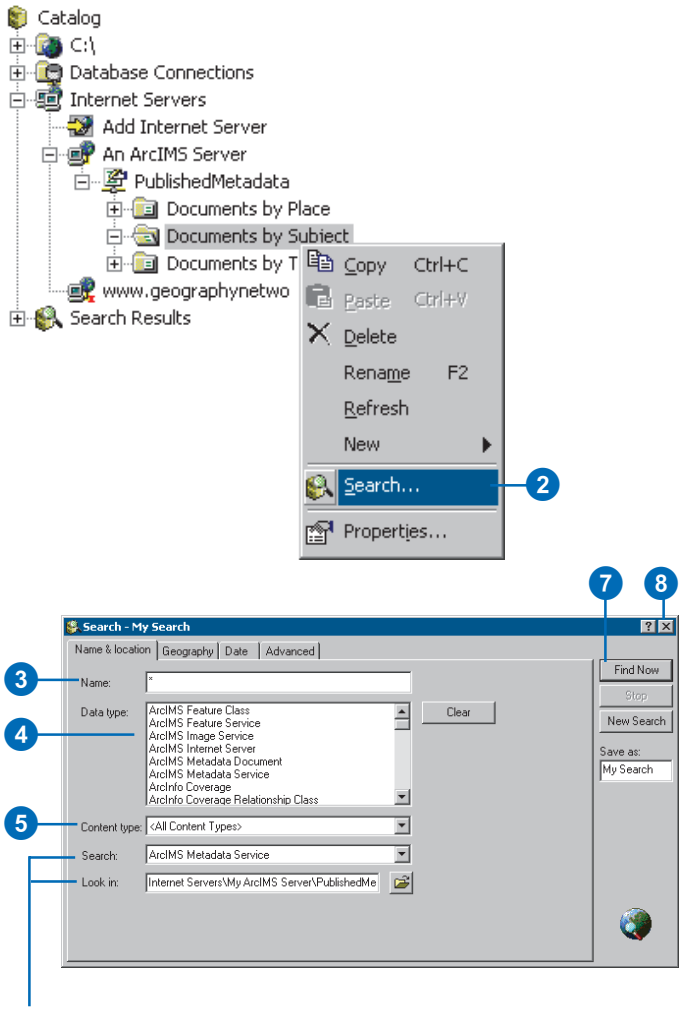
You can also start an ArcCatalog search by clicking the Search button on the standard toolbar. If you start the search this way, choose ArcIMS Metadata Service in the Search dropdown list to take advantage of the Metadata Service's indexes. Click the Browse button next to the Look in text box and navigate to the ArcIMS Metadata Service you want to search.

Searching published metadata documents in ArcCatalog

1. In ArcCatalog, right-click the folder in the Metadata Service in which you want to start looking for data.
2. Click Search.

   The Search dialog box appears. The type of search and the place to start looking are set for you.
3. To search by name, type all or part of the name you're looking for into the Name text box. Use an asterisk (*) to represent one or more letters.
4. To search for specific types of data, such as shapefiles, press the Ctrl key while clicking the types of items you want to find in the Data type list.
5. To search by content type, such as Live Data and Maps, choose the appropriate type from the Content type dropdown list.
6. Use the other tabs to define additional search criteria.

   For example, with the Geography tab you can specify the geographic location in which to search. With the Advanced tab you can look for a word in a ▸



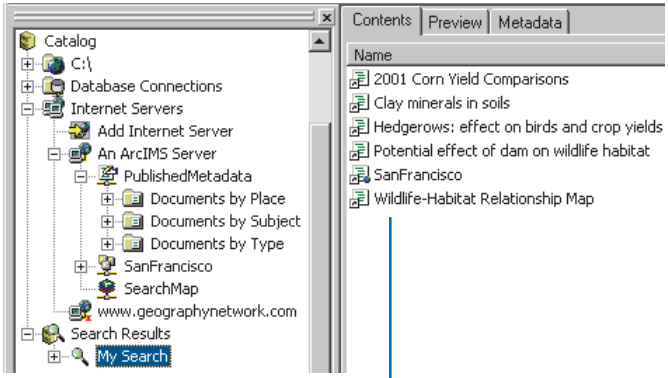
When you start a search by right-clicking an ArcIMS Metadata Service, the type of search and the place to start searching are set for you.

specific metadata element, such as the abstract, or do a full text search to look for the word anywhere in the metadata. To search by content theme, pick Data theme in the Metadata element dropdown list on the Advanced tab.

7. Click Find Now.

   The search is saved in the Search Results folder and is selected in the Catalog tree. As metadata documents are found that satisfy the search criteria, shortcuts to those documents are added to the Search Results list.

8. Click the Close button in the upper right corner of the Search dialog box.



As items are found that satisfy your search criteria, shortcuts to those items are added to the Search Results list.

Connecting Z39.50 clients

5

IN THIS CHAPTER

- **Incorporating Z39.50 into your metadata system**
- **What is Z39.50?**
- **Getting started**
- **Changing the service's port**
- **Setting the default Metadata Service**
- **Setting log options**
- **Protocols and restricting access to your metadata**
- **Setting a communication protocol**
- **Limiting access to your metadata**
- **Customizing the connector**
- **Controlling the format of responses**
- **Available Z39.50 resources**

The search and retrieval of information in online databases is not a new concept. Libraries, government agencies, and similar organizations have been organizing metadata into searchable, distributed catalogs for decades. Many of these groups employ the Z39.50 standard communications protocol for information discovery when creating their systems and search mechanisms. This standard is widely used by library systems and research institutions, including the Library of Congress. This standard is also used by the FGDC Geospatial Data Clearinghouse, which is part of the National Spatial Data Infrastructure. The goal of the NSDI is to facilitate the sharing and use of spatial data both inside and outside the U.S. government.

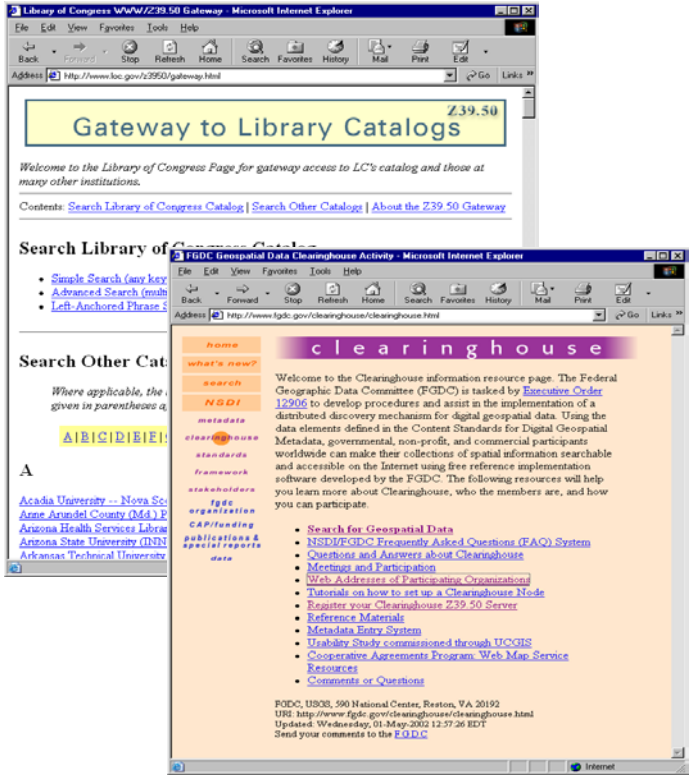
With the ArcIMS Z39.50 Connector, your ArcIMS Metadata Service can participate in an existing library system or the FGDC Clearinghouse. Documents available from your Metadata Service can be searched by any Z39.50 client application.

Incorporating Z39.50 into your metadata system

Adding the Z39.50 Connector to your Metadata Service can make your metadata available to a wider audience with little effort. Many university, college, and public library systems and research institutions use Z39.50 clients to let people search and browse their holdings. This connector lets you seamlessly integrate metadata describing your GIS resources into these existing search and discovery systems, and best of all, these users won't need to learn how to use a new application.

Once you've made your metadata available to Z39.50 clients you can:

- Expose your metadata to users of the NSDI Clearinghouse, which can be searched from the Geography Network Explorer. You must register your Metadata Service to participate as a server, or node, in the NSDI Clearinghouse (http://www.fgdc.gov/clearinghouse/clearinghouse.html).
- Increase visibility for your metadata by registering with the Library of Congress 'Gateway to Library Catalogs' (http://www.loc.gov/z3950/gateway.html).
- Leverage your organization's existing Z39.50 clients, whether they are Web gateways or applications, by allowing them to search Metadata Services exposed by the Z39.50 Connector.
- Implement a complete Z39.50 system that meets the specific information retrieval needs—geospatial or otherwise—of your users and is closely integrated with the Metadata Service you provide. Note: Z39.50 development tools are not provided with ArcIMS.



The Library of Congress and Federal Geographic Data Committee are just two of the many organizations that host Web gateways that enable Z39.50 clients to search registered servers and retrieve information.

What is Z39.50?

The Z39.50 protocol—formally named Information Retrieval (Z39.50); Application Service Definition and Protocol Specification, ANSI/NISO Z39.50-1995—specifies the rules of communication between a search client and an online database. The protocol was initially produced by the National Information Standards Organization (NISO), the standards development group serving the fields of publishing, libraries, and information services. The protocol's name came about because NISO was once the Z39 committee of the American National Standards Institute (ANSI) and it was the fiftieth standard developed by that committee.

The Z39.50 protocol formalizes rules for querying, retrieving records, and asking for those records in a particular format. It doesn't specify implementation details for clients and servers, but rather the language that clients and servers are expected to speak.

Today, the United States Library of Congress hosts the Z39.50 Maintenance Agency, and a group called the Z39.50 Implementors Group (ZIG) continues the standard's development. The specification for the protocol is available at http://lcweb.loc.gov/z3950/agency/ along with other Z39.50 resources.

How the protocol works

Z39.50, just like ArcIMS, follows a client/server model—one computer requests something from another computer, which then processes the request and returns a response. This is a simple proposal when both client and server use the same language, terminology, and database organization, but not that easy in real-world scenarios.

Z39.50 establishes a method of communication that supports information retrieval in a distributed environment. As long as a server understands Z39.50 requests, a client can request information from its database without prior knowledge of its organizational structure. Because the standard doesn't specify implementation details, the server can translate the query into whatever form is appropriate for its database.

Servers generally implement a profile of the Z39.50 protocol. A profile is a subset of the functions available in the Z39.50 specification that support a particular type of application, function, or community. Libraries generally use the bibliographic, BIB-1, profile. Online GIS repositories generally follow the GEO Profile. Established by the FGDC, the GEO Profile extends the bibliographic profile to support searching for geospatial metadata. The GEO Profile is the language used to search servers participating in the NSDI Clearinghouse.

How the Z39.50 Connector works

The ArcIMS Z39.50 Connector operates as the server in the Z39.50 framework; it supports the GEO Profile. Requests sent by Z39.50 clients are translated from the GEO Profile into ArcXML, the language used by ArcIMS. The ArcXML request is then sent to a Metadata Service and a response is generated. The Z39.50 Connector formats the response according to the client's request and sends it back to the Z39.50 client. This process is discussed in more detail in 'Customizing the connector' later in this chapter. For more information on ArcXML, refer to *ArcXML Programmer's Reference Guide*.

Getting started

Getting started with the Z39.50 Connector is easy. You simply need to install the connector and start the service. Refer to *ArcIMS Installation Guide* for step-by-step instructions on installing the connector. Once the service has been started, check the connector's console window to determine if it has started successfully. Alternatively, you can look for this information in the Z39.50 Connector's log files.

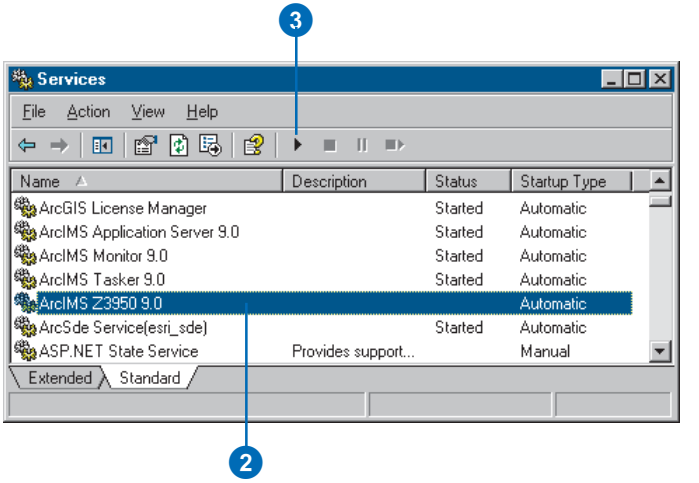
For example, if your organization has an existing Z39.50 server, such as ISite, running, the console window will show a conflict on port 210. Port 210 is typically used by Z39.50 servers to communicate, and is the default port used by the Z39.50 Connector. When more than one server attempts to use port 210, a conflict occurs. To resolve this problem, either stop the other Z39.50 service or modify the port used by one of the services. If you use ISite and you choose to stop that service, be aware that its service is set by default to start automatically; change that service's startup type to manual to avoid having a conflict on port 210 each time the system is restarted. ►

Starting the service on Windows

1. On Windows NT, click Start > Settings > Control Panel > Services.

   On Windows 2000, click Start > Settings > Control Panel > Administrative Tools > Component Services, and click Services on the Tree tab.

   On Windows XP, click Start > Control Panel > Performance and Maintenance > Administrative Tools > Component Services, and click Services on the Tree tab.
2. Click the ArcIMS Z3950 9.0 service in the Services window.
3. Click Start.



Starting the service on UNIX/Linux

1. Execute the file z3950service.sh.
2. Start the ArcIMS Z3950 9.0 service listed in the Services window.

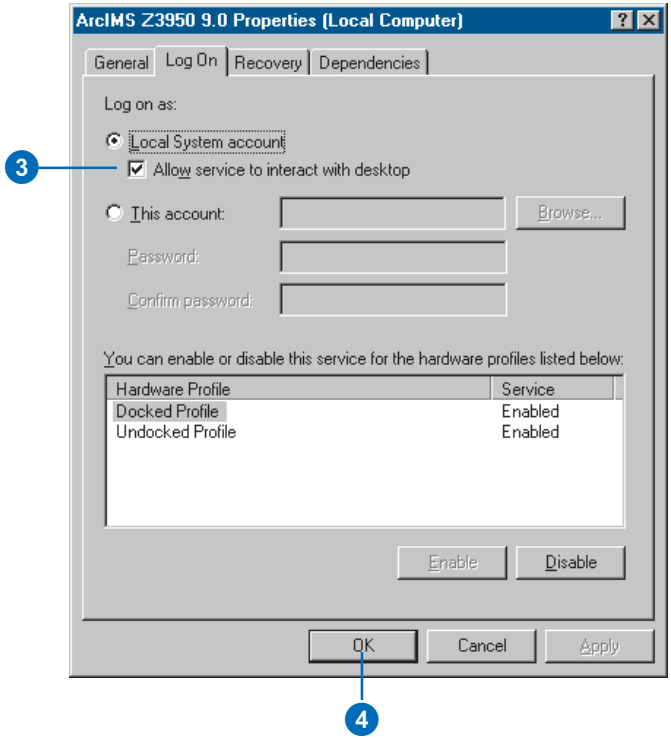
Once the Z3950 service has started successfully and the Z39.50 Connector has been configured to access a Metadata Service, Z39.50 clients can search and browse the available documents. You can test this using a Z39.50 client application. If you don't have one, see 'Available Z39.50 resources' later in this chapter for information about where to get one.

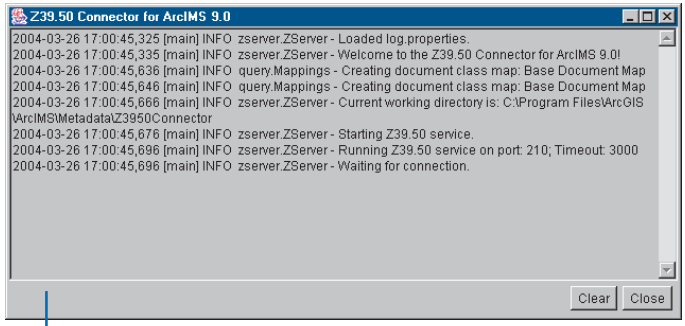
Leave the console window for the Z3950 service open to monitor its status. If you close the window, you can always open it again later; on a Windows system you would do this by stopping and restarting the service.

Showing the console window on Windows

1. In the Services window, right-click the ArcIMS Z3950 9.0 service and click Properties.
2. In the ArcIMS Z3950 9.0 Properties dialog box, click the Log On tab.
3. Under the Local System account option, check Allow service to interact with desktop.
4. Click OK.
5. Stop and restart the ArcIMS Z3950 9.0 service in the Services window.

   The Z39.50 Connector for ArcIMS 9.0 console window appears. Messages in the window indicate whether or not the service is running successfully.





The Z39.50 Connector for ArcIMS 9.0 console window appears each time the service is started. Messages indicate if the service is running successfully.

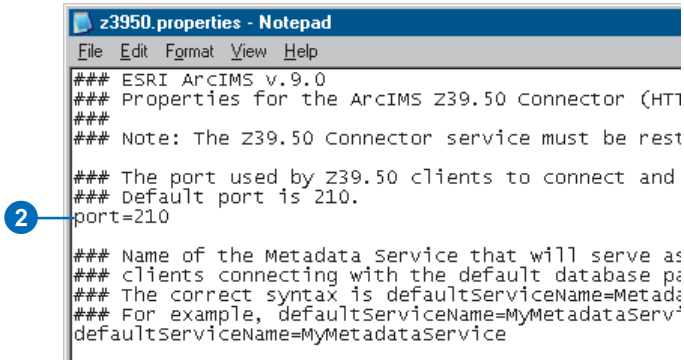
Changing the service's port

Z39.50 servers typically communicate with Z39.50 clients using port 210; this is the default port used by the Z39.50 Connector. If you have more than one Z39.50 server running and both of them are trying to communicate on port 210, a conflict will likely arise. For example, you might currently be using an ISite server to make your metadata available on the NSDI Clearinghouse.

If there is a conflict on the port used by the Z39.50 Connector, the ArcIMS Z3950 service won't start. To resolve this problem, either stop the other service that is trying to use the port or change the port used by one of the services. To change the port used by the Z39.50 Connector, set the port property in the z3950.properties file to the port that should be used. The z3950.properties file is always located in the Z39.50 Connector's install directory at <ArcIMS install directory>\Metadata\Z3950Connector. If you choose to stop the other service, check its startup type; if the service is set to start automatically, there will be a conflict on the port each time the system is restarted.

1. Edit the z3950.properties file in a text editor. This file is located in the Z39.50 Connector's install directory.
2. The default port number for typical Z39.50 servers is 210, but it can be modified depending on your network configuration.

   `port=210`
3. Save and close the file.
4. Stop and restart the ArcIMS Z3950 9.0 service in the Services window.



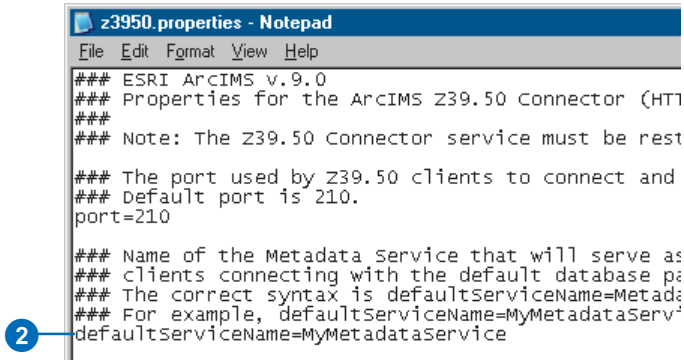
Setting the default Metadata Service

Once the Z3950 service is running successfully, the next step is to configure the Z39.50 Connector to access a Metadata Service. In Z39.50 terms, a client accesses a database that is available on the server. For the Z39.50 Connector, a Metadata Service is a database.

Z39.50 clients can access specific databases, or Metadata Services, by name. If users don't know the name of a Metadata Service available on your server, they can request to access a default database. To specify a Metadata Service as the default database for the Z39.50 Connector, set the defaultServiceName property in the z3950.properties file to the name of the appropriate Metadata Service. If a Z39.50 client requests access to a database other than the default and the database name specified by the client matches the name of another Metadata Service available from the ArcIMS server, the client will be able to access that service.

1. Edit the z3950.properties file in a text editor. This file is located in the Z39.50 Connector's install directory.
2. Edit the defaultServiceName property to match your Metadata Service name.

   `defaultServiceName=MyMetadataService`
3. Save and close the file.
4. Stop and restart the ArcIMS Z3950 9.0 service in the Services window.



Setting log options

A variety of options are available to monitor the status of the Z39.50 Connector and to troubleshoot issues that may arise. Logging options are set by modifying the log.properties file; this file is always located in the Z39.50 Connector's install directory at <ArcIMS install directory>\Metadata\Z3950Connector.

Log information can be sent to a variety of locations, including standard output, the console window, a log file, and a separate debug log file. By default, information is logged to all locations except the debug log file. To stop or start one of the logs, comment or uncomment the appropriate section in the log.properties file. Usually it's most convenient to check the service's startup status using the console window; you can close it when you know the service is running. To learn how to open the console window on Windows, see the section 'Getting started' earlier in this chapter.

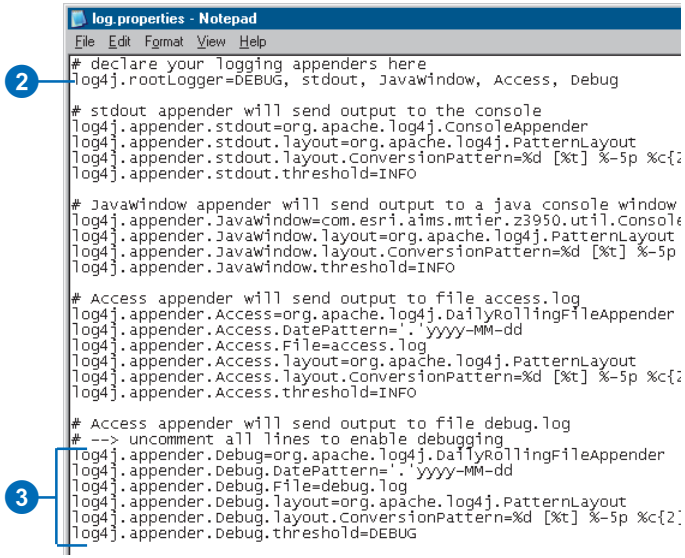
Four levels of logging can be used to record different amounts of information: ERROR, WARN, INFO, or DEBUG. The level of logging that is available to all logs is set on the first line of the log.properties file; by default, ▸

Starting a debug log file

1. Edit the log.properties file in a text editor. This file is located in the Z39.50 Connector's install directory.
2. Change the overall level of logging to the debug level by changing the first argument of the rootLogger property to DEBUG, for example:

   `log4j.rootLogger=DEBUG, stdout, JavaWindow, Access, Debug`
3. Uncomment the debug log file section to start recording information in that location.
4. Save and close the file.
5. Stop and restart the ArcIMS Z3950 9.0 service in the Services window.

   Only the debug log file will record information at the DEBUG level because the thresholds of the other logs remain set at the INFO level.



it's set to the INFO level, which records:

- Service startup status
- Connections made
- Queries received from Z39.50 clients
- Error messages

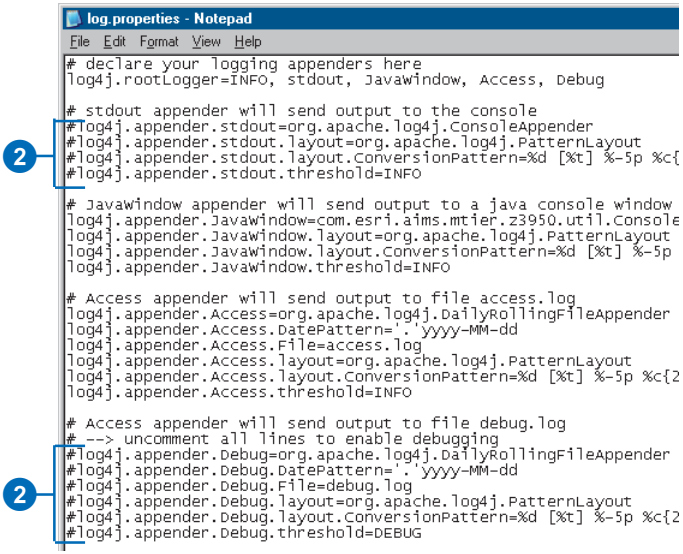
At the ERROR level, only error messages are recorded. At the WARN level, only warnings and error messages are recorded; for example, a warning is issued when the debug log section in the log.properties file is commented out. At the DEBUG level, all messages sent to and from the connector are recorded, including requests passed to and responses received from ArcIMS.

Each log can record information at a different level. To change a log's level, set its threshold property in the log.properties file. No log can record more information than the overall level, which is specified in the first argument on the first line of the file. For example, if the overall level is INFO and the debug log's threshold is set to DEBUG, the debug log file can't record more than the INFO level of information. With the overall level set to DEBUG, everything is recorded in the debug log file; the other logs will continue at their specified thresholds.

By default, the log file, named "access.log", and the debug ▸

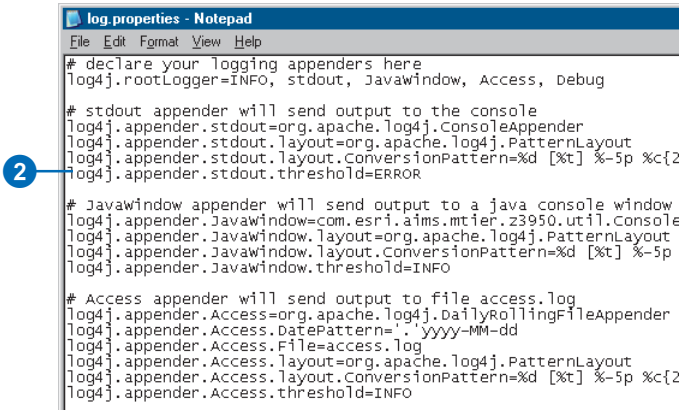
Stopping a log

1. Edit the log.properties file in a text editor. This file is located in the Z39.50 Connector's install directory.
2. Comment out the appropriate section to stop recording information in that location. For example, to stop logging to standard output and to the debug log file, comment out those sections.
3. Save and close the file.
4. Stop and restart the ArcIMS Z3950 9.0 service in the Services window.



Changing a log's threshold

1. Edit the log.properties file in a text editor. This file is located in the Z39.50 Connector's install directory.
2. Set a log's threshold property to the appropriate level. For example, to send only errors to standard output, change its threshold from INFO to ERROR.
3. Save and close the file.
4. Stop and restart the ArcIMS Z3950 9.0 service in the Services window.

