# APPLICATION OF MASTER OF ENGINEERING - GEOGRAPHIC INFORMATION SYSTEMS (GIS) SPECIALTY COURSEWORK TO SEVERAL PROJECTS FOR IMPROVING THE ADMINISTRATION OF A COLORADO STATE AGENCY ENTERPRISE GIS

by

Robert Sacco

A Capstone Project Report submitted in partial fulfillment of the requirements for the degree of

Master of Engineering – Geographic Information Systems Specialty

University of Colorado - Denver

2012

UNIVERSITY OF COLORADO - DENVER

ABSTRACT

Application of Master of Engineering - Geographic Information Systems (GIS) Specialty Coursework
to Several Projects For Improving the Administration of a Colorado State Agency Enterprise GIS

by Robert Sacco

| | |
|---|---|
| Chairperson of the Supervisory Committee: | Professor Dr. Bruce Janson<br>Department of Engineering |

A Capstone project report detailing how the skills learned and developed in courses taken toward the completion of the Master of Engineering - Geographic Information Systems (GIS) Specialty directly applied to several completed or on-going projects for improving the administration of a Colorado state agency enterprise GIS. These projects include development of a GIS Unit intranet site for the purpose of communicating important information about the enterprise GIS to end users; development of a database and intranet web application for capturing and displaying species data collection efforts occurring within the agency; development of a GIS web mapping application to display species data to replace printed maps; schema changes to existing data structures to make the data more relevant and credible; development of a spatial database to capture travel locations to improve scheduling logistics for traveling to capture GIS edits; and development of Python scripts and ESRI Add-Ins to automate making schema changes to enterprise databases, as well as other administrative tasks, and capturing the date and user making edits to enterprise data features.

# TABLE OF CONTENTS

# LIST OF FIGURES

# PREFACE

I matriculated the University of Colorado Denver Civil Engineering Department in the Spring of 2009 having chosen the Master's of Engineering - Geographic Information Systems (GIS) Specialty largely due to the partnership between the Civil Engineering Geographic Information System program and the Business School's System Development and Implementation program.

This partnership appealed to me as I have had a career in the GIS field for a number of years but was interested in completing course work that would develop or improve my technical skills in the area of web programming, GIS web applications in particular; and enterprise GIS database administration.

My intent in developing these skills was to be able to improve upon several aspects of the administration of the enterprise GIS of my employer, Colorado Parks and Wildlife (CPW).

My hope is that this Capstone report demonstrates how I was able to directly take the skills I have developed in each of the courses I completed to fulfill the requirements of the MEng - GIS Specialty degree and apply them to projects for CPW that improved upon some aspect of its enterprise GIS.

## ACKNOWLEDGMENTS

I wish to express my deepest thanks to Jon Kindler, who initially approved my enrollment in the program and provided authorization of training dollars from his budget so I could take these courses, and to Seth McClean who continued to authorize the use of his budget so I could continue my coursework and complete the degree requirements, even through a period of time when I was not technically his employee. Without Jon and Seth's support I would not have been able to even think about enrolling in or completing the program.

I also wish to express sincere appreciation to Dr. Janson for guiding me through the final stages of the program. I'd also like to thank Professor Mannino in the Business School for starting me on the way with web application development and design, as well as database warehousing and Oracle programming; Professor Gunther for sharing his background and expertise on the GeoWeb; Professor Steigerwald for furthering my Python skills and introducing me to ArcObjects and Add-Ins; and all my other professors from the Civil Engineering Department who contributed to helping me fulfill my goal of increasing my knowledge and understanding of GIS through their dedication to providing relevant and up to date course material. This is an industry that is constantly changing and the effort they all have obviously put in to revising their course curriculum's to keep up with these changes needs to be recognized and commended.

I also need to thank my wife Kristine and my daughter Aucklynn for both their patience and support while I pursued this degree. They have been more understanding than I could ever have expected when I had to sacrifice spending time with them while working on studying, midterms, finals, and projects. I owe them both so much now.

Finally, I would like to dedicate this report to Greg Bryant. After having attended a presentation Greg gave at an ESRI User Conference a number of years ago, speaking with him at GIS In The Rockies, and exchanging a couple of e-mails, I felt we shared a common viewpoint about GIS and its role within organizations. Greg played a large part in my interest in enrolling at the University of Colorado Denver and I am sorry I did not enroll in his class sooner and have a chance to get to know him better and learn from his extensive experience in the field.

# ISMG 6240 – WEBSITE DEVELOPMENT PRACTICE / CPW INTRANET

## INTRODUCTION

When an agency or organization begins the development of an enterprise Geographic Information System (GIS) infrastructure, the focus of managing the project is on technical details. The initial discussion and documentation for starting an enterprise GIS project centers on software, hardware, storage requirements, network transmission, security, as well as a host of other important technical factors. However, a key component of the enterprise GIS infrastructure that is generally overlooked in its development is, how will the operation of the enterprise GIS be communicated to the personnel who will have a need to use it. Examples of aspects of the enterprise GIS that need to be communicated to users include, how to install software and location of installation and service pack files, instructions on connecting to enterprise databases with descriptions of the data they contain, on-going updates to both software and data, availability of training resources, how-to's on common functions, as well as other relevant information on how the enterprise GIS supports the specific needs of the organization's end users.

Colorado Parks and Wildlife (CPW) has gone through this evolution of establishing an enterprise GIS and then struggling with an effective means of communicating critical information of how that system operates to the end users that the system was intended to benefit. CPW found that by not having considered how information about its enterprise GIS would be communicated to end users during its inception resulted in an under utilization of the system.

Initial communication about the various aspects of the agency's enterprise GIS consisted of demonstrations at agency section meetings, articles in an inconsistently e-mail delivered quarterly newsletter, and targeted e-mails. The problems with these methods were that they are limited in both their ability to keep up with frequent updates, and the scope of the audience they can reach. New versions of software and service packs, availability of data and updates, and employees are constantly changing which requires a system of communication that is readily available to all staff with the most current and relevant information.

With this in mind, I was interested in developing an intranet presence for the CPW GIS Unit that would be a platform for the timely communication of relevant enterprise GIS information for all CPW staff interested in using it.

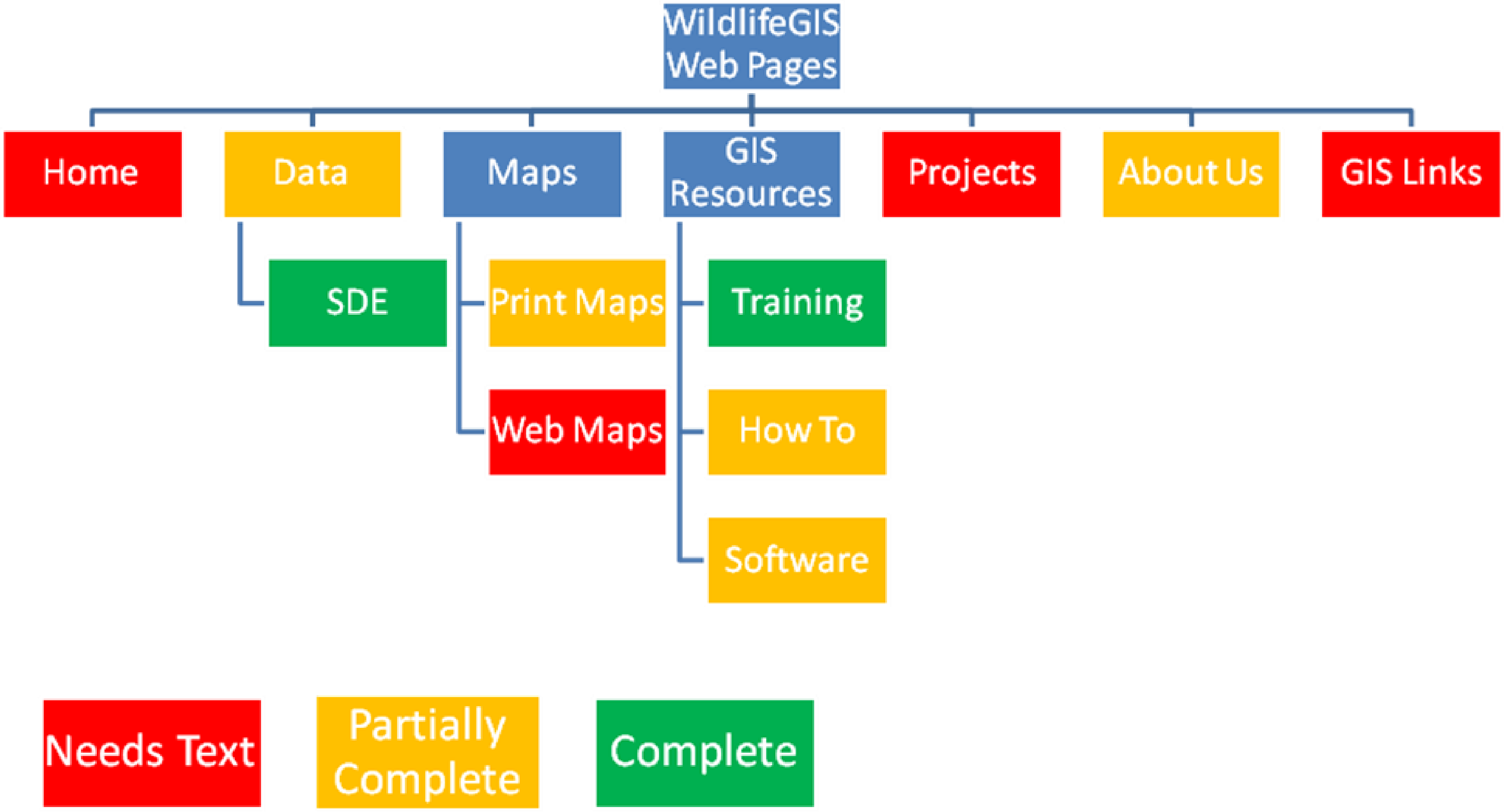
## METHODS

To complete this project with no prior experience with web programming, I enrolled in ISMG 6240 – Website Development Practice as my first course in the program to gain the skills that would be necessary. This course provided a sound foundation and practice using Extensible Hypertext Markup Language (XHTML), Cascading Style Sheets (CSS), and VB.NET, as well as accepted web design principles.

Using what I learned from this class, I first assembled a team of stakeholders from the GIS Unit and scheduled several meetings to brainstorm what the content the GIS Unit wanted to communicate to other CPW staff would be on these intranet pages and how that content would be organized. During the kickoff meeting, index cards were used to quickly create content page headings and listing the sections that would be hosted on these content pages as staff came up with them. Once consensus was reached on the content and its layout, I developed an information architecture using MS PowerPoint from the index cards to visually demonstrate the general content organization for the website (Figure 1). At subsequent meetings this information architecture was used to further add new content and reorganize existing content so it would be easier for end users to navigate to appropriate content. Through the use of color coding the various content pages, the information architecture also provided a quick progress report of the completion status for the web pages throughout the development cycle.

I worked with staff proficient in graphic design to come up with an image for the header of all pages unique for the GIS Unit but which contained the CPW logo and text. I used the predominant colors in the header image to develop a color scheme for the pages by finding complementary colors using Adobe's Kuler web site. I created storyboards that showed the design layout of the site with header, footer, side navigation, and main content sections along with the header image and color design. The storyboard was also presented at meetings to discuss these intranet pages and changes were iteratively made based on continuous feedback.

**Figure 1. Web Design Information Architecture.**



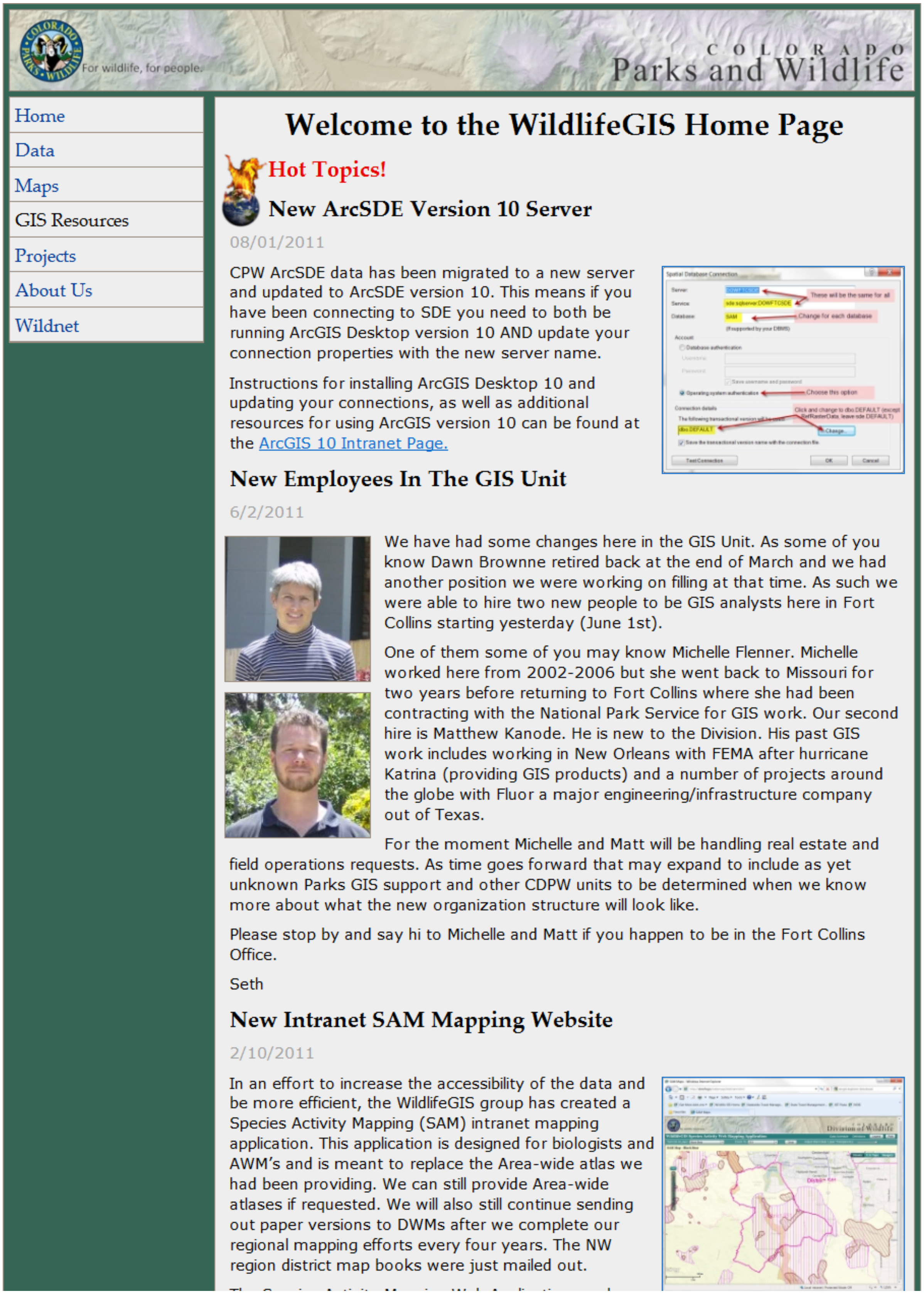
Once the content, organization, and layout were agreed to by staff participating in these web development meetings, I began to work on coding the web site. I used MS Visual Studio 2008 to create the ASP.NET version 3.5 framework web pages. The first step was to develop a Master Page using XHTML that contained the elements that would be found on all pages including the header, footer, side navigation, and main content (Appendix A). Within the main content element a placeholder was added for which the unique content for each page would be placed. A CSS stylesheet was developed to control formatting of the navigation menu and all other XHTML elements (Appendix B). Finally, individual content pages were developed based on the Master template according to the information architecture with text and other content developed both by me, as well as other CPW GIS Unit staff.

## RESULTS

The result of this project was a series of web pages that provide an intranet presence for the Colorado Parks and Wildlife GIS Unit to communicate information about its personnel, on-going projects, training, and other GIS related information. Most importantly, there are pages that provide CPW employees, including biologists, law enforcement officers, engineers, real estate professionals, and others that are interested in using GIS information on what software is available to them, how they can install the appropriate versions of that software and its most recent service packs, as well as instructions for connecting to CPW hosted enterprise data and how to make connections to various relevant external data sources. Figures 2-4 show examples of the site's Home Page, SDE Connection instruction page, and ArcGIS Installation page.

By working with the agency's IT Unit, a link to these pages was added to the organization's main intranet page. This has allowed new employees interested in using GIS to quickly find and access the relevant information for getting them started with installing software and accessing data. In the past they would have had to find out from word of mouth, or by digging around, someone to contact in the GIS Unit for providing them with software installation and data connection instruction documents. Having these documents already posted to these web pages has resulted in gained efficiencies through saving GIS Unit staff time in e-mailing individuals these documents. In addition,

Figure 2. CPW Intranet Home Page.



## Welcome to the WildlifeGIS Home Page

Hot Topics!

### New ArcSDE Version 10 Server

08/01/2011

CPW ArcSDE data has been migrated to a new server and updated to ArcSDE version 10. This means if you have been connecting to SDE you need to both be running ArcGIS Desktop version 10 AND update your connection properties with the new server name.

Instructions for installing ArcGIS Desktop 10 and updating your connections, as well as additional resources for using ArcGIS version 10 can be found at the ArcGIS 10 Intranet Page.

Spatial Database Connection

Server: DOWFTCSDE — These will be the same for all

Service: sde:sqlserver:DOWFTCSDE

Database: SAM — Change for each database

(If supported by your DBMS)

Account

○ Database authentication

Username:

Password:

☑ Save username and password

● Operating system authentication — Choose this option

Connection details

The following transactional version will be used:

dbo.DEFAULT — Click and change to dbo.DEFAULT (except RefRasterData, leave sde.DEFAULT)

☑ Save the transactional version name with the connection file.

### New Employees In The GIS Unit

6/2/2011

We have had some changes here in the GIS Unit. As some of you know Dawn Brownne retired back at the end of March and we had another position we were working on filling at that time. As such we were able to hire two new people to be GIS analysts here in Fort Collins starting yesterday (June 1st).

One of them some of you may know Michelle Flenner. Michelle worked here from 2002-2006 but she went back to Missouri for two years before returning to Fort Collins where she had been contracting with the National Park Service for GIS work. Our second hire is Matthew Kanode. He is new to the Division. His past GIS work includes working in New Orleans with FEMA after hurricane Katrina (providing GIS products) and a number of projects around the globe with Fluor a major engineering/infrastructure company out of Texas.

For the moment Michelle and Matt will be handling real estate and field operations requests. As time goes forward that may expand to include as yet unknown Parks GIS support and other CDPW units to be determined when we know more about what the new organization structure will look like.

Please stop by and say hi to Michelle and Matt if you happen to be in the Fort Collins Office.