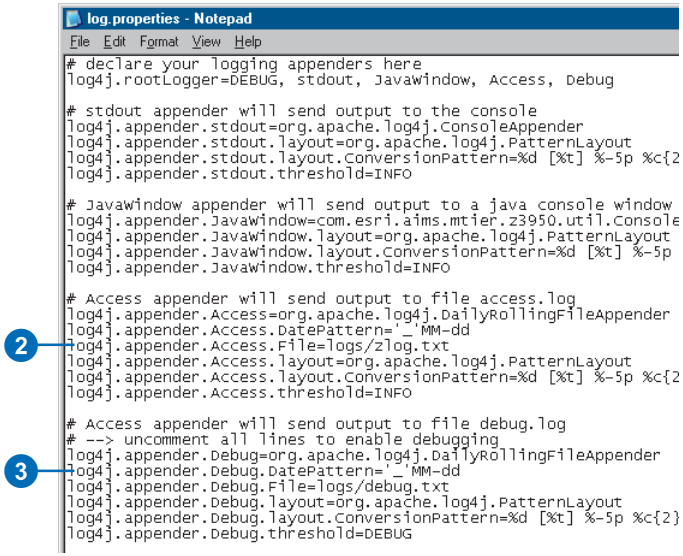


log file, named “debug.log”, are stored in the location where the Z39.50 Connector is installed. You can change the name and location of these files by changing the File property in the appropriate section of the log.properties file. For example, if you create a directory named “logs” in the location where the Z39.50 Connector was installed and change the log file’s File property to “logs/zlog.txt”, the log file, named “zlog.txt”, will be placed in the logs directory; the directory must already exist. The location specified is relative to where the connector is installed unless you specify a complete file path, for example, “C:/Z Logs/access.log”. Forward slashes (/) must be used in the file path.

Every day new log files are started and the previous day’s log files are renamed; if today is June 1, yesterday’s debug log file would be “debug.log.2004-05-31”, by default; this is accomplished by defining the date format as “yyyy-MM-dd”. To change the date format used to rename old log files, change the DatePattern property in the appropriate section of the log.properties file. With the DatePattern “‘_’MM-dd”, old debug logs would be re-named— “debug.log_05-31”, for example.

Changing a log file’s name and date

1. Edit the log.properties file in a text editor. This file is located in the Z39.50 Connector’s install directory.
2. To change a log file’s name or location, modify its File property appropriately. For example, to rename the log file “zlog.txt” and place it in a “logs” directory under the location where the connector is installed, change its File property to “logs/zlog.txt”. Slashes in the file path must be forward slashes (/).
3. To change the date format used to rename old log files, modify the DatePattern property appropriately. For example, to rename old log files following the format, “zlog.txt_04-24”, change the DatePattern property to “‘_’MM-dd”.
4. Save and close the file.
5. Stop and restart the ArcIMS Z3950 9.0 service in the Services window.



Protocols and restricting access to your metadata

The Z39.50 Connector lets people search and browse the documents published to a Metadata Service using Z39.50 clients. Access to your Metadata Services may already be restricted for browsing or searching when they are accessed using ArcCatalog or Metadata Explorer. You may also want to restrict access to your Metadata Services when they are accessed using the Z39.50 Connector. However, your ability to restrict access to Z39.50 clients is affected by the communication protocol used by the connector.

Communication protocols

The Z39.50 Connector can communicate with clients via HTTP, HTTPS, or TCP/IP. HTTP is the standard used in the majority of Web site applications; HTTPS is used for Web sites offering a greater degree of security; and TCP/IP is the choice of most traditional Z39.50 implementations.

Each protocol communicates with ArcIMS components in a different manner.

- With HTTP or HTTPS, client requests are initially sent to the Web server, passed through the ArcIMS Servlet Connector to the ArcIMS Application Server, and finally forwarded to the appropriate ArcIMS Spatial Server for processing before making the return trip with the query results.
- With TCP/IP, client requests bypass the Web server and Servlet Connector and are sent directly to the Application Server where they are forwarded to the appropriate Spatial Server for processing before the query results are returned.

Choosing a protocol for your implementation can be as easy as determining the type of connection needed by your core clients. However, another important factor in your decision involves the accessibility of your metadata.

Restricting access to your metadata

Metadata Services provide two levels of restrictions for accessing published documents—service level and document level. At the service level, access to an entire Metadata Service is controlled using authentication. At the document level, access to specific documents is restricted if they are private.

Service-level restrictions

Browse and search access to an entire Metadata Service can be restricted using authentication. Authentication is the process of verifying usernames and passwords against an access control list. When authentication is enabled and a username and password are required to access a Metadata Service, a Z39.50 client must provide valid login information to browse and search published documents.

Depending on the protocol used by the connector, authentication is enforced either by the ArcIMS Servlet Connector or the Z39.50 Connector. In both cases, the ACL can be either file-based or JDBC-based. Creating an ACL and assigning roles for accessing Metadata Services are discussed in ‘Creating usernames and passwords’ in Chapter 3.

With HTTP or HTTPS, authentication is enforced by the ArcIMS Servlet Connector. All ArcIMS requests received using the HTTP or HTTPS protocol are routed through the Servlet Connector, including those sent by ArcCatalog, Z39.50 clients, and possibly your own custom viewer or client application. If authentication has been enabled and restrictions have been placed on the service being accessed, the Servlet Connector validates the login information included with the request against its ACL. Once validated, the request is forwarded for processing. Enabling authentication in the Servlet Connector affects all ArcIMS services, not just Metadata Services.

With TCP/IP, requests sent by Z39.50 clients bypass the Web server and Servlet Connector. Authentication is enforced by the

Z39.50 Connector itself. If authentication has been enabled and restrictions have been placed on the Metadata Service being accessed, the Z39.50 Connector validates the login information included with the request against its ACL before forwarding the request.

Regardless of the protocol used, if authentication is not enabled, all services available on the ArcIMS server will be available to all users. Everyone will be able to access your Metadata Services.

Document-level restrictions

Once a document has been published, the owner of the document can make it private using ArcCatalog. Users can only see private documents if they are logged in using the same username and password as the owner of the document. For example, each department might access a Metadata Service using the same login information; a private document might be available to everyone in a department but not to the rest of the organization.

Document-level restrictions are controlled solely by the Servlet Connector. Therefore, they are only valid when a client communicates with the Metadata Service using HTTP or HTTPS. When the Z39.50 Connector is configured to use the TCP/IP protocol, communication bypasses the Servlet Connector and, therefore, everyone will be able to see private documents that exist in your Metadata Services.

Default username and password

Both levels of access restrictions—service and document—are dependent on authentication. If access to a Metadata Service is restricted, the client request must include a username and password to access its contents.

However, in most cases, Z39.50 clients are not built to include these parameters in their requests. For example, when you search the NSDI Clearinghouse, there is no mechanism for providing login information as part of your search. Modifying Z39.50 clients to accept and send username and password parameters is not within the scope of this book. Developers should refer to the Z39.50 specification for more information on building clients that perform this function.

When the Z39.50 Connector uses HTTP or HTTPS, you can configure the connector to use a default username and password when client requests are processed; this means the Z39.50 client isn't required to include login information in its requests. The default username and password must be defined in the ACL. You can't provide default login information when the Z39.50 Connector is using TCP/IP.

Protocol and metadata accessibility options

	HTTP and HTTPS	TCP/IP
Service-level restrictions	Yes	Yes
Default username/ password option	Yes	No
Document-level restrictions	Yes	No
Default username/ password option	Yes	Not applicable

This table summarizes the relationship between your choice of protocol and the accessibility of your metadata discussed in this section.

Putting it all together

Based on the example ACL file to the right, an organization has several unrestricted ArcIMS services, those available to the user, “*”, and one restricted service. Authentication has been enabled for Metadata Explorer, ArcCatalog, and the Z39.50 Connector, and all three use the same ACL. Other ArcIMS services can’t be accessed because they haven’t yet been added to the ACL.

Three of the unrestricted services support Metadata Explorer: SearchMap, Gazetteer, and OrgMetadata. Metadata Explorer lets the public search the Metadata Service OrgMetadata, which contains documents describing the unrestricted ArcIMS services and other data that is available for download. Metadata Explorer doesn’t have a login since it only exposes unrestricted services.

The Metadata Service DeptMetadata is used internally to post information about data under development. With ArcCatalog, only users who log in can access this service. Members of team A log in as “teamA”, publish information about data they are working on, and make those documents private until they are ready for internal consumption; team B does the same. Each team can see its own private documents and all public documents. Employees logged in as “internal” can see only public documents. Managers logged in as “manager” can see all public and private documents because they have the role metadata_browser_all.

The Z39.50 Connector is configured to use OrgMetadata as the default Metadata Service. If the user knows the name of the internal service and provides it as the database name along with appropriate login information, the user will be able to browse and search the DeptMetadata service using a Z39.50 client.

With HTTP or HTTPS and the Servlet Connector, when a default login isn’t provided, users must log in to access the DeptMetadata service if it’s specified as the database. If a default login for the user “internal” is provided in the z3950.properties file, users can access the OrgMetadata service and won’t have to log in to access the DeptMetadata service if it’s specified as the database. Users accessing the DeptMetadata service would only see public documents unless other login information is provided.

With TCP/IP and the Z39.50 Connector, users can access the OrgMetadata service and must log in to access the DeptMetadata service if it’s specified as the database. A default login can’t be provided. However, anyone who gains access to the DeptMetadata service will be able to see all public and private documents.

```
<?xml version="1.0" ?>
<AIMSACL>
<USER name="*" services="SearchMap, Gazetteer,
   OrgMetadata, SanFrancisco, California, USA"
   roles="metadata_browser"/>
<USER name="internal" password="employee"
   services="DeptMetadata"
   roles="metadata_browser_all"active="1"/>
<USER name="teamA" password="A.publisher"
   services="DeptMetadata"
   roles="metadata_publisher" active="1" />
<USER name="teamB" password="publisher.B"
   services="DeptMetadata"
   roles="metadata_publisher" active="1" />
<USER name="manager" password="big.office"
   services="DeptMetadata"
   roles="metadata_browser_all"active="1"/>
</AIMSACL>
```

In this example ACL file, the username “” indicates that access to the listed services is unrestricted. Users must log in to access the Metadata Service “DeptMetadata”; your ability to view private documents depends on the login information you provide.*

Setting a communication protocol

The Z39.50 Connector can implement the HTTP, HTTPS, or TCP/IP protocol for communicating with Z39.50 clients. You configure the connector to use a protocol by setting a series of properties in the z3950.properties file. By default, the connector is set to use HTTP; however, you may prefer to use a different protocol. Before changing the protocol, you must consider the consequences—HTTP or HTTPS and TCP/IP have different capabilities for restricting access to your metadata. For details, see ‘Protocols and restricting access to your metadata’ in this chapter.

To use the HTTP or HTTPS protocol with the Z39.50 Connector, set the connectionType property in the z3950.properties file to “http”; this is the default value. Set the url property to the Internet location of the machine that hosts the ArcIMS services; the URL will be either an http or an https address, depending on the protocol you are using.

To use the TCP/IP protocol, set the connectionType property ►

Setting the protocol to HTTP or HTTPS

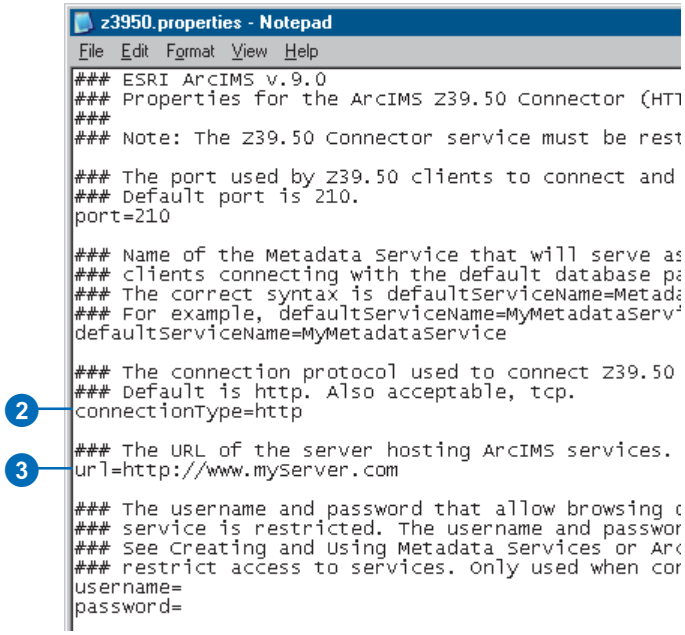
1. Edit the z3950.properties file in a text editor. This file is located in the Z39.50 Connector’s install directory.
2. Set the connectionType property to “http”. Use the same value for the HTTP and HTTPS protocol.

   `connectionType=http`
3. Set the url property to the Internet address of the server that hosts the ArcIMS Metadata Service. For example, for HTTP:

   `url=http://<localhost>.<domain>.com`

   Or for HTTPS:

   `url=https://<localhost>.<domain>.com`
4. Save and close the file.
5. Stop and restart the ArcIMS Z3950 9.0 service in the Services window.



to “tcp” and set the appServerMachine property to the name of the machine that hosts the ArcIMS Application Server. If the ArcIMS Application Server was configured to run on a port other than 5300, you must also modify the appServerPort property in the z3950.properties file.

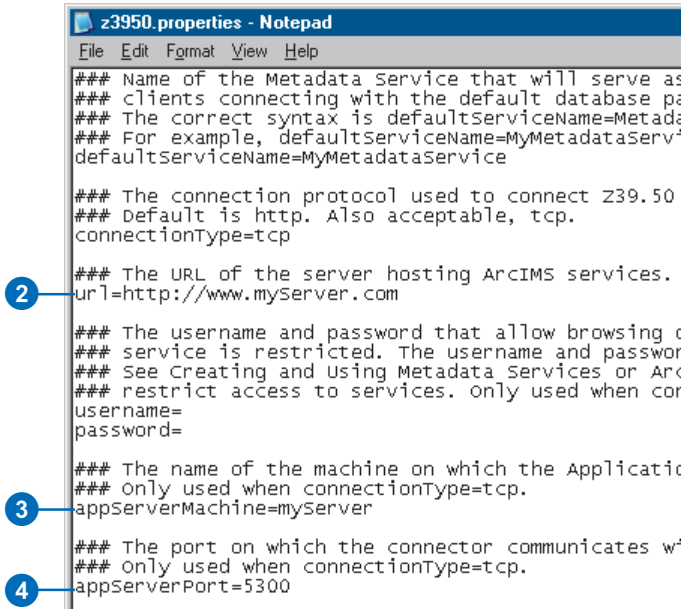
Setting the protocol to TCP/IP

1. Edit the z3950.properties file in a text editor. This file is located in the Z39.50 Connector’s install directory.
2. Set the connectionType property to “tcp”.

   `connectionType=tcp`
3. Set the appServerMachine property to the name of the computer where the ArcIMS Application Server is installed.

   `appServerMachine=hostname`
4. The default port on which the ArcIMS Application Server runs is 5300. If necessary, modify the appServerPort property.

   `appServerPort=5300`
5. Save and close the file.
6. Stop and restart the ArcIMS Z3950 9.0 service in the Services window.



Limiting access to your metadata

You may want to restrict access to your Metadata Service when accessed using the Z39.50 Connector, particularly if you are already restricting access to your Metadata Service using Metadata Explorer or ArcCatalog. Your ability to restrict access to Z39.50 clients and the method you use to enable authentication are affected by the communication protocol used by the connector.

With HTTP and HTTPS, authentication is enforced in the Servlet Connector; you must edit the Esrimap_prop file installed with your servlet engine to enable authentication and define which ACL to use. If you can access the Metadata Service from ArcCatalog, you've already done this—the Servlet Connector enforces authentication for ArcCatalog as well. You must edit the existing ACL, if modifications are required, to restrict access to Metadata Services for Z39.50 clients. If you haven't already configured authentication in the Servlet Connector, follow the instructions in 'Creating usernames and passwords' in Chapter 3 to create an ACL and modify the Esrimap_prop file. ▸

Providing a default login (for HTTP and HTTPS)

1. Edit the z3950.properties file in a text editor. This file is located in the Z39.50 Connector's install directory.
2. If access to the Metadata Service set in the defaultServiceName property is unrestricted, the username and password properties should not have values, for example:

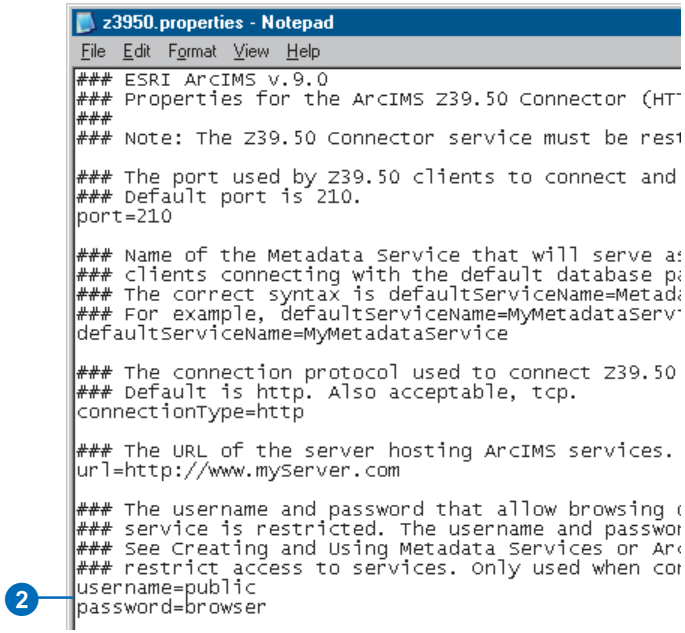
   `username=`

   `password=`

   If you must log in to access the default service and you want to provide a default login, set the username and password appropriately; this information must match a username and password combination in the ACL. For example:

   `username=browseuser`

   `password=browse_all`
3. Save and close the file.
4. Stop and restart the ArcIMS Z3950 9.0 service in the Services window.



In addition, you may want to provide a default login in the z3950.properties file. If access to your Metadata Service is unrestricted, the username and password properties should not have values. However, if users must log in, you can optionally set the username and password properties; they must match a username and password in the ACL. This lets Z39.50 clients, which don't usually require users to log in, access your metadata; the NSDI Clearinghouse Gateway doesn't let users provide login information.

With TCP/IP, authentication is enforced in the Z39.50 Connector; you must edit the Authenticate.properties file to enable authentication and define which ACL to use. That file is located in the connector's install directory. The Z39.50 Connector can share an existing ACL, for example, one used by Metadata Explorer, or have its own. This section describes a file-based implementation. Follow the instructions in 'Creating usernames and passwords' in Chapter 3 to create an ACL. Default login information can't be provided.

See Also

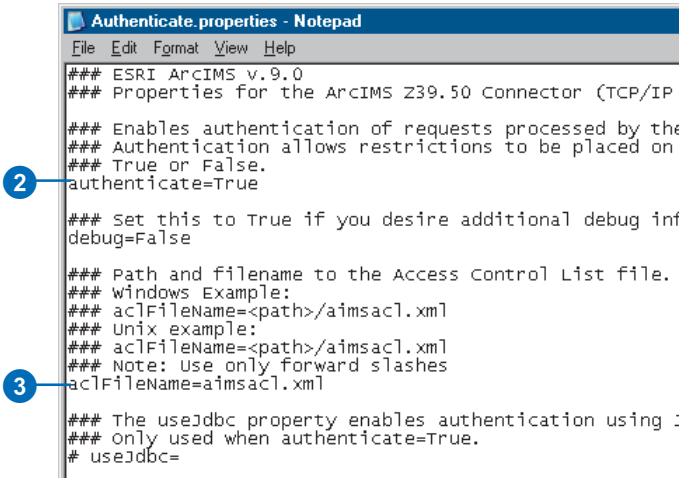
For more information on authentication and using a JDBC-based ACL, see ArcIMS Help.

Turning on authentication in the Z39.50 Connector (for TCP/IP)

1. Edit the Authenticate.properties file in a text editor. This file is located in the Z39.50 Connector's install directory.
2. Enable authentication.

   `authenticate=True`
3. Set the pathname to the ACL. Slashes in the pathname must be forward slashes (/).

   `aclFileName=<full pathname to the ACL file>/aimsacl.xml`
4. Save and close the file.
5. Stop and restart the ArcIMS Z3950 9.0 service in the Services window.



Customizing the connector

You can customize your Z39.50 implementation to meet the needs of your users in two different ways. To customize which metadata elements are available to Z39.50 clients for searching, modify the default definition file, GEO_expanded.xml. To customize the responses that are returned, modify the stylesheets used to format the responses.

Defining the information available for searching

The ArcIMS Z39.50 Connector supports the GEO Profile of the Z39.50 standard; created by the FGDC, it lets you define geospatially relevant search criteria and, therefore, find GIS resources using the Z39.50 protocol.

Z39.50 clients specify the type of information for which they want to search by providing a list of numeric use attribute codes. For example, with the BIB-1 profile, which is used by libraries, the code “4” indicates that the user is looking for a word in the resource’s title. The GEO Profile is built on two existing profiles: BIB-1 and Government Information Locator Service (GILS). This means that in the GEO Profile, “4” is also associated with the title; the code “2060” is only used by the GEO Profile—it indicates the user is doing a spatial search against the bounding coordinates that are defined in the metadata of the resource.

The codes used by a profile and the meaning of those codes are registered with the Z39.50 Maintenance Agency. This means clients and servers agree on what the user is searching for. However, the profile doesn’t specify how the server must implement the search against its database because each database is likely to have a different structure.

The ArcIMS Z39.50 Connector uses a file, GEO_expanded.xml, that maps the use attribute codes into one or more XPaths that identify the location in the published metadata documents where the information can be found. Since ArcCatalog provides both FGDC and ISO metadata editors, this file provides XPaths that locate the same information in both formats.

```
- <MDZCODELOADER>
    <!--  BEGIN: Mandatory use attributes for the GEO Profile   -->
  - <!--
        4 Title
       The name by which the data set is known.
    -->
    <SEMANTIC_PAIR zcode="4" tag="metadata/idinfo/citation/citeinfo/title" />
    <SEMANTIC_PAIR zcode="4" tag="metadata/dataIdInfo/idCitation/resTitle" />
  - <!--
        31 Publication Date
       The date when the data set is published or otherwise made available
    -->
    <SEMANTIC_PAIR zcode="31" type="date" tag="metadata/idinfo/citation/citein
    <SEMANTIC_PAIR zcode="31" type="date" tag="metadata/dataIdInfo/idCitatior
  - <!--
        62 Abstract
       A brief narrative summary of the data set.
    -->
    <SEMANTIC_PAIR zcode="62" tag="metadata/idinfo/descript/abstract" />
    <SEMANTIC_PAIR zcode="62" tag="metadata/dataIdInfo/idAbs" />
```

GEO_expanded.xml maps GEO Profile’s use attribute codes to metadata elements; this defines what information in the published documents can be searched.

Profiles also define some of their codes as mandatory; any client or server that supports a profile must, at a minimum, recognize and support the mandatory codes. All the mandatory GEO Profile codes and several additional codes are supported in the GEO_expanded.xml file, including 1012–Metadata Date, 2017–Ordering Instructions, 2021–Online Linkage, and 2042–Place Keyword.

By modifying the GEO_expanded.xml file, you can customize what content in your published documents is available for searching. For example, to support another use attribute code, add it to the file along with the appropriate mapping. Or, to make a custom metadata element available for searching, map your XPath to an existing use attribute code. Follow the structure used in the GEO_expanded.xml file.

```
GEO_expanded.xml - Notepad
File  Edit  Format  View  Help

<!-- 1012 Metadata Date
The date that the metadata were created or last updated -->
        <SEMANTIC_PAIR zcode="1012" type="date" tag="metadata/metainfo/metd"/>
        <SEMANTIC_PAIR zcode="1012" type="date" tag="metadata/mdDateSt"/>
        <SEMANTIC_PAIR zcode="1012" type="date" tag="metadata/Esri/ModDate"/>

<!-- 1016 Any & 1035 Anywhere -->
        <SEMANTIC_PAIR zcode="1016"/>
        <SEMANTIC_PAIR zcode="1035"/>
```

In this example, the GEO_expanded.xml file is modified to include searching an ESRI-defined metadata element to determine when a metadata document was last modified.

Defining how the Z39.50 response is produced

When a Z39.50 client sends a request to a Z39.50 server, one of the parameters in the request defines how the response must be formatted; in Z39.50 terms, this is the response syntax. The ArcIMS Z39.50 Connector supports the return of records in the formats XML, HTML, Simple Unstructured Text Record Syntax (SUTRS), and Machine-Readable Cataloging (USMARC). SUTRS is essentially plain text. USMARC is the format typically used by libraries.

Another parameter in the request defines the amount of information that should be returned for each document that was found; in Z39.50 terms, this is the element set. The Z39.50 Connector supports the return of records at the full, brief, and summary levels.

The connector uses Extensible Stylesheet Language Transformations (XSLT) stylesheets to convert the response produced by the Metadata Service to a Z39.50 response. The transformations.xml file matches each format and information level combination to a stylesheet that is used to generate the proper response; there are 12 combinations in all. When the connector is set to log messages at the DEBUG level, you can see the output generated by these stylesheets in the debug log file.

```
- <TRANSFORMATIONS>
    <!--  location of stylesheets  -->
    <SOURCE directory="stylesheets" />
    <!--  Record syntax/Element set  -->
    <!-- XML/Full (F)  -->
    <MAP from="metadata" to="xml" stylesheet="metadata_to_xml_full.xsl" />
    <!-- SUTRS/Full (F)  -->
    <MAP from="metadata" to="sutrs" stylesheet="metadata_to_sutrs_full.xsl" />
    <!-- USMARC/Full (F)  -->
    <MAP from="metadata" to="usmarc" stylesheet="metadata_to_usmarc_full.xsl" />
    <!-- HTML/Full (F)  -->
    <MAP from="metadata" to="html" stylesheet="metadata_to_html_full.xsl" />
    <!-- XML/Brief (B)  -->
    <MAP from="metadata" to="xml_brief" stylesheet="metadata_to_xml_brief.xsl" />
```

The transformations.xml file defines which XSLT stylesheets are used to produce the appropriate Z39.50 response.

You can customize the response that is generated by the connector by modifying the stylesheets that are provided with ArcIMS. Or you can modify the transformations.xml file to use your own stylesheets instead. If you prefer to provide a different file that maps the Z39.50 client syntax and element set combinations to your own stylesheets, change the transformationMap property in the z3950.properties file.

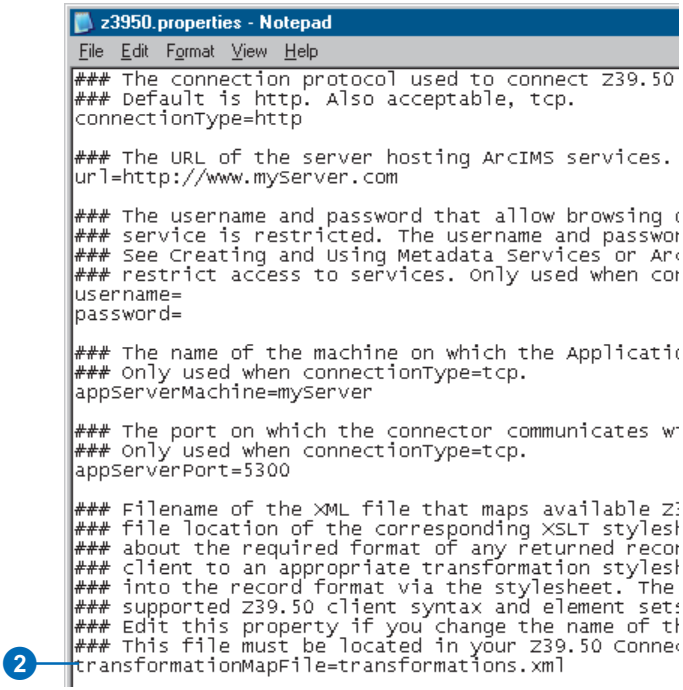
Controlling the format of responses

A Z39.50 client request specifies the format in which documents must be returned and the amount of information that should be included for each document. The format, or syntax, options are XML, SUTRS, USMARC, or HTML. The options for the amount of information, the element set, are full, brief, or summary. These options are discussed in the previous section, ‘Customizing the connector’.