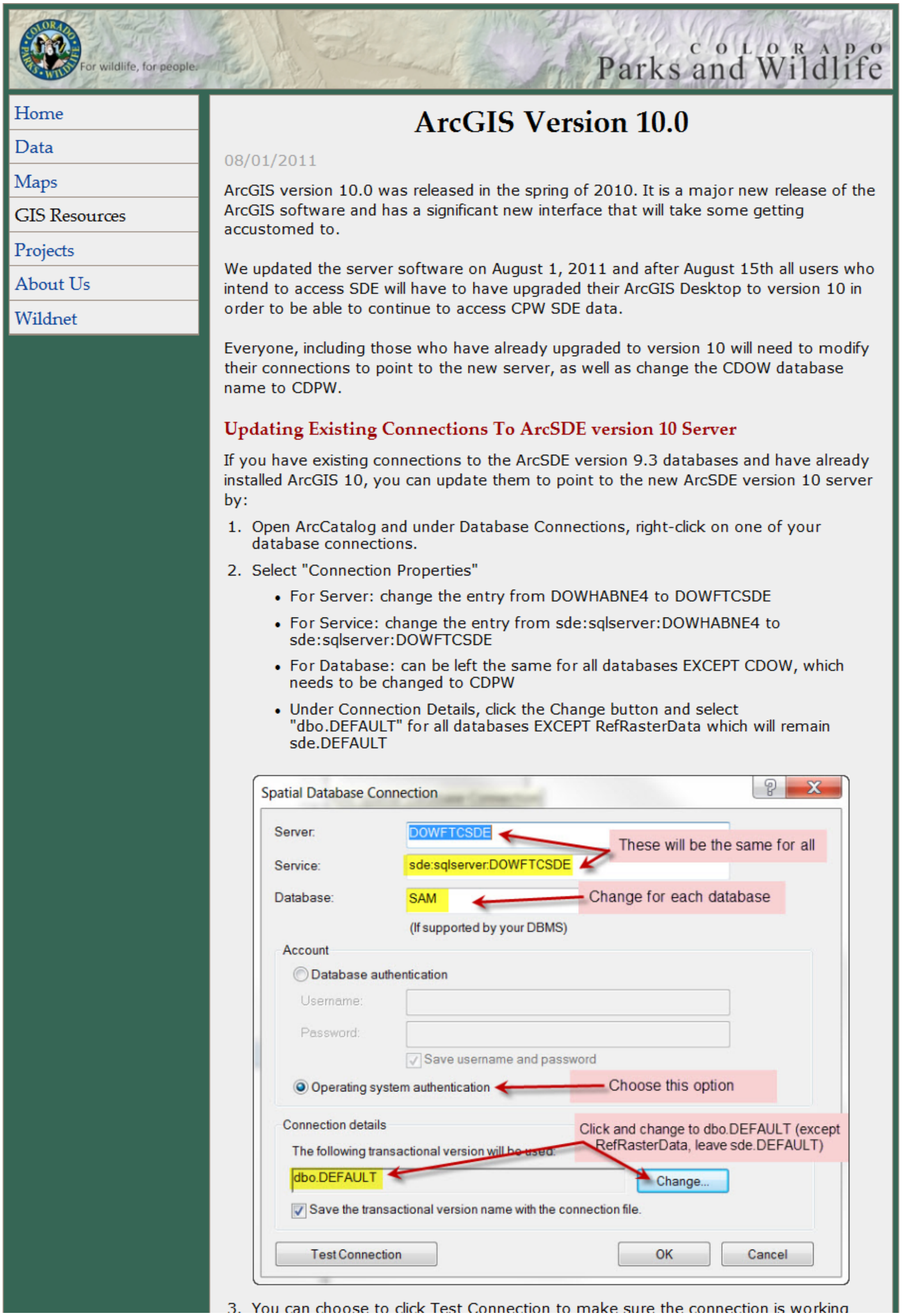
Seth

### New Intranet SAM Mapping Website

2/10/2011

In an effort to increase the accessibility of the data and be more efficient, the WildlifeGIS group has created a Species Activity Mapping (SAM) intranet mapping application. This application is designed for biologists and AWM's and is meant to replace the Area-wide atlas we had been providing. We can still provide Area-wide atlases if requested. We will also still continue sending out paper versions to DWMs after we complete our regional mapping efforts every four years. The NW region district map books were just mailed out.

Figure 3.  CPW Intranet Software Update Page.



# ArcGIS Version 10.0

08/01/2011

ArcGIS version 10.0 was released in the spring of 2010. It is a major new release of the ArcGIS software and has a significant new interface that will take some getting accustomed to.

We updated the server software on August 1, 2011 and after August 15th all users who intend to access SDE will have to have upgraded their ArcGIS Desktop to version 10 in order to be able to continue to access CPW SDE data.

Everyone, including those who have already upgraded to version 10 will need to modify their connections to point to the new server, as well as change the CDOW database name to CDPW.

## Updating Existing Connections To ArcSDE version 10 Server

If you have existing connections to the ArcSDE version 9.3 databases and have already installed ArcGIS 10, you can update them to point to the new ArcSDE version 10 server by:

1. Open ArcCatalog and under Database Connections, right-click on one of your database connections.
2. Select "Connection Properties"
   - For Server: change the entry from DOWHABNE4 to DOWFTCSDE
   - For Service: change the entry from sde:sqlserver:DOWHABNE4 to sde:sqlserver:DOWFTCSDE
   - For Database: can be left the same for all databases EXCEPT CDOW, which needs to be changed to CDPW
   - Under Connection Details, click the Change button and select "dbo.DEFAULT" for all databases EXCEPT RefRasterData which will remain sde.DEFAULT

Spatial Database Connection

Server: DOWFTCSDE

Service: sde:sqlserver:DOWFTCSDE

These will be the same for all

Database: SAM

Change for each database

(If supported by your DBMS)

Account

○ Database authentication

Username:

Password:

☑ Save username and password

● Operating system authentication

Choose this option

Connection details

The following transactional version will be used:

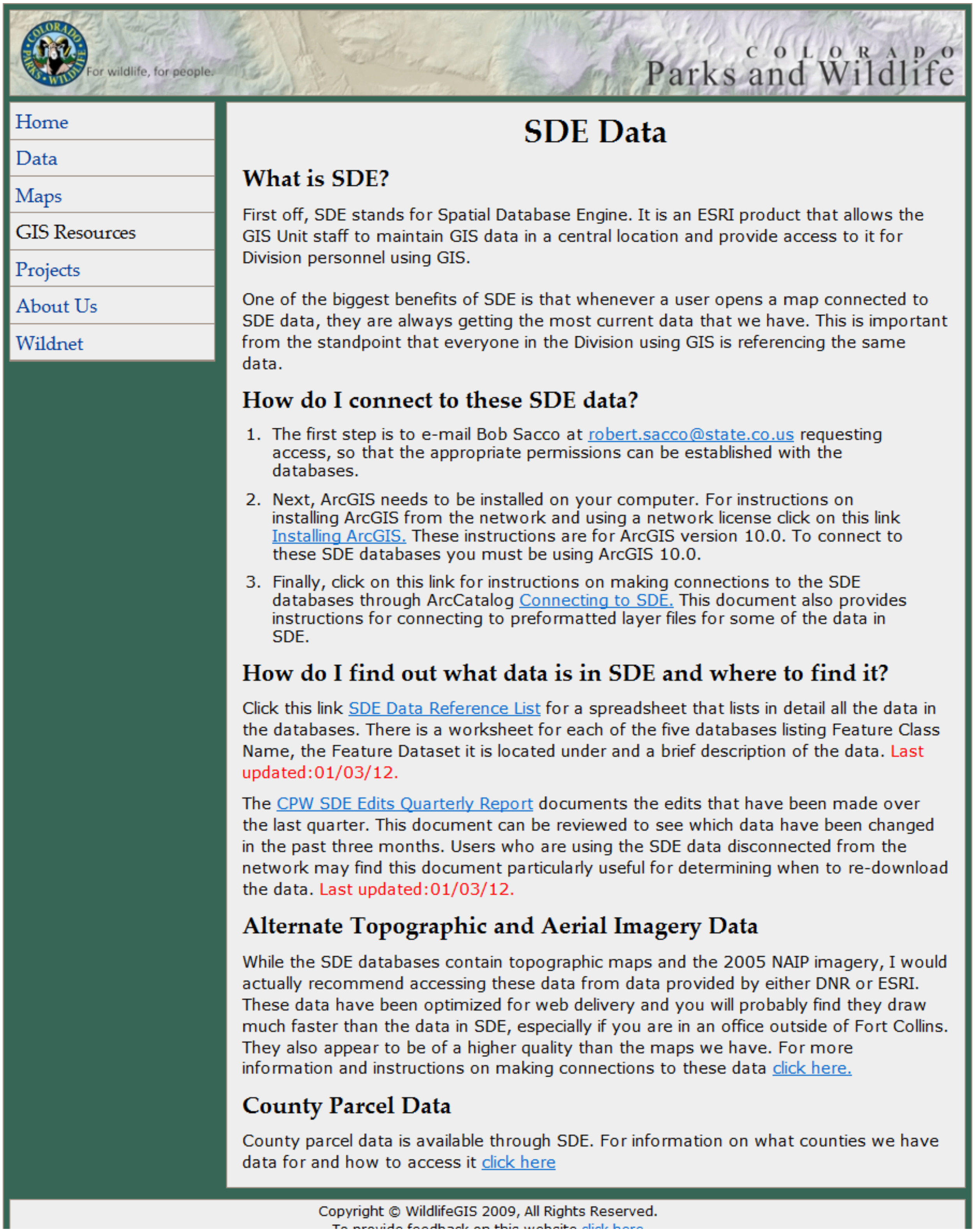
dbo.DEFAULT

Click and change to dbo.DEFAULT (except RefRasterData, leave sde.DEFAULT)

☑ Save the transactional version name with the connection file.

3. You can choose to click Test Connection to make sure the connection is working

**Figure 4. CPW SDE Connection Page.**



# SDE Data

## What is SDE?

First off, SDE stands for Spatial Database Engine. It is an ESRI product that allows the GIS Unit staff to maintain GIS data in a central location and provide access to it for Division personnel using GIS.

One of the biggest benefits of SDE is that whenever a user opens a map connected to SDE data, they are always getting the most current data that we have. This is important from the standpoint that everyone in the Division using GIS is referencing the same data.

## How do I connect to these SDE data?

1. The first step is to e-mail Bob Sacco at robert.sacco@state.co.us requesting access, so that the appropriate permissions can be established with the databases.
2. Next, ArcGIS needs to be installed on your computer. For instructions on installing ArcGIS from the network and using a network license click on this link Installing ArcGIS. These instructions are for ArcGIS version 10.0. To connect to these SDE databases you must be using ArcGIS 10.0.
3. Finally, click on this link for instructions on making connections to the SDE databases through ArcCatalog Connecting to SDE. This document also provides instructions for connecting to preformatted layer files for some of the data in SDE.

## How do I find out what data is in SDE and where to find it?

Click this link SDE Data Reference List for a spreadsheet that lists in detail all the data in the databases. There is a worksheet for each of the five databases listing Feature Class Name, the Feature Dataset it is located under and a brief description of the data. Last updated:01/03/12.

The CPW SDE Edits Quarterly Report documents the edits that have been made over the last quarter. This document can be reviewed to see which data have been changed in the past three months. Users who are using the SDE data disconnected from the network may find this document particularly useful for determining when to re-download the data. Last updated:01/03/12.

## Alternate Topographic and Aerial Imagery Data

While the SDE databases contain topographic maps and the 2005 NAIP imagery, I would actually recommend accessing these data from data provided by either DNR or ESRI. These data have been optimized for web delivery and you will probably find they draw much faster than the data in SDE, especially if you are in an office outside of Fort Collins. They also appear to be of a higher quality than the maps we have. For more information and instructions on making connections to these data click here.

## County Parcel Data

County parcel data is available through SDE. For information on what counties we have data for and how to access it click here

Copyright © WildlifeGIS 2009, All Rights Reserved.
To provide feedback on this website click here

e-mailing these types of documents for updates to software version, servers, and data changes to employees already using the system was inefficient from a number of standpoints.  First, attachments to e-mails resulted in an unnecessary increase in storage being used on e-mail servers.  It was also found that end users receiving a high volume of e-mail would overlook these announcements and documents.  Some staff had difficulty managing these documents and would access older, out-of-date documents that provided instructions that no longer worked causing frustration on their part and taking up more GIS Unit staff time re-emailing the appropriate documents.

Over the past three years I have maintained this web site as the intranet presence for the CPW GIS Unit and have continually added or updated pages as necessary to keep content current and relevant. This site is now the main access point for new employees to learn about and implement the agency's GIS resources available to them.  Instruction documents and other information can more easily be updated and posted to the site.  These documents are only stored once on the server and always have the most current and relevant information.  Having this information more readily available to staff has resulted in a significant increase in the use of, and interest in the agency's enterprise GIS.  It has also reduced the amount of GIS Unit staff time used for individually distributing this type of information, along with reducing storage on e-mail servers for this type of support documentation.

# ISMG 6480 – DATA WAREHOUSING AND ADMINISTRATION / SPECIES DATA INVENTORY

## INTRODUCTION

Another issue organizations need to consider when implementing an enterprise GIS is exactly what data will be stored in the enterprise databases. Ideally, all of an organization's data would be housed in enterprise databases. In reality, the Colorado Parks and Wildlife GIS Unit has had to make decisions regarding what data will be stored and managed within its enterprise GIS. These decisions have had to be made due to the number of field personnel permanently and temporarily employed by the agency who are collecting species field data. These data are often collected on a wide range of species and at a variety of scales ranging through a particular property to an Area within a Region, to a Region within the state of Colorado, to statewide data. The GIS Unit has made the decision to focus the efforts of its enterprise to species that are economically important, or threatened and/or endangered and for which the data is being collected either statewide or for the complete range of the species in Colorado. This compromise has had to be made due to the personnel resources of the GIS Unit and their ability to assist with administrating field collected data into the enterprise relative to the volume of data collection efforts.

However, biologists and other personnel may be collecting data at local levels that are stored in various formats dependent on their technical comfort with various tools such as MS Access databases, Excel spreadsheets, or Word documents. While these data are not stored within the enterprise GIS they hold immeasurable value in a number of ways. Localized studies can be used to contribute knowledge, assist with, and help refine the development of statewide data. Opportunistic data capture on species that may currently be under the radar may become important in the future, if for example, the species were to begin an unforeseeable decline in population. In addition, these studies with already developed methodologies, data structures, and established values can serve as a template for being expanded into other areas and having the ability for the data to be seamlessly integrated together.

For these reasons, the Director of Colorado Parks and Wildlife convened a Data Committee to look into and address how the agency collected, managed, stored, used, and distributed data. As this committee met, it became clear that it first needed to address knowing what data was being collected within the agency before it could begin to tackle any of these other issues. The agency has numerous employees spending considerable effort collecting data that is often stored on individual desktops, laptops, or filing cabinets and which may be used for the reports or papers they were intended to produce but which could be reused in numerous other ways.  Also, when employees retire these data may become obscure and their value lost to the agency.

As a member of this committee and seeing the need for providing a simple way to capture and display the species data collected by the agency I decided to put the skills I learned in the ISMG 6480 Data Warehousing / Administration class to use to address this problem.

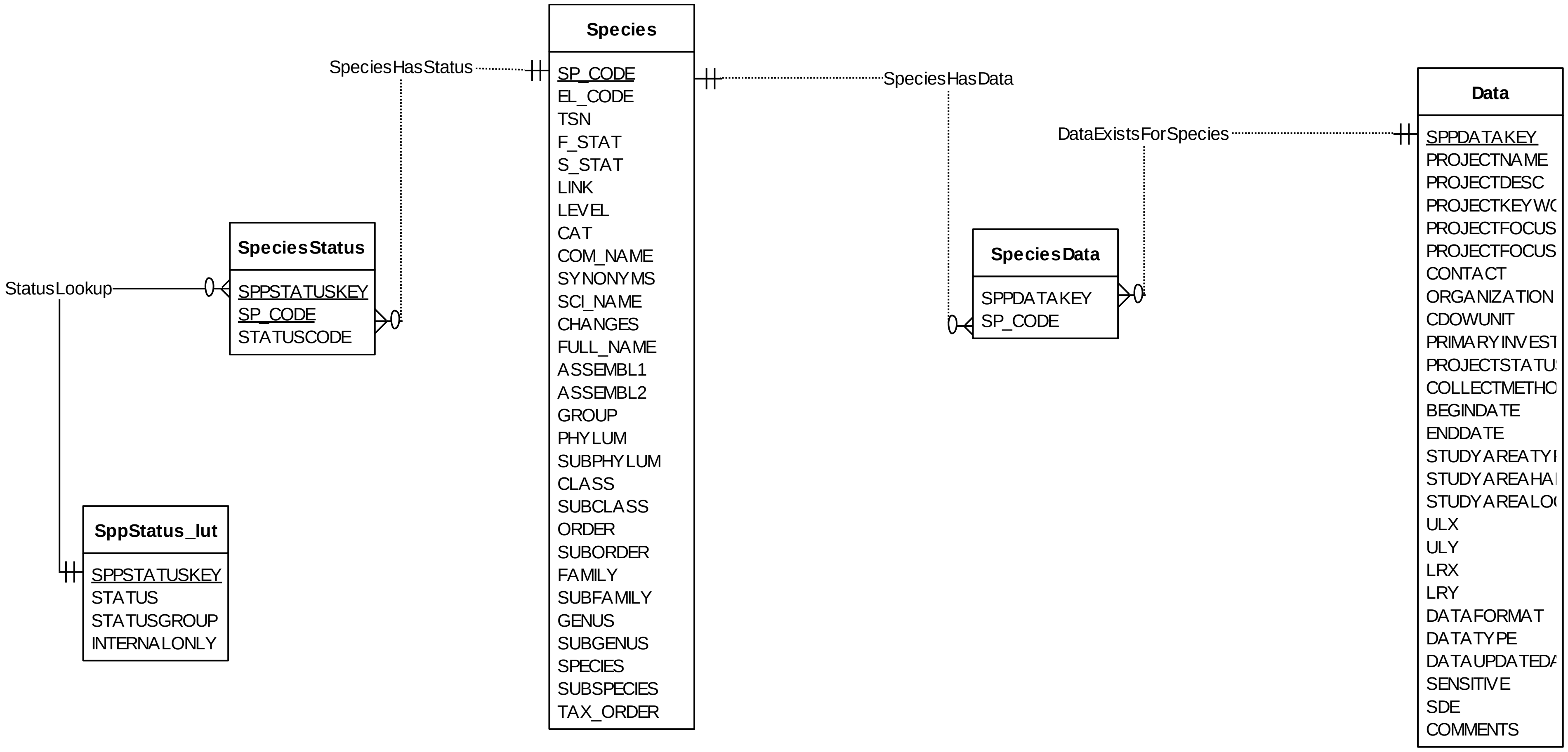
## METHODS

Again, I began by working with a sub-committee of staff from the larger Data Committee along with other interested stakeholders.  The purpose of this group was to identify information that would be important to capture to provide an overview of the data being collected and contact/location information on where the data could be found.  This group also attempted to find a balance of capturing enough information about the data being collected that would be sufficient in providing someone conducting a search of data the ability to determine if the data were relevant to what they were looking for, while keeping the burden of entering this information to the staff maintaining the data to a minimum.

Once this group identified the information to be collected and presented to users, an Entity Relationship (ER) diagram was created to schematically represent tables, attributes, and relationships (Figure 5).  This ER diagram was constructed using ER Assistant and was used to help committee members easily visualize the data structure and information and further refine the database schema.

During this process, it was also recognized that a list of species with both common and scientific names was going to be a cornerstone of this database.  Considerable effort was put into finding and resurrecting an existing table that had been developed in the past and cleaning it up.  Cleaning up this table using a MS Access database included, making updates to taxonomic changes that had occurred since it was first created, adding missing species, working through creating or eliminating subspecies information as appropriate, and creating fields for filtering data to maintain a full species list table while having the ability to limit the list to species that were most currently relevant to CPW.

**Figure 5. ER Diagram for Species Data Inventory Database.**



It was also determined that an additional benefit of this database with a complete species list table would be to identify any important species conservation listings the species may belong to and having the ability to select one of these conservation lists and having it return all the species on it. Previous methods of finding this information were difficult to find and scattered across various web sites. Having them in one easily searchable location would provide an additional benefit to the organization. Tables and relationships to provide this information were also added to the ER diagram.

Once the ER Diagram was finalized, I began work on using Microsoft SQL Server 2008 Management Studio for creating SQL Data Definition Language code to create the tables, keys, relationships, and constraints in Microsoft SQL Server 2008 R2 (Appendix C). With the database structure completed, reference or lookup tables were populated with known values and known existing data was used to populate the main data tables.

Finally, Microsoft Visual Studio 2008 was used to develop a VB.NET web application within the existing structure of the CPW GIS Unit intranet pages to provide tools for enabling multi-user data entry and retrieval of data from the database that were easy to use and widely accessible.

## RESULTS

The result of this project was an enterprise database established in Microsoft SQL Server 2008 that contains a list of wildlife species, information on data collection efforts that have occurred in the past or that are currently on-going for these species, and information about what species exist on conservation lists such as their threatened or endangered status at both the federal and state level. Figure 6 contains a database diagram generated from SQL Server showing the final tables in the database, their attributes and keys, and their relationships. This database is made accessible to staff through an intranet application that allows them to easily enter data about their projects, along with querying species to determine what data exists in the agency for that species, where to access the data, and who to contact for more information about it. The web application also provides the conservation lists a species exists on or allows the user to choose a conservation list and retrieve a list of all the species in Colorado that are on it (Figures 7-11).