Converting the Metadata Service’s response to the requested Z39.50 response format is a two-step process. First, the Z39.50 Connector determines which stylesheet to use, then the appropriate stylesheet is applied to produce the required output. The transformations.xml file determines which stylesheet is used; it maps all supported Z39.50 client syntax and element set combinations to one of the 12 XSLT stylesheets provided with the connector. To customize the connector to use a different mapping file, change the transformationMap property in the z3950.properties file.

1. Edit the z3950.properties file in a text editor. This file is located in the Z39.50 Connector’s install directory.
2. Set the transformationMapFile property to the name of the mapping file. The mapping file must be located in the Z39.50 Connector’s install directory.

   `transformationMapFile=transformations.xml`
3. Save and close the file.
4. Stop and restart the ArcIMS Z3950 9.0 service in the Services window.



Available Z39.50 resources

Learn more about Z39.50

The Z39.50 protocol's specification, related tutorials, and other information are available from the Z39.50 Maintenance Agency site, hosted by the United States Library of Congress at http://lcweb.loc.gov/z3950/agency/.

In addition, the ESRI implementation of the Z39.50 specification is based on the GEO Profile. For information, refer to http://www.blueangeltech.com/standards/GeoProfile/geo22.htm.

A number of Z39.50 Web gateways or clearinghouses are also available, including:

- FGDC NSDI Clearinghouse—http://www.fgdc.gov/clearinghouse
- Library of Congress gateway—http://lcweb.loc.gov/z3950/

Z39.50 client software

If you don't have an existing Z39.50 client, a revised version of the MCNC Center for Networked Information Discovery and Retrieval (CNIDR®) ISite Z39.50 software suite is freely available through the FGDC Clearinghouse and can be used to test your Z39.50-enabled site. This version of ISite, modified to support the GEO Profile, can be downloaded from http://www.fgdc.gov/clearinghouse/tutorials/getisite.html.

The ISite software suite contains clients as well as utilities to index and serve metadata. However, while you can effectively use the clients included with ISite to test your Z39.50 configuration, its server software, like most Z39.50 servers, utilizes port 210. The ArcIMS Z39.50 Connector functions as the Z39.50 server and, by default, also uses port 210. Disable ISite's Z39.50 service or use separate ports to avoid the conflict caused by using both applications.

Configuring Metadata Explorer

A

IN THIS APPENDIX

- **Which ArcIMS services does Metadata Explorer need?**
- **Starting the services**
- **Configuring Metadata Explorer**
- **Do you need a login?**
- **Loading the gazetteer data**
- **Customizing the gazetteer**
- **An unattended Metadata Explorer session will expire**
- **Integrating Metadata Explorer into ArcExplorer**

Metadata Explorer is a Web-based application that lets you browse or search a Metadata Service and find metadata documents that describe GIS resources. The main reason to host Metadata Explorer on your Web site is to let people within your organization access the information published to your Metadata Service. You could also make your Metadata Explorer available to people outside your organization.

You don't need to read this appendix if you just want to find data. Simply point your Web browser at a Metadata Explorer site and follow the instructions in Chapter 2 to find data. However, if you're tasked with getting Metadata Explorer running, then you'll find what you need to know here.

Getting Metadata Explorer up and running is essentially a two-step process. The first step is to install Metadata Explorer and deploy it on your Web server. This includes running the ArcIMS postinstall program to load Metadata Explorer and making sure you have the necessary components installed and configured properly: Java 2 Platform Software Development Kit (SDK), Web server, and servlet engine. The second step involves setting up the necessary ArcIMS services that Metadata Explorer uses, such as the Metadata Service itself, a gazetteer service, and a search map for interactively specifying a search area when looking for data.

The information for completing the first step is covered in the *ArcIMS Installation Guide*. This appendix focuses on completing the second step, including starting the appropriate ArcIMS services; configuration options, such as requiring a password; and loading the gazetteer data.

Which ArcIMS services does Metadata Explorer need?

You can think of Metadata Explorer as an interface to a particular Metadata Service. While you can host several Metadata Services through ArcIMS at one time, Metadata Explorer can only reference one of them. If you want to host two independent Metadata Services and have both accessible through Metadata Explorer, you would need to deploy another Metadata Explorer application on your Web server. For example, you would call the second one Metadata Explorer 2, and it would be configured to access the second Metadata Service.

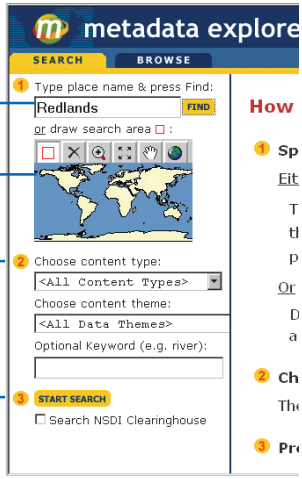
Some of the features available in Metadata Explorer rely directly on services provided by ArcIMS. Thus, for you to host Metadata Explorer, you also need to host the additional ArcIMS services used by Metadata Explorer: a Metadata Service, a search map (Image Service), and a gazetteer (also implemented as a Metadata Service). To get Metadata Explorer working, you must first start the necessary services in ArcIMS, then configure Metadata Explorer to use these services.

The gazetteer helps you find the location in which you want to search for data.

The search map lets you draw a box to specify the location in which you want to search for data.

Define your search criteria.

Click Start Search to find documents that match the search criteria in the Metadata Service.



Starting the services

When you installed ArcIMS, the installation program also installed sample configuration files that you can use to start the services needed by Metadata Explorer. These configuration files are named MetadataServer.axl, SearchMap.axl, and Gazetteer.axl and are located in the Metadata directory of the AXL directory. The default install location for these files is:

- **Windows:** <Drive>:\ArcIMS\Axl\Metadata
- **UNIX:** <ArcIMS install directory>/Metadata/Axl

1. Start a Metadata Service called, for example, Metadata. Use the MetadataServer.axl file to start your Metadata Service.

   Follow the instructions in Chapter 3, ‘Creating Metadata Services’, making sure that you correctly specify the ArcSDE connection parameters, table name prefix, and so on.

2. Start a search map (Image Service) called, for example, SearchMap.

   Use the SearchMap.axl file and start an Image Service. You may need to edit the file in a text editor to ensure that the pathname to the data is correct. Look for the following line in the file and edit the pathname accordingly.

   ```
   <SHAPEWORKSPACE name="shp_ws-0"
   directory="<ArcIMS install directory>
   \Metadata\Data\searchmap"/>
   ```

   The search map data is provided as a starting point for you. You can display your own search map using your own data. Any Image Service will suffice as long as the data resides in geographic coordinates (decimal degrees).

3. Optionally, start the gazetteer (Metadata Service).

   Before you can start the gazetteer service, you must have installed, loaded, and indexed the gazetteer data; see the section below, ‘Loading the gazetteer data’, for instructions. In the meantime, you can configure Metadata Explorer to work without the gazetteer.

When the gazetteer data is ready, edit the Gazetteer.axl file in a text editor to specify the same connection parameters you used when you loaded the gazetteer data into ArcSDE. If you create or modify a different configuration file for the gazetteer service, the table prefix must be “gazetteer”. Then, start another Metadata Service called, for example, Gazetteer.

Although the gazetteer is invoked as a Metadata Service, it is not a true Metadata Service. Don’t attempt to publish metadata to it.

Configuring Metadata Explorer

Once all the services are running, you can configure Metadata Explorer to use the services. For instance, Metadata Explorer needs to know the name of the machine that hosts the services and the names of the services themselves. You provide this information through a property file called aimsmeta.properties. This file is located in the install directory at:

- **Windows:** <ArcIMS install directory>\Metadata\metadataexplorer\WEB-INF\classes\aimsmeta.properties
- **UNIX:** <ArcIMS install directory>/arcims/Metadata/metadataexplorer/WEB-INF/classes/aimsmeta.properties

If you are using JRun, Tomcat, or WebSphere, this file will be located where you installed the Web software in the directory shown below:

- **JRun 4:** <JRun install directory>\ servers\default\SERVER-INF\temp\metadataexplorer.war-0123456789\WEB-INF\classes

  The 10-digit number is generated when the servlet is deployed.
- **Tomcat:** <Tomcat install directory>\webapps\metadataexplorer\WEB-INF\classes
- **Websphere 5.0:** <WebSphere install directory>\AppServer\installedApps\<servername>\metadataexplorer_war.ear\metadataexplorer.war\WEB-INF\classes

Commonly, the services that you run to support Metadata Explorer will be hosted on the same machine running ArcIMS. However, the property file gives you the flexibility to run the services on different machines if your ArcIMS site configuration supports it.

Edit the aimsmeta.properties file in a text editor and follow the steps below to configure Metadata Explorer.

1. Set the name of the Metadata Service, host machine, and port number the service is running on. For example:

   ```
   meta_service_name=Metadata
   meta_host_name=hostname
   meta_port_number=5300
   ```
2. Set the name of the Image Service used as the search map and specify the host machine and port number on which the service is running.

   ```
   search_service_name=SearchMap
   search_host_name=hostname
   search_port_number=5300
   ```
3. If you’ve loaded the gazetteer data and created a Metadata Service for it, enable the gazetteer. Set the name of the Metadata Service representing the gazetteer and specify the host machine and port number on which the service is running.

   ```
   search_gazetteer=true
   gazetteer_name=Gazetteer
   gazetteer_host_name=hostname
   gazetteer_port_number=5300
   ```
4. Optionally, change the number of metadata documents displayed per page as a result of a search.

   ```
   results_batch_size=20
   ```
5. Optionally, restrict access to Metadata Explorer with a login. You can set the login to off, optional, or required. See the next section for more information on logins.

   ```
   login=off
   ```

You should now be able to launch Metadata Explorer. In a Web browser, type the following URL:

http://<localhost>.<domain>.<com>/metadataexplorer

Oracle Application Server requires an extra forward slash (/) at the end of the URL:

http://<localhost>.<domain>.<com>/metadataexplorer/

Do you need a login?

You can configure Metadata Explorer so that anyone who has access to your Web server can access it. Alternatively, if your Metadata Service contains sensitive information that should only be accessed by certain users, you may choose to require a user to log in before Metadata Explorer can be accessed.

There are three levels of access that you can use for Metadata Explorer:

No login is required—Anyone can access the public documents on the Metadata Service. However, if any published documents on the service are flagged as "private" by a metadata publisher, they will not be seen when browsing and searching for documents. (To flag a metadata document as private, in ArcCatalog, right-click the document on the Metadata Service, click Properties, and check Private.)

Login is optional—All public documents are available for viewing without logging in. If metadata publishers choose to login with their username and password, they will see their private documents as well.

When a login is optional, a login link will appear in the upper right corner, next to the Help link, allowing users to log in if they want to access their private documents.

Login is required—All access to Metadata Explorer is restricted. Users must log in to gain access. Once logged in, users will see all public documents and the private documents they published.

As the administrator of Metadata Explorer, you can also create a specific login that can view all private documents.

When a login is required, Metadata Explorer will display a login page before a user can access the site.

If you choose to enable the login feature—by setting login to optional or required in the aimsmeta.properties file—you will also need to enable authentication and set up usernames and passwords to access Metadata Explorer as described below.

Enabling authentication

Metadata Explorer communicates with ArcIMS through the Java Connector. The Java Connector can authenticate, or validate, a user and grant them access to Metadata Explorer based on the username and password they supply. By default, Metadata Explorer doesn't attempt to authenticate any users. Thus, to fully utilize the login feature, you must enable authentication. Once enabled, you define usernames and passwords to access Metadata Explorer in an ACL.

You can implement your ACL using a text file or a Java database (JDBC). Authentication via the Java Connector works the same way as authentication through the Servlet Connector. For more information on authentication, see the topic 'Restricting access to services' in *ArcIMS Help*. The information below illustrates a file-based ACL implementation.

1. Edit the authenticate.properties file in a text editor. This file is in the same location as aimsmeta.properties.
2. Enable authentication by setting authenticate to true.

   ```
   authenticate=True
   ```

3. Specify the location and name of the ACL file. Use forward slashes (/) to separate directories in the pathname.

```
aclFileName=<pathname>/aimsacl.xml
```

Setting up usernames and passwords

You set up usernames and passwords in the access control list. If you've already created a Metadata Service to which you can publish metadata documents, then you're probably already familiar with an ACL file. For instance, in ArcCatalog, you specify a username and password when you connect to your Metadata Service and publish documents.

ArcCatalog and Metadata Explorer communicate to ArcIMS through different connectors. Metadata Explorer communicates through the Java Connector, whereas ArcCatalog communicates through the Servlet Connector. Each connector needs to access username and password information specified in an ACL. You may use the same ACL for both connectors; however, you may choose not to. It just depends on whether or not you think it's alright for ArcCatalog users to see the auxiliary services (search map and gazetteer) that support Metadata Explorer when they connect to your site—these two services must be free services for Metadata Explorer.

The following ACL file serves as an example of how you might set yours up for Metadata Explorer. This example will work when you set the login to optional or required. The services representing the search map and gazetteer should always be listed as free services or services that don't require a username and password. The service representing your Metadata Service should also be listed as a free service when a login is optional.

```
<?xml version="1.0" ?>
<AIMSACL>
<USER name="*" services="SearchMap, Gazetteer,
   Metadata" roles="metadata_browser" />
<USER name="user1" password="pass1"
   services="Metadata" roles="metadata_publisher"
   />
<USER name="user2" password="pass2"
   services="Metadata" roles="metadata_publisher"
   />
<USER name="admin" password="admin"
   services="Metadata"
   roles="metadata_administrator" />
</AIMSACL>
```

If you are using a separate ACL file for the Java Connector and Servlet Connector, you should use the same usernames and passwords. In general, any metadata documents flagged as private can only be viewed in Metadata Explorer by the user who published the document to the Metadata Service. Because the username is stored with the published metadata document, the username in the ACL file must match. However, a user with the role metadata_administrator will be able to view all documents, including those flagged as private.

Loading the gazetteer data

To use the gazetteer functionality in Metadata Explorer, you need to do five things:

- Install the gazetteer data.
- Import the data into ArcSDE.
- Index the gazetteer data.
- Create a Metadata Service based on the gazetteer data.
- Configure Metadata Explorer to access the gazetteer.

If you didn't install the gazetteer data along with ArcIMS, you can run the installation program again to install it. When you run the installation program, choose the Custom installation option and look for Metadata Explorer. Make sure you specify that you want to install the gazetteer data because, by default, it won't be installed.

The ArcIMS installation program installs ArcSDE export files that you must load into ArcSDE:

- gazetteer
- gazetteerd
- gazetteerdr
- gazetteerr
- gazetteeru

You may have noticed that the sdeexport file names look similar to the objects created in the database for a Metadata Service. Aside from using a different table name prefix, the database schema for the gazetteer service is the same as for any Metadata Service. However, the records for the gazetteer do not contain complete metadata—do not attempt to publish metadata documents to the Gazetteer Service from ArcCatalog!

The gazetteer data provided with ArcIMS contains more than 350,000 place names and consumes approximately 300 megabytes of disk space. You must work with your database administrator to plan where and how to load the data. See Appendix B, 'ArcSDE considerations', in this book for detailed information. If you want to add or remove entries from the gazetteer, see the section 'Customizing the gazetteer' below.

Four scripts are also installed. You'll run the one that is appropriate for your database and operating system to import the data using ArcSDE:

- GazetteerImport.bat—Windows version for SQL Server and DB2
- GazetteerImport_Ora.bat—Windows version for Oracle
- GazetteerImport.sh—UNIX version for DB2
- GazetteerImport_Ora.sh—UNIX version for Oracle

Before you begin, you need some information from the database administrator. You need to know the ArcSDE server name, the ArcSDE instance, a username and password, and the database name, if you're using SQL Server or DB2. Because the gazetteer needs to be administered separately from other Metadata Services, you may want to use a different username for the gazetteer from the one you use for other Metadata Services.

Follow the steps below to import the gazetteer data.

1. Verify that your version of ArcSDE meets the system requirements.

   For information on compatible versions of ArcSDE, go to *http://www.esri.com/software/arcims/index.html* and click System Requirements on the left. Specify your configuration, then on the resulting page, click Compatibility Notes in the Quick Links box.

2. Verify that %SDEHOME%\bin ($SDEHOME/bin for UNIX) is in your system's path variable.

3. Open the operating system's command window and navigate to the location where you installed the gazetteer files. By default, they are installed at <ArcIMS install directory>/Metadata/Data/Gazetteer.

4. Run the GazetteerImport script that corresponds to your operating system and database. The arguments for the Oracle version of the script are:

   ```
   GazetteerImport_Ora <service> <server>
   <username> <password>
   ```

   For Oracle, you would type the following on the command line, for example:

   ```
   GazetteerImport_Ora 5151 sdeServer aUser
   aPassword
   ```

   While the script runs, you may see error messages indicating that some objects couldn't be deleted from the database. Ignore these messages if you haven't installed the gazetteer data before—the script attempts to delete any preexisting gazetteer objects from the database.

5. In the command window, navigate to the location where the metadata commands are installed.

6. Run the aimsmetaindx command to index the gazetteer place names so the gazetteer service will work in Metadata Explorer. The arguments for the aimsmetaindx command are:

   ```
   aimsmetaindx <sde_server_machine>
   <sde_instance> <database> <username> <password>
   {table_name_prefix} {logfile} {localcodepage}
   ```

   You must provide the same ArcSDE connection information that you used to load the data. You must also specify the table_name_prefix argument since the prefix used is "gazetteer", not "imsmetadata". For example, with an Oracle database you would type the following on the command line:

   ```
   aimsmetaindx sdeServer 5151 # aUser aPassword
   gazetteer
   ```

   The aimsmetaindx command will finish immediately, but indexing has just begun in the database; it will take a few hours to index the gazetteer place names. See the section 'Manually indexing documents' in Chapter 3 for more information. See the section 'Configuring ArcSDE' in Appendix B for information about indexing issues with DB2 databases.

Once the gazetteer data has been loaded and indexed, you can create a Metadata Service based on that data using the sample configuration file Gazetteer.axl, as described in the section 'Starting the services' above. If you create or modify a different configuration file to start the gazetteer service, the table prefix must be "gazetteer". More information about the aimsmetaindx command and starting Metadata Services can be found in Chapter 3, 'Creating Metadata Services', in this book.

Once the gazetteer service has been started, configure Metadata Explorer to access the gazetteer service as described in the section 'Configuring Metadata Explorer' above.

Customizing the gazetteer

The gazetteer distributed with ArcIMS contains roughly 350,000 place names from around the world. It contains a complete listing of all countries, large- and medium-sized cities, major water bodies, mountain ranges, and so on. Its global scope allows you to use it with any Metadata Service you host.

However, its global scope can also mean that it's not ideally suited to your Metadata Service. For example, searching for Paris in Metadata Explorer yields more than 70 places—including entries for Paris, France, and Paris, Texas, in the United States. If you know that the metadata published to your service will only cover the United States, there will be unnecessary gazetteer entries. Similarly, the level of detail in this global gazetteer may not be sufficient for your application, especially when you expect people to search for data using the names of small towns or local geographic features.

Having some idea beforehand as to what geographic area the metadata on your Metadata Service will occupy lets you to customize your gazetteer to match. When you deploy Metadata Explorer, you can modify the gazetteer so it more closely matches the geographic extent of the metadata published to your service.

There are several ways you can tailor the gazetteer to suit your needs. You can load the gazetteer data distributed with ArcIMS, add some place names to it, and remove those you don't want. For example, you might add places that are inside your area of interest and remove some of the global features that are outside. Alternatively, you can start from scratch and build your own gazetteer, then load those place names into a new, empty gazetteer service.

Whether you're modifying an existing gazetteer or creating a new one, your first step is to compile a list of place names you want the gazetteer to have. To create a new gazetteer, create a table or feature class with the places on your list, start a gazetteer service, and load your places into the database using ArcCatalog. The

same method is used to add places to an existing gazetteer, except the gazetteer service has already been started. To remove place names from an existing gazetteer, you must export the gazetteer's places to a personal geodatabase, edit the data in ArcMap to remove the unwanted places, and load the modified data into a new gazetteer service.

Creating a list of place names

To add entries to a gazetteer, you need to have a list of place names that people should be able to search for. Your list of place names can be stored in any table or feature class supported by ArcGIS—use whichever format is most convenient for you. Don't worry if your list isn't contained in a single data source; some place names might be stored in a text file while others are stored in a shapefile. Place names can be added incrementally to a gazetter, from each data source in turn.

A data source containing place names for a gazetteer service must have the following pieces of information. Store each piece of information in a separate column in a table or feature class.

A place name—One or more keywords that can be used to search for a place. Typically the place name is the name of a country, city, river, or any geographic feature for which people will be able to search. There should be one record for each place; for example, you would have one record for Japan, one for San Francisco, one for Lake Ontario, and so on.

Each record's place name can consist of more than one keyword, separated by a space. For example, the place name, "New York Big Apple" would let people find New York City whether they searched for "New York" or "Big Apple". Keywords are not case sensitive.

A description of the place—A statement clearly identifying the place. The description is shown in Metadata Explorer and must help people select the place in which they are interested. For example, if you try to find "Paris" with the ArcIMS gazetteer, you will find 67 places with that name, including "Paris, Ile-de-France, France" and "Paris, Oregon, United States". The description should indicate the type of feature, such as an airport, lake, glacier, or historical location of a city, for example, "Paris (historical), Oregon, United States".

A rank, or sorting order, of the place—A value from 1 to 25, where a value of 1 indicates a high rank. When a gazetteer search finds many places, higher ranked places will appear at the top of the list.

The geographic location of the place—Either a shape representing the place or, in a table, four coordinates defining the place's extent. With a spatial data source, such as a feature class, the bounding box of each shape will be calculated and loaded into the database. With a table, the bounding box must be defined by specifying the minimum x, minimum y, maximum x, and maximum y coordinates for the extent. Each value must be stored in a separate column.

The location must be specified in decimal degrees. Spatial data sources used to populate the gazetteer must have a spatial reference defined. For instance, shapefiles must have an associated .prj file. When a table is used, the coordinates for the bounding box must be specified in decimal degrees.

Adding place names to a gazetteer

Once you have your list of place names stored in a form that ArcCatalog can use, you're ready to load them into the database. You'll add the place names to a gazetteer service in ArcCatalog using a special tool that you must add to a toolbar. The gazetteer service may contain many places, or it may be empty.

Follow the steps below to add place names to your gazetteer. Before you begin, you need to get some information from the database administrator. You need to know the ArcSDE server name, the ArcSDE instance, a username and password, and the database name, if you're using SQL Server or DB2.

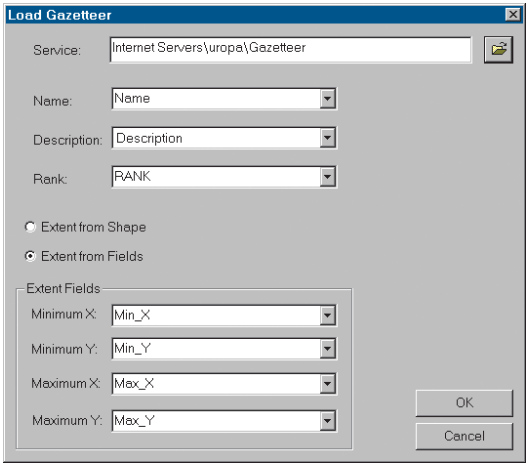
1. If you don't have a gazetteer service running already, use ArcIMS Administrator to start a Metadata Service for your

gazetteer. See the section ‘Starting the services’ earlier in this appendix for more information.

2. To add place names, you must have metadata_publisher access to the gazetteer service defined in the ACL file. Most likely, you will need to temporarily modify your ACL file to provide this access.

3. Start ArcCatalog and add the Load Gazetteer tool to a toolbar. Click the Tools menu and click Customize. In the Customize dialog box, click the Commands tab. On the left side, click the Gazetteer category. On the right side, click the Gazetteer Loader command. Click and drag the command from the Customize dialog box onto an ArcCatalog toolbar. Close the Customize dialog box.

4. In ArcCatalog, select the data source containing the place names that you want to add to the gazetteer service. Then, click the Gazetteer Loader button that you added in step 3. The following dialog box appears.



5. Click the Browse button in the upper right corner of the dialog box next to the Service text box. Navigate to the GIS Servers folder, connect to the ArcIMS server, then select the gazetteer service to which you want to add places. If you don’t see the icon next to the gazetteer service, you don’t have metadata_publisher access to it. You’ll need to edit your ACL file before you can use this tool.

6. Specify the columns in the data source that contain the place name, description, and rank information.

7. If the data source is a table, specify which column contains the appropriate extent coordinate. The South dropdown list should specify the column containing the minimum y coordinate, the East dropdown list should specify the column containing the maximum x coordinate, and so on.

8. Click OK.

   This tool takes a while to run. ESRI testing shows it takes approximately two to three minutes to load 1,000 place names. Your performance may vary.

9. Run the aimsmetaindx command to index the new gazetteer place names so they can be found in Metadata Explorer. The arguments for the aimsmetaindx command are:

```
aimsmetaindx <sde_server_machine>
<sde_instance> <database> <username> <password>
{table_name_prefix} {logfile} {localcodepage}
```

   Provide your ArcSDE connection information for the appropriate arguments. You must also specify the table_name_prefix argument since your gazetteer service will use a prefix in its configuration file, such as “gazetteer”, not “imsmetadata”. For example, with an Oracle database you would type the following on the command line:

```
aimsmetaindx sdeServer 5151 # aUser aPassword
gazetteer
```

   The aimsmetaindx command will finish immediately, but indexing will continue in the database; depending on how many place names you’ve loaded, it may take a few hours to index them. See the section ‘Manually indexing documents’ in

Chapter 3 for more information. See the section 'Configuring ArcSDE' in Appendix B for information about indexing issues with DB2 databases.

Once the new place names have been loaded into and indexed in your gazetteer service, they are ready for use in Metadata Explorer. More information about the aimsmetaindx command can be found in Chapter 3, 'Creating Metadata Services', of this book.

Removing place names from a gazetteer

Removing place names from a gazetteer service requires that you edit the gazetteer data locally to remove the unwanted places. If you created a custom set of place names, edit your original data in ArcMap to remove the unwanted places. Once the appropriate place names have been removed, you must load the edited data into a new gazetteer service.

If you no longer have the original data that you used to create the gazetteer or you want to remove places from the gazetteer data distributed with ArcIMS, you must first export the contents of your gazetteer service to a personal geodatabase. Because the geodatabase currently doesn't support ArcSDE XML columns, which are used by Metadata Services to store and index the contents of XML documents, you can't directly access the feature class associated with the gazetteer service in ArcCatalog. Therefore, to export the contents of your gazetteer service you must first create a view on that feature class using the ArcSDE sdetable command; the view must not include the service's XML column.

Follow the steps below to export the gazetteer data to a personal geodatabase, edit the data, then load the edited data into a new gazetteer service. Before you begin, you need to get some information from the database administrator. You need to know the ArcSDE server name, the ArcSDE instance, a username and password, and the database name, if you're using SQL Server or DB2.

1. Verify that %SDEHOME%\bin ($SDEHOME/bin for UNIX) is in your system's path variable.
2. Open a command window and run the following ArcSDE command to create a view on the feature class associated with your gazetteer service.

   ```
   sdetable -o create_view -T gazetteer_v
   -t gazetteer
   -c docid,service,server,datasetname,footprint
   -a fid,name,rank,description,footprint
   -w "datasetname <> 'Home'" -i <service>
   -s <server_name> -D <database>
   -u <DB_User_name> -p <DB_User_password>
   ```

   Provide your database connection information for the appropriate arguments. Do not include the -D argument if you're using an Oracle database. For example, with a SQL Server or DB2 database:

   ```
   sdetable -o create_view -T gazetteer_v
   -t gazetteer
   -c docid,service,server,datasetname,footprint
   -a fid,name,rank,description,footprint
   -w "datasetname <> 'Home'" -i 5151 -s sdeServer
   -D sde -u aUser -p aPassword
   ```

   If your gazetteer service's configuration file specifies a table prefix other than "gazetteer", provide your prefix for the -t argument and change the -T argument to "<yourPrefix>_v".

   The -c argument specifies the columns in the feature class that will be included in the view—the view must not include the column named "XML". For gazetteer services, only three columns in the service's feature class are important for defining place names: Service, Server, and DatasetName. The Service column contains the place name. The Server column contains the rank. The DatasetName column contains the description. The -a argument specifies aliases for these columns that are more appropriate for gazetteer data than the actual column names.

   The -w argument specifies a SQL where clause that selects which records will be included in the view. This where clause

excludes the gazetteer service's root document. When the edited gazetteer data is loaded into a new gazetteer service, it must not include the original service's root document. The new service will have its own root document—there can be only one root document in a Metadata Service. For more information about the database schema for a Metadata Service, see Appendix B, 'ArcSDE considerations', in this book.

3. Create a personal geodatabase that will store the exported gazetteer feature class. In ArcCatalog, right-click a folder, point to New, and click Personal Geodatabase. A personal geodatabase is created in that folder. Type a new name for the geodatabase, such as "gazetteer", and press Enter.

   If you will be modifying the gazetteer data distributed with ArcIMS and the view was created as described in step 2, the exported data will require approximately 160 MB of space. If you included additional columns in the view, more disk space will be required.

4. Connect to the ArcSDE database that contains the gazetteer service's objects. In ArcCatalog, double-click the Database Connections folder, then double-click Add Spatial Database Connection. Provide the same connection information that you used in step 2 to access the database. Double-click the new database connection in the Contents tab to see a list of its contents.

5. In the Contents tab, select the feature class named "gazetteer_v", or the name you provided in the -T argument in step 2. In ArcCatalog, the view will appear the same as any other feature class because it includes a Geometry column.

6. Export the view's contents to the personal geodatabase. Right-click the view, point to Export, and click To Geodatabase (single). The Feature Class to Feature Class tool appears. Click the Browse button next to Output location, navigate to and select the personal geodatabase created in step 3, then click Add. Type a name into the Output Feature Class Name text box, such as "gazetteer". Click OK.

7. Edit the exported data in ArcMap to remove the places you don't want. If you only want to remove specific features, you can select them by their attribute values. If you want to remove whole geographic regions, it may be easier to select these features interactively in the map. Delete the selected features and save your edits. See the book *Editing in ArcMap* for more information about editing data.

8. Start a new Metadata Service for the edited gazetteer data. When you create a new configuration file for this service, you must specify a different table name prefix. Otherwise, if you use an existing table prefix, such as "gazetteer", you will append the edited places to a gazetteer service that still contains the records you wanted to remove.

9. Select the feature class containing the edited places in ArcCatalog, then click the Gazetteer Loader button. Load the edited places into the new gazetteer service following the steps in the earlier section, 'Adding place names to a gazetteer'. If you haven't used the Gazetteer Loader before, you first need to add it to a toolbar in ArcCatalog.

Once you finish loading the gazetteer, it's ready for use with Metadata Explorer.

An unattended Metadata Explorer session will expire.

When a user accesses Metadata Explorer with a username and password, the session will expire if no activity occurs between the user and the Web server after approximately 30 minutes. The specific length of time will depend upon your Web server and servlet engine. You can explicitly set the session time-out by editing the web.xml file located at <ArcIMS install directory>/Metadata/MetadataExplorer/WEB-INF.

Edit the web.xml file in a text editor and insert into the file the following lines shown in bold text. Set the session-timeout to the desired number of minutes.

```
<web-app>
  <session-config>
    <session-timeout>30</session-timeout>
  </session-config>
</web-app>
```

Integrating Metadata Explorer into ArcExplorer

When you add data to ArcExplorer, you may have noticed that there is a button at the top of the dialog box that lets you launch the Geography Network. If you want to, you can have that button launch your Metadata Explorer instead. Locate your aimsclient.properties file. On Windows, this file resides in your Profiles directory; on UNIX, it resides in your login directory. Edit the file in a text editor and change the GNHome parameter.

```
# GNHome - (optional) Location of the Geography
# Network portal for

GNHome=http://<your server>/metadataexplorer
```

This will replace Geography Network with Metadata Explorer in your ArcExplorer.

ArcSDE considerations

B

IN THIS APPENDIX

- **Specifying ArcSDE connection information**
- **Metadata Service database schema**
- **Removing Metadata Service objects from the database**
- **Configuring an Oracle database**
- **Configuring a DB2 database**
- **Configuring a SQL Server database**
- **Configuring ArcSDE**
- **Increasing the ArcIMS time-out for ArcSDE**
- **Getting the best performance from the database**

ArcIMS relies on the capabilities of ArcSDE and a relational database to store, search for, retrieve, and index metadata documents. Whenever ArcIMS receives a request to perform an operation, such as finding a metadata document or publishing a new document, the request is translated into SQL and sent through ArcSDE to the database. The result of the operation is passed back from the database and out through ArcIMS. For example, the result might list the metadata documents that matched the search criteria or indicate the success or failure of publishing a document.

The database is a critical component of a Metadata Service. As the database administrator, you must ensure the database is configured properly to support text indexing before a Metadata Service can be started. You also have some control over how the Metadata Service's tables are stored in the database and how documents published to the metadata service are indexed. There are some additional ArcSDE configuration and maintenance topics in this appendix that you should be aware of in connection with ArcIMS Metadata Services.

An ArcSDE XML column is used to store metadata documents that are published to a Metadata Service. ArcSDE XML columns are supported with Oracle, SQL Server, and DB2 databases and require text indexing functionality to be installed and configured in those databases. Metadata Services are not supported with Informix databases.

Specifying ArcSDE connection information

The first time a Metadata Service is started, ArcIMS creates a set of objects in the relational database to store and manage metadata documents. For ArcIMS to create these objects, you must provide ArcSDE connection information in the configuration file for the Metadata Service. This includes the name of the ArcSDE server; the instance; the username; the password; and if appropriate, the database. For example, with a SQL Server or DB2 database, your configuration file will have an entry like this:

```
<SDEWORKSPACE name="aName" server="sdeServer"
instance="port:5151" database="sde" user="mdUser"
password="mdPassword" />
```

A sample configuration file, MetadataServer.axl, is provided to illustrate how a Metadata Service should be defined.

The user specified in the configuration file will own the Metadata Service's objects; this user must be granted the same level of database permissions as other GIS users on your system. For more information on creating users, see the *ArcSDE Configuration and Tuning Guide* for your database.

Metadata Service database schema

For each Metadata Service, one feature class, or ArcSDE layer, and four tables are created in the database; for convenience, these objects are often referred to as a set of tables. These objects are named using the table name prefix specified in the Metadata Service's configuration file. For example, your configuration file will have an entry like this:

```
<TABLE_NAME prefix="imsmetadata" />
```

You can specify any prefix you like in the configuration file. The length of the prefix is ultimately limited by the length of table names allowed by the relational database you're using and will vary. In general, keep the name short—15 characters or less.

When you start a Metadata Service, ArcIMS checks to see if objects with the given prefix exist in the database. If they don't exist, the objects are created; if they do exist, the existing objects are used.

Given a prefix of "imsmetadata", the following objects will be created in the database:

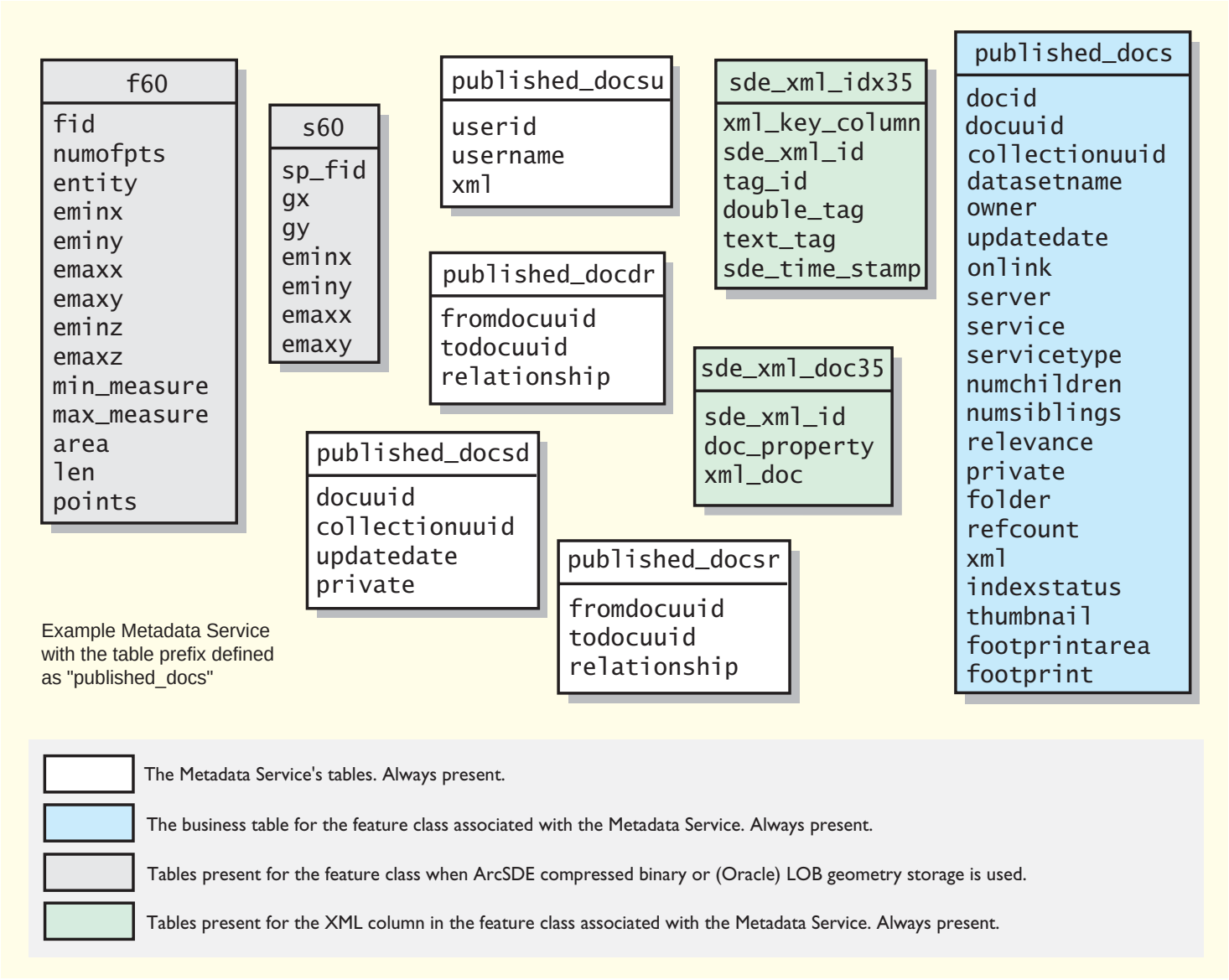
- IMSMETADATA
- IMSMETADATAD
- IMSMETADATADR
- IMSMETADATAR
- IMSMETADATAU

"IMSMETADATA", or in other words, "<prefix>", in the list above is the feature class. The other objects in the list are the tables created by ArcIMS: "<prefix>D" is the deleted table, "<prefix>DR" is the deleted relationships table, "<prefix>R" is the relationships table, and "<prefix>U" is the users table. The database schema for each object is described below.

An ArcSDE XML column is used in the feature class to store published metadata documents similar to the way the feature class's Geometry column stores feature data. To store and manage the feature data and XML documents, ArcSDE uses several side tables in addition to the objects listed above; these ArcSDE side tables are described in detail in the *ArcSDE Configuration and Tuning Guide* for your database.

Business table: <prefix>

The business table associated with the Metadata Service's feature class, or ArcSDE layer, contains one record for each folder or document. An additional record represents the Metadata Service itself; this is the service's root document. Each record has an XML document, a feature representing the extent in decimal



Example physical schema for a single Metadata Service

degrees of the resource described by the document, and additional information about the document.

Some search criteria are evaluated using the information in this table rather than the information stored in the document. Information in this table is also used to assemble the responses generated by the Metadata Service.

The columns in the business table are described below. The ArcSDE data type follows the column's name.

DocID (SE_INT32)—The table's primary key.

DocUUID (SE_UUID)—The globally unique identifier (GUID) that uniquely identifies a published metadata document. A foreign key reference used to join the business table to the relationships, deleted, and deleted relationships tables and to an administrative table, if one exists.

When a document is published, a GUID is generated if one doesn't already exist in the PublishedDocID element; the new GUID is stored in the original document, if possible.

CollectionUUID (SE_UUID)—The GUID that uniquely identifies the Metadata Service to which the document was originally published. This column supports harvesting. The CollectionUUID for all documents directly published to a Metadata Service will match that service's GUID; documents harvested from another Metadata Service will have a different CollectionUUID.

A Metadata Service's GUID is defined when its tables are created. If a service's tables are deleted and recreated, a new GUID will be generated even if the configuration file is unchanged. If the MDUpgrade command is used, the Metadata Service's GUID won't change.

DatasetName (SE_STRING)—The name of the published document; "Home" for the root document. The default name for a document is the title defined in the metadata, if one exists. Otherwise, a default name will be derived from the object that was published. A document can be renamed using ArcCatalog—this won't change the title in the metadata document.

Owner (SE_INT32)—The user who published the document. This is a user defined in the access control list used by ArcIMS. A foreign key reference used to join the business table to the Users table.

UpdateDate (SE_INT32)—The date and time when the document was last published or edited. This value is the number of seconds elapsed since midnight, January 1, 1970, coordinated universal time, according to the system clock.

Onlink (SE_STRING)—The location of the resource or information about the resource described by the document. This value may be a URL; it's derived from the Online Linkage element in the document.

Server (SE_STRING)—The URL of the ArcIMS server. This column is only used when the document describes an ArcIMS Image, Feature, or WMS service.

If the document is published by dragging and dropping the service in ArcCatalog, this information is derived directly from the service. If a standalone XML document is published, such as for WMS services, this information is derived from the Server element in the document.

Service (SE_STRING)—The name of the ArcIMS service. This column is only used when the document describes an ArcIMS Image, Feature, or WMS Service or an ArcIMS feature class. For feature classes, the value is its name; its type, for example, polygon; and the name of its Feature Service.

If the document is published by dragging and dropping the service or feature class in ArcCatalog, this information is derived directly from the object. If a standalone XML document is published, such as for WMS services, this information is derived from the Service element in the document.

ServiceType (SE_STRING)—The content type of the resource described by the document. The content type codes are:

- A—Live Data and Maps referencing an ArcIMS Image Service
- B—Live Data and Maps referencing an ArcIMS Feature Service or ArcIMS Feature Class
- C—Live Data and Maps referencing an ArcIMS WMS Service
- D—Live Data and Maps specified in the document but the object published wasn't an ArcIMS service
- E—Downloadable Data
- F—Offline Data
- G—Static Map Images
- H—Other Documents
- I —Applications
- J—Geographic Services
- K—Clearinghouses
- L—Map Files
- M—Geographic Activities
- U—Unclassified

NumChildren (SE_INT32)—The number of documents and folders contained by a folder.

NumSiblings (SE_INT32)—The number of documents that are related to this document. Related documents can be defined from a document's Properties dialog box in ArcCatalog.

Relevance (SE_INT32)—The relevance of the document.

Private (SE_INT32)—Indicates if a document is private. Private documents are only accessible to the user who owns them. By default, documents are publicly available.

Folder (SE_INT32)—Indicates if a document is a document, a folder, or the root document. Folders are documents that can contain other documents.

RefCount (SE_INT32)—The number of references to a document that exist in the Metadata Service. If a document is published to one folder and then copied to another folder, there will be two references to that document.

Xml (SE_XML)—An XML column. A foreign key reference used to join the business table to the XML document table, which contains the published metadata documents.

IndexStatus (SE_INT32)—Currently not used.

Thumbnail (SE_BLOB)—An image in JPEG format that illustrates the resource described by the document.

FootprintArea (SE_FLOAT)—The area of the document's footprint.

Footprint (SE_SHAPE)—A spatial column. A foreign key reference used to join the business table to the feature table, which contains the footprints of the resources described by the metadata documents.

Each footprint represents the extent in decimal degrees of the resource; the rectangle's coordinates are defined within the document. This column supports spatial searches for published metadata documents.

Deleted table: <prefix>D

The deleted table records which documents have been deleted from the Metadata Service. This table supports harvesting. When a national Metadata Service harvests a state Metadata Service, documents that have been deleted from the state service since the last update will be removed from the national service.

The columns in the deleted table are described below. The ArcSDE data type follows the column's name.

DocUUID (SE_UUID)—The GUID that identifies which document was deleted from the Metadata Service.

CollectionUUID (SE_UUID)—The GUID that identifies the Metadata Service to which the document was originally published.

UpdateDate (SE_INT32)—The date and time when the document was deleted. This value is the number of seconds elapsed since midnight, January 1, 1970, coordinated universal time, according to the system clock.

Private (SE_INT32)—Indicates if a document was private.

Deleted relationships table: <prefix>DR

The deleted relationships table records which relationships no longer exist in the Metadata Service. This table supports harvesting. When a national Metadata Service harvests a state Metadata Service, relationships that have been deleted from the state service since the last update will be removed from the national service.

The columns in the deleted relationships table are described below. The ArcSDE data type follows the column's name.

FromDocUUID (SE_UUID)—The GUID that identifies the From document in the deleted relationship.

ToDocUUID (SE_UUID)—The GUID that identifies the To document in the deleted relationship.

Relationship (SE_INT32)—The relationship between the From document and the To document, which was deleted. The codes used in this column are:

- 1—Parent–Child
- 2—Sibling
- 3—Ancestor–Descendant

For example, if a document named "population" was moved from folder A to folder B, the parent–child relationship between folder A (the To document) and "population" (the From document) was deleted and will be recorded in this table. The parent–child relationship between folder B and "population" currently exists and will be recorded in the relationships table.

Relationships table: <prefix>R

The relationships table stores the relationships between documents, folders, and the Metadata Service. There are three types of relationships: parent–child, ancestor–descendant, and sibling.

For a document in a folder, there is a parent–child relationship between the folder and the document. When folder A contains folder B, documents in folder B have an ancestor–descendant relationship with folder A. All documents have either a parent–child or an ancestor–descendant relationship with a Metadata Service. Child and descendant relationships are added when documents are published, when folders are created, or when they are copied or moved.

Sibling relationships are added when two documents are defined as being related to each other, for example, using a document's Properties dialog box in ArcCatalog. Sibling relationships are logical and manually defined based on document content; they are not related to the hierarchical structure used to browse documents in a Metadata Service. For a document describing soil test data, a document describing a soil map might be specified as a related document. The soil map document becomes a sibling of the soil test document; the reverse relationship will not be assumed—the soil test document won't automatically become a sibling of the soil map document. In the Metadata Explorer, links to a document's siblings appear at the bottom of the Details page.

The columns in the relationships table are described below. The ArcSDE data type follows the column's name.

FromDocUUID (SE_UUID)—The GUID that identifies the From document in the relationship. For parent–child and ancestor–descendant relationships, the folder or Metadata Service is the From document. For sibling relationships, this is the source document.

ToDocUUID (SE_UUID)—The GUID that identifies the To document in the relationship. For parent–child and ancestor–

descendant relationships, the contained document or folder is the To document. For sibling relationships, this is the target document for the link that appears when you examine the source document in the Metadata Explorer.

Relationship (SE_INT32)—The relationship between the From and To documents. The codes used in this column are:

- 1—Parent–Child
- 2—Sibling
- 3—Ancestor–Descendant

For example, if a folder “California” exists at the root level of a Metadata Service, “California” has a parent–child relationship with the Metadata Service. If a document “Redlands streets” exists in the folder “California”, “Redlands streets” has a parent–child relationship with “California” and an ancestor–descendant relationship with the Metadata Service.

Users table: <prefix>U

The users table stores the names of all users who have sent requests to the Metadata Service. Users may have browsed or searched the contents of the Metadata Service, or they may have published, copied, moved, or deleted documents. These users are defined in the access control list used by the ArcIMS server that hosts the Metadata Service.

The columns in the users table are described below. The ArcSDE data type follows the column’s name.

UserID (SE_INT32)—The table’s primary key.

UserName (SE_STRING)—The users who have sent requests to the Metadata Service.

Xml (SE_BLOB)—Currently not used.

Removing Metadata Service objects from the database

There are three ways to remove a Metadata Service’s objects from the database: in ArcCatalog, using ArcSDE commands, and sending the ArcXML RESET request.

Regardless of the method you use, before deleting the Metadata Service’s objects, you must remove any views you may have created on those objects. For example, you might have created a view on the feature class associated with a Metadata Service to export and customize the gazetteer data.

In ArcCatalog, create a database connection using the ArcSDE connection information specified in the SDEWORKSPACE element in the Metadata Service’s configuration file. Delete any views associated with the service, then select and delete the five objects whose names begin with the TABLE_NAME prefix specified in the configuration file. For example, if the prefix is “Gazetteer”, delete views, such as “Gazetteer_v”, then delete the feature class “Gazetteer” and the tables “GazetteerD”, “GazetteerDR”, “GazetteerR”, and “GazetteerU”.

Deleting the Metadata Service’s objects can also be accomplished from the operating system’s command line using the sdetable -o delete command; the arguments are:

```
sdetable -o delete -t <table_name> -i <service>
-s <server_name> -u <DB_User_name>
-p <DB_User_password> -D <database>
```

Specify the same connection information used in the service’s configuration file for the appropriate arguments. For example, if the prefix for the Metadata Service is “IMS”, delete the feature class like this:

```
sdetable -o delete -t ims -i 5151 -s sdeServer
-u aUser -p aPassword -D sde
```

To delete the Metadata Service’s tables, run the same command four more times, changing the table name each time to “IMSD”, “IMSDR”, “IMSR”, and “IMSU”.

After using either ArcCatalog or the ArcSDE commands to delete the objects, the next time the Metadata Service is started, ArcIMS will re-create the objects in the database.

Instead of using ArcCatalog or ArcSDE commands, you can send the ArcXML RESET request to the Metadata Service. For example, you might do this if you're managing the service using a custom Web application. This request deletes and re-creates the Metadata Service's objects. For more information about the RESET request, see the *ArcXML Programmer's Reference Guide*.

Regardless of the method you use to delete the objects, when the Metadata Service's objects are re-created, the service will have a new root document and, therefore, a new collection identifier. If another Metdata Service harvests your documents, the next time documents are imported, their collection identifier will change in the harvester's repository. This may affect the harvester's strategy for searching and managing harvested documents.

Configuring an Oracle database

ArcSDE XML columns require Oracle Text for Oracle9*i*™ or InterMedia Text for Oracle8*i*™ to be installed and configured. The database administrator must accomplish step 1 and step 2 below before a Metadata Service can be started.

Step 1: Install the text component.

1. Follow the instructions in the Oracle documentation to install the appropriate text component for your version of the database.

   Oracle Text is installed by default with a typical installation of Oracle9*i*. However, if you upgraded an earlier version of the database and InterMedia Text was not installed, Oracle Text will not be installed as part of the upgrade; it must be installed as an additional component after the upgrade is completed.

2. Follow the instructions in the Oracle documentation to verify the text component has been installed and is working properly.

Step 2: Grant privileges for the Metadata Service.

After logging in to SQL*Plus as the ctxsys user, you must grant privileges to the user who will own the Metadata Service by issuing the command:

```
sql>grant execute on ctx_ddl to <mdUsername>;
```

where <mdUsername> is the user who will own the Metadata Service's objects.

Step 3: Set the optimizer parameters.

Metadata Services require a few optimizer parameters to be set in the database for searches to perform well. Setting these parameters will not adversely affect other client software that accesses your ArcSDE database. Once this has been accomplished, review the section 'Configuring ArcSDE' later in this appendix.

Oracle8*i*

For Oracle8*i*, the following parameters must be added to your Oracle server's init.ora file:

```
OPTIMIZER_INDEX_CACHING = 90
OPTIMIZER_INDEX_COST_ADJ = 10
OPTIMIZER_MODE=CHOOSE
```

Refer to the Oracle8*i* documentation for a complete discussion of these parameters. In general, setting these parameters as specified directs the Oracle8*i* optimizer to favor index scans and nested loop execution plans over a full table scan or a sort merge.

Oracle9*i*

For Oracle9*i*, the following parameter must be added to your Oracle server's init.ora file:

```
OPTIMIZER_MODE=CHOOSE
```

Only the optimizer mode must be set; the other parameters used with Oracle8*i* aren't required.

Configuring a DB2 database

ArcSDE XML columns require the DB2 Net Search Extender to be installed and configured. The database administrator must accomplish steps 1 and 2 below before a Metadata Service can be started.

Step 1: Install the Net Search Extender.

1. Follow the instructions in the IBM documentation to install DB2 Net Search Extender.
2. Install the Net Search Extender FixPak 4, which is available for download from http://www-1.ibm.com/support/docview.wss?rs=0&q=DB2+Net+Search+Extender&uid=swg24005753&loc=en_US&cs=utf8&cc= us&lang=en.
3. Follow the instructions in the IBM documentation to verify the extender has been installed and is working properly.

Step 2: Enable the database for Net Search Extender.

Once the DB2 Net Search Extender and FixPak 4 have been installed and verified, you must enable a database for Net Search Extender in the same way that you enable a database for Spatial Extender. This can be accomplished either using the control center or by issuing DB2 commands from the operating system's command line. A database only needs to be enabled once, before the first ArcSDE XML column is created in that database.

Control center

In the DB2 Control Center, right-click the database in which the Metadata Service's objects will be created, point to Net Search Extender, and click Enable Database for Text.

Once this has been accomplished, review the section 'Configuring ArcSDE' later in this appendix; you do not need to issue additional commands from the command line to enable the database.

Command line

If you choose not to enable the database using the control center, you can do so from the operating system's command line following the steps below.