**Figure 6. SQL Server Database Diagram of final Species Inventory Database.**

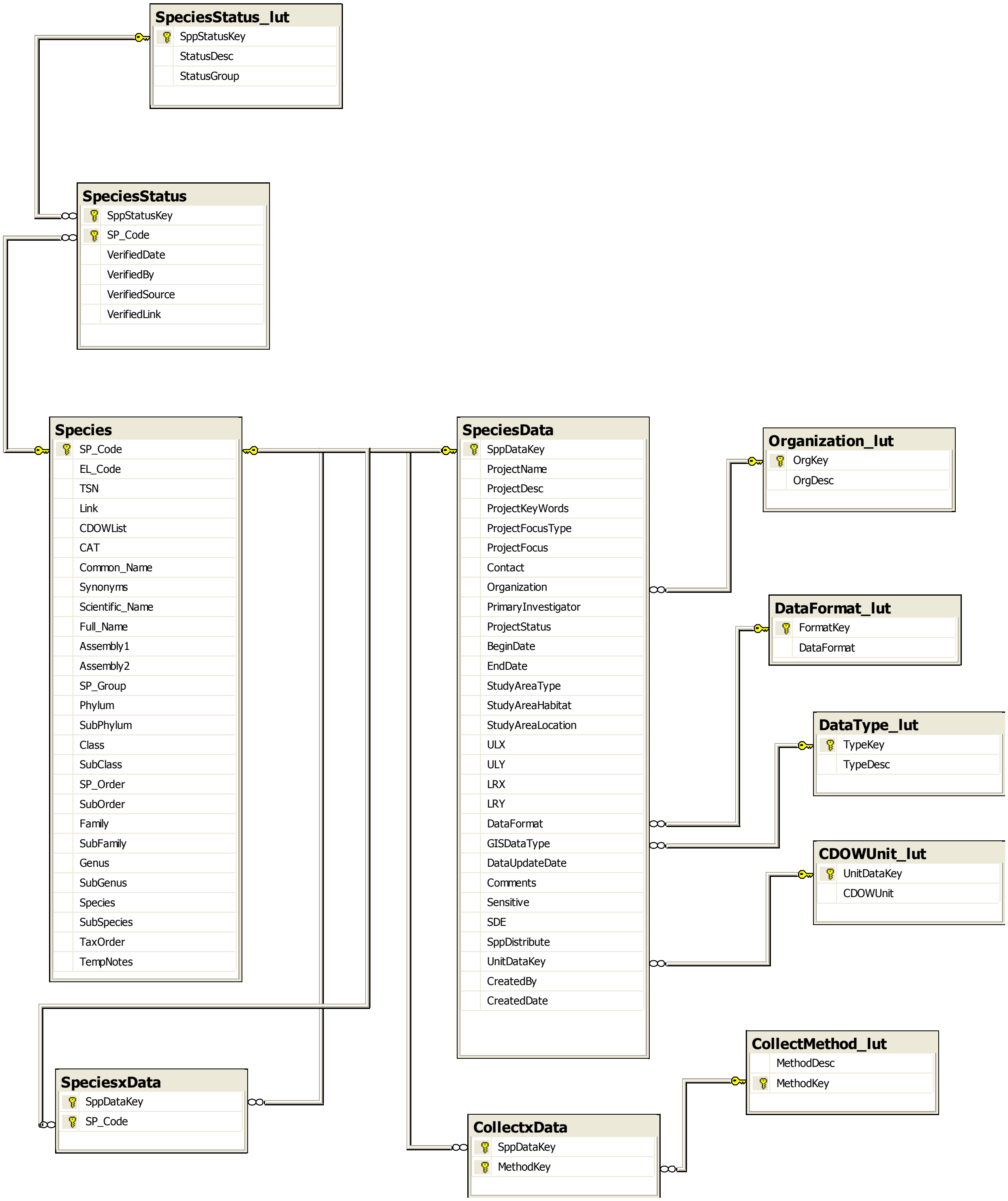
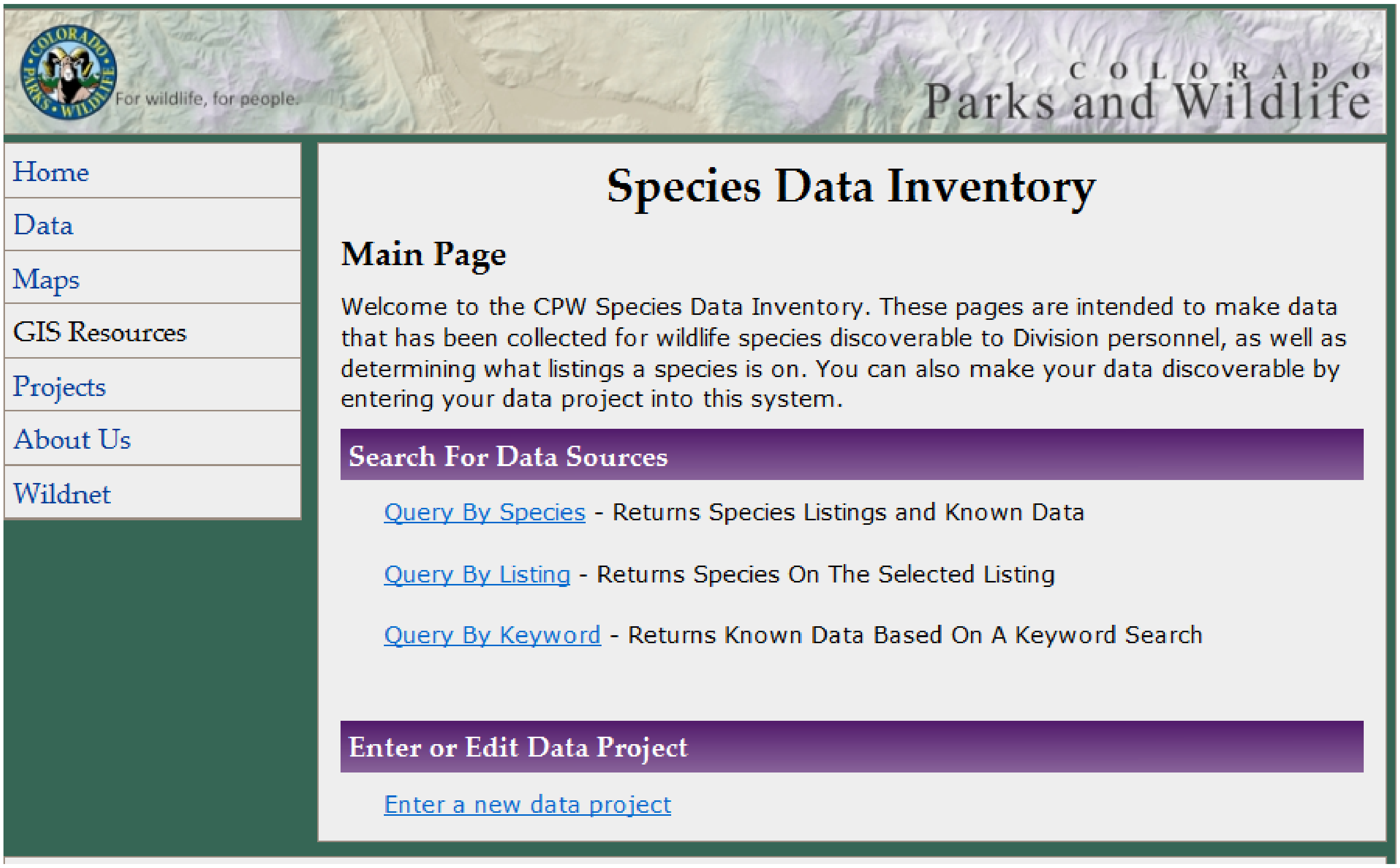


**Figure 7.  Species Data Inventory Database Application Main Page.**



# Species Data Inventory

## Main Page

Welcome to the CPW Species Data Inventory. These pages are intended to make data that has been collected for wildlife species discoverable to Division personnel, as well as determining what listings a species is on. You can also make your data discoverable by entering your data project into this system.

### Search For Data Sources

Query By Species - Returns Species Listings and Known Data

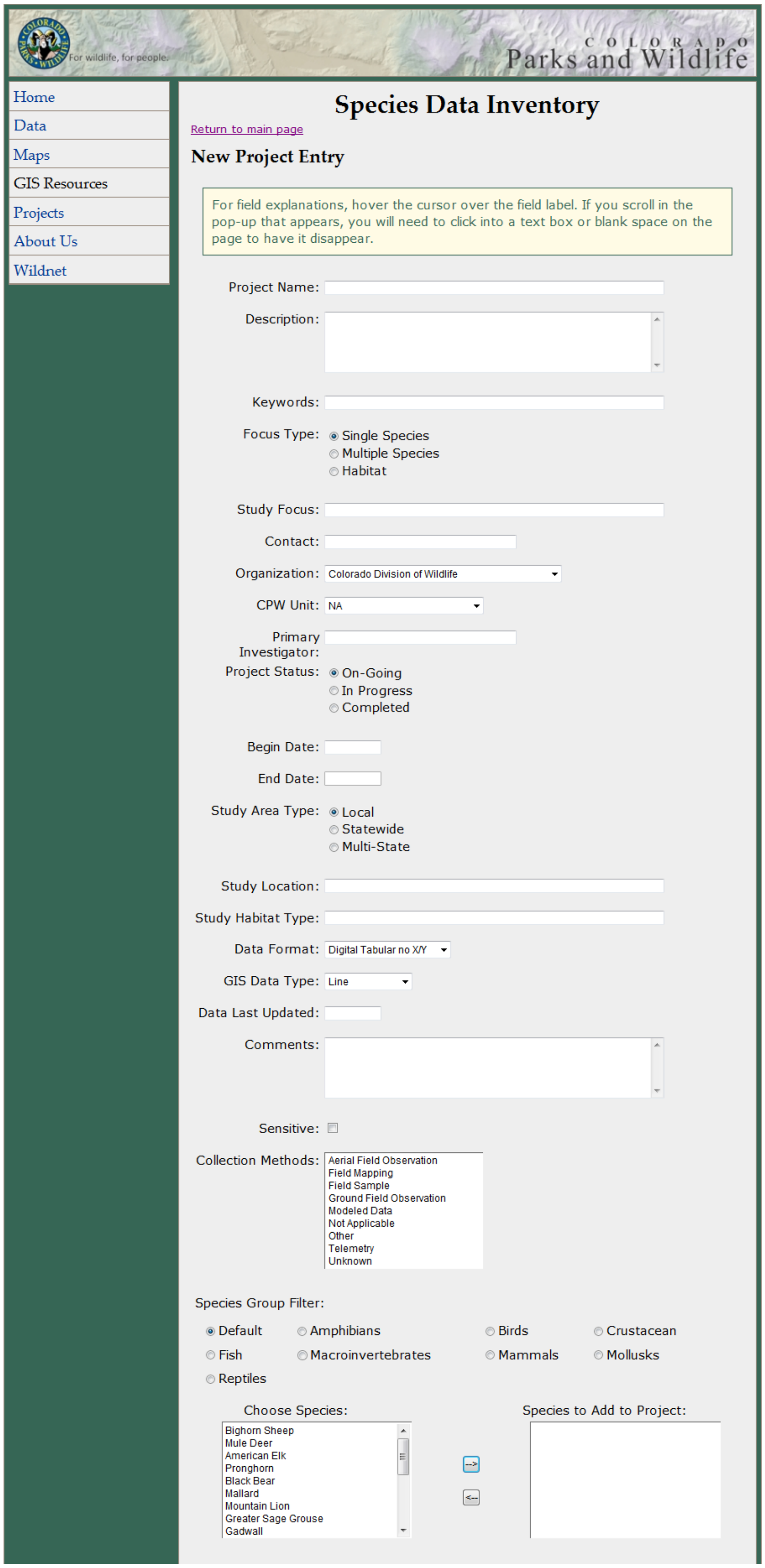
Query By Listing - Returns Species On The Selected Listing

Query By Keyword - Returns Known Data Based On A Keyword Search

### Enter or Edit Data Project

Enter a new data project

Figure 8.  Species Data Inventory Database Application Data Entry Page.



Species Data Inventory

Return to main page

New Project Entry

For field explanations, hover the cursor over the field label. If you scroll in the pop-up that appears, you will need to click into a text box or blank space on the page to have it disappear.

Project Name:

Description:

Keywords:

Focus Type: ● Single Species ○ Multiple Species ○ Habitat

Study Focus:

Contact:

Organization: Colorado Division of Wildlife

CPW Unit: NA

Primary Investigator:

Project Status: ● On-Going ○ In Progress ○ Completed

Begin Date:

End Date:

Study Area Type: ● Local ○ Statewide ○ Multi-State

Study Location:

Study Habitat Type:

Data Format: Digital Tabular no X/Y

GIS Data Type: Line

Data Last Updated:

Comments:

Sensitive: □

Collection Methods: Aerial Field Observation, Field Mapping, Field Sample, Ground Field Observation, Modeled Data, Not Applicable, Other, Telemetry, Unknown

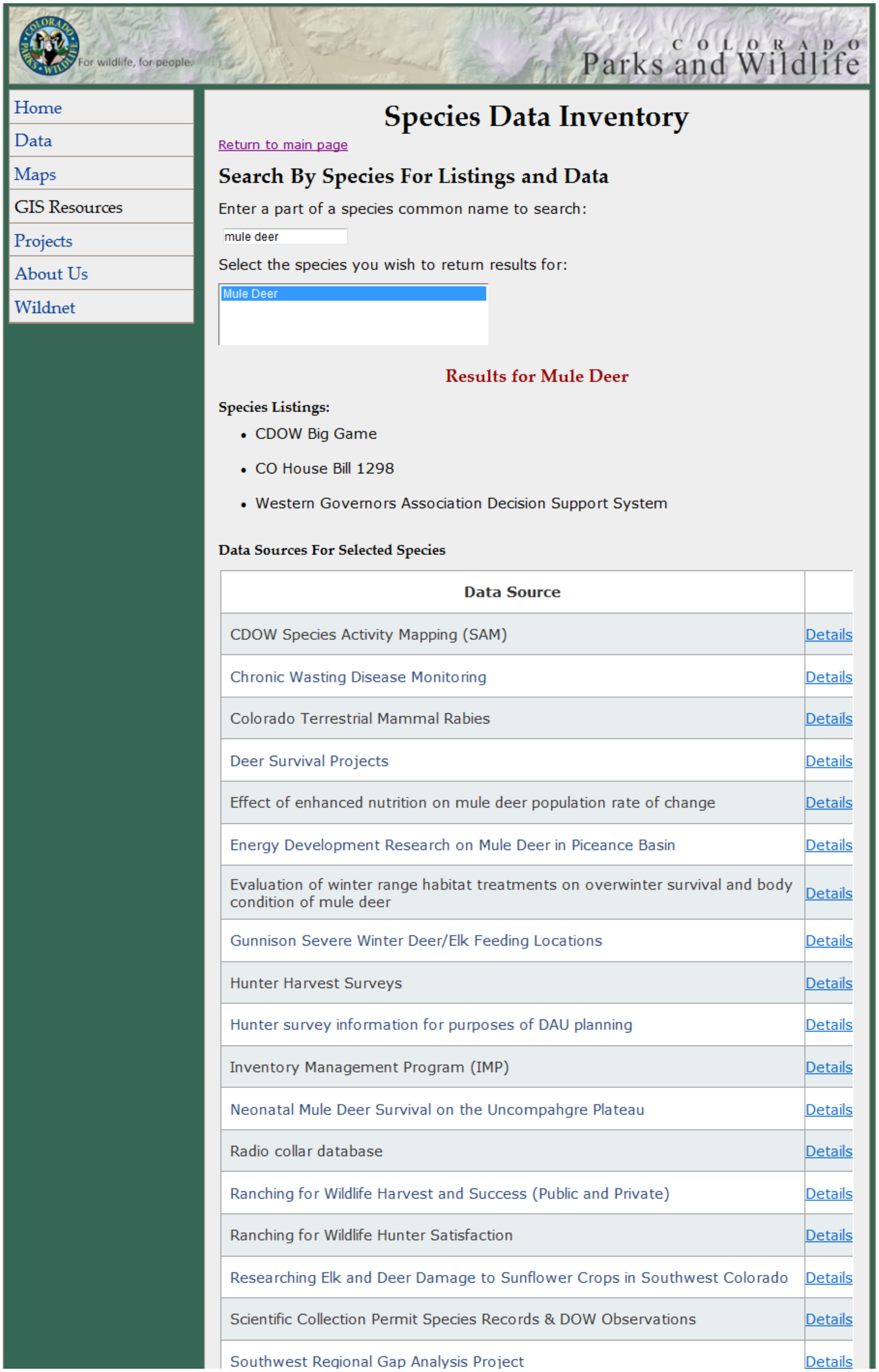
Species Group Filter:

● Default ○ Amphibians ○ Birds ○ Crustacean
○ Fish ○ Macroinvertebrates ○ Mammals ○ Mollusks
○ Reptiles

Choose Species: Bighorn Sheep, Mule Deer, American Elk, Pronghorn, Black Bear, Mallard, Mountain Lion, Greater Sage Grouse, Gadwall

Species to Add to Project:

**Figure 9. Species Data Inventory Database Application Species Search Page.**



## Species Data Inventory

Return to main page

### Search By Species For Listings and Data

Enter a part of a species common name to search:

mule deer

Select the species you wish to return results for:

Mule Deer

**Results for Mule Deer**

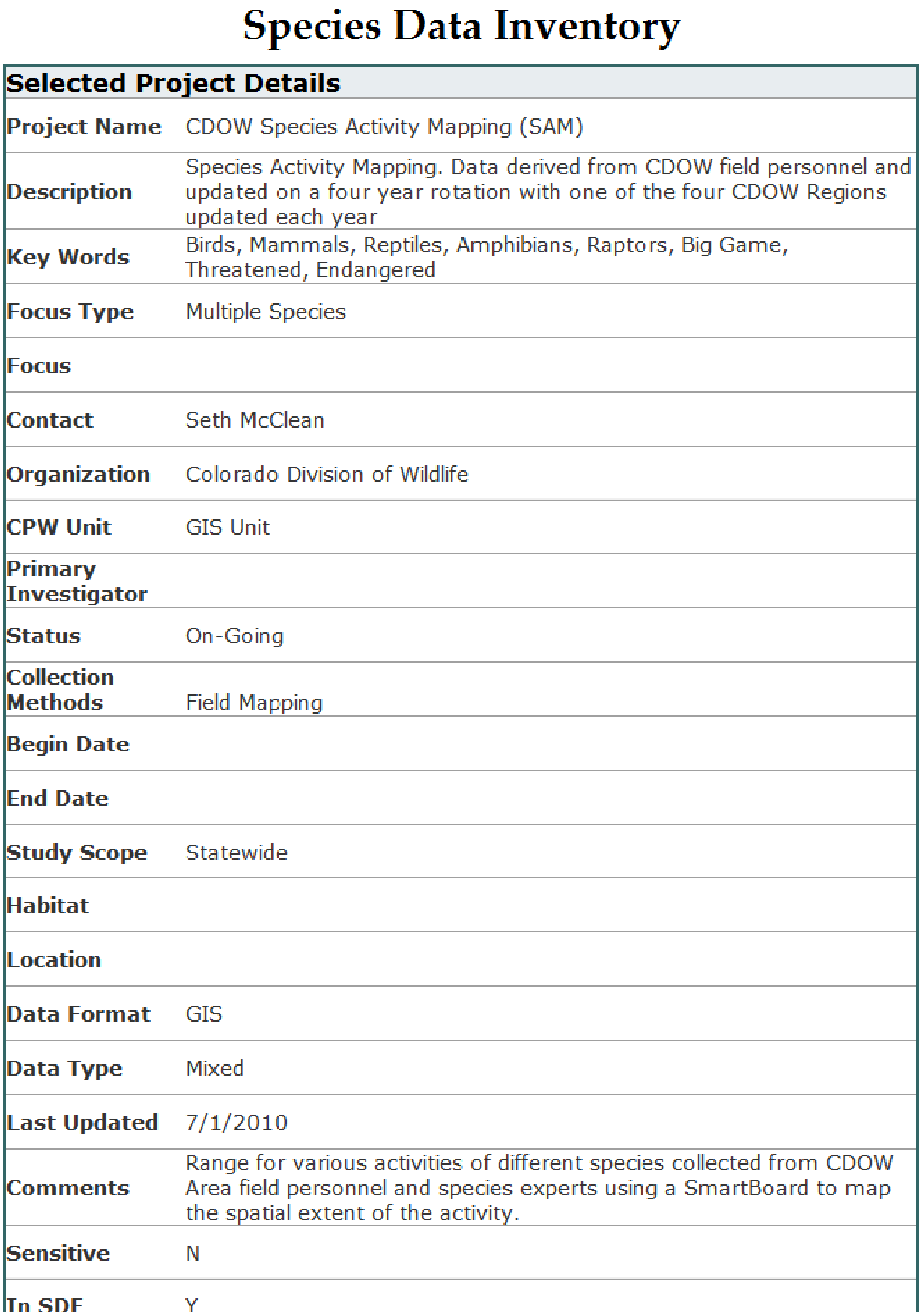
**Species Listings:**

- CDOW Big Game
- CO House Bill 1298
- Western Governors Association Decision Support System