1. Start the Net Search Extender Instance Service as follows:

   ```
   db2text "START"
   ```

2. Enable a database for Net Search Extender as follows:

   ```
   ENABLE DATABASE FOR TEXT CONNECT TO <sde>
   ```

   where "<sde>" is the name of the database in which the Metadata Service's objects will be created.

Review the section 'Configuring ArcSDE' later in this appendix.

Configuring a SQL Server database

ArcSDE XML columns require the full text search engine to be installed and configured. The database administrator must accomplish steps 1 through 3 below before a Metadata Service can be started. The full text search engine is not supported with Microsoft SQL Server Desktop Engine (MSDE).

Step 1: See if the full text search engine is installed.

The full text search engine is installed by default with a typical installation of SQL Server 2000. However, if you upgraded SQL Server and full text search was not installed in SQL Server 7.0, full text search will not be installed as part of the upgrade; it must be installed as an additional component after the upgrade is completed.

Find out if the full text search engine has been installed properly by running the following query using the Query Analyzer:

```
SELECT
fulltextserviceproperty('IsFulltextInstalled')
```

If the value 0 or an error is returned, you must install the full text search engine following the instructions in step 2. If the value 1 is returned, the full text search engine is installed; skip to step 3.

Step 2: Install the full text search engine.

If the full text search engine hasn't already been installed, install it by rerunning the SQL Server installation.

1. Run the SQL Server setup.
2. Click SQL Server 2000 Components.
3. Click Install Database Server.
4. Click Next.
5. Specify the computer on which the full text search engine will be installed. By default it will be installed on the local computer, but you can specify a remote computer instead. Click Next.
6. Click Upgrade, remove, or add components to an existing instance of SQL Server and click Next.
7. Specify the instance on which the full text search engine will be installed. By default it will be installed on the default SQL Server instance, if one is detected on the computer, but you can specify a named instance instead. Click Next.
8. Click Add components to your existing installation and click Next.
9. Click Server Component in the Components list on the left, then check Full-Text Search in the Sub-Components list on the right. Leave all other options the same. Click Next.
10. Follow the remaining instructions to complete the installation.

Once the installation has completed, verify that the component was installed correctly by running the query specified in step 1.

Step 3: Create a full text catalog.

Once the full text search engine has been correctly installed, you can use either the Enterprise Manager or the Query Analyzer to create a full text catalog in the database in which the Metadata Service's objects will be created.

Enterprise Manager

In Enterprise Manager, create a full text catalog following the steps below.

1. In the tree, click the plus sign (+) next to Microsoft SQL Servers, then SQL Server Groups, then the appropriate SQL Server, then the plus sign next to Databases.
2. Right-click the database in which the Metadata Service's tables will be created.

3. Point to New and click New Full-Text Catalog.
4. Type a name for the full text catalog into the Name text box. If you provide a name other than “SDE_DEFAULT_CAT”, you must modify the XML_IDX_FULLTEXT_CAT keyword in the SDE_dbtune table; see the section ‘Configuring ArcSDE’ later in this appendix for more information.
5. Click the Browse button (...), navigate to the location where you want the full text catalog to be stored, and click OK.
6. Click OK in the New Full-Text Catalog Properties dialog box.

Once a full text catalog has been created, review the section ‘Configuring ArcSDE’; you do not need to issue additional commands using Query Analyzer to enable the database or create a full text catalog.

Query Analzyer

If you choose not to create the full text catalog using Enterprise Manager, you can do so using Query Analyzer following the steps below.

1. In Query Analyzer, select the database in which the Metadata Service’s objects will be created using the dropdown list on the toolbar or by executing the USE command. For example:

   ```
   USE sde
   ```

2. Enable full text indexing in the database by issuing the sp_fulltext_database command. The value of the action parameter must be “enable”. For example:

   ```
   EXEC sp_fulltext_database 'enable'
   ```

   Use this command carefully. A database only needs to be enabled once, before the first full text catalog is created in the database. If the database is already full text enabled and this command is run again, existing full text catalogs will be dropped and recreated.

3. Create a full text catalog in the database by issuing the sp_fulltext_catalog command. The value of the ftcat parameter defines the catalog’s name. If you provide a name other than “SDE_DEFAULT_CAT”, you must modify the XML_IDX_FULLTEXT_CAT keyword in the SDE_dbtune table; see the following section ‘Configuring ArcSDE’ for more information. The value of the action parameter must be “create”. Optionally, the path parameter can be used to set the location on disk where the full text catalog will be created. For example:

   ```
   EXEC sp_fulltext_catalog 'SDE_DEFAULT_CAT',
   'create', 'C:\Fulltext'
   ```

Review the section ‘Configuring ArcSDE’.

Configuring ArcSDE

This section describes how you should configure ArcSDE before you start a Metadata Service. The information below applies to all databases, unless otherwise specified.

Publishing metadata documents with large thumbnails

A metadata document’s thumbnail is stored in a BLOB column in the Metadata Service’s feature class. Thumbnails created using ArcCatalog are embedded within a metadata document and are very small; there will be no trouble publishing documents with these types of thumbnails.

When a document is published using ArcCatalog and a thumbnail is not embedded within a metadata document, the document’s content will be checked to see if there is a reference to an external image file; specifically, the FGDC metadata element Browse Graphic File Name will be checked. If a filepath or HTTP address is found that references a JPEG image and ArcCatalog can access the image, it will be loaded into the Metadata Service as the document’s thumbnail. ArcCatalog will only attempt to access the first JPEG image referenced in the metadata.

External image files referenced by a metadata document may be large. If a referenced image file is larger than approximately one

megabyte in size, publishing the document to the Metadata Service may fail. You will need to adjust the default value of the MAXBLOBSIZE parameter in ArcSDE if you want to support publishing larger images as thumbnails.

The MAXBLOBSIZE parameter can be found in the giomgr.defs file, located in the etc directory of your ArcSDE installation. For information on how to adjust this parameter, see the section on managing BLOB data in Chapter 3, 'Configuring ArcSDE Services', of *Managing ArcSDE Services*. You may also want to refer to Appendix E, 'ArcSDE Initialization Parameters', in the same guide.

Metadata Service DBTUNE keywords

DBTUNE configuration keywords let you control how objects are created in the database. They are stored in the SDE_dbtune table and specify an object's properties, such as where a table will be stored and how it will be indexed.

You don't have to edit any DBTUNE parameters to start a Metadata Service, but you may want to do so to specify where the service's objects will be stored, for example. When ArcIMS creates a Metadata Service's objects in the database, the DEFAULTS keyword in the SDE_dbtune table is used to specify the properties of those objects. It isn't possible to specify a different DBTUNE keyword to use with a Metadata Service. If you want to edit the parameters used to create a Metadata Service, you must make those changes to the DEFAULTS keyword. However, you can specify a DBTUNE keyword if you migrate your Metadata Service to ArcIMS 9 using the MDUpgrade command; follow the instructions in Appendix E.

Keep in mind that DBTUNE configuration keywords are only used when objects are created in the database. If you've already started a Metadata Service, edits to the SDE_dbtune table will only be used if you create a second Metadata Service or if you remove a Metadata Service's tables from the database and restart the service.

After a Metadata Service has been created, it is possible to specify a DBTUNE keyword for the ArcSDE side tables associated with a Metadata Service's XML column. This is important if you need to change the XML parameters that control how and when the published documents are indexed. A few of the XML parameters are discussed in the following sections; for more information about them, see the *ArcSDE Configuration and Tuning Guide* for your database.

To specify a DBTUNE keyword for the ArcSDE XML tables, you must drop and rebuild those tables using the MDAlterIndex command—provide the appropriate keyword in the config_keyword argument. Then, run the aimsmetaindx command to rebuild the index in the database so the documents can be found by a search. For more information about these commands, see Chapter 3, 'Creating Metadata Services', in this book.

For more information about DBTUNE keywords and their parameters, see the chapter 'Configuring DBTUNE storage parameters' in the *ArcSDE Configuration and Tuning Guide* for your database.

Gazetteer DBTUNE keywords

By default, several IMS_METADATA keywords are included in the SDE_dbtune table. The gazetteer data is provided with ArcIMS in sdeexport format. The data is loaded into ArcSDE using the gazetteerImport script, which uses ArcSDE commands to load and index the data. Three of the IMS_METADATA keywords are used by the gazetteerImport script: IMS_METADATA for the feature class, "gazetteer"; IMS_METADATARELATIONSHIPS for the table, "gazetteerr"; and IMS_METADATAUSERS for the table, "gazetteeru".

The remaining keywords are obsolete because the database schema for Metadata Services has changed. The obsolete keywords are:

- IMS_METADATATAGS
- IMS_METADATATHUMBNAILS

- IMS_METADATAVALUES
- IMS_METADATAWORDINDEX
- IMS_METADATAWORDS

The gazetteer feature class contains 350,000 features. Before loading the gazetteer data, make sure these objects will be stored where there is enough room for them. You may need to edit the IMS_METADATA keywords so that the objects are stored in a different location than specified in DEFAULTS.

Because keywords aren't provided by default for the deleted and deleted relationships tables, "gazetteerd" and "gazetteerdr", respectively, the gazetteerImport script loads these tables into ArcSDE using the DEFAULTS keyword. If you modify the IMS_METADATA keywords, you may want to edit the gazetteerImport script to use the IMS_METADATARELATIONSHIPS keyword or another custom keyword to load the deleted and deleted relationships tables.

For more information about the gazetteer, see Appendix A, 'Configuring Metadata Explorer', in this book.

DB2 DBTUNE XML keywords

DB2 has a known limitation for the Net Search Extender that affects how documents published to an ArcSDE XML column are indexed. Currently, only a user with DB2ADM privileges can index documents.

The Metadata Service's configuration file has an index_words attribute on the METADATA_CONTENT element. When set to "automatic", the default, a command is sent to the database to update the text index immediately after the documents are published. When set to "manual", the text index is updated when the aimsmetaindx command is run from the command line.

The user who owns the Metadata Service's objects in the database generally won't have DB2ADM privileges. Because of this known limitation, the "automatic" setting in a Metadata Service's configuration file won't work. Also, with the "manual" setting, only a user with DB2ADM privileges will be able to run the aimsmetaindx command to update the text index. The arguments for the aimsmetaindx command are:

```
aimsmetaindx <sde_server_machine> <sde_instance>
<database> <username> <password>
{table_name_prefix} {logfile} {localcodepage}
```

If the DB2ADM user runs this command to index published documents, the table_name_prefix argument must be specified and it must be prefaced with the user who owns the Metadata Service's objects. For example:

```
aimsmetaindx sdeServer 5151 sde sdeUser
sdePassword aUser.imsmetadata
```

One solution for this problem is for the DB2ADM user to schedule the aimsmetindx command to run periodically, for example using a cron job. Another solution is to set the XML_IDX_FULLTEXT_UPD_FREQUENCY parameter in the SDE_dbtune table to set a schedule in the database for updating the text index. For all documents published or updated during the day to be indexed each night, for example, you should not specify a minimum number of documents with the XML_IDX_FULLTEXT_UPD_MINIMUM parameter.

For information about how to set the XML_IDX_FULLTEXT_UPD_FREQUENCY and XML_IDX_FULLTEXT_UPD_MINIMUM parameters in the SDE_dbtune table, see the documentation for DB2 Net Search Extender. You should review these parameters before creating a Metadata Service.

See Chapter 3, 'Creating Metadata Services', in this book for more information about the index_words attribute in the configuration file and the aimsmetaindx command.

SQL Server DBTUNE XML keywords

The XML parameters in the SDE_dbtune table affect how documents published to an ArcSDE XML column are indexed by the database. For SQL Server databases, the following XML

parameters should be reviewed before creating a Metadata Service: XML_IDX_FULLTEXT_CAT and XML_IDX_FULLTEXT_UPDATE_METHOD. Also, if your Metadata Service will store documents written in a language other than English, you will want to review the XML_IDX_FULLTEXT_LANGUAGE parameter in addition to verifying the language settings in the database.

XML_IDX_FULLTEXT_CAT

Verify that the value of the XML_IDX_FULLTEXT_CAT parameter is the same as the full text catalog's name. If the full text catalog's name is different, update the value in the SDE_dbtune table to match.

A multidatabase instance of ArcSDE has one ArcSDE instance, one central Spatial Database Engine™ (SDE®) database, and all non-sde databases containing spatial data are dependent upon the sde database. If you have a multidatabase instance of ArcSDE, all spatial databases that will contain a Metadata Service or another object with an ArcSDE XML column must have a full text catalog. In this configuration, each spatial database references the central SDE_dbtune table, thus each full text catalog's name must match the value of the central XML_IDX_FULLTEXT_CAT parameter.

If you have a single spatial database instance of ArcSDE (where there is one SQL Server database per ArcSDE instance), the name of the full text catalog in each database must match the value of the XML_IDX_FULLTEXT_CAT parameter in the SDE_dbtune table within that database.

For Metadata Services, changes to the XML_IDX_FULLTEXT_CAT parameter must be made for the DEFAULTS keyword.

XML_IDX_FULLTEXT_UPDATE_METHOD

The frequency and method with which the full text catalog will be updated in the database needs to be considered based on the amount and rate of change of the documents published to your Metadata Service, the hardware resources available to the database, and whether or not it is important for a newly published document to be immediately available for a search. Documents can only be found by a search once they have been indexed by the database; that is, once the full text catalog has been updated.

Two different settings affect how documents are indexed: one is in the Metadata Service's configuration file, and the other is in the SDE_dbtune table.

The Metadata Service's configuration file has an index_words attribute on the METADATA_CONTENT element. When set to "automatic", the default, a command is sent to the database to update the full text catalog immediately after the documents are published. When set to "manual", the full text catalog is updated when the aimsmetaindx command is run from the command line. See Chapter 3, 'Creating Metadata Services', in this book for more information about these settings.

In the SDE_dbtune table, if the value of the XML_IDX_FULLTEXT_UPDATE_METHOD parameter is "CHANGE_TRACKING MANUAL", the database expects the full text catalog to be updated manually. With this DBTUNE setting, the "automatic" or "manual" setting in the configuration file will work and the indexing behavior of the various Metadata Service commands will be as expected. The full text catalog will be updated when the Metadata Service sends the command to do so.

However, by default the XML_IDX_FULLTEXT_UPDATE_METHOD parameter is set to "CHANGE_TRACKING BACKGROUND". This instructs the database to manage when the full text catalog is updated and to do so using a background process; the only control you have is over the priority of the background process in the database. With this DBTUNE setting, the "automatic" or "manual" setting in the configuration file is irrelevant, and all Metadata Service commands that add or modify documents will trigger background indexing.

Background updates will be affected by the resources available on the database server, the priority of the background process, and the load of other operations on the database. Manual updates will have a higher priority in the database, and thus will probably be faster, but they will affect the performance of searches and other operations in the database.

If a moderate number of updates are made daily to a Metadata Service and updated documents don't need to be immediately available for searches, you may get the best combination of performance and convenience by accepting the default settings in both the SDE_dbtune table and the Metadata Service's configuration file: "CHANGE_TRACKING BACKGROUND" and "automatic". While documents are indexed in the background, repeated searches with the same criteria may find more documents each time the search is issued.

If only a few documents are updated at a time, the database has adequate resources, and it is important for published documents to be immediately available for searching, set the SDE_dbtune table to "CHANGE_TRACKING MANUAL" and index_words in the configuration file to "automatic". Publishing large sets of documents should be planned during periods of low database activity.

If documents are published frequently, large numbers of documents are published at once, and it is imperative to maintain the best performance for searches against your Metadata Service, set the SDE_dbtune table to "CHANGE_TRACKING MANUAL" and index_words in the configuration file to "manual". Updates to the full text catalog should be scheduled during windows for database maintenance, for example, nightly. Either run the aimsmetaindx command manually when necessary or schedule it to run periodically with the operating system's tools, such as a cron job. Or use the SQL Server database's tools to schedule when the full text catalog will be updated.

Review the SQL Server Books Online topics 'Full Text Indexing Support', 'Maintaining Full Text Indexes', 'Full Text Search Recommendations', and 'Performing Infrequent Tasks' for more information about updating full text catalogs. ESRI recommends that you always use change tracking with your full text catalog to ensure the best performance for indexing XML documents.

XML_IDX_FULLTEXT_LANGUAGE

If your Metadata Service will store documents written in a language other than English, you may need to set the XML_IDX_FULLTEXT_LANGUAGE parameter in the SDE_dbtune table. The XML_IDX_FULLTEXT_LANGUAGE parameter is used to set the language of the column that is indexed in the full text catalog. This setting affects the linguistic analysis used to index the text in the published documents—if the published documents are written in Japanese but the language of the column is set to English, the text won't be indexed properly and you will have trouble finding documents with a search.

If you are using a localized version of SQL Server, the default full text language option will be set to the language of the server. For example, if you are using a Japanese version of SQL Server, the language of the database, the full text catalog, and the column to be indexed should all be set to Japanese because the default full text language option will be set to Japanese. You should not have to change the value of the XML_IDX_FULLTEXT_LANGUAGE parameter in the SDE_dbtune table.

However, if you are using an international version of SQL Server, the default full text language option will be set to U.S. English. Even if you specifically set the language of the database and the full text catalog to Japanese, the language of the column that will be indexed will default to U.S. English. Set the XML_IDX_FULLTEXT_LANGUAGE parameter in the SDE_dbtune table to the appropriate value for the language of the published documents; this value is used to set the language of the column. See the SQL Server Books Online help topic 'sp_fulltext_column' for the appropriate value; for example, for Japanese you would set the value of the

XML_IDX_FULLTEXT_LANGUAGE parameter in the SDE_dbtune table to be “0x0411”.

When a column is set to a specific language, such as Japanese, you will be able to publish and search documents in both Japanese and English. When a Metadata Service will contain documents in many languages or an unsupported language, the language of the column to be indexed should be set to neutral; you may have trouble searching with some text.

See the SQL Server Books Online help topics ‘default full-text language Option’, ‘Column-Level Linguistic Analysis’, and ‘sp_fulltext_column’ for more information.

Increasing the ArcIMS time-out for ArcSDE

If ArcIMS and ArcSDE are running on the same machine, ArcIMS will start before ArcSDE when the machine is restarted. This means that ArcIMS will be up and have services trying to connect to the ArcSDE server before it has started. To prevent problems due to this lag time, the time-out parameters in the aimsdeio.cfg file should be modified.

The aimsdeio.cfg file is located at <ArcIMS installation directory>\server\etc. The contents of this file look like:

```
<DATAWORKSPACE
      tag="SDEWORKSPACE"
      description="sde data workspace"
      exe="DataAccess\aimssdeio.dll"
      timeout="15" />
```

The time-out parameter should be changed to “60”.

Getting the best performance from the database

The following discussions will help you keep your Metadata Service operating at peak performance.

Updating the database statistics

Over time, as people publish documents to, update documents in, and delete documents from your Metadata Service, the statistics in the database for the Metadata Service’s objects will become stale. When this happens, the users of your Metadata Service may start to complain that searches are slow.

To keep your metadata database operating at peak performance, you should periodically update the statistics for the Metadata Service’s objects. You should do this whenever approximately 20 percent of the records in the Metadata Service have changed—because of adding, updating, or deleting documents.

Update statistics using the ArcSDE sdetable command from the operating system’s command line. Execute the appropriate version of this command for your database.

Oracle

```
sdetable -o update_dbms_stats -t <table_name>
-m compute -i <service> -s <server_name>
-u <DB_User_name> -p <DB_User_password>
```

DB2 and SQL Server

```
sdetable -o update_dbms_stats -t <table_name>
-i <service> -s <server_name> -u <DB_User_name>
-p <DB_User_password> -D <database>
```

Statistics must be updated for all the Metadata Service’s objects. Run the above command once for each object, changing the <table_name> argument to “<prefix>”, “<prefix>D”, “<prefix>DR”, “<prefix>R”, and “<prefix>U” each time; <prefix> is the table name prefix defined in the Metadata Service’s

configuration file. Modify the remaining arguments to provide the ArcSDE connection information.

For more information about the sdetable command, look at the Administrative Command References at ArcSDE Online; from *support.esri.com*, click Software > ArcSDE > ArcSDE Developer Help > Administrative Command References.

Fine-tuning performance in Oracle

As the number of documents published to your Metadata Service increases, you may begin to notice that searches take a bit longer. If you see a decrease in performance, run the script below. This script builds statistics that the database can use to increase performance.

There is no specific rule for when you should run this script. It will depend on the number of metadata documents published to your Metadata Service and how large those documents are. In general, you might run this if your database contains more than 3,000 metadata documents. However, running this script will not decrease the performance of smaller databases; you can run it at any time. This script is only applicable to Oracle8*i* and Oracle9*i* databases.

Run this script once for each user who owns a Metadata Service. Follow the steps below to create and run the script.

1. In a text editor, enter the following script. Replace "<insert owner name>" in the first line with the name of the user who owns the Metadata Service.

```
define owner=<insert owner name>

begin

dbms_stats.set_table_stats('&owner',
'sde_logfile_data', NULL, NULL, NULL, 13591,
15, 11, NULL);

dbms_stats.set_column_stats('&owner',
'sde_logfile_data', 'logfile_data_id', NULL,
NULL, NULL, 5, 0.2, 0, NULL, 3, NULL, NULL);

dbms_stats.set_column_stats('&owner',
'sde_logfile_data', 'sde_row_id', NULL, NULL,
NULL, 13565, 0.000073719, 0, NULL, 4, NULL,
NULL);

dbms_stats.set_index_stats('&owner',
'SDE_LOGFILE_DATA_IDX1', NULL, NULL, NULL,
13591, 42, 13591, 1, 1, 13, NULL, NULL, NULL);

dbms_stats.set_index_stats('&owner',
'SDE_LOGFILE_DATA_IDX2', NULL, NULL, NULL,
13591, 32, 13565, 1, 1, 59, NULL, NULL, NULL);

end;

/
```

2. Open a command window and connect to Oracle using SQL*Plus. For example:

```
sqlplus myuser/mypassword
```

3. Run the SQL script. For example, if the script is named "myscript":

```
@myscript
```

Harvesting documents

C

IN THIS APPENDIX

- **Collecting documents to create a community GIS portal**
- **Exporting and importing documents between Metadata Services**
- **Adding the OAI-PMH Connector to a Metadata Service**
- **Harvesting documents using the OAI-PMH Harvester**

When you're looking for data or other GIS resources, you may have to search several organizations' Metadata Services before you find what you need. A community can build a central GIS portal together to better serve its members. The host of the portal can collect information describing the resources available from many organizations and assemble this information into one Metadata Service that powers the GIS portal. This way, there is just one Metadata Service to search. However, the resources themselves would continue to be offered or hosted by the individual contributors, not the host of the central Metadata Service. This appendix describes the harvesting process and discusses the methods available to create a central Metadata Service with ArcIMS.

Collecting documents to create a community GIS portal

Every GIS uses data produced by different organizations. The goal of a spatial data infrastructure (SDI), such as the NSDI, is to promote sharing geographic information between organizations. Metadata Services are important in an SDI because they promote and enable the sharing of GIS resources.

People who know about a Metadata Service can use it to look for the resource they need. The problem is that they may search several sites before they find it. GIS portals are key because they collect relevant resources from many organizations and make them available to the community they serve. States, agencies, regional councils, and large urban centers can benefit from hosting a GIS portal.

A Metadata Service is one part of a community GIS portal; it lets people in the state, for example, more efficiently find the GIS resources they need. GIS portals and the Metadata Services behind them are discussed in more detail in the white paper *Implementing a Metadata Catalog Portal in a GIS Network*, available from *support.esri.com.*

The benefits of harvesting

There are two ways to build a collection of documents for a GIS portal. With *distributed searching*, each participating organization runs its own service to support searching its resources, such as a Z39.50 service. When a person searches the GIS portal, that search is sent to the participating services. Each service generates results and sends them to the portal, which collates and presents the results. This is how the NSDI Clearinghouse works.

With *harvesting*, documents describing resources are collected from the participating organizations and copies of them are stored and indexed in a central location. One central search service with access to the entire collection, supports the GIS portal. The advantages of this strategy include:

- Searches always have access to the entire collection, even if an individual organization's service becomes unavailable.
- Results can be generated more quickly without the additional Internet traffic required to access the database and without having to wait for the slowest participating service to respond before results can be generated.
- People have a common experience searching all services. The administrator of the portal can ensure that all information collected includes the information required to support the available searches.
- Participating services aren't searched every time people search the GIS portal. Individual servers don't have to manage additional load.
- With frequent harvesting, the information within the central service can be as up-to-date as if the participating services were searched directly.

Harvesting is a process, not a single event. The administrator of the central service, the *harvester*, and the contributing organization, the *provider*, can together determine how frequently the provider's documents should be harvested based on how often they change. After an initial transfer of documents, changes to the provider's service might be harvested daily, weekly, or quarterly.

Building a central Metadata Service

There are two ways to provide documents from an individual Metadata Service and harvest documents into a central Metadata Service. First, documents can be provided by exporting documents using the MDExport command and harvested by importing documents using the MDImport command. Second, documents can be provided using the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) Connector and

harvested using the OAI-PMH Harvester application. OAI-PMH was developed by the digital library community to promote interoperability between metadata collections and as an alternative to distributed search protocols, such as Z39.50.

ArcIMS Metadata Services internally track when documents are added, changed, and deleted; the changes in relationships as documents are moved from one folder to another; and when related document links are altered. With importing and exporting, the export files include information about all changes that have occurred in the provider's Metadata Service since a given date. After importing, the provider's documents in the central Metadata Service will mirror the provider's local Metadata Service.

However, the MDExport and MDImport commands don't interact with the Metadata Service through ArcIMS and ArcXML requests. Instead, they communicate directly with the ArcSDE database. Thus, these commands must be run from the operating system's command line on a machine that has access to the database over the local area network; they can't be run across the Internet. In addition, because these commands don't communicate with ArcIMS, authentication is not performed.

Therefore, the harvester can't directly harvest documents from the provider's Metadata Service. A harvesting arrangement must be established between the two organizations. The exporter can use a cron job, for example, to periodically generate export files and place them on an FTP site, for example. The harvester can also use a cron job to get the export files from the FTP site and import them to the central collection.

When the provider is using the OAI-PMH Connector, the harvester can access that service and retrieve the provider's documents over the Internet. The OAI-PMH Harvester lets the harvester retrieve documents that have been added or updated within a given time period. The documents can be directly published to the central Metadata Service or stored in a folder on the harvester's file system. Harvested documents might be stored in a folder so the documents can be published later, during off-peak hours. The OAI-PMH Harvester has the option to respect the folder structure that has been created in the provider's service or to ignore those folders.

The OAI-PMH Connector won't currently provide information to the harvester about the documents that have been deleted from the provider's service. Information about deleted relationships won't be translated to the central Metadata Service, for example, if a document is moved from one folder to another in the provider's collection. Also, information about related documents won't be sent to the central service. If documents in the provider's Metadata Service are completely restructured, the harvester could delete and reharvest all of the provider's documents.

OAI-PMH services exist that communicate with digital library archives rather than ArcIMS Metadata Services. Records describing library resources may be harvested into a central Metadata Service. However, library resources won't have a meaningful spatial extent and can only be found with full text searches. By the same token, with the OAI-PMH Connector added to an ArcIMS Metadata Service, records describing GIS resources may by harvested into a digital library collection. However, metadata describing GIS resources will likely lack codes used by libraries to support various searches.