**Data Sources For Selected Species**

| Data Source | |
|---|---|
| CDOW Species Activity Mapping (SAM) | Details |
| Chronic Wasting Disease Monitoring | Details |
| Colorado Terrestrial Mammal Rabies | Details |
| Deer Survival Projects | Details |
| Effect of enhanced nutrition on mule deer population rate of change | Details |
| Energy Development Research on Mule Deer in Piceance Basin | Details |
| Evaluation of winter range habitat treatments on overwinter survival and body condition of mule deer | Details |
| Gunnison Severe Winter Deer/Elk Feeding Locations | Details |
| Hunter Harvest Surveys | Details |
| Hunter survey information for purposes of DAU planning | Details |
| Inventory Management Program (IMP) | Details |
| Neonatal Mule Deer Survival on the Uncompahgre Plateau | Details |
| Radio collar database | Details |
| Ranching for Wildlife Harvest and Success (Public and Private) | Details |
| Ranching for Wildlife Hunter Satisfaction | Details |
| Researching Elk and Deer Damage to Sunflower Crops in Southwest Colorado | Details |
| Scientific Collection Permit Species Records & DOW Observations | Details |
| Southwest Regional Gap Analysis Project | Details |

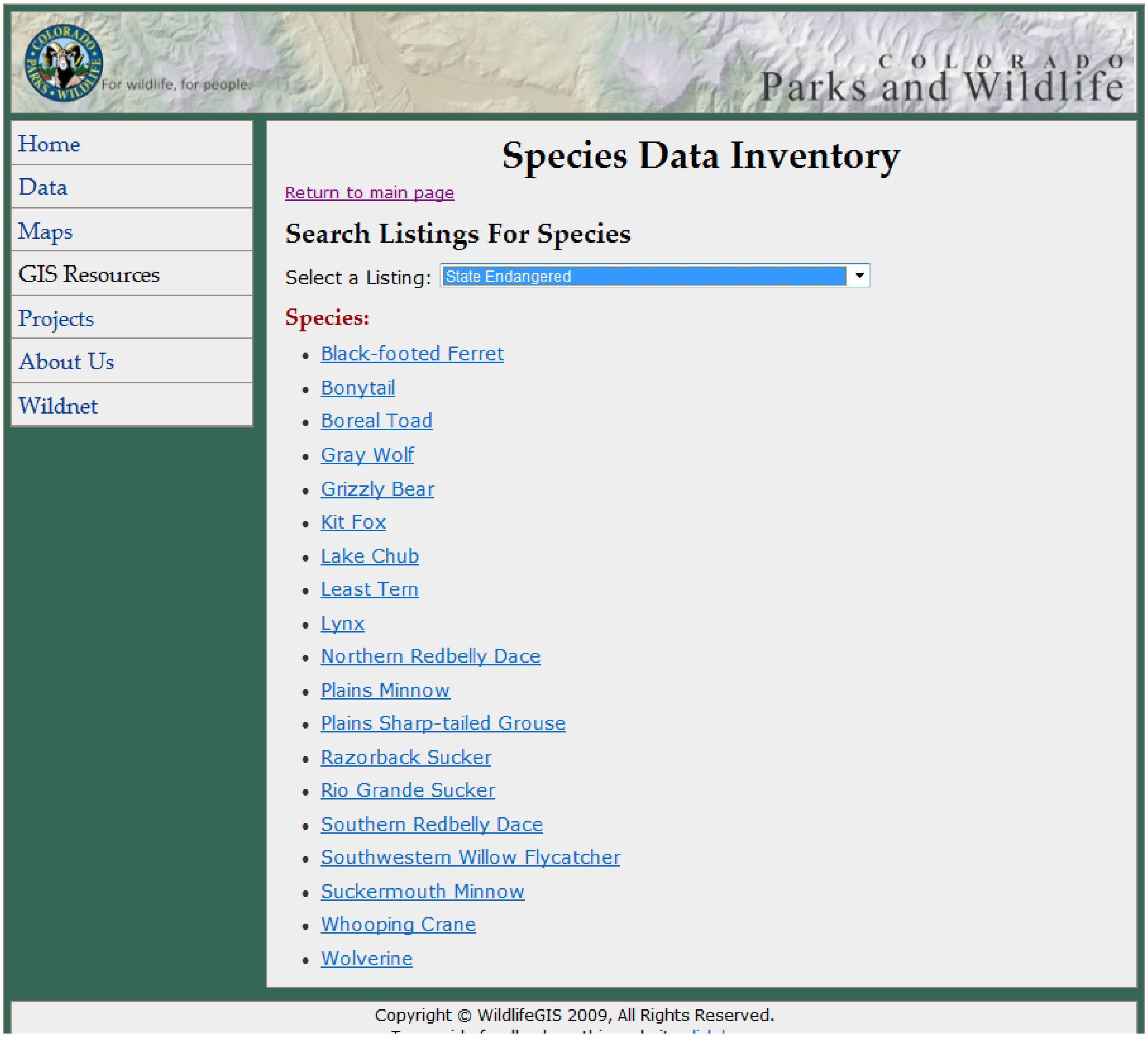
**Figure 10. Species Data Inventory Database Application Data Source Details.**



## Species Data Inventory

| Selected Project Details | |
|---|---|
| **Project Name** | CDOW Species Activity Mapping (SAM) |
| **Description** | Species Activity Mapping. Data derived from CDOW field personnel and updated on a four year rotation with one of the four CDOW Regions updated each year |
| **Key Words** | Birds, Mammals, Reptiles, Amphibians, Raptors, Big Game, Threatened, Endangered |
| **Focus Type** | Multiple Species |
| **Focus** | |
| **Contact** | Seth McClean |
| **Organization** | Colorado Division of Wildlife |
| **CPW Unit** | GIS Unit |
| **Primary Investigator** | |
| **Status** | On-Going |
| **Collection Methods** | Field Mapping |
| **Begin Date** | |
| **End Date** | |
| **Study Scope** | Statewide |
| **Habitat** | |
| **Location** | |
| **Data Format** | GIS |
| **Data Type** | Mixed |
| **Last Updated** | 7/1/2010 |
| **Comments** | Range for various activities of different species collected from CDOW Area field personnel and species experts using a SmartBoard to map the spatial extent of the activity. |
| **Sensitive** | N |
| **In SDF** | Y |

**Figure 11. Species Data Inventory Database Application Species Listing Search.**



## Species Data Inventory

### Search Listings For Species

Select a Listing: State Endangered

**Species:**

- Black-footed Ferret
- Bonytail
- Boreal Toad
- Gray Wolf
- Grizzly Bear
- Kit Fox
- Lake Chub
- Least Tern
- Lynx
- Northern Redbelly Dace
- Plains Minnow
- Plains Sharp-tailed Grouse
- Razorback Sucker
- Rio Grande Sucker
- Southern Redbelly Dace
- Southwestern Willow Flycatcher
- Suckermouth Minnow
- Whooping Crane
- Wolverine

Copyright © WildlifeGIS 2009, All Rights Reserved.

Having this type of database and application provides more value to data collection efforts that have occurred in the past or are on-going in the agency by making the existence of these data more well known and easily accessible to staff. Historic or localized studies can be used to determine where species have occurred and guide locations for new studies. In addition, knowing that there are already established methodologies and data structures for a certain species can greatly assist someone looking to conduct similar research in another location. Basing new research on existing methodologies will help the agency be more efficient since the new researcher will not have to recreate the wheel by developing their own methodologies. Having knowledge of existing data structures and basing new studies on them also results in adding value to the data through having the ability to combine data and perform analysis over larger areas of the landscape and being able to appropriately compare data collected at different time scales.

# CVEN 5800 – WEB DEPLOYMENT OF GIS / SAM WEB GIS APPLICATION

## INTRODUCTION

Colorado Parks and Wildlife (CPW) collects and manages Geographic Information System (GIS) data on threatened and endangered, and economically important wildlife species. Depending on the species, anywhere from 1 to 12 critical life cycle activities are collected and updated each year for each species. These data are managed in an ESRI ArcSDE Geodatabase comprised of approximately 182 feature classes organized in 37 feature datasets.

These data are updated each year in one of four administrative regions in the state of Colorado through a process named Species Activity Mapping (SAM). SAM entails members of the CPW GIS Unit traveling to each of the CPW Area offices in a Region and working with each game warden and biologist to perform updates on these data by editing the feature classes.

Traditionally, at the completion of these edits the updates were printed into map books and distributed to game warden and biologist field personnel. It has been estimated that approximately 1,500 paper pages were printed each year with a significant effort required to print and compile these books. Often temporary employees would be hired to complete this printing task, but they could have been used accomplishing other tasks. In addition, GIS Unit staff time was needed to train the temporary staff in completing the task. This effort also consumed a large amount of printing resources including paper, toner, and wear and tear on printers. If data were updated anytime in between the four year cycle of updating a region, the paper maps became obsolete. However, these data are extremely important for land use comments ranging from county zoning and development to state legislated identification of wildlife resources at proposed oil and gas development locations.

As part of my term project for CVEN 5800 I developed the idea of replacing these printed maps with a web mapping application that would fulfill the same benefits of the printed maps while providing additional benefits such as allowing the user to navigate the maps at different scales, viewing different backgrounds, and printing site specific maps as needed. Other benefits of this shift in data access from printed maps to a web mapping application would be freeing permanent and temporary staff time for completing other tasks, and allowing the flexibility to more dynamically update the maps and instantly provide these changes to the end users without waiting to reprint the maps at the end of the next four year cycle.

The objectives of this project were to mashup operational species data GIS layers with various basemap data layers provided by third parties within an ESRI Flex API web mapping application having a similar layout to the existing CPW GIS Unit intranet pages. The application would provide tools enabling the user to turn on specific species layers and zoom to a District administrative boundary of interest. It would also provide functions for panning and zooming in the map, returning to the extent of the state of Colorado, turning on a legend for the layers being viewed, and providing documentation of the definitions for the layers and maps of the data update schedule. One of the most important criteria of the application was that it be easy to use. The purpose of this application was not to recreate ArcGIS in a web format, but to provide a focused, easy to use application that enabled field personnel who do not regularly use computing resources the ability to easily access the species mapping information that they regularly need without having to provide paper maps.

This project also demonstrates another stage in the evolution of the administration and accessibility of an enterprise GIS. At CPW the evolution has consisted of moving local file server data into the enterprise ArcSDE database system with access provided to a small number of core GIS staff. The next stage was expanding the use of the system to all GIS users in the agency including biologists, engineers, and other staff with experience and interest in using GIS software to view data, create maps and perform analysis. The next stage is making these enterprise data accessible to managers and other agency personnel who do not have the interest or time to gain the expertise in using GIS software to access data, but who may have a need to see and use the data in specific ways. To serve these needs, the information stored in the enterprise GIS databases needs to be published as web services so that it can be consumed in web applications that are easy to use and provide views to the specific data that these individuals need to be able to access.

## METHODS

When this project began, it was initially developed using ArcGIS Server and ArcSDE version 9.3.1, but it has recently been updated to version 10. First, ArcGIS mxd map documents were created for each of the 37 species whose data is collected during the SAM sessions using Layer files containing standard symbology used for each of the activity layers. These map document files were then saved as ArcGIS optimized msd map service document files which were then published as an ArcServer map service. Each of these map web services was used to create and reference tiled map caches to increase the performance of these services. The performance increase is achieved through serving the data as pre-generated image files with resolutions appropriate at various scales as opposed to streaming the raw vector files over the network. This was an important part of the project because many of the field staff referencing these data will be doing so from remote locations with limited network bandwidths. When version 10 of ArcServer was installed, these tiled map caches were converted to the new Compact format which maximized the disk usage these files took up, as well as reducing the number of files in the cache so that these files could be more efficiently copied to other servers if necessary. In addition to the species data, important agency administrative boundaries were also published as tiled map cache services so they could also be displayed in this application.

Once these web services were created, Flash Builder 4 was used with the ESRI Flex API version 2.1 to develop an application that could be formatted independently of browser compatibility as long as the users had the Flash client installed, which is the case for all CPW staff. This application was developed from the ground up as opposed to using the ESRI Flex Viewer template in order to gain a more in depth knowledge of Flex development, and to customize the application to better match the layout of the existing intranet ASP.NET pages. This application was developed using MXML for formatting and Actionscript for the scripting functions (Appendix D). Once the layout was established for the application, dropdowns were coded for the user to select a species and administrative district they were interested in viewing. Event handlers were created to turn on the web service for the chosen species and zoom to the administrative district chosen with a different symbology displayed to highlight the district boundary with a different graphic. Another button was created that when clicked zooms the user to the extent of the state of Colorado, which was also set as the initial extent. A button bar was added allowing the user to select a choice of basemaps to display including Streets, U.S. Topo, and Imagery backgrounds. Since the project was developed for CVEN 5800 several enhancements to the application have been made including buttons for turning on a legend for the visible layers, printing, and adding additional data including Game Management Units and property ownership, which are important for field personnel using this application to be able to view. A slider bar for adjusting layer transparency was also added at the request of the end users.

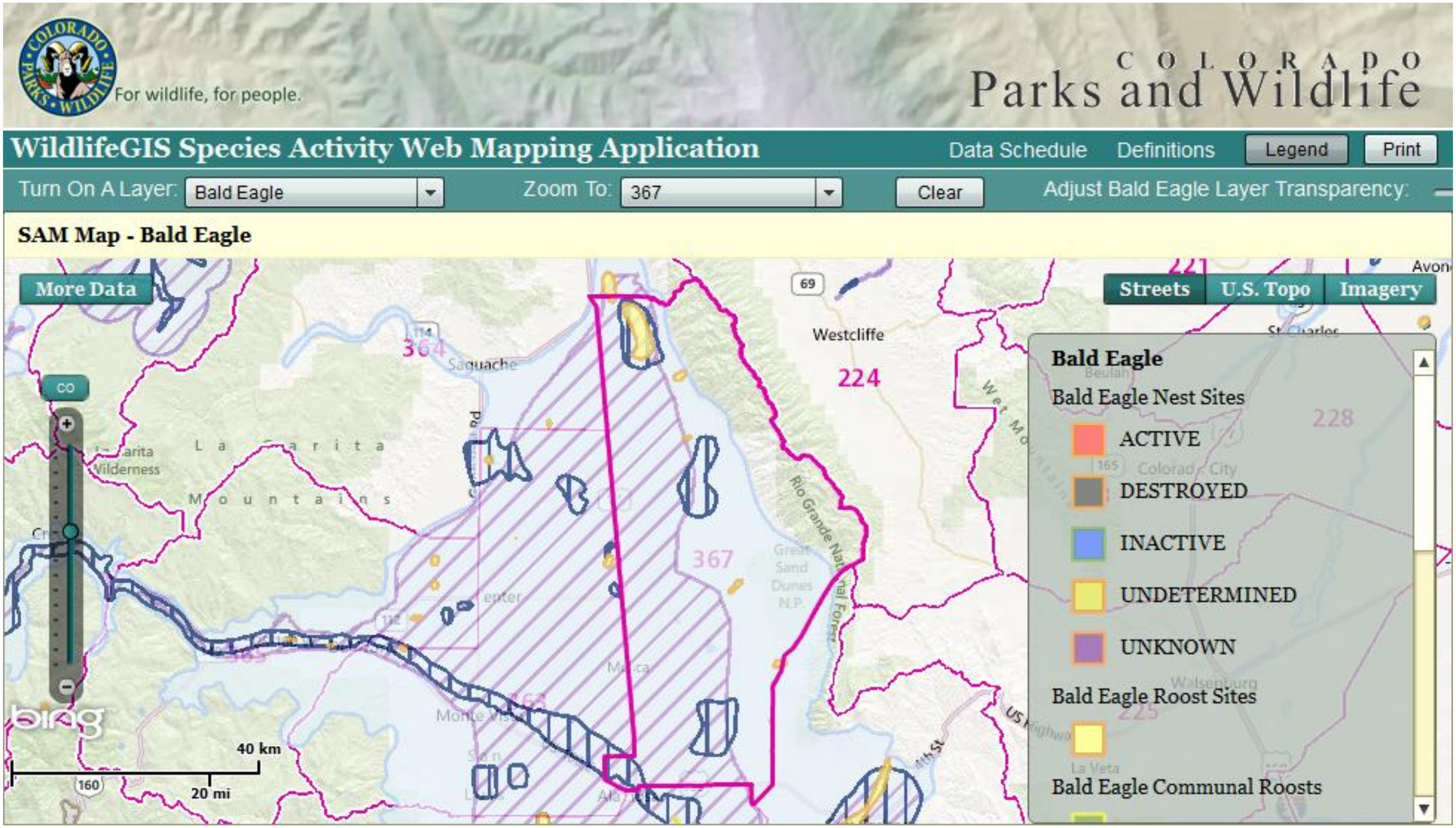
ArcSDE versioning reconcile and post operations are monitored for updates to species data at quarterly intervals and tiled map caches are updated for species which have had data updated during the previous three months. Field personnel are then able to see these changes reflected in the application within three months of the changes being made in the enterprise database versus traditionally having to wait four years when the paper maps were reprinted. If needed earlier due to business needs, these tiled map caches can be updated even sooner than the established quarterly basis.

## RESULTS

The result of completing this project is a web application (Figure 12) that is able to provide field staff the ability to view species activity data that had previously been provided to them in four year intervals with printed paper map books. This application has eliminated the need to print these map books which has resulted in a tremendous savings of staff time, paper, toner, and printer wear and tear. This application has also given the GIS Unit the ability to more quickly distribute and make accessible data that has become more dynamic than in the past with updates occurring more regularly outside the normal four year cycle.

While this application has limited functionality, its target audience is not regular GIS users (or even regular computer users). By keeping the application focused with functions that are available and described with text rather than accessed through icons, it has proven to be easily used by a wide number of users who have a need to view these data.

**Figure 12. SAM Web Mapping Application.**



WildlifeGIS Species Activity Web Mapping Application

Turn On A Layer: Bald Eagle

Zoom To: 367

**SAM Map - Bald Eagle**

**Bald Eagle**

Bald Eagle Nest Sites

🟥 ACTIVE

⬛ DESTROYED

🟦 INACTIVE

🟨 UNDETERMINED

🟪 UNKNOWN

Bald Eagle Roost Sites

🟨

Bald Eagle Communal Roosts

## CVEN 5385 – GIS RELATIONAL DATABASES / SAM FIELDS & TRAVEL DATABASE

### INTRODUCTION

As described in the previous section of this report, CVEN 5800 – Web Deployment of GIS / SAM Web GIS Application, Colorado Parks and Wildlife conducts what is called "Species Activity Mapping" (SAM) to update its GIS data. SAM consists of an enterprise database containing data on 37 species with Feature Classes for between 1 to 12 defined activities for each of these species. Data is updated by members of the GIS Unit traveling to each CPW Area administrative unit in one of four administrative Regions in the state each year. The GIS Unit staff members then work with game wardens and biologists to add or update data as necessary to these roughly 182 feature classes.

Two parts of projects completed for CVEN 5385 have recently either been applied or are being worked on to improve how these enterprise SAM data are collected. First, as part of my midterm project in researching ESRI Data Models, I chose NatureServe's Biodiversity Model. While this model was somewhat outdated, I was able to develop ideas from this data model that could be added to CPW's existing data structures to enhance them to make the data collected more meaningful to end users and improve the credibility of analysis.

Second, this process of collecting these data across the state requires travel on the part of GIS Unit staff with the same cities being visited every four years but possibly with different staff. As each new update period starts, questions around the logistics of travel, such as where to stay, where to eat, and what to do continually arise. As part of my final project for developing a spatial database I came up with the idea of a travel locations spatial database to capture hotels, restaurants, and points of interest for staff to capture new locations and provide updated reviews.

### METHODS

While reviewing the NatureServe Biodiversity geodatabase model I came across some of their documentation from July 2, 2001. In this documentation, there was a Simple Diagram of Biodiversity and Conservation Features. In this diagram several feature diagrams were identified as Historic Distribution, Potential Distribution, Predicted Distribution, and Known Distribution. While I conducted extensive research to find where these features were described, both in the metadata of the downloaded geodatabase model, as well as through internet searches, I could not find how NatureServe defined these different features. In mapping species activity ranges, one of the most important aspects is clearly defining the activity. Without clearly defining the intent of the area to be mapped, misinterpretations can easily occur resulting in unacceptable inconsistencies in the data. It would have been ideal to determine how NatureServe defined these distribution types to match already accepted definitions established in the conservation community and allow for future alignment with the data collected by CPW with others using these same distribution types. However, having no luck finding definitions for these distribution types left me having to develop my own definitions to at least maintain consistency between the different staff mapping these data at CPW. The definitions I developed were:

- **Known Distribution** – "Species is known to be using the area for the defined activity based on current observations."
- **Predicted Distribution** – "Species is believed to be using the area for the defined activity but no direct observations have been made."
- **Potential Distribution** – "Habitat is sufficient for the species to use the area for the defined activity but is not currently being used by the species."
- **Historic Distribution** – "Species is known to have used the area in the past for the defined activity but is currently extirpated from the area."

A text Coded Value Domain was added to the enterprise ArcSDE database containing the SAM feature classes and the values Known, Predicted, Potential, and Historic were added to this Domain. The Domain would be associated with the field Distribution Type so that editing the attributes for this field will result in a dropdown of these values. Having the dropdown available makes entry easier and maintains a consistency in the data allowing it to be easily filtered as necessary when viewing or performing analysis on these data. This field would also have "Known" designated as the default value so that no entry would be necessary when adding a new feature when it was a Known Distribution Type, which is the overwhelming majority of these data.

While making this database schema change in the SAM database, it was also determined that the data could also be made more useful by adding auditing fields that captured who made changes to the spatial and attribute values of the features and when those changes were made. The main challenge of making these field changes was the tedious aspect of adding these 5 fields to each of the nearly 182 feature classes in the SAM database, associating the Domain, and designating the Default value for the Distribution Type field. This was accomplished through the use of Python scripting and is described in the next section, CVEN 5386 – GIS Lab / Python Automation & Add-In Auditing.

To develop the Travel Locations spatial database, similar methods as described in the section ISMG 6480 – Data Warehousing and Administration / Species Data Inventory were employed. First, several interested GIS Unit staff were gathered together to discuss the type of data that would be most beneficial in providing future travelers information to help them schedule their trip. Next, this information was then initially outlined in a Conceptual Database Design using simple flowchart tools (Figure 13). This Conceptual Database Design provided a general overview of the data that would be collected in the database. Once the stakeholders reviewed, made changes to, and accepted the Conceptual Design, the ideas were further structured into a Logical Database Design using Entity Relationship structures in the ER Assistant software tool (Figure 14). The Logical Database Design organized the ideas from the Conceptual Design into actual table entities with attributes and relationships between tables. Again, the Logical Database Design went through the iterative process of having the stakeholders review, make changes to, and finally accepting the design. Once the Logical Design was accepted, a UML Database Design was constructed using ArcGIS Diagrammer (Figure 15). Using this software tool to model the UML Database Design had the advantage of best being able to represent spatial database structures unique to the ESRI Geodatabase that the database would ultimately be developed in, including Spatial Features, Domains, and Subtypes.

-

Figure 13. Travel Location Spatial Database Conceptual Design.

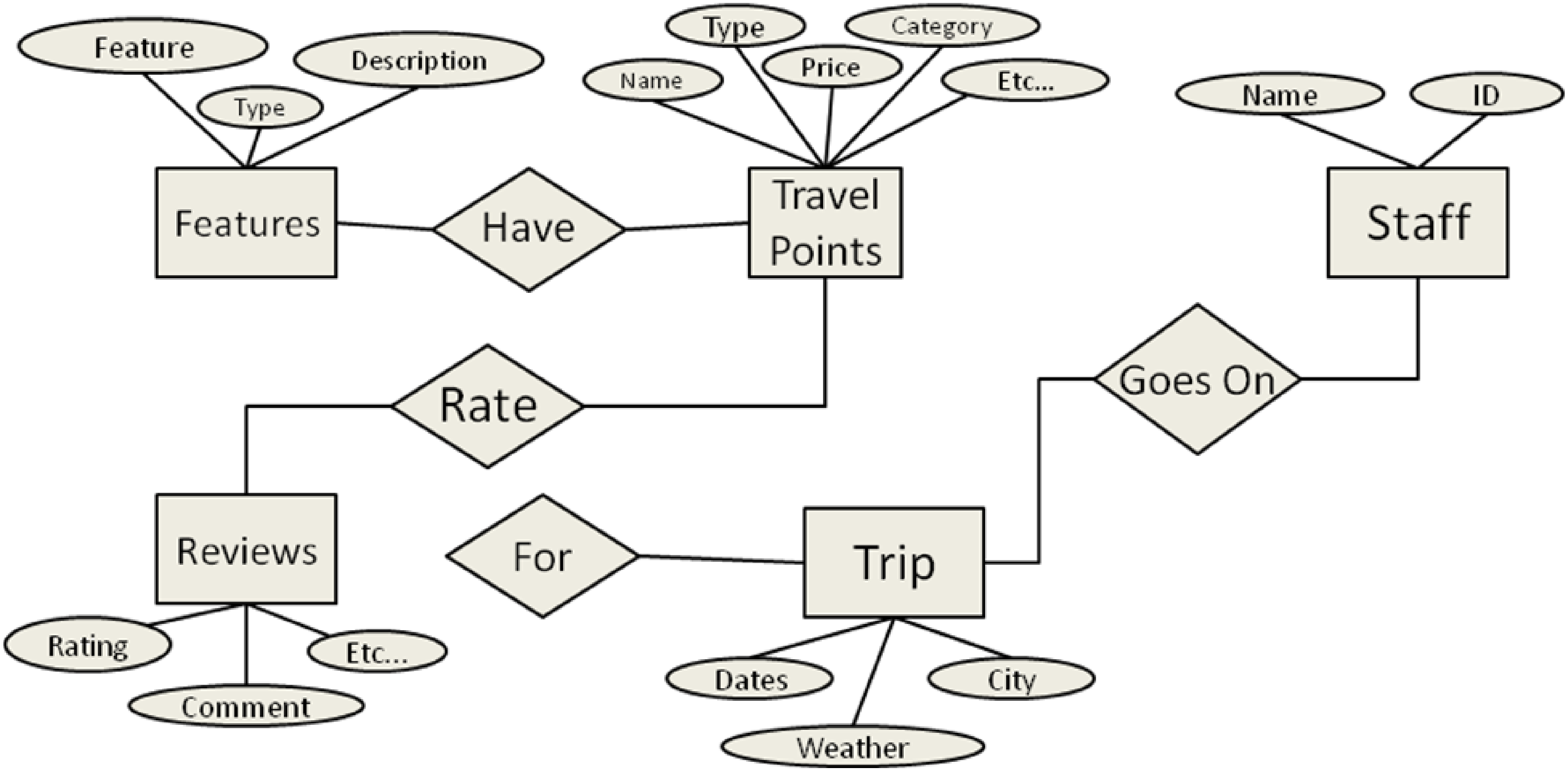


**Figure 14. Travel Location Spatial Database Logical Design.**

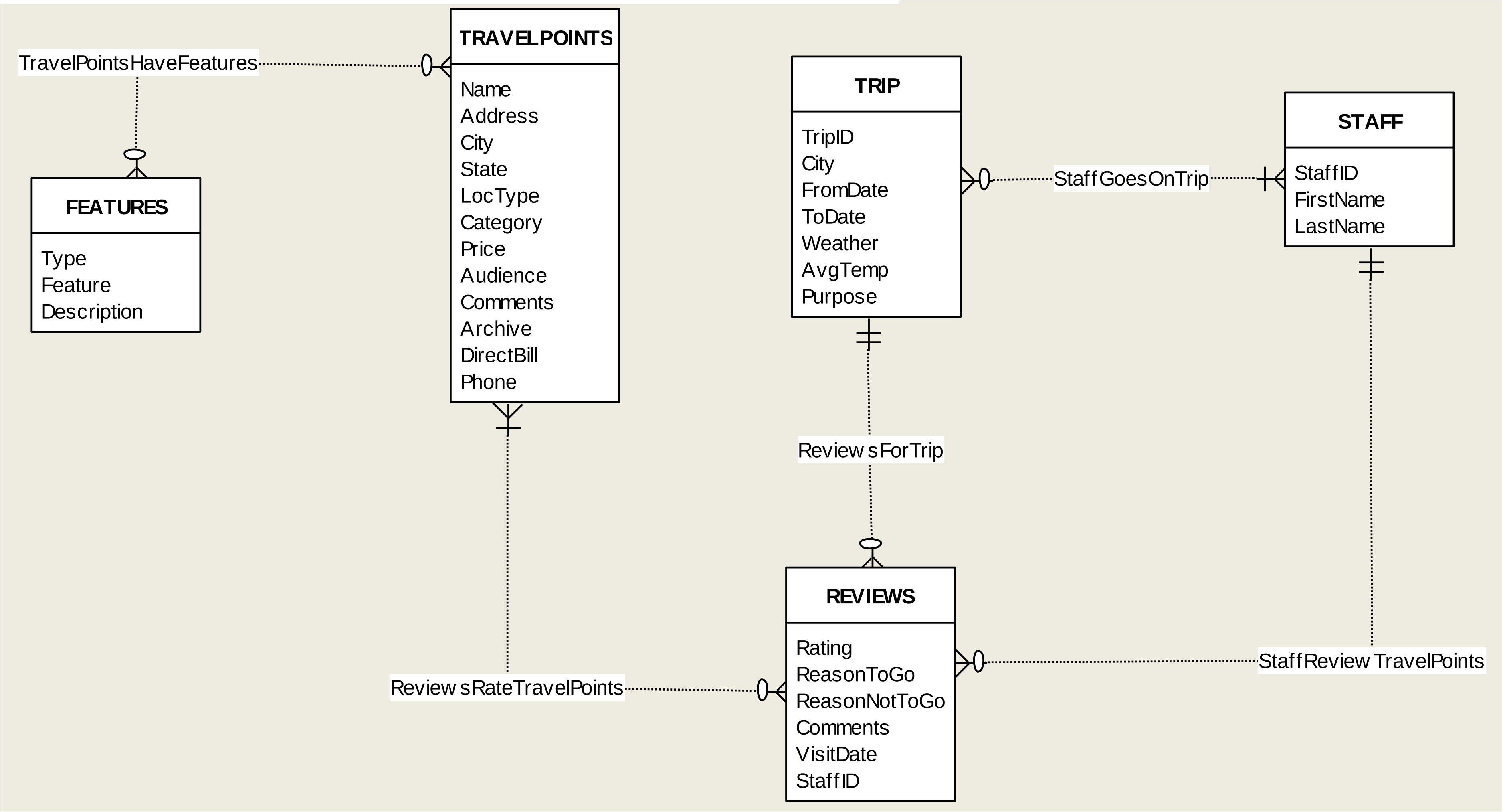
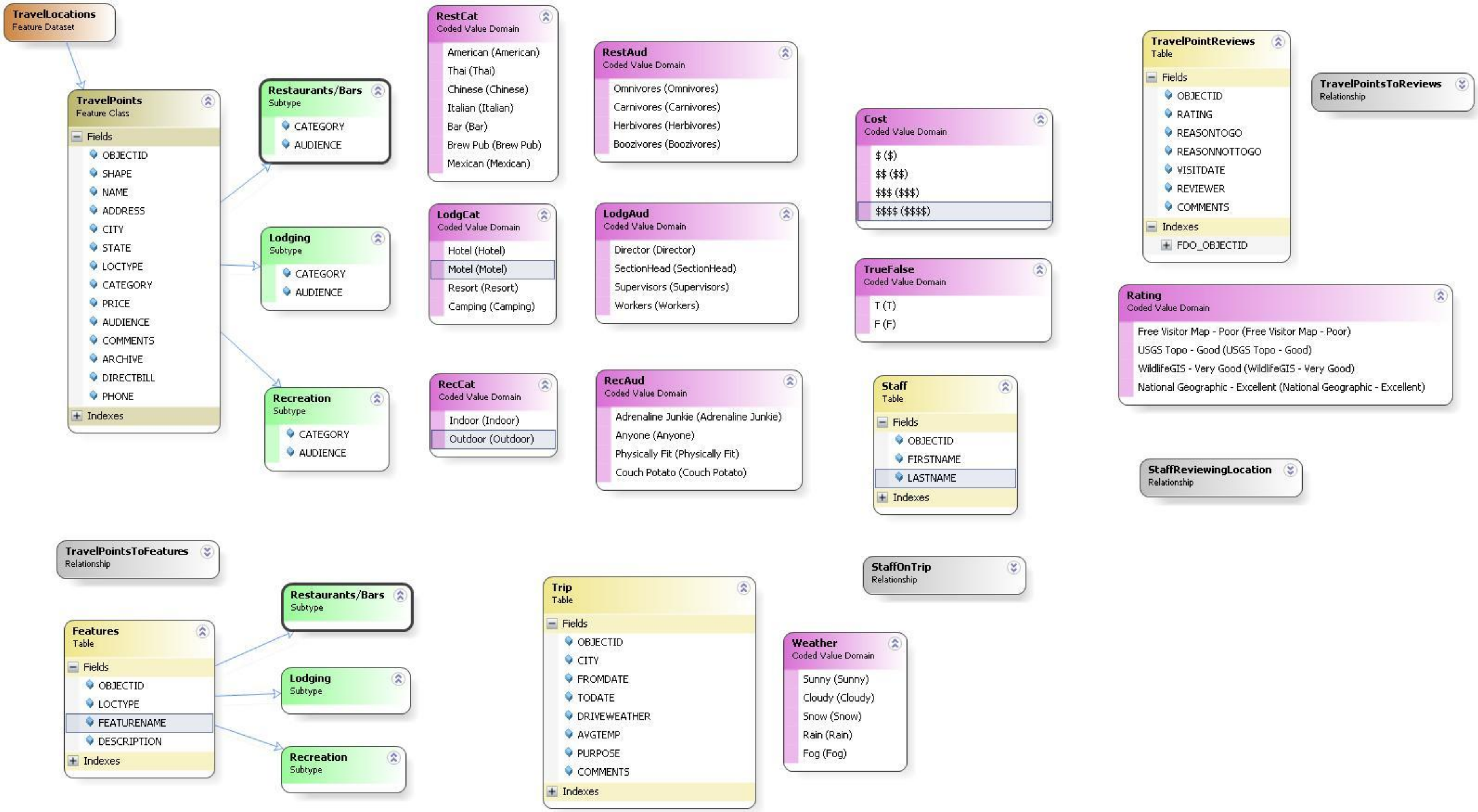


Figure 15.  Travel Location Spatial Database UML Design.



## RESULTS

Adding the Distribution Type field represents a relatively simple change to the CPW enterprise GIS, however, I believe it will provide a significant increase in value to these data. Until this field was added there was often no way to distinguish subtle differences in the use of the defined species activities and all captured range features were assumed to be used in the same way. During the SAM edit sessions I would often see the game wardens and biologists thinking out loud and debating themselves about whether to include an area or not in a specific range activity. These self debates most often revolved around whether to include areas that have been used in the past by the species but have not been used in recent years (Historic), areas that have not been witnessed to be used by the species but are assumed to be used based on other observed areas of use (Predicted), or areas that have the requisite attributes to support the species for the designated range activity but are not currently being used by the species (Potential). When the biologist or game warden decides to keep these areas in the data features without having the ability to distinguish these different types of use, the data loses some of its credibility. Having this field available for them to distinguish these different distribution types will allow end users to filter the data based on the appropriate need and increase the credibility of data views and analysis performed on these data.

Similarly, data captured during SAM edit sessions has focused nearly singularly on spatial features with close to no attribute data being captured. This limitation is often the result of time constraints and the tight schedules of these editing sessions. In addition, the capture of descriptive attributes for the data has not been a stated priority and as a result is neglected by editors trying to complete the edits as quickly as possible. However, often during reviews of the data in trying to determine the validity or timeliness of a feature, the first questions asked are when the data were added and who added them. In order to capture these data, four new fields were added to the data schema for each of the Feature Classes. Fields were added to capture who and when spatial features and attribute data were added or edited. To address the issue that editors were not likely to complete these fields, an ESRI Add-In was developed for automatically capturing the system user and time and populating these fields from them as features were added or edited. Details on this ESRI Add-In are described in the next section, CVEN 5386 – GIS Lab / Python Automation & Add-In Auditing.

These fields were added as part of the database schema in the spring of 2012 in time to be used during the 2012 SAM edit sessions. They have been successfully used during the first administrative Area of the year. There were several instances where a game warden was conflicted about whether a species range distribution should be kept in the data or eliminated because the features have not been used by the species for some time. The game wardens were going to leave the data in because there were extenuating circumstances that were responsible for the species no longer using the area that could be reversed and they did not want to remove the features as part of the range which would indicate that the species does not use the area. Having the Distribution Type field allowed these features to be designated as Historic. In the past if these data had been left in, they would have been assumed to be the same as every other part of the captured range as an observed known use of the area. Having the Historic designation tagged to these areas will now alert end users that this has been an important part of the species range but that the species is not currently using the area. In addition, the ESRI Add-In was found to successfully capture the system time and user that made edits to spatial features and their attributes captured at the feature level in their attribute fields. These simple database schema changes will add considerable value to the data by distinguishing important differences in the species distribution types of their range activities and by being able to determine when features were added or edited and who to contact if questions arise about the features.

The Travel Location spatial database was developed as part of the final project for CVEN 5385. However, during testing of the database it was found that while the database was developed with strict adherence to relational database principles that were learned during this class, these structures are difficult to use within the ArcGIS software. In particular, the many-to-many table relationships are extremely confusing to develop the data within the editing environment and not very apparent on how to access and use them to view the data. Editing and viewing these structures was confusing to even the most advanced ArcGIS users that these data were tested on and would be completely unusable to intermediate or novice users. In order to make this database functional in a real working environment, compromises to the database structure need to be made, including flattening the several many-to-many relationship structures and finding other structures to replace them. The novel aspect of this project has resulted in having a lower priority compared to other projects. However, as this year's SAM sessions have begun with new personnel it has become realized that this database would have considerable value at providing efficiencies for staff scheduling the logistics of their travel to the cities where editing occurs.  The current objective for this project is to complete improving the database structure and testing editing within an ArcGIS environment by the summer of 2012 in time for users to add the travel locations and their reviews for the cities visited during the 2012 SAM sessions. Further improvement for the database would be a web application for viewing and editing these data. This web application project would be a good case for learning the ESRI Javascript API and its editing environment which could serve as a template for other GIS web editing projects.

## CVEN 5386 – GIS LAB / PYTHON AUTOMATION & ADD-IN AUDITING

### INTRODUCTION

As described in the previous section of this report, CVEN 5800 – Web Deployment of GIS / SAM Web GIS Application, Colorado Parks and Wildlife conducts what is called "Species Activity Mapping" (SAM) to update its enterprise GIS species data. SAM consists of an enterprise database containing data on 37 species with Feature Classes for between 1 to 12 defined range activities for each of these species. Data is updated by members of the GIS Unit traveling to each CPW Area administrative unit in one of four administrative Regions in the state each year. The GIS Unit staff members then work with game wardens and biologists to add or update data as necessary to these roughly 182 feature classes.

The technical process behind SAM is comprised of checking out the data from the ArcSDE Geodatabase to Personal Geodatabases for each species, which are copied to two laptops. Check out of the data to local copies on the laptops is necessary due to either the lack of network availability or slow network speeds impacting the performance of drawing time for the data at the CPW Area offices. Edits are performed on these local copies of the data by projecting an ESRI ArcMap document to a registered SmartBoard and having the game wardens and/or biologists update the data directly on the SmartBoard. The game wardens and biologists work closely with a GIS Analyst who is responsible for setting up the data edit sessions and assisting the game wardens and biologists with the technical aspects of editing and navigating the map document. After completing the edits for an Area, the data from the Personal Geodatabases are checked back in to the ArcSDE Geodatabase. The check out/check in process is repeated 4-5 times annually.

The purpose of this section is to describe and document how this course developed skills for developing processes that were used to automate several steps of data management for CPW's SAM, as well as creating an automated audit trail for capturing when a feature was last updated and the GIS Analyst who oversaw the edit. Skills learned in CVEN 5386 using Python scripting language and ESRI Add-Ins using C# programming language will be used to automate these business processes. Specifically, the tasks performed for this project included:

1. Use the Python scripting language to automate the one time process of adding the 5 new fields described in the previous section, CVEN 5385 – GIS Relational Databases / SAM Fields & Travel Database, to each of the 182 feature classes for further defining features along with capturing a feature level audit trail. The new fields to be added include:

    a. DistributionType – captures information for further describing the type of species activity range type for a feature. This is a Text data type field of 20 characters defined by a Domain with the values "Known", "Potential", "Predicted", and "Historic" and the Default value of "Known".

    b. AttLastUpdateBy – captures the username of the last person to make an attribute edit. Text data type of 15 characters.

    c. AttLastUpdateDate – captures the date of the last edit to an attribute field. Date data type.

    d. ShpLastUpdateBy – capture the username of the last person to make a spatial edit to a feature. Text data type of 15 characters.

    e. ShpLastUpdateDate - captures the date of the last spatial edit of a feature. Date data type.