Exporting and importing documents between Metadata Services

To share documents between two ArcIMS Metadata Services, one option is to export documents from one service and import them into another. For example, the administrator of a local Metadata Service might establish an arrangement with the administrator of a statewide Metadata Service to make information about local resources available to a wider audience.

The process for harvesting documents with this method is:

- The provider runs the MDExport command, which creates a set of files representing the individual Metadata Service's contents.
- The provider places the export files in a location accessible to the harvester, for example, an FTP site.
- The harvester transfers the export files to its system.
- The harvester runs the MDImport command to update the provider's documents in the central Metadata Service.

Importing and exporting might be preferable because it can be automated using a cron job or the at command, for example, if a significant effort has been made to establish relationships between the published documents, or if documents are frequently deleted or reorganized. The disadvantage is that the harvester can't directly harvest the provider's Metadata Service. The provider and harvester must generate and consume the export files at the expected intervals, otherwise there could be problems keeping the central Metadata Service up to date.

Exporting documents

The following steps detail how to export documents from your Metadata Service.

1. Open the operating system's command window and navigate to the location where the ArcIMS metadata commands are installed: <ArcIMS install directory>/Metadata/Commands.
2. Run the MDExport command to export documents from your Metadata Service. The usage for the MDExport command is:

   ```
   MDExport <output_directory>
   <sde_server_machine> <sde_instance> <database>
   <username> <password> {table_name_prefix}
   {documents_since_date}
   {include_imported_documents}
   {include_private_documents} {localcodepage}
   ```

   Mandatory arguments are shown in "<>" and optional arguments are shown in "{}". Optional arguments may be replaced with "#".

   You must specify the same ArcSDE connection information as in the Metadata Service's configuration file. The table_name_prefix argument must match the table prefix specified in the configuration file. If appropriate, make sure the localcodepage argument is set to "true".

   For example, with a SQL Server database, you would type the following on the operating system's command line to export all documents updated since January 1, 2004:

   ```
   mdexport exports\Q1_04 sdeServer 5151 aDatabase
   aUser aPassword tablePrefix 01/01/2004
   ```

   output_directory—The name and location of the directory in which the exported files will be placed; this directory must not already exist. By default, the output directory is created in the Commands directory. Either provide a full path or a path relative to the Commands directory. If there is a space in the path, enclose it within double quotes.

   sde_server_machine—The name of your ArcSDE server.

   sde_instance—The ArcSDE instance or port number. For example, 5151. Or the instance name, for example, "esri_sde".

   database—The ArcSDE database name. The database name is required for SQL Server and DB2 databases. For Oracle, this

information isn't appropriate; skip this parameter by specifying "#" instead.

username—The database username.

password—The password associated with the username.

table_name_prefix—The table name prefix specified in the Metadata Service's configuration file. By default, "imsmetadata".

documents_since_date—A date in the format MM/DD/YYYY. Only documents updated after the specified date will be exported. By default, all documents are exported.

include_imported_documents—"True" or "false"; false by default. When true, documents imported from another Metadata Service using the MDImport command will be included in the output. When imported, there will be no way to distinguish between documents that were published directly to the service and documents that were imported.

include_private_documents—"True" or "false"; false by default. When true, private documents are included in the output. When imported, there will be no way to distinguish between public and private documents.

localcodepage—"True" or "false"; false by default. Set to true for Metadata Services where localcodepage is set to true in the configuration file. Localcodepage should be true for services that manage documents authored using 8-bit languages, such as Russian and Thai, or multibyte languages, such as Chinese and Japanese. See Appendix D, 'Publishing non-English metadata', for details.

Once the output directory has been created, share it with the administrator who is harvesting your Metadata Service.

Importing documents

Before running the MDImport command, you must have already started a Metadata Service. If you've customized your service to use an administrative table, records won't be added to the administrative table corresponding to the imported documents.

Follow the steps below to import documents into your Metadata Service.

1. Open the operating system's command window and navigate to the location where the ArcIMS metadata commands are installed: <ArcIMS install directory>/Metadata/Commands.
2. Run the MDImport command to import documents from your Metadata Service. The usage for the MDImport command is:

   ```
   MDImport <input_directory> <sde_server_machine>
   <sde_instance> <database> <username> <password>
   {table_name_prefix} {owner} {folder}
   {make_private} {localcodepage}
   ```

   Mandatory arguments are shown in "<>" and optional arguments are shown in "{}". Optional arguments may be replaced with "#".

   You must specify the same ArcSDE connection information as in the Metadata Service's configuration file. The table_name_prefix argument must match the table prefix specified in the configuration file. If appropriate, make sure the localcodepage argument is set to "true".

   For example, with a SQL Server database you would type the following on the operating system's command line to import documents into a folder named "providerA":

   ```
   mdimport C:\providerA\Q1_04 sdeServer 5151
   aDatabase aUser aPassword tablePrefix #
   providerA
   ```

   input_directory—The name and location of the directory that contains the exported files. By default, the input directory is assumed to exist in the Commands directory. Either provide a full path or a path relative to the Commands directory. If there is a space in the path, enclose it within double quotes.

   sde_server_machine—The name of your ArcSDE server.

sde_instance—The ArcSDE instance or port number. For example, 5151. Or the instance name, for example, “esri_sde”.

database—The ArcSDE database name. The database name is required for SQL Server and DB2 databases. For Oracle, this information isn’t appropriate; skip this parameter by specifying “#” instead.

username—The database username.

password—The password associated with the username.

table_name_prefix—The table name prefix specified in the Metadata Service’s configuration file. By default, “imsmetadata”.

owner—The name of the user who will own the imported documents, by default, “admin”. If another user is specified, the user must be defined in the Metadata Service’s ACL and must have previously connected to the service to publish or browse documents; that is, the username must already be present in the service’s users table.

folder—The folder into which the documents will be imported; the folder must already exist. If a folder isn’t specified, the documents will be loaded at the root level of the Metadata Service. Use a forward slash “/” as the path separator. For example, the folder “California/Redlands” indicates that the documents will be imported into the Redlands folder, which is contained by the California folder, which exists at the root level of the Metadata Service. If there is a space in the path, enclose it within double quotes. Once imported, you can rearrange the documents using ArcCatalog.

make_private—“True” or “false”; false by default. When true, all imported documents will be private and, therefore, only the owner of the imported documents will be able to view them. Set to true if you want to review the imported documents before making them public, for example, to ensure they meet the requirements of your Metadata Service.

localcodepage—“True” or “false”; false by default. Set to true for Metadata Services where localcodepage is set to true in the configuration file. Localcodepage should be true for services that manage documents authored using 8-bit languages, such as Russian and Thai, or multibyte languages, such as Chinese and Japanese. See Appendix D, ‘Publishing non-English metadata’, for details.

3. Run the aimsmetaindx command to add the newly imported documents to the text index; they must be indexed before they can be searched. The arguments for the aimsmetaindx command are:

   ```
   aimsmetaindx <sde_server_machine>
   <sde_instance> <database> <username> <password>
   {table_name_prefix} {logfile} {localcodepage}
   ```

   You must specify the same ArcSDE connection information as in the Metadata Service’s configuration file. The table_name_prefix argument should match the new table prefix specified in the configuration file. For example, with a SQL Server database you would type the following on the operating system’s command line:

   ```
   aimsmetaindx sdeServer 5151 aDatabase aUser
   aPassword tablePrefix
   ```

   The aimsmetaindx command will finish immediately, but indexing has just begun in the database; see the section ‘Manually indexing documents’ in Chapter 3 for more information. See the section ‘Configuring ArcSDE’ in Appendix B for information about indexing issues with DB2 databases.

Documents will change over time in the provider’s Metadata Service. Work with the providers to ensure that they periodically provide you with new exports so you can include their updates in your Metadata Service. Each time you receive updates you’ll have to run the MDImport and then the aimsmetaindx commands. If you’ve made any changes to your version of the imported documents, those changes will be lost if new versions of those documents are provided in a future update.

The imported documents will have a different collection identifier from documents published directly to your Metadata Service; the collection identifier is unique to each Metadata Service. If you're using an administrative table, be aware that records representing the imported documents won't be added to the administrative table. If you want to manage imported documents in the same way as documents that are native to your Metadata Service, you could use a database trigger to add records to the administrative table when records with a different collection identifier are added to the service's feature class.

If you export and import documents to move them from one service to another within your organization, you'll lose some properties of individual documents. On import, all documents can be set as public or private and all documents can be assigned the same owner, but these properties can't be set for groups of documents. The original information about a document's owner and whether it was public or private will be lost. Only a document's owner can edit it or make it private. The owner or a user with metadata_administrator privileges can delete it. The only way to change the owner of a document once it has been imported is by sending the ArcXML request CHANGE_OWNER to the Metadata Service.

Format of the exported files

After running the MDExport command, the output directory will contain the following:

Metadata—A subdirectory containing the exported documents. These documents are XML files, stored in UTF-8 encoding. Each file is named using the document identifier.

Thumbnails—A subdirectory containing graphic images that illustrate the content of the resources described by the documents. These images are JPEG files. Each file is named using the identifier of the associated document.

Documents.xml—A file that lists all exported metadata documents. The list references the XML files stored in the Metadata directory.

Relationships.xml—A file that lists information about the relationships between the exported documents. For example, this file defines which documents are contained within a given folder.

DeletedDocuments.xml—A file that lists the documents that have been deleted from the provider's Metadata Service. When updates are imported, these files will be deleted from the harvester's Metadata Service.

DeletedRelationships.xml—A file that lists the relationships that no longer exist in the provider's Metadata Service. When updates are imported, these relationships will be deleted from the harvester's Metadata Service. For example, if a deleted document previously existed within a folder, the relationship between the deleted document and the folder will be removed when the updates are imported.

The Documents.xml, Relationships.xml, and DeletedDocuments.xml files are described in detail below. If you have metadata stored in another system, you could use this export format to import your legacy metadata into an ArcIMS Metadata Service. Use these descriptions to create your own files following the ArcIMS Metadata Service's export format.

Documents.xml

This file is associated with the Metadata Service's business table. It contains information about documents in the provider's Metadata Service.

There is one AIMSEF_DOCUMENT element for each document. Each AIMSEF_DOCUMENT element will have the following attributes, which are associated with a similarly named column in the business table. For example, the name attribute is associated with the DatasetName column.

- docuuid
- collectionuuid
- name
- updatedate
- onlink
- server
- service
- servicetype
- relevance
- folder
- minx
- miny
- maxx
- maxy

The server and service attributes are only included if the document describes an ArcIMS Image, Feature, or WMS service. The minx, miny, maxx, and maxy attributes contain the appropriate coordinates in decimal degrees describing the rectangle stored in the Footprint column. For details about the information stored in the columns associated with each attribute, see Appendix B, ‘ArcSDE considerations’.

The Documents.xml file will look similar to the example below:

```
<?xml version="1.0" encoding="UTF-8"?>
<AIMSEF_DOCUMENTS>
   <AIMSEF_DOCUMENT docuuid="{xxxxxxxx-xxxx-xxxx-
      xxxx-xxxxxxxxxxx1}"
      collectionuuid="{xxxxxxxx-xxxx-xxxx-xxxx-
      xxxxxxxxxxxA}" name="Hydrologic Boundaries"
      updatedate="1017796668" folder="false"
      relevance="0" minx="-179.1" miny="17.6"
      maxx="179.7" maxy="71.3"
      onlink="http://www.esri.com"
      servicetype="E" />
   <AIMSEF_DOCUMENT docuuid="{xxxxxxxx-xxxx-xxxx-
      xxxx-xxxxxxxxxxx2}"
      collectionuuid="{xxxxxxxx-xxxx-xxxx-xxxx-
      xxxxxxxxxxxA}" name="Population 2000"
      updatedate="1018085750" folder="false"
      relevance="0" minx="-175.9" miny="38.4"
      maxx="175.0" maxy="59.9"
      onlink="http://www.esri.com"
      server="http://www.esri.com" service="Pop2k"
      servicetype="A" />
</AIMSEF_DOCUMENTS>
```

Relationships.xml

This file is associated with the Metadata Service’s relationships table. It contains information about the relationships between documents in the provider’s Metadata Service. For example, a relationship establishes which documents are contained in a folder.

There is one AIMSEF_RELATIONSHIP element for each relationship included in this file. Each AIMSEF_RELATIONSHIP element will have the following attributes, which are associated with the similarly named columns in the relationships table. For example, the reltype attribute is associated with the Relationship column.

- fromdocuuid
- todocuuid
- reltype

Only parent–child and sibling relationships are recorded in this file. When the documents are imported into the harvester’s Metadata Service, the ancestor–descendant relationships that existed in the provider’s Metadata Service no longer apply.

For the fromdocuuid attribute, “Root” will be substituted for the identifier representing the root document in the provider’s Metadata Service. The identifier for the harvester’s Metadata Service will be substituted on import. For the reltype attribute, “C” is used to denote parent–child relationships, and “S” is used

to denote a sibling relationship. See Appendix B for more information about relationships.

The Relationships.xml file will look similar to the example below:

```
<?xml version="1.0" encoding="UTF-8"?>
<AIMSEF_RELATIONSHIPS>
   <AIMSEF_RELATIONSHIP fromdocuuid="Root"
      todocuuid="{xxxxxxxx-xxxx-xxxx-
      xxxx-xxxxxxxxxxx1}" reltype="C" />
   <AIMSEF_RELATIONSHIP fromdocuuid="xxxxxxxx-
      xxxx-xxxx-xxxx-xxxxxxxxxxx1"
      todocuuid="{xxxxxxxx-xxxx-xxxx-
      xxxx-xxxxxxxxxxx2}" reltype="S" />
</AIMSEF_RELATIONSHIPS>
```

DeletedDocuments.xml

This file is associated with the Metadata Service's deleted table. It contains information about the documents that have been removed from the provider's Metadata Service.

There is one AIMSEF_DELETEDDOCUMENT element for each deleted document. Each AIMSEF_DELETEDDOCUMENT element will have the following attributes, which are associated with the similarly named columns in the deleted table. For details about the information stored in the columns associated with each attribute, see Appendix B, 'ArcSDE considerations'.

- docuuid
- collectionuuid
- updatedate
- private

If no documents have been deleted, there will be no AIMSEF_DELETEDDOCUMENT elements in the file. Therefore, no documents will be deleted from the harvester's Metadata Service on import.

The Deleted.xml file will look similar to the example below:

```
<?xml version="1.0" encoding="UTF-8"?>
<AIMSEF_DELETEDDOCUMENTS>
   <AIMSEF_DELETEDDOCUMENT docuuid="{xxxxxxxx-
      xxxx-xxxx-xxxx-xxxxxxxxxxx3}"
      collectionuuid="{xxxxxxxx-xxxx-xxxx-
      xxxx-xxxxxxxxxxxA}" updatedate="1023144231"
      private="false" />
   <AIMSEF_DELETEDDOCUMENT docuuid="{xxxxxxxx-
      xxxx-xxxx-xxxx-xxxxxxxxxxx4}"
      collectionuuid="{xxxxxxxx-xxxx-xxxx-
      xxxx-xxxxxxxxxxxA}" updatedate="1023144245"
      private="false" />
</AIMSEF_DELETEDDOCUMENTS>
```

DeletedRelationships.xml

This file is associated with the Metadata Service's deleted relationships table. It contains information about the relationships that have been removed from the provider's Metadata Service. For example, for each deleted document there will be at least one deleted relationship.

There is one AIMSEF_DELETEDRELATIONSHIP element for each deleted relationship. Each AIMSEF_DELETEDRELATIONSHIP element will have the same attributes and values as the AIMSEF_RELATIONSHIP elements in the Relationships.xml file. The columns and values in the deleted relationships table are the same as in the relationships table. For details about the information stored in these tables, see Appendix B, 'ArcSDE considerations'.

If no relationships have been deleted, there will be no AIMSEF_DELETEDRELATIONSHIP elements in the file. Therefore, no relationships will be deleted from the harvester's Metadata Service on import.

The DeletedRelationships.xml file will look similar to the example below:

```
<?xml version="1.0" encoding="UTF-8"?>
<AIMSEF_DELETEDRELATIONSHIPS>
   <AIMSEF_DELETEDRELATIONSHIP fromdocuuid="Root"
      todocuuid="{xxxxxxxx-xxxx-xxxx-
      xxxx-xxxxxxxxxxx3}" reltype="C" />
   <AIMSEF_DELETEDRELATIONSHIP fromdocuuid="Root"
      todocuuid="{xxxxxxxx-xxxx-xxxx-
      xxxx-xxxxxxxxxxx4}" reltype="C" />
   <AIMSEF_DELETEDRELATIONSHIP
      fromdocuuid="xxxxxxxx-xxxx-xxxx-xxxx-
      xxxxxxxxxxx3" todocuuid="{xxxxxxxx-xxxx-
      xxxx-xxxx-xxxxxxxxxxx4}" reltype="S" />
</AIMSEF_RELATIONSHIPS>
```

Adding the OAI-PMH Connector to a Metadata Service

To share documents between two ArcIMS Metadata Services, one option is for the provider to use the OAI-PMH Connector and for the harvester to use the OAI-PMH Harvester. Getting the OAI-PMH Connector up and running is a two-step process. The first step is to deploy it on your Web server. The second step is to configure the OAI Connector to access and describe your Metadata Service.

For the first step, the necessary components for ArcIMS must be installed and configured properly before the OAI-PMH Connector can be deployed: Java 2 Platform SDK, Web server, and servlet engine. The OAI-PMH Connector is installed by default when you install ArcIMS. However, it won't be deployed and configured as part of the postinstall program; this must be done manually following the instructions in the readme.html file located at <ArcIMS install directory>\Metadata\OAI-PMH. In the first section, 'Configure your Web server for the OAI-PMH Connector', click the appropriate link for your Web server and servlet engine combination, follow those instructions to deploy the OAI-PMH Connector, then restart your Web server.

For the second step, you must configure the OAI-PMH Connector for your Metadata Service. You provide this information using a property file named aims_oai.properties. This file is located in the install directory at:

- **Windows:** <ArcIMS install directory>\Metadata\OAI-PMH\aimsharvester\Web-inf\config\aims_oai.properties
- **UNIX:** <ArcIMS install directory>/arcims/Metadata/OAI-PMH/aimsharvester/Web-inf/config/aims_oai.properties

If you are using JRun, Tomcat, or WebSphere, this file will be located where you installed the Web software in the directory shown below:

- **JRun 4:** <JRun install directory>\ servers\default\SERVER-INF\temp\aimsharvester.war-0123456789\Web-inf\config

  The 10-digit number is generated when the servlet is deployed.
- **Tomcat:** <Tomcat install directory>\webapps\aimsharvester\Web-inf\config
- **Websphere 5.0:** <WebSphere install directory>\AppServer\installedApps\<servername>\aimsharvester_war.ear\aimsharvester.war\Web-inf\config

Edit the aims_oai.properties file in a text editor and follow the steps below to configure the OAI-PMH Connector.

1. Set the communication protocol that the OAI-PMH Connector will use to communicate with ArcIMS. By default, the connector is set to use HTTP; however, you may prefer to use TCP/IP instead. Set the connectionType property to "http" or "tcp" as apropriate. For example:

   `connectionType=http`
2. Set the name of the Metadata Service that can be harvested using the OAI-PMH Connector; additional Metadata Services running on the same ArcIMS server can't be harvested. For example:

   `defaultServiceName=MyMetadataService`

3. If the HTTP protocol is being used, set the url property to Internet address of the server that hosts the ArcIMS Metadata Service. You can optionally provide a default login for accessing the Metadata Service. If access is unrestricted, the username and password properties should not have values. If you must login to access the default service and you want to provide a default login, set the username and password properties to match a username and password combination in the Metadata Service's access control list. For example:

   ```
   url=http://<localhost>.<domain>.<com>
   ```

   ```
   username=
   ```

   ```
   password=
   ```

4. If the TCP/IP protocol is being used, set the appServerMachine property to the name of the computer where the ArcIMS Application Server is installed. The default port on which the ArcIMS Application Server runs is 5300; if necessary, modify the appServerPort property. For example:

   ```
   appServerMachine=hostname
   ```

   ```
   appServerPort=5300
   ```

   Several additional properties need to be set to identify and describe your server when this information is requested using OAI-PMH.

5. Provide a name for your Metadata Service. For example,

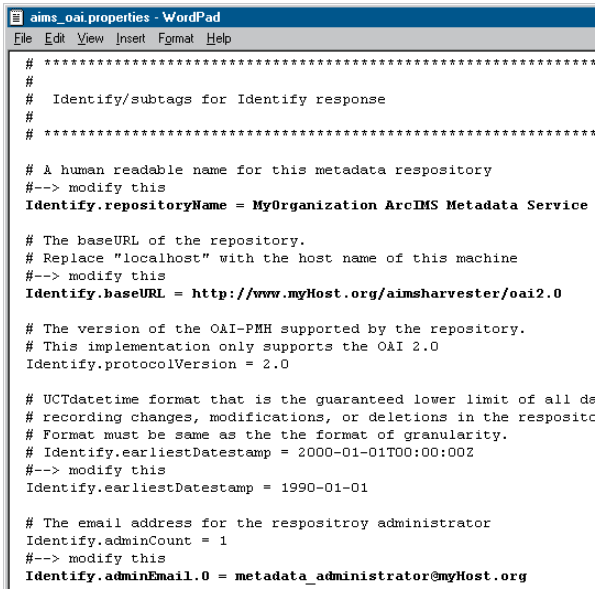
   ```
   Identify.repositoryName = MyOrganization ArcIMS
   Metadata Service
   ```

6. Modify the URL at which the OAI-PMH Connector can be accessed to reference the server that hosts your ArcIMS Metadata Service. This is the URL to which OAI-PMH requests can be sent. For example,

   ```
   Identify.baseURL = http://
   <localhost>.<domain>.<com>/aimsharvester/oai2.0
   ```

7. Provide an e-mail address at which the administrator of the Metadata Service can be contacted. For example,

   ```
   Identify.adminEmail.0 =
   metadata_administrator@<domain>.<com>
   ```



8. Provide an identifier representing the server that hosts your ArcIMS Metadata Service; this should not be prefaced by "http://". For example,

   ```
   Identify.repositoryIdentifier =
   <localhost>.<domain>.<com>
   ```

9. The value of the description property must be a valid XML document. Modify the default description value, changing the word "localhost" to match the value of the repositoryIdentifier property as defined above. The word localhost appears twice in two different XML elements. For example, this is illustrated below with the repositoryIdentifier property set to "www.myHost.org".

```
aims_oai.properties - WordPad
File  Edit  View  Insert  Format  Help

# Supports the OAI description
# For details see: http://www.openarchives.org/OAI/2.0/guidelines-
# For example: Identify.repositoryIdentifier = myhost.org
#--> modify this
Identify.repositoryIdentifier = www.myHost.org

# Other to describe the respository.
# Notes : description string should be well-formatted XML string
Identify.descriptionCount = 0
Identify.description.0=<description><oai-identifier xmlns=
\"http://www.openarchives.org/OAI/2.0/oai-identifier\" xmlns:xsi=
\"http://www.w3.org/2001/XMLSchema-instance\" xsi:schemaLocation=
\"http://www.openarchives.org/OAI/2.0/oai-identifier
http://www.openarchives.org/OAI/2.0/oai-identifier.xsd\"><scheme>(
<repositoryIdentifier>www.myHost.org</repositoryIdentifier><delim:
<sampleIdentifier>oai:www.myHost.org:ee05444a-1dd1-11b2-ad9e-c04b:
</oai-identifier></description>


# ****************************************************************
#
#  Log setting
#
# ****************************************************************
# Sets the filelog level, should one of value of
# ALL, FATAL, ERROR, WARN, INFO, DEBUG, OFF
# Please do not use DEBUG mode, it will slow the application.
Logger.level = INFO
```

The OAI-PMH Connector will create a log file named aims_oai_log in the logs directory at the same location as the config directory, which contains the aims_oai.properties file. Each time the Web server and servlet engine are restarted, a new log file named aims_oai_log will be created and the previous log file will be renamed aims_oai_log.1.

There are several increasing levels of information that can be reorded in the log file: ERROR, WARN, INFO, DEBUG, and OFF. By default, the connector is set to log information at the INFO level, which includes information such as the status of the connector, connections made, requests sent from OAI-PMH clients, warnings generated by the connector, and errors that occur. At the ERROR level, only error messages are recorded. At the WARN level, only warnings and error messages are recorded. At the DEBUG level, all messages are recorded including the ArcXML requests sent to and received from ArcIMS.

You change the level of logging to record more or less information, as appropriate. For example, change the level to DEBUG to troubleshoot a problem, if one occurs.

10. If desired, set the level of information that will be recorded in the OAI-PMH Connector's log file. For example, as illustrated in the above graphic,

    `Logger.level=DEBUG`

11. Save and close the file.

12. Stop and restart your servlet engine and Web server.

You should now be able to verify that the OAI-PMH connector has been successfully configured. In a Web browser, type the following URL:

http://<localhost>.<domain>.<com>/aimsharvester/oai2.0?verb=Identify

Oracle Application Server requires an extra forward slash (/) at the end of the URL:

http://<localhost>.<domain>.<com>/aimsharvester/oai2.0?verb=Identify/

If the OAI-PMH Connector has been correctly configured, it will return an XML document showing some of the information that you specified in the aims_oai.properties file.