2. Use the Python scripting language to automate the on-going check out process of the data from the ArcSDE system to Personal Geodatabases for each species.

3. Use the ESRI Add-In Editor Extension to automate the process of capturing the username of the person logged into the computer making edits, as well as the current date, and populating the appropriate attribute or shape audit fields for the feature being edited.

4. Use the Python scripting language to automate the on-going check in process of the data from the checked out Personal Geodatabases back to the ArcSDE Geodatabase.

## METHODS

The first step necessary in creating an audit trail for capturing usernames and dates for when features are updated and capturing features distribution type includes adding the fields to the feature classes that will store these data. Adding these five fields to 182 feature classes by either entering the Properties of the feature classes or using the ArcToolbox tool for adding fields would be a time consuming process. While this is only a one-time process, it would be more efficiently performed by writing a Python script to iterate through the database and add the fields to the feature classes automatically. The steps that this script needs to perform include:

1. Set the workspace to the SAM ArcSDE Geodatabase

2. Retrieve a list of the feature datasets for each species and iterate through this list

3. For each feature dataset retrieve a list of the feature classes and iterate through this list

4. For each feature class retrieve a list of the fields that contain the string "update"

5. Perform a conditional statement testing if the list of fields is greater than zero. If the field list contains 1 or more items then it indicates that the audit fields already exist in the feature class and the new audit fields do not need to be added.

6. If the field list contains 0 items then use the AddField geoprocessing tool to add the four audit fields and distribution type to the feature class with the data type and length as listed in the Introduction.

A Python file was created and saved using the Python IDLE program to perform these steps (Appendix E).

Next, because the SAM data is checked out from the ArcSDE Geodatabase four to five times each year, automating this process through a Python script would also result in a considerable savings of time. The steps that this script would perform include:

1. Determining if previously checked out Personal Geodatabases exist in the check out directory and delete them if they do.

2. Retrieve a list of the feature datasets for each species and iterate through this list

3. For each feature dataset

    a. Truncate the full ArcSDE feature dataset name to just the species name

    b. Create a Personal Geodatabase with the species name in version 10.0 database format

    c. Check out all the feature classes in the feature dataset using the Create Replica tool. Parameters include the feature dataset to be checked out, the check_out replica type, the newly created Personal Geodatabase to check out the data to, the name identifying the replica, and the access type.

    d. Set the check out versions in the ArcSDE Geodatabase to private so that they cannot be seen by the ArcSDE users and change the name of the version to the species being checked out

e. Set the ArcSDE edit version to Protected from Public to prevent edits being made to the ArcSDE version once data has been checked out. This eliminates the need to reconcile any conflicts between edits made in the field on the Personal Geodatabases and users making edits in the office on the ArcSDE Geodatabase.

A Python file was created and saved using the Python IDLE program to perform these steps (Appendix F).

An ESRI Add-In was developed using C# and the ESRI Editing Extension for capturing the username of the person logged in, as well as the system date and populating the new audit fields with these values for features being edited. For newly created features, all four of the audit fields are populated. For features that are updated, edits are determined to be either changes to the attributes or changes to the shape. For attribute changes, only the attribute audit fields are updated. For geometry changes, only the shape audit fields are updated.

The Add-In was created by the following steps:

1. Starting Visual Studio 2010 Professional

2. Creating a New Project and selecting the Project Type of Visual C# → ArcGIS →Desktop Add-Ins. The ArcMap Add-in template was selected and the project was provided a name and location.

3. Using the ArcGIS Add-Ins Wizard, the following was completed:

   a. Metadata for the XML file was completed on the Welcome screen

   b. On the Available Add-in Components page the Editor Extension was selected and a Class Name was provided.

4. In the code file, the OnStartEditing and OnStopEditing editor events were wired to the OnStartUp() method. The onChangeFeature and onCreateFeature edit events are wired to the OnStartEditing event if the edit workspace path is in the directory where the SAM checked out data exists on the editing laptop.

5. Two functions were created for getting the system logged in user and the system date. The system user name is truncated to remove the domain name and the first and last characters are converted to upper case to match the database user name convention.

6. The onCreateFeature edit event calls the user and date functions to retrieve that information from the system and then populates all four of the audit fields with those values.

**7.** The onChangeFeature edit event also calls the user and date functions to retrieve that information. This edit event then iterates through fields whose values have changed and tests whether the attribute fields being changed are related to geometry changes and if they aren't, populates the attribute audit fields with the system user and date. It also tests whether a feature's shape has changed and if it has it updates the shape audit fields with the system user and date.

The Visual Studio code for this ESRI C# Add-In can be found in Appendix G.

Finally, because the SAM data is checked in from the Personal Geodatabases back to the ArcSDE database four to five times each year, automating this process through a Python script would also result in a considerable savings of time. The steps that this script would perform include:

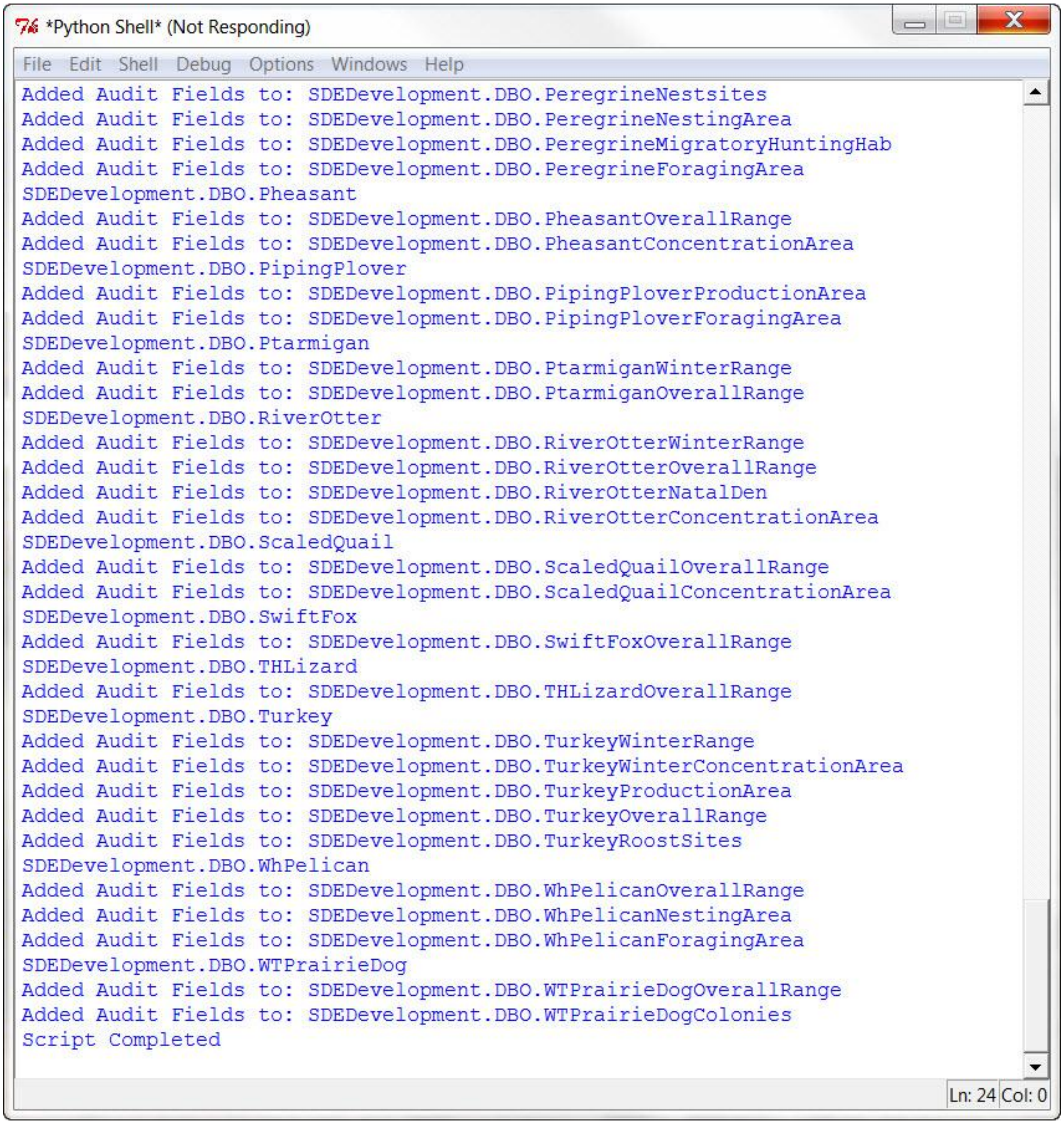
1. Iterating through the list of feature datasets in the ArcSDE SAM database and for each species/feature dataset
    a. Truncate the full ArcSDE feature dataset name down to just the species name to match the checked out Personal Geodatabase names.
    b. Checking in the data from the Personal Geodatabase to the ArcSDE Geodatabase using the Synchronize Changes tool with parameters designating the geodatabase workspaces, the checkout version name, synchronizing from the Personal Geodatabase to the ArcSDE Geodatabase, with the Personal Geodatabases being favored during reconciliation defined by object, and automatically reconciling with the parent version if no conflicts are detected
2. Changing the ArcSDE edit version back to Public so that authorized users can edit the data in the ArcSDE Geodatabase.

A Python file was created and saved using the Python IDLE program to perform these steps (Appendix H).

## RESULTS

The result of running the Add Fields Python script was automatically adding the four audit fields and one distribution type field to each of the 182 feature classes in the SAM ArcSDE Geodatabase if they had not previously been added to the feature class (Figure 16).

**Figure 16. Python IDLE Shell Display Documenting Progress of Add Field Script.**



*Python Shell* (Not Responding)
File Edit Shell Debug Options Windows Help
Ln: 24 Col: 0

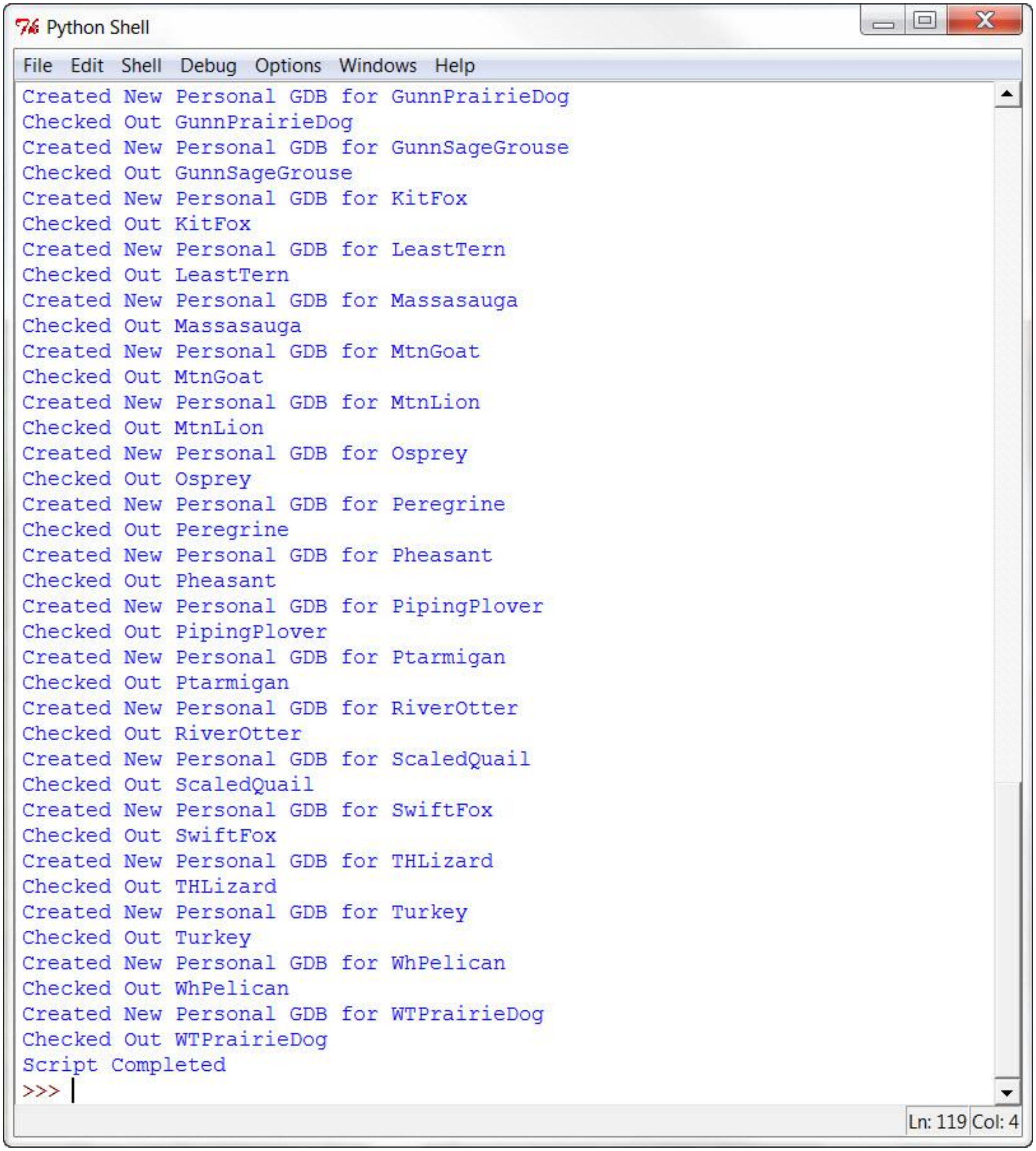
```
Added Audit Fields to: SDEDevelopment.DBO.PeregrineNestsites
Added Audit Fields to: SDEDevelopment.DBO.PeregrineNestingArea
Added Audit Fields to: SDEDevelopment.DBO.PeregrineMigratoryHuntingHab
Added Audit Fields to: SDEDevelopment.DBO.PeregrineForagingArea
SDEDevelopment.DBO.Pheasant
Added Audit Fields to: SDEDevelopment.DBO.PheasantOverallRange
Added Audit Fields to: SDEDevelopment.DBO.PheasantConcentrationArea
SDEDevelopment.DBO.PipingPlover
Added Audit Fields to: SDEDevelopment.DBO.PipingPloverProductionArea
Added Audit Fields to: SDEDevelopment.DBO.PipingPloverForagingArea
SDEDevelopment.DBO.Ptarmigan
Added Audit Fields to: SDEDevelopment.DBO.PtarmiganWinterRange
Added Audit Fields to: SDEDevelopment.DBO.PtarmiganOverallRange
SDEDevelopment.DBO.RiverOtter
Added Audit Fields to: SDEDevelopment.DBO.RiverOtterWinterRange
Added Audit Fields to: SDEDevelopment.DBO.RiverOtterOverallRange
Added Audit Fields to: SDEDevelopment.DBO.RiverOtterNatalDen
Added Audit Fields to: SDEDevelopment.DBO.RiverOtterConcentrationArea
SDEDevelopment.DBO.ScaledQuail
Added Audit Fields to: SDEDevelopment.DBO.ScaledQuailOverallRange
Added Audit Fields to: SDEDevelopment.DBO.ScaledQuailConcentrationArea
SDEDevelopment.DBO.SwiftFox
Added Audit Fields to: SDEDevelopment.DBO.SwiftFoxOverallRange
SDEDevelopment.DBO.THLizard
Added Audit Fields to: SDEDevelopment.DBO.THLizardOverallRange
SDEDevelopment.DBO.Turkey
Added Audit Fields to: SDEDevelopment.DBO.TurkeyWinterRange
Added Audit Fields to: SDEDevelopment.DBO.TurkeyWinterConcentrationArea
Added Audit Fields to: SDEDevelopment.DBO.TurkeyProductionArea
Added Audit Fields to: SDEDevelopment.DBO.TurkeyOverallRange
Added Audit Fields to: SDEDevelopment.DBO.TurkeyRoostSites
SDEDevelopment.DBO.WhPelican
Added Audit Fields to: SDEDevelopment.DBO.WhPelicanOverallRange
Added Audit Fields to: SDEDevelopment.DBO.WhPelicanNestingArea
Added Audit Fields to: SDEDevelopment.DBO.WhPelicanForagingArea
SDEDevelopment.DBO.WTPrairieDog
Added Audit Fields to: SDEDevelopment.DBO.WTPrairieDogOverallRange
Added Audit Fields to: SDEDevelopment.DBO.WTPrairieDogColonies
Script Completed
```

The result of running the Data Check Out Python script included deleting any existing Personal Geodatabases from the checkout directory, iterating through each species feature dataset in the SAM ArcSDE Geodatabase and creating a Personal Geodatabase that all the data in the feature datasets are checked out to (Figure 17). In addition, versions created for each feature dataset checkout were renamed and set to Private so end users could not view them from ArcSDE. The main edit version for ArcSDE users was set to Protected so that the data could be viewed but not edited to eliminate the need to work through reconciling conflicts between data edited in the field and concurrent edits made to the ArcSDE version.

The result of the C# program was an ESRI Add-In that will be copied to SAM editing laptops and installed to work with the editing mxd documents on those laptops (Figure 18). The Add-In will test if the editing workspace is the directory where SAM data is checked out so that it will not attempt to wire onCreateFeature or onChangeFeature events when the user starts an editing session on a different workspace. If the user starts an edit session on the SAM checked out data directory, then the Add-In will populate both the attribute and shape user and date audit fields for the edited feature class for newly created features (Figure 19). For updated features the Add-In will test if only the feature's attribute data has changed and will only populate the attribute audit fields (Figure 20). If only the feature's shape or geometry has been updated then only the shape audit fields will be updated (Figure 21).

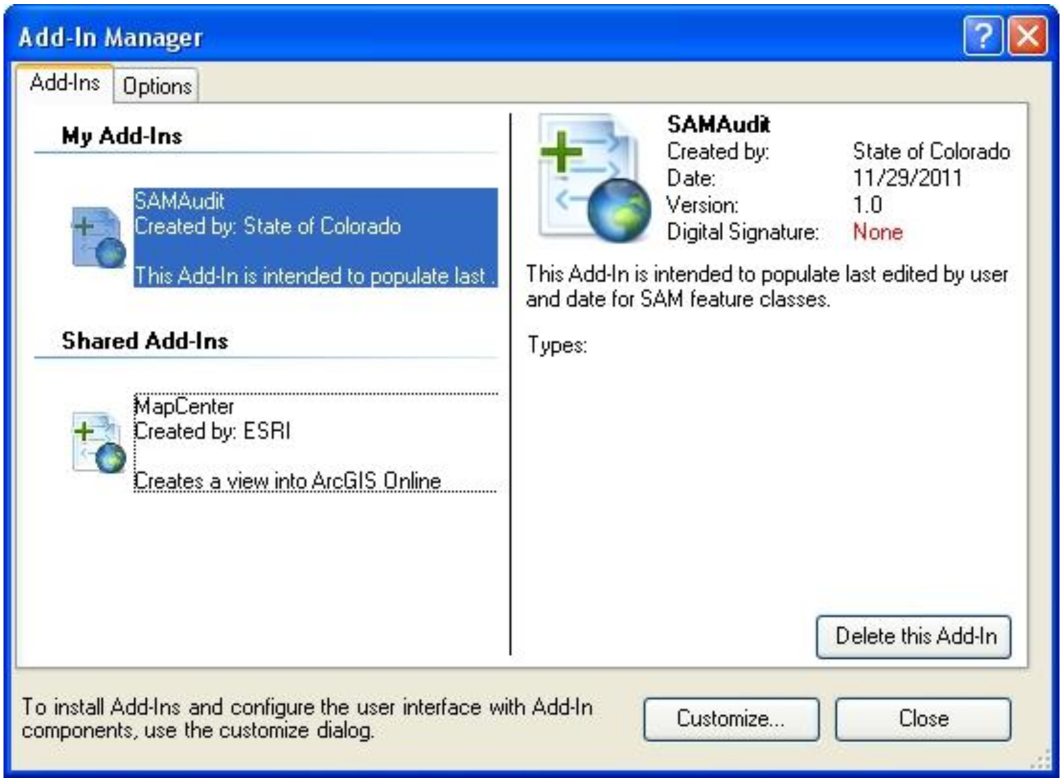
The result of the Python Check In script was synchronizing or checking the data back in from the Personal Geodatabases edited in the field to the ArcSDE Geodatabase version the data was checked out from (Figure 22). Check out versions were removed from the ArcSDE Geodatabase and the primary edit version was set back to Public access so that authorized users could begin editing features in the ArcSDE Geodatabase again.

**Figure 17. Python IDLE Shell Display Documenting Progress of Check Out Script.**



```
Created New Personal GDB for GunnPrairieDog
Checked Out GunnPrairieDog
Created New Personal GDB for GunnSageGrouse
Checked Out GunnSageGrouse
Created New Personal GDB for KitFox
Checked Out KitFox
Created New Personal GDB for LeastTern
Checked Out LeastTern
Created New Personal GDB for Massasauga
Checked Out Massasauga
Created New Personal GDB for MtnGoat
Checked Out MtnGoat
Created New Personal GDB for MtnLion
Checked Out MtnLion
Created New Personal GDB for Osprey
Checked Out Osprey
Created New Personal GDB for Peregrine
Checked Out Peregrine
Created New Personal GDB for Pheasant
Checked Out Pheasant
Created New Personal GDB for PipingPlover
Checked Out PipingPlover
Created New Personal GDB for Ptarmigan
Checked Out Ptarmigan
Created New Personal GDB for RiverOtter
Checked Out RiverOtter
Created New Personal GDB for ScaledQuail
Checked Out ScaledQuail
Created New Personal GDB for SwiftFox
Checked Out SwiftFox
Created New Personal GDB for THLizard
Checked Out THLizard
Created New Personal GDB for Turkey
Checked Out Turkey
Created New Personal GDB for WhPelican
Checked Out WhPelican
Created New Personal GDB for WTPrairieDog
Checked Out WTPrairieDog
Script Completed
>>> 
```

Figure 18. Add-In Manager Showing SAMAudit Add-In Available In ArcMap.



SAMAudit
Created by: State of Colorado

This Add-In is intended to populate last .

MapCenter
Created by: ESRI

Creates a view into ArcGIS Online

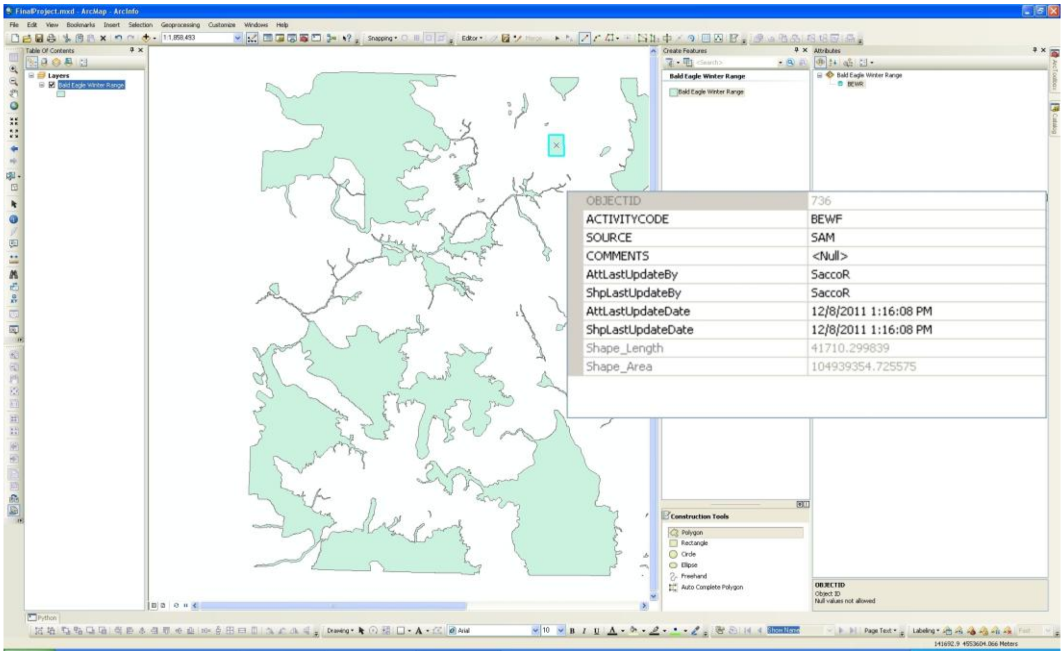
SAMAudit

| | |
|---|---|
| Created by: | State of Colorado |
| Date: | 11/29/2011 |
| Version: | 1.0 |
| Digital Signature: | None |

This Add-In is intended to populate last edited by user and date for SAM feature classes.

To install Add-Ins and configure the user interface with Add-In components, use the customize dialog.

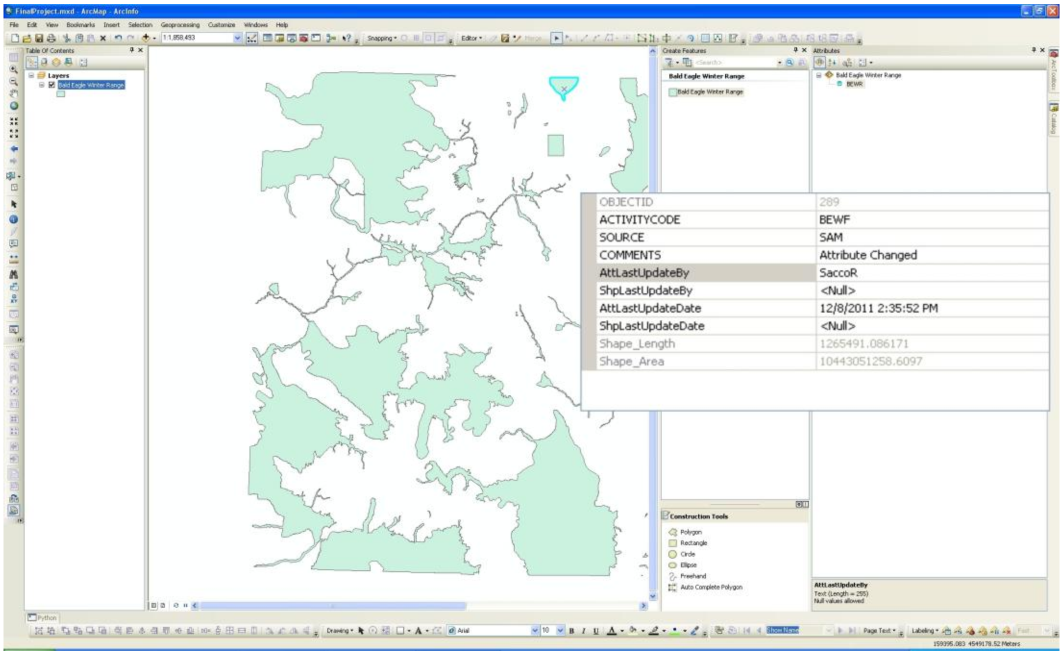
Figure 19. All Audit Fields Populated When A New Feature Is Created.



| | |
|---|---|
| OBJECTID | 736 |
| ACTIVITYCODE | BEWF |
| SOURCE | SAM |
| COMMENTS | <Null> |
| AttLastUpdateBy | SaccoR |
| ShpLastUpdateBy | SaccoR |
| AttLastUpdateDate | 12/8/2011 1:16:08 PM |
| ShpLastUpdateDate | 12/8/2011 1:16:08 PM |
| Shape_Length | 41710.299839 |
| Shape_Area | 104939354.725575 |

OBJECTID
Object ID
Null values not allowed

## Figure 20. Attribute Only Audit Fields Populated When Attributes Are Edited.

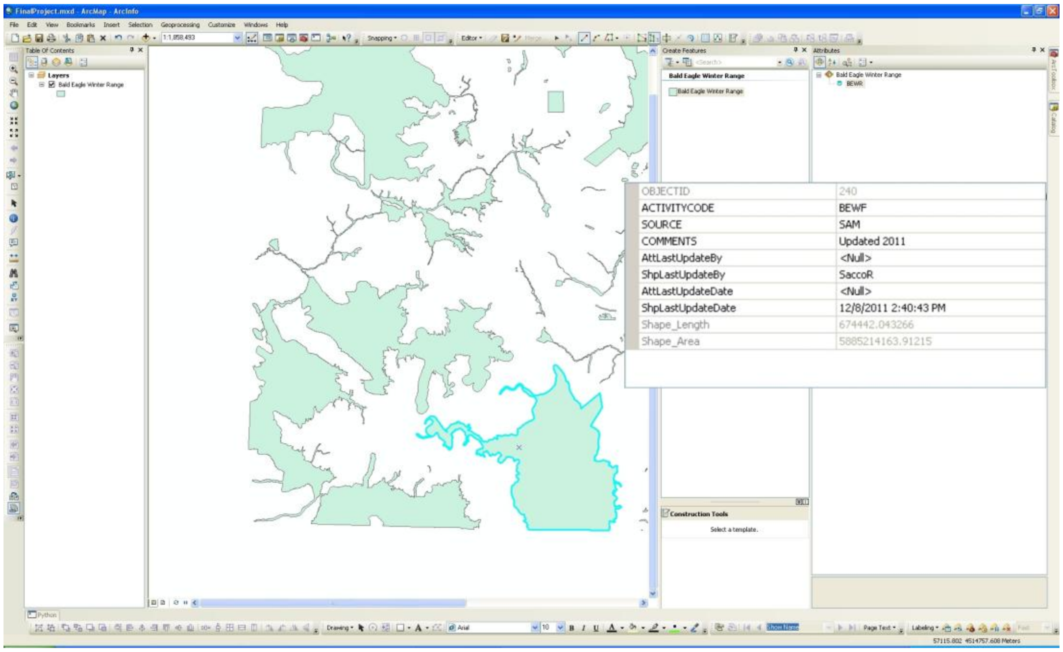


| | |
|---|---|
| OBJECTID | 289 |
| ACTIVITYCODE | BEWF |
| SOURCE | SAM |
| COMMENTS | Attribute Changed |
| AttLastUpdateBy | SaccoR |
| ShpLastUpdateBy | <Null> |
| AttLastUpdateDate | 12/8/2011 2:35:52 PM |
| ShpLastUpdateDate | <Null> |
| Shape_Length | 1265491.086171 |
| Shape_Area | 10443051258.6097 |

AttLastUpdateBy
Text (Length = 255)
Null values allowed

159395.083 4549178.52 Meters

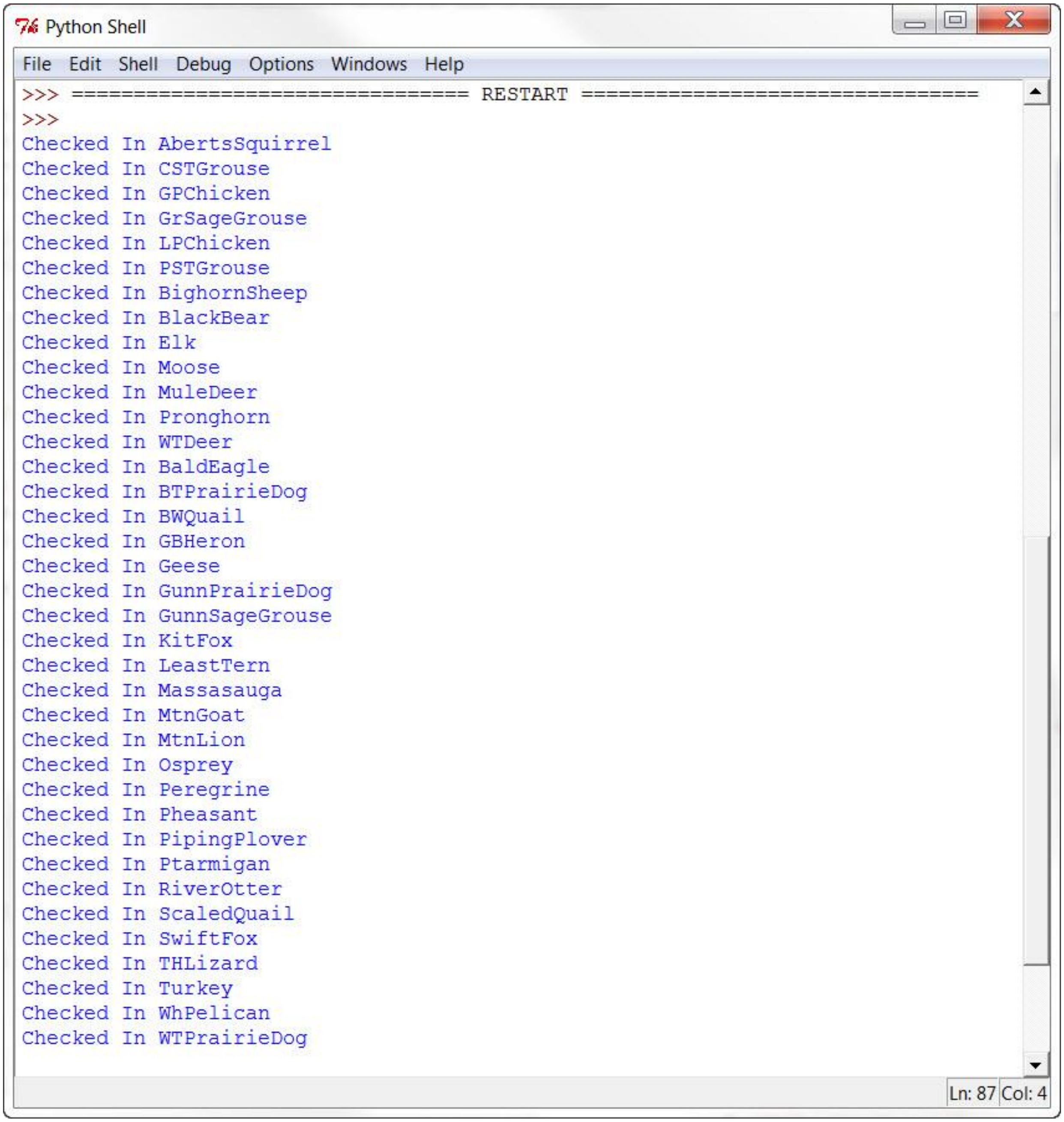
## Figure 21. Shape Only Audit Fields Populated When Geometry Is Edited.



| | |
|---|---|
| OBJECTID | 240 |
| ACTIVITYCODE | BEWF |
| SOURCE | SAM |
| COMMENTS | Updated 2011 |
| AttLastUpdateBy | <Null> |
| ShpLastUpdateBy | SaccoR |
| AttLastUpdateDate | <Null> |
| ShpLastUpdateDate | 12/8/2011 2:40:43 PM |
| Shape_Length | 674442.043266 |
| Shape_Area | 5885214163.91215 |

57115.802 4514757.608 Meters

**Figure 22. Python IDLE Shell Documenting The Progress Of Data Check In Script.**



```
>>> ================================ RESTART ================================
>>> 
Checked In AbertsSquirrel
Checked In CSTGrouse
Checked In GPChicken
Checked In GrSageGrouse
Checked In LPChicken
Checked In PSTGrouse
Checked In BighornSheep
Checked In BlackBear
Checked In Elk
Checked In Moose
Checked In MuleDeer
Checked In Pronghorn
Checked In WTDeer
Checked In BaldEagle
Checked In BTPrairieDog
Checked In BWQuail
Checked In GBHeron
Checked In Geese
Checked In GunnPrairieDog
Checked In GunnSageGrouse
Checked In KitFox
Checked In LeastTern
Checked In Massasauga
Checked In MtnGoat
Checked In MtnLion
Checked In Osprey
Checked In Peregrine
Checked In Pheasant
Checked In PipingPlover
Checked In Ptarmigan
Checked In RiverOtter
Checked In ScaledQuail
Checked In SwiftFox
Checked In THLizard
Checked In Turkey
Checked In WhPelican
Checked In WTPrairieDog
```

## DISCUSSION

This Capstone report intended to demonstrate how over the past three years the skills learned and developed taking courses toward meeting the requirements of the Master of Engineering – GIS Specialty were directly applied to projects I have developed and worked on for my employer, Colorado Parks and Wildlife (CPW).

The first two courses I completed were ISMG courses in the Business School's System Development and Implementation program. The skills learned and developed in these courses led to the completion of projects at CPW that provided better and more efficient methods of communicating various aspects of CPW's enterprise GIS to end users which has resulted in an improvement in the understanding and utilization of the system. The more the enterprise GIS is used in CPW, the better equipped staff are with tools for analyzing, understanding, communicating, and distributing information on the issues that the agency is tasked to address. The first course, Web Development and Practice provided me the skills to design and develop a complete intranet site to represent the GIS Unit of the agency with no prior web programming experience. The web pages of this intranet site allow the GIS Unit to disseminate the constantly changing information on the use of its enterprise system to end users quickly and efficiently. The second course, Database Warehousing and Administration strengthened my database administration and development skills. The result of this course was a project at CPW to develop a new enterprise database for capturing the myriad of data that the agency collects. This database design and development project was also combined with the web application skills learned in the previously described course to create an intranet application that allows end users to enter their wildlife species data collection efforts, as well as providing a comprehensive view of the available species data within the agency and where to find it.

After completing the two ISMG courses, I focused on completing the Civil Engineering GIS courses. In each of the CVEN courses I took, I worked towards applying the skills I learned in them to projects for making improvements in the CPW's Species Activity Mapping (SAM) enterprise GIS database and data collection process.

In Introduction to the GeoWeb, as part of my final project I worked towards developing an easy to use GIS web application for non-GIS technical staff to be able to access various range data for over 37 species. Through the implementation of this project, the CPW GIS Unit was able to eliminate the printing of over 1,500 map pages a year. In addition, this application allows for delivery of real-time edits to staff along with allowing them to zoom to scales appropriate to their needs with several choices for displaying basemaps.

The midterm project for GIS Relational Databases led me to research a NatureServe ESRI Data Model where I discovered aspects of their database schema that would be appropriate to add to the SAM database models. The discovery and implementation of this schema would go a long way to addressing a limitation in the collection of these data that had been considered for some time and would drastically improve the value of these data and its credibility. While considering this schema change I also spent time considering adding other fields to the schema for auditing the data to capture when the data was last edited and by whom. For the final project in the GIS Relational Databases course, I developed a database using several spatial database design tools for capturing the locations and descriptions of places of interest for field staff traveling to remote locations to conduct the SAM edit sessions. Having a database that captures hotels, restaurants, and places of interest and peer recommendations for them at the locations of Area Field Offices will make scheduling these trips quicker and easier, as well as providing a more pleasurable experience visiting these areas.

Finally, one of the reasons there was a reluctance to make schema changes to the SAM database was the fact that there are over 182 feature classes and the changes would need to be made to each of them. This dilemma was solved through the development of Python skills in the GIS Lab course. As part of my final project, I developed a Python script that iterated through all the feature classes and added the appropriate new fields, as well as assigning data types, lengths, domains, and default values to the fields. This represented a significant savings of time versus having to accomplish the task one feature class at a time. Using ArcObjects in a C# Add-In also allowed me to automate the capture of the system time and user who made edits to attributes, geometries, or both and update the audit fields for a feature being edited. The automation of this task solved the problem of the time limitations that were preventing staff from entering update times and users. Having these audit data populated provides a level of accountability for the data, as well as answering common questions about when and who edited data. Finally, an additional two Python scripts were developed to automate the check out and check in of data for disconnected versioning edits. Having these scripts automate this process that used to be accomplished manually also represents another considerable savings of time to the agency in the administration of its enterprise GIS.

The CPW GIS Unit intranet site, Species Inventory database and web application, and SAM Web Mapping application have all been completed and successfully implemented within the agency and have demonstrated marked improvements in communicating information about the use of the enterprise GIS and data collected by the agency along with making GIS data more accessible to non-GIS staff. The Python automation projects for changing data schema's to existing feature classes also successfully added these new fields in a matter of hours running in the background, freeing staff to pursue other projects while they ran. Similarly, automating the check out and check in of disconnected editing data through Python scripts was also run successfully and will represent an on-going savings of staff time as this process is run several times a year, year after year. The SAM schema changes for capturing species distribution types and the audit fields were made recently prior to the start of the SAM editing sessions this year. During the first field edit session, the ESRI Add-In worked as expected and captured the date and user making edits to data without burdening the staff conducting the edits with typing in these attributes. The species distribution type was also used in several instances to capture important differences in a species use of the landscape that will make these data more useful and credible. Time will tell the impact that these changes will have in improving these data but initial testing indicates that these projects should be successful. Finally, the travel database was completed as part of the final course project. However, while the database met the structure and standards of attribute only relational databases and captured the data intended by the stated requirements, ESRI continues to confound end users with their implementation of relational databases in the ArcGIS desktop software. Testing the database with advanced ArcGIS users left them confused about how to work with the data to view the information they needed and the procedures for editing the data were so convoluted as to be unusable. This was the only project that has not been completely implemented within CPW as of the completion of this paper. I plan to pursue making corrections to the database schema, as well as perhaps customizing an online application in order to make this database usable.