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
    <responseDate>2004-05-15T22:34:31Z</responseDate>
    <request verb="Identify">http://www.myHost.org/aimsharvester/oai2.0</request>
  - <Identify>
      <repositoryName>MyOrganization ArcIMS Metadata Service</repositoryName>
      <baseURL>http://www.myHost.org/aimsharvester/oai2.0</baseURL>
      <protocolVersion>2.0</protocolVersion>
      <earliestDatestamp>1990-01-01</earliestDatestamp>
      <deletedRecord>no</deletedRecord>
      <granularity>YYYY-MM-DD</granularity>
      <adminEmail>metadata_administrator@myHost.org</adminEmail>
    </Identify>
  </OAI-PMH>
```

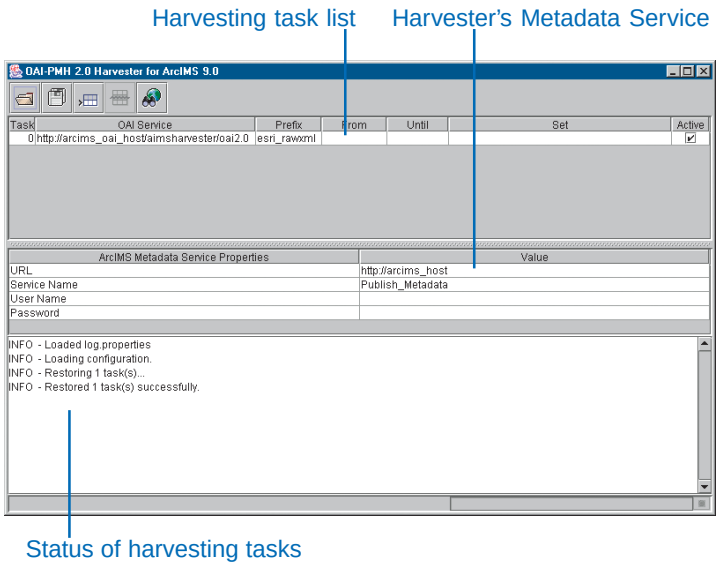
It's important to note that the OAI-PMH Connector is a Web application that is meant to be accessed programmatically using the OAI-PMH protocol. It does not provide HTML pages or a user interface.

Harvesting documents using the OAI-PMH Harvester

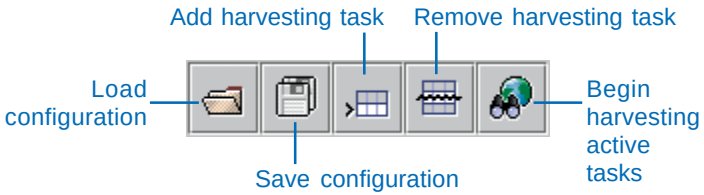
Once an organization participating in a central Metadata Service has successfully configured the OAI-PMH Connector, the harvester can harvest documents from the provider's Metadata Service using the OAI-PMH Harvester.

To start the OAI-PMH Harvester, open the operating system's command window and navigate to <ArcIMS install directory>/Metadata/OAI-PMH. Type "run" and press Enter. The OAI Harvester dialog box appears.

The OAI Harvester dialog box is divided into three panels. At the top is the list of harvesting tasks; each task identifies an OAI-PMH Connector from which you want to harvest and the documents you want to retrieve. In the middle is information about your Metadata Service; the harvested documents will be published directly to the specified service. At the bottom you can see the status of your harvesting tasks; initially it will show that the default configuration file for the OAI-PMH Harvester has been successfully loaded.



Use the toolbar to add and remove harvesting tasks, save the OAI Harvester's configuration, load a new configuration file, and start the harvesting process.



When the OAI-PMH Harvester opens, the default configuration file, OAIServers.xml, is loaded. It includes a default harvesting task to help you get started. Edit the default harvesting task to identify an OAI-PMH Connector from which you want to harvest, specify the appropriate information to publish the harvested documents to your Metadata Service, and begin harvesting following the instructions below.

1. In the task list, double-click the default value in the OAI Servers column, "http://arcims_oai_host/aimsharvester/oai2.0". Change "arcims_oai_host" to reflect the URL of the provider's ArcIMS server. For example, if the ArcIMS server is "http://www.providerHost.org", the address for the provider's OAI-PMH Connector will be "http://www.providerHost.org/aimsharvester/oai2.0".

   The value in the Prefix column should remain "esri_rawxml"; this indicates that you want the entire document to be retrieved as-is from the provider. The first time you harvest from a connector you'll want to retrieve all documents—don't specify values in the From, Until, and Set columns.

2. Make sure this task has a check mark in the Active column. Only active tasks will be harvested.

3. Press Enter, or click elsewhere in the task list for the information specified in the OAI Servers column to take effect.

   Now that a harvesting task has been defined, you need to provide information about the Metadata Service to which the harvested documents will be published.

4. In the second panel, double-click in the Value column to the right of URL and specify the URL of your ArcIMS server, for example, “http://www.harvesterHost.com”.

5. Double-click in the Value column to the right of Service Name and specify the name of the Metadata Service to which the harvested documents will be published.

6. Double-click in the Value column to the right of User Name and specify the username to use to publish the harvested documents.

   This user will own the harvested documents. Only a document’s owner can edit it or make it private. The owner or a user with metadata_administrator privileges can delete it. The only way to change the owner of a document once it has been harvested is by sending the ArcXML request CHANGE_OWNER to the Metadata Service.

7. Double-click in the Value column to the right of Password and specify the password to use to publish the harvested documents.

   The username and password specified in the Value column must match a username and password combination in the access control list that has permission to publish to the Metadata Service.

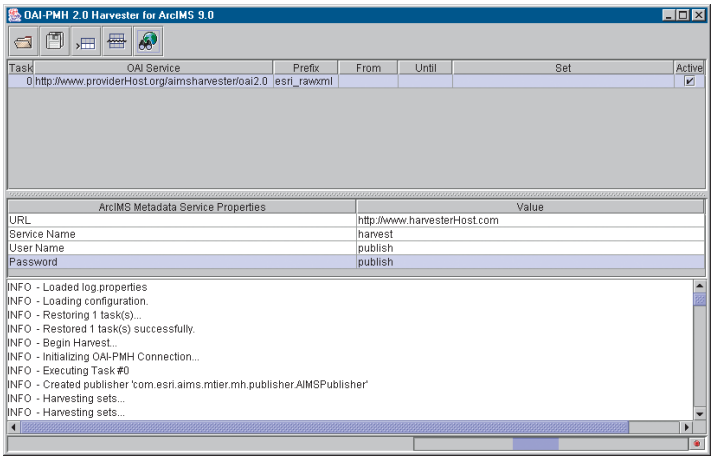
8. Press Enter, or click elsewhere in the harvester panel for the password you’ve specified to take effect.

9. Click Save to save the changes you’ve made to the OAI-PMH Harvester’s configuration. In the Save dialog box, navigate to the <ArcIMS install directory>/Metadata/OAI-PMH directory. Save your changes to the configuration file OAIServers.xml; your settings will be loaded each time the OAI-PMH Harvester is started.

10. Click Begin harvesting active tasks to start harvesting documents.

11. In the message box that appears, click Yes to start harvesting.

In the status panel, messages appear indicating the progress of the harvesting tasks; scroll down to see the latest messages. The status bar also indicates that harvesting is in process. Click the red dot at the bottom right of the dialog box to stop harvesting at any time.



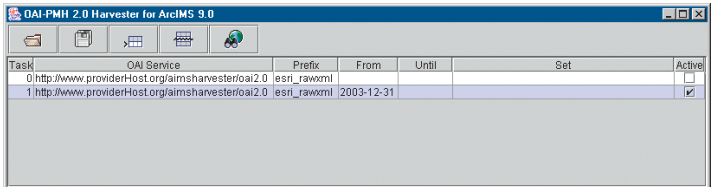
By default, if a problem occurs when the harvested documents are being published, an error message will appear and harvesting will be stopped. For example, this might happen if you did not click elsewhere in the second panel after specifying the password, or if the user doesn’t have permission to publish to the specified Metadata.Service.

Harvesting documents by date

The first time you harvest a provider's documents you'll want to retrieve all of them. For subsequent harvests, you'll typically want to retrieve only the documents that have been added or modified since the last time the provider was harvested. Do this by specifying the date when you last harvested the provider in the From column. For example, if the last successful harvest was on December 31, 2003, harvest all documents modified since then by specifying "2003-12-31" in the From column. Dates must be provided in the format YYYY-MM-DD.

Any combination of dates can be specified in the From and Until columns as appropriate. To harvest documents modified in a specific time period, specify a date in both the From and Until columns. To harvest documents modified before a given date, specify a date only in the Until column. Be sure to click elsewhere in the task list after specifying a date so that your changes to the task list take effect.

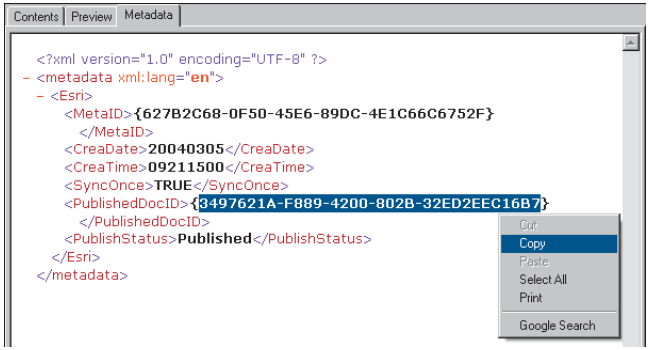
You can modify the existing task in the task list for this provider and make that task active. Or, uncheck the existing task in the Active column to deactivate it, click Add a harvesting task, then define the new task to retrieve the latest updates from that provider. Only active tasks will be executed during the harvesting process.



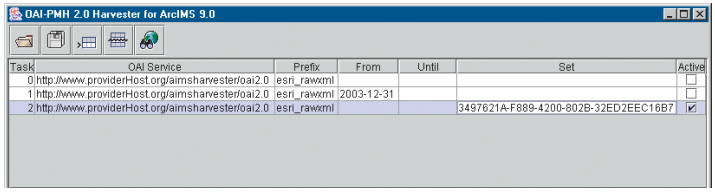
Harvesting documents from a folder in a Metadata Service

Suppose you only want to harvest documents that reside in a specific folder in the provider's Metadata Service. To specify that folder as the set that you want to harvest, put its document identifier in the Set column in the task list.

To get the document identifier for a folder, select the folder in ArcCatalog, click the Metadata tab, then click Xml in the Stylesheets dropdown list on the Metadata toolbar. The value of the PublishedDocID element is the folder's document identifier. For OAI-PMH services, the value used to specify a set is the document identifier excluding the braces; select this information with your mouse, right-click the selected text, then click Copy.



In the OAI-PMH Harvester, paste this value into the Set column in the task list. Press Enter or click elsewhere in the task list for your changes to take effect.



Customizing the harvesting configuration

While the most commonly used properties for the harvester and the provider can be specified in the OAI-PMH Harvester dialog box, a few properties can only be specified by editing the configuration file in a text editor. The configuration file is named OAIServers.xml and is located at <ArcIMS install directory>/Metadata/OAI-PMH.

Your custom settings will be loaded each time the OAI-PMH Harvester is started. However, they won't appear in the OAI-PMH Harvester dialog box. If you customize the settings for a harvesting task, remove that task, and save your changes, your custom settings will be lost. If you add a task, you'll have to modify its settings in a text editor if you want it to use the same custom settings as the other tasks.

When you open the configuration file in a text editor, you can see there is one repository element for each provider. Additional properties can be set for a provider as follows.

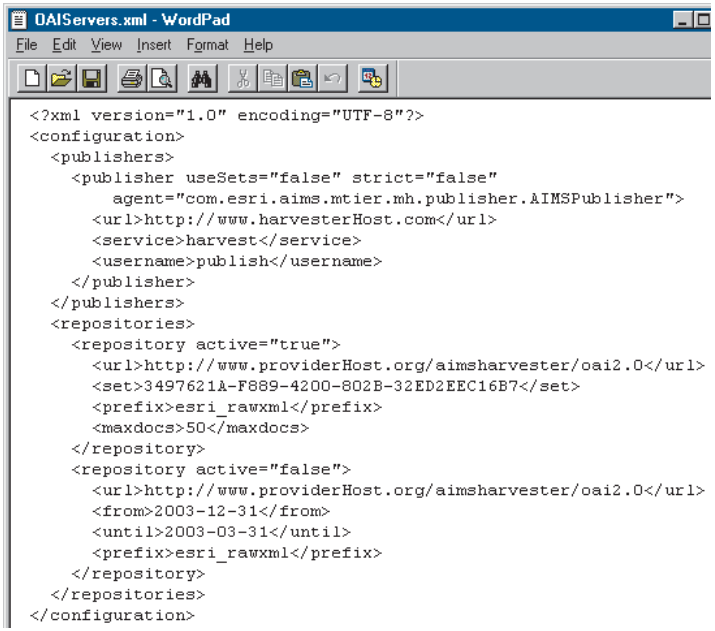
Maximum number of documents—To set the maximum number of documents that you want to harvest from a provider, add the maxdocs element into the appropriate repository element and set its value to an appropriate number. For example, you would add "<maxdocs>50</maxdocs>".

Login to access provider's Metadata Service—If you must log in to access documents in the provider's Metadata Service, add username and password elements into the appropriate repository element and set their values appropriately. For example, you would add "<username>harvesterUser</username>" and "<password>harvesterPass</password>".

In the configuration file there is one publisher element for the harvester. Additional properties can be set for the harvester as follows.

Copying the provider's folder structure—The provider's Metadata Service may organize documents into a hierarchy of folders so they can be more easily browsed. By default, folders matching the provider's folder structure will be created at the root level of the Metadata Service into which the documents are harvested. Instead, you can choose not to copy the provider's folder structure; with this option, all documents will be placed at the root level of the harvester's Metadata Service. To not replicate the provider's folders, add the useSets attribute to the provider element and set it to "false", for example, "<provider useSets='false'>". If the useSets attribute isn't present, the value "true" will be assumed.

Publishing documents to a folder on disk—Instead of publishing the harvested documents directly to the specified Metadata Service, they can be stored on disk. You might prefer to review or



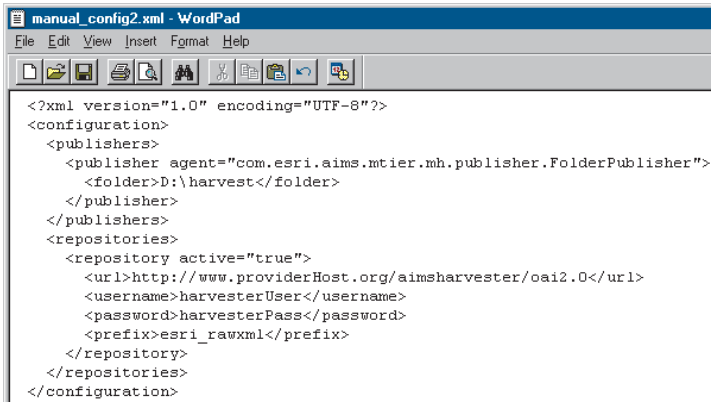

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <publishers>
    <publisher useSets="false" strict="false"
        agent="com.esri.aims.mtier.mh.publisher.AIMSPublisher">
      <url>http://www.harvesterHost.com</url>
      <service>harvest</service>
      <username>publish</username>
    </publisher>
  </publishers>
  <repositories>
    <repository active="true">
      <url>http://www.providerHost.org/aimsharvester/oai2.0</url>
      <set>3497621A-F889-4200-802B-32ED2EEC16B7</set>
      <prefix>esri_rawxml</prefix>
      <maxdocs>50</maxdocs>
    </repository>
    <repository active="false">
      <url>http://www.providerHost.org/aimsharvester/oai2.0</url>
      <from>2003-12-31</from>
      <until>2003-03-31</until>
      <prefix>esri_rawxml</prefix>
    </repository>
  </repositories>
</configuration>
```

In this example, one of the providers has been customized to harvest only the first 50 documents found. The harvester's information has also been customized. The folder hierarchy used in the provider's Metadata Service won't be duplicated in the harvester's Metadata Service, and an attempt will be made to publish all harvested documents even if problems are encountered during publishing.

process the documents on disk before publishing them; this might be desirable if your Metadata Service doesn't use an administrative table that would keep the harvested documents from becoming public immediately after harvesting.

For this option, remove the url element from the provider element, set the agent attribute on the provider element to "com.esri.aims.mtier.mh.publisher.FolderPublisher", and add a folder element with a value indicating the full path where the harvested documents will be stored. For example, "<provider agent='com.esri.aims.mtier.mh.publisher.FolderPublisher'>" and "<folder>D:\harvest</folder>". With these modifications, when the configuration file is loaded the information in the harvester panel will appear empty; the folder information won't be shown.

The folders used to organize documents in the provider's Metadata Service won't be replicated when documents are harvested to disk, even if the useSets attribute is set to "true".




```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <publishers>
    <publisher agent="com.esri.aims.mtier.mh.publisher.FolderPublisher">
      <folder>D:\harvest</folder>
    </publisher>
  </publishers>
  <repositories>
    <repository active="true">
      <url>http://www.providerHost.org/aimsharvester/oai2.0</url>
      <username>harvesterUser</username>
      <password>harvesterPass</password>
      <prefix>esri_rawxml</prefix>
    </repository>
  </repositories>
</configuration>
```

In this example, the provider's information has been customized to retrieve documents from the OAI-PMH Connector with a username and password. Also, the harvester's information has been customized. The harvested documents will be stored in a folder on disk rather than directly publishing them to the harvester's Metadata Service, and thumbnails will be retrieved from the provider's Metadata Service.

Publishing harvested documents—By default, if any problems are encountered when publishing the harvested documents, the harvesting process will stop. Instead, you can choose to attempt to publish each harvested document; depending on the problem, some documents may be published successfully. To attempt to publish all documents, add the strict attribute to the provider element and set it to false, for example, "<provider strict='false'>". If the strict attribute isn't present, the value "true" will be assumed.

Setting log options

The log options for the OAI-PMH Harvester are similar to the options for the Z39.50 Connector. Logging options are set by modifying the log.properties file, which is located at <ArcIMS install directory>/Metadata/OAI-PMH.

Log options can be sent to several locations: standard output, the status panel in the OAI-PMH Harvester, a log file, and a separate debug log file. By default, information is logged only to standard output and the status panel. Usually it's most convenient to check the status of the harvesting process in the status panel of the dialog box. Use the debug log file to troubleshoot any problems you may encounter. Because the standard output for the OAI-PMH Harvester is sent to the command window from which it was started, you may choose not to send log information to that location.

Four levels of logging can be used to record increasing amounts of information: ERROR, WARN, INFO, or DEBUG. The level of logging available to all logs is set on the first line of the log.properties file. By default, it's set to the INFO level, which records the status of the OAI-PMH Harvester and the harvesting process, warnings that are generated, and any errors that occur. At the ERROR level, only error messages are recorded. At the WARN level, only warnings and error messages are recorded. At the DEBUG level, all messages are recorded.

Each log can record information at a different level. To change a log's level, set its threshold property in the log.properties file. No

log can record more information than the overall level, which is specified in the first line of the file. For example, if the overall level is INFO and the debug log's threshold is set to DEBUG, the debug log file can't record more than the INFO level of information. With the overall level set to DEBUG, everything is recorded in the debug log file; the other logs will continue at their specified thresholds.

If you start a log file and a debug log file, files named "access.log" and "debug.log", respectively, will be created by default in the <ArcIMS install directory>/Metadata/OAI-PMH directory. Settings in the log.properties file let you specify different file names and locations. They also let you include a date in the log file's name.

The steps for starting and stopping a log, changing a log's threshold, and changing a log file's name and date are the same for the OAI-PMH Harvester as for the Z39.50 Connector. See 'Setting log options' in Chapter 5 for detailed instructions about how to complete these tasks.

log.properties - Notepad

File Edit Format View Help

```
# Set root logger level from INFO to DEBUG to debug harvester client application
log4j.rootLogger=DEBUG, stdout, PanelAppender, Access, Debug

# stdout
#log4j.appender.stdout=org.apache.log4j.ConsoleAppender
#log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
#log4j.appender.stdout.layout.ConversionPattern=%d [%t] %-5p %c{2} - %m%n
#log4j.appender.stdout.threshold=INFO

# Panelwindow
log4j.appender.PanelAppender=com.esri.aims.mtier.mh.ui.MessagePanelAppender
log4j.appender.PanelAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.PanelAppender.layout.ConversionPattern=%-5p - %m%n
log4j.appender.PanelAppender.threshold=INFO

# Access log to file
#log4j.appender.Access=org.apache.log4j.DailyRollingFileAppender
#log4j.appender.Access.DatePattern='.'yyyy-MM-dd
#log4j.appender.Access.File=harvesterLogs/access.log
#log4j.appender.Access.layout=org.apache.log4j.PatternLayout
#log4j.appender.Access.layout.ConversionPattern=%d [%t] %-5p %c{2} - %m%n
#log4j.appender.Access.threshold=INFO

# Debug log to file
log4j.appender.Debug=org.apache.log4j.DailyRollingFileAppender
log4j.appender.Debug.DatePattern='.'yyyy-MM-dd
log4j.appender.Debug.File=harvesterLogs/debug.log
log4j.appender.Debug.layout=org.apache.log4j.PatternLayout
log4j.appender.Debug.layout.ConversionPattern=%d [%t] %-5p %c{2} - %m%n
log4j.appender.Debug.threshold=DEBUG
```

In this example, log information is only sent to the status panel in the OAI-PMH Harvester dialog box and to a debug log file; the standard output and log file options are commented out. The overall threshold is set to DEBUG, but the individual logs have different thresholds. The debug log file will be placed in a directory named harvesterLogs in the location where the OAI-PMH Harvester application was installed.

Publishing non-English metadata

D

IN THIS APPENDIX

- **What languages will a Metadata Service support?**
- **Configuring your Metadata Service**
- **Indexing and harvesting considerations**

Do you expect people to publish metadata documents in languages other than English to your Metadata Service? If so, you should read this section for information on how many different languages your Metadata Service can support and how to configure your Metadata Service to do so.

What languages will a Metadata Service support?

The languages supported by a Metadata Service depend on how your database is configured. If the character set of the database is set to an 8-bit language, such as Russian or Hebrew, or a multibyte language, such as Chinese, Japanese, or Korean, the Metadata Service will support the publishing of metadata documents written in English and in the language used by your database. See the section 'Configuring ArcSDE' in Appendix B and your database's documentation for information about configuring your database and its text index in different languages.

Configuring your Metadata Service

If your database is set to use an 8-bit or multibyte character set, the localcodepage attribute must be set in the Metadata Service's configuration file following the steps below.

1. Edit your Metadata Service configuration file, for example, MetadataServer.axl, in a text editor.
2. Locate the SDEWORKSPACE element and append this attribute and value to it: localcodepage="true". For example:

   ```
   <SDEWORKSPACE name="mySDE" server="sdeServer"
   instance="port:5151" database="aDatabase"
   user="aUser" password="aPassword"
   localcodepage="true"  />
   ```

3. Save the configuration file and start your Metadata Service.

The localcodepage attribute adjusts how the Metadata Service connects to the database. This attribute must be set in the configuration file when you first create your service or before any non-English metadata documents are published to it.

If you previously created a Metadata Service and published non-English metadata documents to it, you should not change the localcodepage attribute and then restart the same Metadata Service. If you do, the existing non-English metadata documents will not be displayed properly. To change the localcodepage setting so that non-English metadata documents are handled correctly, you must create a new Metadata Service with the localcodepage attribute set to "true", then republish your documents to the new service.

Indexing and harvesting considerations

Whenever you set the localcodepage attribute to "true" in a Metadata Service's configuration file as described in the previous section, you must also set the localcodepage argument to "true" whenever you run indexing or harvesting commands on that service. The localcodepage argument usually appears at the end of the command's usage; the usage for the aimsmetaindx command is:

```
aimsmetaindx <sde_server_machine> <sde_instance>
<database> <username> <password>
{table_name_prefix} {logfile} {localcodepage}
```

The value you use for the localcodepage argument must match the value of the localcodepage attribute in the Metadata Service's configuration file. For example, with a Japanese database, you would index published documents like this:

```
aimsmetaindx sdeServer 5151 aDatabase aUser
aPassword aPrefix # true
```

Harvesting documents from another Metadata Service is equivalent to publishing the individual metadata documents directly to your service. If the Metadata Service from which you harvest contains metadata documents written in languages that your Metadata Service doesn't support, you may experience some errors during indexing. In addition, the text of the document may not display properly. You will probably need to delete these foreign language metadata documents from your Metadata Service using ArcCatalog.

What's new with ArcIMS 9?

E

IN THIS APPENDIX

- **Changes for Metadata Services with ArcIMS 9**
- **Migrating an existing Metadata Service to ArcIMS 9**

For ArcIMS 9, improvements were made to Metadata Services for better scalability and performance. Metadata Services now use the database's native text indexing capability to index and search published documents. There were no changes to Metadata Explorer or ArcCatalog; no changes are required to any custom clients that access a Metadata Service.

This new design required changes to the database schema for Metadata Services. Any services that currently exist won't work once you have installed ArcIMS 9. You can upgrade the objects in the database associated with your existing Metadata Service using the MDUpgrade command, or you can republish your documents to a new Metadata Service. However, you must first install and configure the text-indexing component for your database management system following the instructions in Appendix B, 'ArcSDE considerations'.

Changes for Metadata Services with ArcIMS 9

For ArcIMS 9, improvements were made to Metadata Services for better scalability and performance. The changes primarily involve how metadata documents are stored and indexed. Metadata Services now rely on ArcSDE to manage the documents and the database's native text indexing capability to index and search their contents. There were no changes to Metadata Explorer or ArcCatalog. An overview of the changes is provided below.

Changes in the database

The most significant change to Metadata Services is the database schema. Published documents are no longer stored in a BLOB column. ArcSDE 9 supports columns of the type SE_XML for storing XML documents in the database. An ArcSDE XML column is now used to store published documents. This change has implications in how both the relational database and ArcSDE are configured.

First, because ArcSDE XML columns are not supported with Informix databases, Metadata Services are also no longer supported with Informix databases. Second, Metadata Services now require the appropriate text indexing component installed and configured properly in Oracle, SQL Server, and DB2 databases before you can create a Metadata Service. Also, in ArcSDE, consideration needs to be given to settings for the XML parameters in the SDE_dbtune table since they control the properties of the text index that is created in the database.

Because of these changes, the Metadata Service's database schema has changed significantly. If you currently manage Metadata Services, including gazetteer services, with ArcIMS versions 4.0 or 4.0.1, those existing services will not work with ArcIMS 9. If you are using ArcIMS 4.0.1, you must upgrade your existing service's objects in the database—a utility is provided with ArcIMS to do this. Alternatively, you must start a new Metadata Service with ArcIMS 9 and republish your documents. If you are using ArcIMS version 4.0, you won't be able to upgrade your existing Metadata Service; you must create a new service with ArcIMS 9 and republish your documents.

The new database schema used by Metadata Services is discussed in detail in Appendix B, 'ArcSDE considerations', in this book. Upgrading an ArcIMS 4.0.1 Metadata Service for 9.0 is discussed in detail later in this appendix.

As a final note, the administrator for a Metadata Service will not be able to see the contents of the feature class associated with the service in ArcCatalog. For example, you might do that to troubleshoot problems when a Metadata Service fails to start. Because ArcSDE XML columns are not supported in the geodatabase with ArcGIS 9, you can't access the feature class's properties, preview its contents, or export its data. Instead you must use the database's tools to inspect the contents of the business table associated with the feature class.

Alternatively, you can create a view on the feature class associated with a Metadata Service using ArcSDE commands. If the view excludes the XML column, you can see its contents in ArcCatalog. However, if you later want to remove the Metadata Service's objects from the database you *must* delete the view first! Otherwise, the database will lock the data used by the view and you won't be able to remove the feature class from the database.

Indexing changes

ArcSDE provides the ability to choose which XML elements in the published documents will be indexed. Unless you have a specific need to do so, it generally does not make sense to index information, such as the number of features in a feature class, attribute and spatial reference information, and so on. For example, you might choose to index just the 15 metadata elements that provide the most useful information when searching, such as the title, abstract, keywords, scale, and important dates.

ArcSDE also lets you control whether an element's value will be indexed as text or as a number. By default, everything is indexed as text, but for date searches dates must be indexed numerically. New commands provided with ArcIMS let the administrator define the indexing scheme for a Metadata Service and list any errors that occurred during indexing. An indexing error would occur when an element containing a date is set to be indexed numerically, but a document contains text in that element instead. For example, "20040114" would be successfully indexed, but "01/14/04" would result in an error. The indexing scheme used by a Metadata Service can be changed at any time by dropping the current index and rebuilding it based on a new definition file.

Z39.50 Connector changes

Changes to the database schema for Metadata Services required some changes to the Z39.50 Connector. At ArcIMS 4.0.1, configuring this connector required loading some Z39.50 use attribute code definitions into the database. This is no longer required. Also, improvements were made in how the connector logs information and errors.

ArcXML changes

The changes in how published documents are indexed required some minor changes to the configuration files used to define a Metadata Service. With ArcIMS 4.0.1, the METADATA_CONTENT element used the attributes minwordlength and index_numbers to specify how published documents would be indexed. These attributes are no longer used. If they exist in a configuration file, they will be ignored.

Because of changes in the Z39.50 Connector, the zcode attribute is no longer supported on the TAGTEXT and TAGVALUE elements, which are used in search requests sent to Metadata Services. Also, the PUT_METADATA_SEMANTIC and SEMANTIC_PAIR elements are no longer supported in publish requests.

With the new database schema used by Metadata Services, information can no longer be gathered from ArcIMS about the words that have been indexed. The GET_COLLECTION_INFO element can still be used to request information about the elements that have been indexed; the only valid value for the collection attribute is "tags". However, the following attributes on that element aren't supported, since they were only used when retrieving the indexed words: orderby, tag, and zcode.

Harvesting changes

In addition to the existing import and export mechanism for moving published documents between ArcIMS servers, support has been added for harvesting based on the Open Archives Initiative Protocol for Metadata Harvesting. An OAI-PMH Connector, similar to the Z39.50 Connector, can be created that lets documents published to a Metadata Service be harvested following OAI-PMH. Also, the OAI-PMH Harvester application lets an administrator harvest documents from any OAI-PMH Connector and directly publish those documents to a Metadata Service.

Migrating a Metadata Service to ArcIMS 9

Because the database schema has changed significantly, Metadata Services that you are currently running won't work with ArcIMS 9. If you're currently using ArcIMS version 4.0, you can't upgrade existing Metadata Services; you must create a new service with ArcIMS 9 and republish your documents to the new service.

If you're currently using ArcIMS 4.0.1, you can upgrade objects in the database associated with your existing Metadata Services using the MDUpgrade command as described below. None of the information in your existing service will be changed or lost. The objects in the database will simply be reorganized to follow the

new database schema. Or you can start a new service with ArcIMS 9 and republish your documents to the new service.

Once your Metadata Service is running successfully with ArcIMS 9, remove the objects associated with the 4.0.1 version of the service from the database.

Any gazetteer services you are using with Metadata Explorer must also be upgraded for ArcIMS 9 because they are Metadata Services. You can upgrade a gazetteer service using the MDUpgrade command the same way you would upgrade any Metadata Service.

If your gazetteer uses the data provided with ArcIMS, you can delete the existing gazetteer objects from the database and load the data as described in 'Loading the gazetteer data' in Appendix A. The gazetteer data provided with ArcIMS 9 is the same as with previous versions, but it has been upgraded to follow the new database schema. If you've customized the gazetteer and you have the original data used to define your custom places, delete the existing gazetteer objects from the database, create a new Metadata Service, and load your custom data as described in the section 'Customizing the gazetteer' in Appendix A.

Upgrading Metadata Services

You can use the MDUpgrade command to migrate your Metadata Service from ArcIMS 4.0.1 to ArcIMS 9 following the steps below.

1. Back up all objects in the database associated with Metadata Services. With ArcIMS 4.0.1, one feature class and nine tables were associated with each Metadata Service. If the table prefix specified in a Metadata Service's configuration file was "IMSMETADATA", the following objects would be associated with that service:
   - IMSMETADATA (feature class)
   - IMSMETADATADELETED
   - IMSMETADATADELETEDREL
   - IMSMETADATARELATIONSHIPS
   - IMSMETADATATAGS
   - IMSMETADATATHUMBNAILS
   - IMSMETADATAUSERS
   - IMSMETADATAVALUES
   - IMSMETADATAWORDINDEX
   - IMSMETADATAWORDS

   If the prefix in the configuration file was "Gazetteer", the objects in the database would be Gazetteer, GazetteerDeleted, GazetteerDeletedRel, and so on.
2. Delete all Metadata Services that are saved as part of your ArcIMS site, for example, using ArcIMS Administrator. Metadata Services created with previous versions of ArcIMS and saved in the site startup file, ArcIMSSite.sez, won't start once ArcIMS 9 has been installed.
3. Uninstall ArcIMS 4.0.1 and install ArcIMS 9.
4. On the ArcSDE server used by your Metadata Services, uninstall ArcSDE 8.3 and install ArcSDE 9.
5. Make sure the text indexing component has been installed and configured properly for your database on the server used by your Metadata Services. Follow the instructions in Appendix B, 'ArcSDE considerations', in this book.
6. Set the ArcSDE DBTUNE parameters appropriately for Metadata Services after reviewing the information in Appendix B. The settings for XML columns affect how and when published documents will be indexed.

   Before you can continue, you must get information from the database administrator about how to connect to the ArcSDE 9 server.
7. Open the operating system's command window and navigate to the location where the ArcIMS metadata commands are installed: <ArcIMS install directory>/Metadata/Commands.

8. Run the MDUpgrade command to upgrade your Metadata Service. The usage for the MDUpgrade command is:

```
MDUpgrade <sde_server_machine> <sde_instance>
<database> <username> <password>
<src_table_name_prefix> <trg_table_name_prefix>
{index_document} {localcodepage} {keyword}
```

Mandatory arguments are shown in "<>" and optional arguments are shown in "{}". Optional arguments may be replaced with "#".

You must specify the same ArcSDE connection information as in the Metadata Service's configuration file. The src_table_name_prefix argument must match the table prefix specified in the existing configuration file. Provide a different prefix for the trg_table_name_prefix argument; this prefix must be used in the new configuration file. For DB2, the index_document argument must be "n"; as discussed in the 'Configuring ArcSDE' section in Appendix B, only the database administrator has permission to index documents. If appropriate, make sure the localcodepage argument is set to "true".

For example, with a DB2 database you would type the following on the operating system's command line:

```
mdupgrade sdeServer 5151 aDatabase aUser
aPassword oldPrefix newPrefix N
```

sde_server_machine—The name of your ArcSDE server.

sde_instance—The ArcSDE instance or port number. For example, "5151". Or the instance name, for example, "esri_sde".

database—The ArcSDE database name. The database name is required for SQL Server and DB2 databases. For Oracle, this information isn't appropriate; skip this parameter by specifying "#" instead.

username—The database username.

password—The password associated with the username.

src_table_name_prefix—The table prefix specified in the configuration file for the ArcIMS 4.0.1 Metadata Service. By default, "imsmetadata".

trg_table_name_prefix—The new table prefix to use for the ArcIMS 9 Metadata Service. This value can't be the same as the src_table_name_prefix argument. The existing configuration file must be modified to use this prefix instead of src_table_name_prefix.

index_document—"Y" or "n"; the default is "y". When "y", the published documents are indexed in the database immediately after the objects are upgraded. When "n", you must run the aimsmetaindx command later to index the documents. You can't search published documents until they have been indexed.

localcodepage—"True" or "false"; false by default. Set to true for Metadata Services where localcodepage is set to true in the configuration file. Localcodepage should be true for services that manage documents authored using 8-bit languages, such as Russian and Thai, or multibyte languages, such as Chinese and Japanese. See Appendix D, 'Publishing non-English metadata', for details.

keyword—The DBTUNE configuration keyword to use when the new ArcIMS 9 Metadata Service's objects are created in the database. If a keyword isn't specified, "DEFAULTS" will be used.

9. Modify the configuration file for your Metadata Service using a text editor.

Change the prefix attribute on the TABLE_NAME element to match the value you specified in the trg_table_name_prefix argument for the upgrade command. Remove the minwordlength and index_numbers attributes from the METADATA_CONTENT element; you may also want to change the value of the index_words attribute, depending on how the database and ArcSDE were configured. If

appropriate, make sure the localcodepage attribute on the SDEWORKSPACE element is set to “true”.

10. Start your Metadata Service using the updated configuration file, for example, using ArcIMS Administrator.

11. Repeat steps 9 through 11 for each Metadata Service, including gazetteer services, that requires upgrading.

12. If you are using a DB2 database or you chose not to index documents during the upgrade, run the aimsmetaindx command or update the text index using the database’s tools during a window of time when the database is available for maintenance. The documents must be indexed before the Metadata Service can be searched. The arguments for the aimsmetaindx command are:

    ```
    aimsmetaindx <sde_server_machine>
    <sde_instance> <database> <username> <password>
    {table_name_prefix} {logfile} {localcodepage}
    ```

    You must specify the same ArcSDE connection information as in the Metadata Service’s configuration file. The table_name_prefix argument should match the new table prefix specified in the configuration file. For example, with a DB2 database, the DB2ADM user would type the following on the operating system’s command line:

    ```
    aimsmetaindx sdeServer 5151 sde sdeUser
    sdePassword aUser.newPrefix
    ```

    The aimsmetaindx command will finish immediately, but indexing has just begun in the database; see the section ‘Manually indexing documents’ in Chapter 3 for more information. See the section ‘Configuring ArcSDE’ in Appendix B for more information about indexing issues with DB2 databases.

13. Configure Metadata Explorer to use the new Metadata Services, if appropriate, following the instructions in Appendix A, ‘Configuring Metadata Explorer’, in this book.

If you previously used an administrative table with your Metadata Service and you want to continue using the same table, no changes are required to the table or to the configuration file. However, if you want to create and use a new administrative table, you must populate the table yourself with identifiers for the documents in the ArcIMS 9 service and update the configuration file with information about the new table.

Once you’ve verified that the documents in your Metadata Services have been upgraded successfully, delete the objects from the database that were associated with the 4.0.1 version of those services. To do this, connect to ArcSDE in ArcCatalog using the same connection information specified in the configuration file. Then, delete the appropriate objects from the database; these are described in step 1 above.

Index

A

Access control list
  creating an ACL file 28, 31
  defining usernames and passwords 28
  described 28
  for ArcCatalog 29
  for Metadata Explorer 30, 86–87
  for Z39.50 Connector 71–72, 73, 76–77
  roles 28, 31, 32, 53, 56–57, 87
  sample file 28, 73, 87
  using with different connectors 29, 87
ACL. *See* Access control list
Add to ArcExplorer, from Metadata Explorer 14
Add to ArcMap, from Metadata Explorer 14
ADMIN_TABLE element 39
Administrative and political boundaries 43
Administrative tables
  and importing documents 119
  and upgrading 138
  creating 39
  defining in configuration file 24, 39
  described 23–25
Agriculture and farming 43
Aims_oai.properties file 122
Aimsmeta.properties file 85
Aimsmetaindx command
    38, 89, 91, 107, 118, 132, 138
Applications 11, 44
ArcCatalog
  browsing published documents 58
  configuring Metadata Services for 29
  connecting to an ArcIMS server 58
  editing metadata 42. *See also* Authoring metadata
  finding GIS resources 58, 59
  managing Metadata Services
    browsing published documents 58
    copying published documents 54, 55
    creating folders 54
    deleting published documents 55
    private documents 57
ArcCatalog (continued)
  managing Metadata Services (continued)
    related documents 56, 57
    renaming published documents 57
  metadata synchronizers 44, 48
  publishing metadata 53. *See also* Publishing documents
  role in metadata system 4
  searching by
    Content Theme 60
    Content Type 59
    geographic location 59
    keyword 60
ArcExplorer
  add data from Metadata Explorer 14
  starting Metadata Explorer 94
ArcMap
  add data from Metadata Explorer 14
  starting Metadata Explorer 8
ArcSDE
  ArcIMS time-out parameter 110
  configuring, for Metadata Services 105–110
  connection information 96
  creating a view 92–93, 101
  database performance 110–111
  DBTUNE keywords
    defining 106–110
    using 34–35, 137
  deleting Metadata Service objects 101–102
  MAXBLOBSIZE parameter 31, 105
  Metadata Service database schema 92, 96–101
  role in Metadata Service 4, 20
  sdetable command 92–93
  sdexml command 36
  XML columns
    20, 35, 92, 96, 97, 99, 102, 103
ArcXML elements
  ADMIN_TABLE 39
  CHANGE_OWNER 119
  METADATA_CONTENT 26, 37, 108
  SDEWORKSPACE 26, 96, 132

ArcXML elements (continued)
TABLE_NAME 26, 96
Atmosphere climatology and meteorology 43
Authenticate.properties file 86
for Metadata Explorer 30, 86
for Z39.50 Connector 77
Authentication
and Z39.50 communication protocols 71–73, 76–77
Authenticate.properties file 30, 77, 86
basic or digest 29
described 20, 28
enabling
in Java Connector 30, 86
in Servlet Connector 29, 71–72, 76
in Z39.50 Connector 77
Esrimap_prop file 29, 31
Authoring metadata
automatically 42, 44, 48
browse graphics 49
choosing a metadata editor 46
creating an ISO-only document 48–49
creating thumbnails 52
for ArcIMS services 32, 47
preparing documents for publication 46–50
process 42
removing FGDC documentation hints 49
with FGDC editor 46–49
with ISO editor 49–51

B

Basic authentication 29
Biologic and ecologic 43
BLOB size parameter 31, 105
Browse graphics 31, 49
Browsing published documents 12, 54, 58
Business and economic 43

C

Cadastral and land planning 43
Character sets 132
Cjk parameter. *See* Localcodepage
Clearinghouse
as Content Type 11, 44
FGDC NSDI. *See* NSDI Clearinghouse
Collection identifiers 98, 102, 119
Connectors, enabling authentication
Java 30, 86
Servlet 29, 71–72, 76
Z39.50 76–77
Content Theme
categories
Administrative and political boundaries 43
Agriculture and farming 43
Atmosphere climatology and meteorology 43
Biologic and ecologic 43
Business and economic 43
Cadastral and land planning 43
Cultural society and demographic 43
Elevation and derived products 43
Environmental and conservation 43
Facilities and structures 43
Geologic and geophysical 43
Human health and disease 43
Imagery basemaps and land cover 44
Inland water resources 44
Locations and geodetic networks 44
Military and intelligence 44
Oceans and estuaries 44
Transportation networks 44
Utility and communication networks 44
defining in metadata
with FGDC editor 46, 48
with ISO editor 46, 50
described 11, 42
searching by 13, 60
Content Type
categories
Applications 11, 44
Content Type (continued)
categories (continued)
Clearinghouses 11, 44
Downloadable Data 10, 44
Geographic Activities 11, 44
Geographic Services 11, 44
Live Data and Maps 10, 44
Map Files 11, 44
Offline Data 10, 44
Other Documents 11, 44
Static Map Images 11, 44
defining in metadata
with FGDC editor 47, 49
with ISO editor 46, 51
described 10, 44
searching by 13, 59
Copying published documents 54, 55
Coverage area, viewing 14, 15
Cultural society and demographic 43

D

Database management system. *See* DBMS
DB2
configuration, for ArcSDE XML columns 103
database performance 110
DBTUNE keywords 107
indexing issues 107
DBMS
configuration, for ArcSDE XML columns
DB2 103
Oracle 102–103
SQL Server 103–105
Metadata Service database schema
administrative tables 23–25, 39, 119
creating a view 92–93, 101
deleted relationships table 100
deleted table 99–100
feature class 92, 96–99
gazetteer 88

DBMS (continued)
Metadata Service database schema (continued)
overview 96, 97
relationships table 100–101
removing objects from the database 101
users table 101–102
with ArcIMS 4.0.1 136
performance 110–111
role in Metadata Service 20
statistics 110
support 95
DBTUNE keywords
and gazetteer 106–107
and Metadata Services 106
DB2, for ArcSDE XML columns 107
IMS_METADATA 106–107
SQL Server, for ArcSDE XML columns 107–110
Deleting metadata, for ArcIMS services 32
Deleting published documents 55
Digest authentication 29
Document identifiers 24, 39, 45
Downloadable Data 10, 44

E

Editing metadata. *See* Authoring metadata
Elevation and derived products 43
Environmental and conservation 43
Esrimap_prop file 29, 31
Exporting published documents 116. *See also* Harvesting documents
Extent of resource, viewing 14, 15

F

Facilities and structures 43
Feature services
creating metadata 32, 47
deleting metadata 32
publishing metadata 53
Federal Geographic Data Committee. *See* FGDC
FGDC
Clearinghouse. *See* NSDI Clearinghouse
metadata
adding Content Theme 43, 46, 48
adding Content Type 47, 49
browse graphics 31, 49
publishing preparation 46–50
removing documentation hints 49
validation 23, 26, 42
Z39.50 search protocol
client software 81
gateway. *See* NSDI Clearinghouse
GEO Profile 63, 78, 81
ISite server 64, 66, 81
Finding GIS resources
from Z39.50 clients 61, 62
in ArcCatalog 58, 59
in Metadata Explorer 9, 12, 13
Folders, creating in Metadata Services 54

G

Gazetteer, for Metadata Explorer
adding places 90–92
configuration file, sample 84
customizing 89
database schema 88
defining places 90
enabling 85
indexing places 33, 89, 91–92
installing 87
loading provided places 88–89
removing places 92–93
searching with 10, 13
starting the service 84
upgrading for ArcIMS 9 136
Gazetteer.axl 84
GEO Profile 63, 78, 81
GEO_expanded.xml 78–79
Geographic Activities 11, 44
Geographic Services 11, 44
Geography Network
described 2
Explorer, searching NSDI Clearinghouse 13
sharing data on 3
Geologic and geophysical 43

H

Harvesting documents
changes to a Metadata Service 115
collection identifiers 98, 102, 119
described 114
exporting published documents 115, 116, 119
importing published documents 102, 115, 116, 117, 119
MDExport command 115, 116
MDImport command 115, 117
non-English metadata documents 132
OAI-PMH Connector
aims_oai.properties file 122
configuring 122–124
deploying the Web application 122
described 115
log file 124
verifying configuration 124
OAI-PMH Harvester
configuring 125–130
described 115, 125
harvesting by date 127
harvesting by folder 127
harvesting process 126
harvesting tasks 125
log options 129–130
log.properties file 129. *See also* Log.properties file: Z39.50 Connector
manual configuration 128–129
OAIServers.xml file 125, 126, 128
starting 125

Human health and disease 43

I

Image services
creating metadata 32, 47
deleting metadata 32
publishing metadata 53
Imagery basemaps and land cover 44
Importing published documents 116. *See also* Harvesting documents
IMS_METADATA DBTUNE keywords 106–107
Indexing documents
aimsmetaindx command 38, 89, 91, 118, 132, 138
and gazetteer places 89, 91
and importing documents 118
and publishing 21, 37. *See also* Publishing documents
and upgrading 138
automatically 21
configuration 22, 27, 33
definition file 22, 33
described 21
in DB2 databases 21, 107, 138
in SQL Server databases 21, 108–109
index types 22, 33, 34–35
indexing conflicts 23, 36
indexing numbers 22, 33, 36
indexing words 23
manually 21, 34, 37–38
MDAlterIndex command 34, 35
MDTagConflictList command 36
non-English documents 23, 36, 38, 132
performance 21, 23, 31
troubleshooting 36
Inland water resources 44
International Organization for Standardization. *See* ISO metadata
Internationalization 131
ISite Z39.50 server 64, 66, 81
ISO metadata
adding Content Theme 46, 50
adding Content Type 46, 51
creating an ISO-only document 48–49
publishing preparation 46–50
validation 23, 26, 42

J

Java Connector, enabling authentication 30, 86

L

Live Data and Maps
as Content Type 10, 44
printing, in Metadata Explorer 17
viewing services 14, 16, 58
Localcodepage
defining in configuration file 132
specifying in commands 34, 36, 38, 116, 117, 132, 137
Locations and geodetic networks 44
Log.properties file
OAI-PMH Harvester 129
Z39.50 Connector 68, 69, 70
Logging in, to access Metadata Services 8, 58, 86
Login page, configuring for Metadata Explorer 30, 85, 86

M

Map Files 11, 44
MAXBLOBSIZE parameter 31, 105
MDAlterIndex command 34
MDExport command 115, 116
MDImport command 115, 117
MDTagConflictList command 36
MDUpgrade command 137
Metadata
browse graphics 49
creating 42. *See also* Authoring metadata
deleting, for ArcIMS services 32
described 1, 41
document identifiers 24, 39, 45
publishing 53. *See also* Publishing documents
standards 23, 42
synchronizers 44, 48
validating content 23, 26, 42
viewing, in Metadata Explorer 14, 15
Metadata Explorer
adding data to ArcExplorer 14
adding data to ArcMap 14
browsing documents 12, 54
configuring
access control list 86–87
aimsmeta.properties 85
Authenticate.properties file 30, 86
authentication 30, 86–87
gazetteer 84, 85. *See also* Gazetteer
login page 30, 85, 86
metadata service 85
search map 84, 85
search results size 85
session time-out 94
described 4, 7
finding GIS resources 9, 13
installing 83
logging in 8, 86
printing from 17
related documents 56
search results, viewing 14
searching by
Content Theme 11
Content Type 10
geographic location 10
keyword 11
starting
from ArcExplorer 94

Metadata Explorer (continued)
starting (continued)
from ArcMap 8
from Web browser 8
viewing
coverage area 14, 15
Live Data and Maps 14, 16
metadata 14, 15
spatial extent of resource 14, 15
Metadata Service commands
aimsmetaindx 38, 89, 91, 107, 118, 132
MDAlterIndex 34, 35
MDExport 115, 116
MDImport 115, 117
MDTagConflictList 36
MDUpgrade 137
Metadata Services
collection identifiers 98, 102, 119
components 3
configuring
access control list 20, 28
administrative tables 23–25, 39, 119
ArcSDE 105–110
ArcSDE connection information 96
authentication 20, 28
configuration file, sample 84
DBMS 102, 103
indexing documents 21, 27, 33–36, 108
internationalization 131
localcodepage 132
table name prefix 26, 96
validation 23, 26
database performance 110–111
database schema 92, 96–101
described 2, 20
document identifiers 24, 39, 45
export format 119
exporting published documents 116. *See also* Harvesting documents
harvesting documents 114. *See also* Harvesting documents
Metadata Services (continued)
importing published documents 116. *See also* Harvesting documents
log file 31
managing published documents
copying documents 54, 55
creating folders 54
deleting documents 55
indexing conflicts 36
indexing documents 38. *See also* Indexing documents
private documents 57, 72
related documents 56, 57
renaming documents 57
permissions 28. *See also* Access control list: roles
publishing documents 53. *See also* Publishing documents
removing objects from database 101–102
roles 28. *See also* Access control list: roles
starting 26, 27
troubleshooting 31
upgrading for ArcIMS 9 135
what's new for ArcIMS 9 134
METADATA_CONTENT element 26, 37, 108
MetadataServer.axl 26, 84
MetaIDChecker utility 45
Migrating to ArcIMS 9
upgrading 136
what's new 134
Military and intelligence 44

N

National Spatial Data Infrastructure (NSDI) 61, 114. *See also* NSDI Clearinghouse
NSDI Clearinghouse
described 61, 62, 81, 114
participating in, with Z39.50 Connector 62, 77
searching 13

O

OAI-PMH Connector 115. *See also* Harvesting documents: OAI-PMH Connector
OAI-PMH Harvester 115. *See also* Harvesting documents: OAI-PMH Harvester
OAIServers.xml file 125, 126, 128
Oceans and estuaries 44
Offline Data 10, 44
Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) 114
Oracle
configuration, for ArcSDE XML columns 102–103
database performance 110–111
Other Documents 11, 44

P

Panning and zooming
exploring Live Data and Maps 14
setting the search extent 10
Passwords. *See* Access control list
Permissions. *See* Access control list: roles
Prefix. *See* Table name prefix
Printing, from Metadata Explorer 17
Private documents 57, 72
Publishing documents
and indexing 21. *See also* Indexing documents
in ArcCatalog 53
non-English documents 131
performance 21, 23
preparing documents
creating ISO-only documents 48–49
removing FGDC hints 49
with FGDC editor 47–49
with ISO editor 49–51
referencing large browse graphics 49, 105
roles, in access control list 28, 53, 56

Publishing documents (continued)
  troubleshooting 31
  validation 23, 26, 42

R

Related documents 56, 57
Relational database management system. *See* DBMS
Renaming published documents 57
Roles, in access control list
  metadata_administrator 28, 57, 87
  metadata_browser 28
  metadata_browser_all 28
  metadata_publisher 28, 53, 56
  metadata_service_author 28, 32

S

Sdetable command 92–93
SDEWORKSPACE element 26, 96, 132
Sdexml command 36
Search map, for Metadata Explorer
  enabling 85
  sample configuration file 84
  searching with 10
  starting the service 84
Search results, in Metadata Explorer
  exploring 14
  printing 17
  setting number per page 85
Searching by
  Content Theme 13, 60
  Content Type 13, 59
  geographic location 13, 59
  keyword 13, 60
Searching for GIS resources
  from Z39.50 clients 61, 62
  in ArcCatalog 58, 59
  in Metadata Explorer 9, 12, 13
SearchMap.axl 84
Servlet Connector, enabling authentication 29, 76
Spatial Data Infrastructure (SDI) 114
SQL Server
  and non-English documents 109–110
  configuration, for ArcSDE XML columns 103–105
  database performance 110
  DBTUNE keywords 107–110
  indexing options 108–109
Standards, for metadata 23, 42
Static Map Images 11, 44
Synchronizers, in ArcCatalog 44, 48

T

Table name prefix
  and database schema 88, 96, 97, 101
  defining in configuration file 26, 137
  specifying in commands 34, 36, 38, 89, 91, 92, 137
TABLE_NAME element 26, 96
Thumbnails 31
Time-out parameter 110
Transformations.xml 79, 80
Transportation networks 44
Troubleshooting
  indexing issues 36
  publishing issues 31
  Z39.50 Connector issues 64, 68

U

Upgrading Metadata Services 137
Usernames. *See* Access control list
Utility and communication networks 44

V

Validation 23, 26, 42
Views, creating in DBMS 92–93, 101

W

Web browser
  searching for data from 9
  starting Metadata Explorer 8

X

XML columns, in ArcSDE 20, 35, 92, 96, 97, 99, 102, 103
XML documents, publishing 23, 42

Z

Z39.50 Connector
  accessing Metadata Services 67, 72, 77
  and private documents 72
  ArcIMS Z3950 service
    console window 64, 65, 68
    starting the service 64
  configuring
    access control list 71–72, 73, 76–77
    Authenticate.properties file 77
    authentication 71–73, 76–77
    communication protocol 71–73, 74, 76–77
    default login 76, 77
    default Metadata Service 67
    GEO_expanded.xml 78–79
    log.properties file 68–70
    logs 68–70
    port, for ArcIMS Z3950 service 64, 66
    transformations.xml 79
    with HTTP or HTTPS 74
    with TCP/IP 75
    z3950.properties file 79
  customizing 78–79
  described 61, 63
  how it works 63, 78–79
  installing 64
  participating in NSDI Clearinghouse 62, 77

Z39.50 Connector (continued)
  response formats 79, 80
  role in Metadata Service 61, 62, 63
  testing 65
  troubleshooting 64, 68
  use attribute codes 78
  Z39.50 support 63, 78, 79
Z39.50 search protocol
  client software 62, 72, 81
  described 63
  GEO Profile 63, 78, 81
  institutions using 61
  ISite server 64, 66, 81
  profiles 63
  resources 63, 81
  search gateways
    Library of Congress 62
    NSDI Clearinghouse 62. *See also* NSDI Clearinghouse
  specification 81
  Z39.50 Implementors Group (ZIG) 63
  Z39.50 Maintenance Agency 63, 78, 81
Z3950.properties file 66, 67, 76, 77, 79, 80