As I complete my studies at the University of Colorado Denver, I have these observations to make. I hope this Capstone report demonstrates that my time here has been a success in accomplishing the goals I set out when I first enrolled. The partnership between the Business School and the Civil Engineering Department that attracted me to the program was exactly what I was looking for in combining IT skills with GIS. The strengths of the ISMG courses were in their structured methodology of choosing a particular technology and providing the students with in-depth technical knowledge on how to use that technology through focused exercises, readings of technical manuals, homework, and projects. Limitations to this approach are the lack of exposure to alternative technologies that serve similar functions and the ability to apply the lessons learned to projects of the students choosing. In contrast the CVEN courses provided what I consider to be a higher level overview of the course topics and exposed students to a wide range of technologies being deployed in a particular field. More often, after completing a CVEN course I was left feeling that I learned a little about a wide range of topics but did not master technical expertise in any one area. However, I did greatly appreciate in the CVEN courses the ability to choose my own midterm and final projects which provided the opportunity to choose a specific technology and apply it to a relevant work problem. Perhaps the two schools could learn from one another with the ISMG courses perhaps allowing students to choose their own projects and CVEN limiting courses to slightly more focused topics and technologies.

# REFERENCES

"Add Field (Data Management)". ArcGIS 10 Help. ESRI. 13 Mar. 2012, Web. 08 Dec 2011. http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//001700000047000000

"API Reference". ArcGIS Resource Center – ArcGIS API For Flex. ESRI. n.d., Web. 01 Dec. 2010. http://help.arcgis.com/en/webapi/flex/apiref/index.html

"Biodiversity Model". ESRI Understanding Our World. ESRI & NatureServe. 6 Jul 2001, Web. 01 May 2011. http://support.esri.com/en/downloads/datamodel/detail/4

Boehm, A. 2008. Murach's ASP.NET 3.5 Web Programming With VB 2008. United States of America: Mike Murach & Associates, Inc.

"Building add-ins for ArcGIS Desktop". ArcGIS Resource Center. ArcObjects SDK 10 Microsoft .NET Framework. ESRI. n.d., Web. 08 Dec. 2011. http://help.arcgis.com/en/sdk/10.0/arcobjects_net/conceptualhelp/index.html#/Building_add_ins_for_ArcGIS_Desktop/0001000000w2000000/

Cummins, Jay. "Determine which field is being modified during OnChangeFeature event". GIS Stack Exchange. 8 Jul 2011, Web. 08 Dec. 2011. http://gis.stackexchange.com/questions/11971/determine-which-field-is-being-modified-during-onchangefeature-event

"ESRI Basics of Python". ESRI Training. ESRI. n.d., Web. 08 Dec. 2011. http://training.esri.com/gateway/index.cfm

Evjen, B., Hanselman, S., & Rader, D. 2008. Professional ASP.NET 3.5 In C# and VB. Indianapolis, IN: Wiley Publishing, Inc.

Freeman, E., & Freeman, E. 2006. Head First HTML with CSS & XHTML. Sebastopol, CA: O'Reilly Media, Inc.

Gassner, David. 2010. Flash Builder 4 and Flex 4 Bible. Indianapolis, IN: Wiley Publishing, Inc.

"Kuler". Adobe. n.d., Web. 01 Dec. 2009. http://kuler.adobe.com/#themeID/15325

Mannino, M.V. 2009. Database Design, Application Development, and Administration. 4th ed. Ingram Book Company.

"Samples". ArcGIS Resource Center – ArcGIS API For Flex. ESRI. n.d., Web. 01 Dec. 2010. http://help.arcgis.com/en/webapi/flex/samples/index.html

"Visual C# Programming". Home & Learn. n.d., Web. 08 Dec. 2011. http://www.homeandlearn.co.uk/csharp/csharp.html

W., Phillip. "New Functionality and Tools With Bing Maps (formerly Virtual Earth) in ArcGIS 10". For ArcGIS Server 10. Support Services Blog. ESRI. 19 Nov. 2010, Web. 01 Dec. 2010. http://blogs.esri.com/Support/blogs/supportcenter/archive/2010/11/19/new-functionality-and-tools-with-bing-maps-formerly-virtual-earth-in-arcgis-10.aspx

Watrall, E., & Siarto, J. 2009. Head First Web Design. Sebastopol, CA: O'Reilly Media, Inc.

# APPENDIX A: XHTML CODE FOR INTRANET MASTER PAGE

```
<%@ Master Language="VB" CodeFile="WildlifeGISWeb.master.vb" Inherits="MasterPage"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <link rel="SHORTCUT ICON" href="Images/DOWLOGO.ICO" />
    <title></title>
<asp:ContentPlaceHolder id="head" runat="server">
    </asp:ContentPlaceHolder>
    <link href="App_Themes/WildlifeGISWebMaster.css" rel="stylesheet"
        type="text/css" />
</head>
<body>
    <form id="form1" runat="server">
    <div id="allcontent">
       <div id="header">
            <img src="Images/HeaderGraphic.jpg" alt="Colorado Parks & Wildlife GIS Unit Intranet Pages"
style="vertical-align:middle; padding:0px; margin:0px"/>
       </div>
       <div id="main">
            <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
            </asp:ContentPlaceHolder>
       </div>
       <div id="sidebar">
        <ul id="mainnavmenu">
            <li><a title="Home" href="Default.aspx">Home</a></li>
            <li><a title="Data" href="Data.aspx">Data</a>
                <ul id="datanavmenu" class="secnavmenu">
                    <li><a title="SDE Data" href="SDE.aspx">SDE Data</a></li>
                    <li><a title="Sensitive Data" href="DOWSensitiveDataProcedures.aspx">Sensitive
Data</a></li>
                    <li><a title="Species Data Inventory" href="SpeciesDataInventory.aspx">Species Data
Inventory</a></li>
                    <li><a title="Observation / Sci Coll Data" href="ObservationData.aspx">Observation
/ Sci Coll Data</a></li>
                    <li><a title="Topos/Aerial Photos" href="ToposAerial.aspx">Topos / Aerial
Photos</a></li>
                    <li><a title="County Parcels" href="CountyParcelData.aspx">County Parcels</a></li>
                    <li><a title="External Data" href="ExternalData.aspx">External Data</a></li>
                </ul>
            </li>
            <li><a title="Maps" href="Maps.aspx">Maps</a>
                <ul id="datanavmenu2" class="secnavmenu secnavmenu2">
                    <li><a title="Print Maps" href="PrintMaps.aspx">Print Maps</a></li>
                    <li><a title="Web Maps" href="WebMaps.aspx">Web Maps</a></li>
                </ul>
            </li>
            <li title="GIS Resources">GIS Resources
                <ul id="datanavmenu3" class="secnavmenu secnavmenu3">
                    <li><a title="Training" href="Training.aspx">Training</a></li>
                    <li><a title="How To" href="HowTo.aspx">How To</a></li>
                    <li><a title="Software" href="Software.aspx">Software</a></li>
                    <li><a title="GIS Links" href="GISLinks.aspx">GIS Links</a></li>
                    <li><a title="GPS" href="GPSMain.aspx">GPS</a></li>
                </ul>
            </li>
            <li><a title="Projects" href="Projects.aspx">Projects</a>
                <ul id="datanavmenu4" class="secnavmenu secnavmenu4">
                    <li><a title="BLM Lease Sales" href="BLMLeaseSale.aspx">BLM Lease Sales</a></li>
                </ul>
            </li>
            <li><a title="About Us" href="AboutUs.aspx">About Us</a></li>
            <li><a title="Wildnet" href="http://wildnet/">Wildnet</a></li>
            </ul>
            </div>
       <div id="footer">
            <p>
            Copyright &copy; WildlifeGIS 2009, All Rights Reserved.
            </p>
            <p>
                To provide feedback on this website
            <a href="mailto:robert.sacco@state.co.us?subject=Web Site Feedback">click here</a>
            </p>
      </div>
    </div>
    </form>
</body>
</html>
```

## APPENDIX B: CSS CODE FOR INTRANET FORMATTING

```
* {
    margin: 0px 0px 0px 0px;
    padding: 0px;
}
body
{
    font-family:Arial;
    color:#000000;
    font-size: medium;
    font-family: Verdana, Tahoma, Geneva, Arial, Sans-Serif;
}

#main, #sidebar, #header, #footer
{
    margin:5px;
    background-color:#EEEEEE;
    border:solid 2px #998B7F;
}

#sidebar
{
    width:200px;
    padding-bottom: 0px;
    margin-bottom: 0px;

}

#main
{
    float:right;
    width:715px;
    margin-top:0px;
    padding: 5px;
}

#allcontent
{
    background-color:#366655;
    width: 950px;
    margin: 0 auto;
    border:solid 2px #998B7F;
}

#header
{

}

 #footer
 {
     clear:both;
     text-align:center;
     font-size:small;
 }

 #HotTopics
 {
     background-image: url(../Images/EarthOnFlames.png);
     background-repeat:no-repeat;
     background-position:top left;
 }

#GISUnit
{
    width:450px;
    border-right: dotted 2px #998B7F;
    float:left;
}

#HotLinks
{
    width:250px;
    margin: 10px 0px 0px 450px;

}

 p
 {
     margin:10px;
     line-height: 1.3em;
 }

 li
 {
    margin-bottom: 10px;
    line-height: 1.1em;
}

 ul, ol
 {
    padding: 0px;
    margin: 0px 40px 0px 40px;
}

 #mainnavmenu
 {
     margin: 0px;
     padding: 0px;
     width: 200px;
     font-size: 1.2em;
     font-family: "Palatino Linotype", Palatino, "Book Antiqua", Serif;
     position:relative;
     left:0px;
     list-style-type: none;

 }
 #mainnavmenu li

{

    border-bottom: 1px solid #998B7F;
    padding: 5px;
    width: 190px;
    margin:0px;
    display:block;
    line-height:normal;
}



#mainnavmenu li a
{
    text-decoration: none;
    color:#003893;

}


#datanavmenu, #datanavmenu2, #datanavmenu3, #datanavmenu4

{
    margin:0px;
}

#datanavmenu li, #datanavmenu2 li, #datanavmenu3 li, #datanavmenu4 li
{
    width: 150px;

}

#footer p
{
    vertical-align:middle;
    margin:1px 1px 1px 1px;
}

 li ul
 {
     display:none;


 }

 li:hover > ul
 {
     display:block;

 }

.secnavmenu
{
    position:absolute;
    left:100%;
    top:14%;
    width:160px;
    z-index:598;
    background-color:#EEEEEE;
    border:solid 1px #998B7F;

}

.secnavmenu2
{
    top:29%;

}

.secnavmenu3
{
    top: 44%;

}

.secnavmenu4
{
    top: 57%;
}

 .insetdiv
 {
     background:#FFFBE5;
     margin:25px;
     color:#366655;
     border:solid 1px #366655;
 }

 h1,h2,h3,h4
 {
     font-family: "Palatino Linotype", Palatino, "Book Antiqua", Serif;
     font-weight:bold;

 }

 h1
 {
     text-align:center;
     font-size: 2em;
 }

 h2, h3, h4
 {
     margin:10px;

 }

h2
```

```
{
    font-size: 1.5em;
}

h3
{
    font-size: 1.2em;
    color: #960000 /*404040 */;
}

h4
{
    font-size: 1.0em;
}

.Date
{
    color:#A0A0A0;
}

table.trainingschedule
{
    text-align:center;
    margin: 10px 10px 50px 100px;
    border: solid 1px black;
}

table.trainingschedule td, table.trainingschedule th
{
    padding: 5px;
    width: 200px;
}

table.GeoRefServers
{
    margin:auto;
    text-align:center;
    border: solid 1px black;
}

table.GeoRefServers td, table.GeoRefServers th
{
    padding: 5px;
    width: 200px;
    border: solid 1px;
}

table.GeoRefServers th
{
    border-color:White;
    background-color:Black;
    color:White;
}

ul.districtthumbs
```

```
{
    list-style-type:none;
    margin:0px 10px 0px 25px;
    width:675px;
    display:inline-block;

}

ul.districtthumbs a
{
    width:150px;
    margin:0px 20px 25px 10px;
}

ul.districtthumbs img
{
    float:left;
    margin-right: 10px;
}

ul.districtthumbs li
{
    margin: 0px 20px 100px 0px;
}

.webmapstable
{
    margin-left:10px;
}

.webmapstable tr
{
    vertical-align:middle;
}

#GISUnit h2, #HotLinks h2
{
    font-size:1.2em;
}

#GISUnit p, #GISUnit ul, #HotLinks ul
{
    font-size:0.9em;
    margin-right:5px;
}

ol.roman
{
    list-style:upper-roman;
}

ul.blmindentbullets li
{
    margin-left:30px;
    list-style-type:circle;
}

ul.indentbullets li
{
    margin-left:40px;
}

.tooltips
{
    background-color:Yellow;
    color:Navy;
    border:solid 1px Navy;
    text-align:left;
    display:none;
}

.purpleheading
{
    background-image: url('../Images/background-purple.gif');
    background-repeat: repeat;
    padding: 5px;
    color: White;}
```

# APPENDIX C: SQL DDL CODE FOR CREATING DATABASE

```
--Script to create CDOWData Species tables

USE CDOWData

IF OBJECT_ID ('dbo.SpeciesStatus', 'U') IS NOT NULL
 DROP TABLE SpeciesStatus

IF OBJECT_ID ('dbo.SpeciesxData', 'U') IS NOT NULL
 DROP TABLE SpeciesxData

IF OBJECT_ID ('dbo.SpeciesStatus_lut', 'U') IS NOT NULL
 DROP TABLE SpeciesStatus_lut

IF OBJECT_ID ('dbo.Species', 'U') IS NOT NULL
 DROP TABLE Species

IF OBJECT_ID ('dbo.SpeciesData', 'U') IS NOT NULL
 DROP TABLE SpeciesData

IF OBJECT_ID ('dbo.Organization_lut', 'U') IS NOT NULL
 DROP TABLE Organization_lut

IF OBJECT_ID ('dbo.CollectMethod_lut', 'U') IS NOT NULL
 DROP TABLE CollectMethod_lut

IF OBJECT_ID ('dbo.DataFormat_lut', 'U') IS NOT NULL
 DROP TABLE DataFormat_lut

IF OBJECT_ID ('dbo.DataType_lut', 'U') IS NOT NULL
 DROP TABLE DataType_lut

GO


CREATE TABLE SpeciesStatus_lut
(
 SppStatusKey varchar(10) NOT NULL
  CONSTRAINT PKSpeciesStatusSppStatusKey PRIMARY KEY(SppStatusKey),
 StatusDesc varchar(100) NOT NULL,
 StatusGroup varchar(50)
);


CREATE TABLE Species
(
 SP_Code varchar(6) NOT NULL
  CONSTRAINT PKSpeciesSP_Code PRIMARY KEY(SP_Code),
 EL_Code varchar(10),
 TSN varchar(6),
 Link varchar(6),
 CAT varchar(9),
 Common_Name varchar(50),
 Synonyms varchar(100),
 Scientific_Name varchar(45) NOT NULL,
 Full_Name varchar(75),
 Assembly1 varchar(30) NOT NULL,
 Assembly2 varchar(30) NOT NULL,
 SP_Group varchar(15) NOT NULL,
 Phylum varchar(20),
 SubPhylum varchar(20),
 Class varchar(20),
 SubClass varchar(20),
 SP_Order varchar(20),
 SubOrder varchar(20),
 Family varchar(20),
 SubFamily varchar(20),
 Genus varchar(20),
 SubGenus varchar(20),
 Species varchar(30),
 SubSpecies varchar(90),
 TaxOrder decimal
);


CREATE TABLE SpeciesStatus
(
 SppStatusKey varchar(10) NOT NULL
  CONSTRAINT FKSpeciesStatus_lutSppStatusKey FOREIGN KEY(SppStatusKey)
REFERENCES SpeciesStatus_lut(SppStatusKey) ON UPDATE CASCADE ON DELETE
CASCADE,
 SP_Code varchar(6) NOT NULL
  CONSTRAINT FKSpeciesSp_Code FOREIGN KEY (SP_Code) REFERENCES
Species(SP_Code) ON UPDATE CASCADE ON DELETE CASCADE,
 VerifiedDate datetime,
 VerifiedBy varchar(100),
 VerifiedSource varchar(255),
 VerifiedLink varchar(255),

 CONSTRAINT PKSpeciesStatus PRIMARY KEY (SppStatusKey, SP_Code)

);

CREATE TABLE Organization_lut
(
 OrgKey varchar(10) NOT NULL
  CONSTRAINT PKOrganizationOrgKey PRIMARY KEY(OrgKey),
 OrgDesc varchar(100)
);

CREATE TABLE CDOWUnit_lut
(
 UnitDataKey int IDENTITY(1,1)
  CONSTRAINT PKCDOWUnitDataKey PRIMARY KEY (UnitDataKey),
 CDOWUnit varchar(50) NOT NULL
);
```

```
CREATE TABLE CollectMethod_lut
(
 MethodKey int IDENTITY(1,1)
  CONSTRAINT PKCollectMethodMethodKey PRIMARY KEY (MethodKey),
 MethodDesc varchar(50)
);

CREATE TABLE DataFormat_lut
(
 FormatKey varchar(5) NOT NULL
  CONSTRAINT PKDataFormatFormatKey PRIMARY KEY (FormatKey),
 DataFormat varchar(50)
);

CREATE TABLE DataType_lut
(
 TypeKey varchar(5) NOT NULL
  CONSTRAINT PKDataTypeTypeKey PRIMARY KEY(TypeKey),
 TypeDesc varchar(50)
);

CREATE TABLE SpeciesData
(
 SppDataKey int IDENTITY(1,1)
  CONSTRAINT PKSpeciesDataSppDataKey PRIMARY KEY (SppDataKey),
 ProjectName varchar(100) NOT NULL,
 ProjectDesc varchar(255),
 ProjectKeyWords varchar(255),
 ProjectFocusType varchar(25),
 ProjectFocus varchar(100),
 Contact varchar(50),
 Organization varchar(10)
  CONSTRAINT FKOrganization FOREIGN KEY (Organization) REFERENCES
Organization_lut(OrgKey) ON UPDATE CASCADE,
 UnitDataKey int
  CONSTRAINT FKSpeciesDataUnit FOREIGN KEY (UnitDataKey) REFERENCES
CDOWUnit_lut(UnitDataKey) ,
 PrimaryInvestigator varchar(50),
 ProjectStatus varchar(50),
 BeginDate datetime NULL,
 EndDate datetime NULL,
 StudyAreaType varchar(25),
 StudyAreaHabitat varchar(255),
 StudyAreaLocation varchar(255),
 ULX int,
 ULY int,
 LRX int,
 LRY int,
 DataFormat varchar(5)
  CONSTRAINT FKFormat FOREIGN KEY (DataFormat) REFERENCES
DataFormat_lut(FormatKey) ON UPDATE CASCADE,
 GISDataType varchar(5)
  CONSTRAINT FKType FOREIGN KEY (GISDataType) REFERENCES
DataType_lut(TypeKey),
 DataUpdateDate datetime,
 Comments varchar(255),
 Sensitive varchar(1),
 SDE varchar(1),
 SppDistribute varchar(1),
 CreatedBy varchar(20),
 CreatedDate datetime NULL,

 CONSTRAINT ValEndDate CHECK ((BeginDate = NULL OR EndDate = NULL) OR (EndDate
>= BeginDate)),
 CONSTRAINT DomainFocusType CHECK (ProjectFocusType IN('Single
Species','Multiple Species','Habitat')),
 CONSTRAINT DomainStatus CHECK (ProjectStatus IN('In Progress', 'On-Going',
'Completed', NULL)),
 CONSTRAINT DomainType CHECK (StudyAreaType IN('Local', 'Statewide', 'Multi-
State')),
 CONSTRAINT DomainSensitive CHECK (Sensitive IN ('Y', 'N', NULL)),
 CONSTRAINT DomainSDE CHECK (SDE IN ('Y', 'N', NULL)),
 CONSTRAINT DomainSppDistribute CHECK (SppDistribute IN ('Y', 'N', NULL))
);

CREATE TABLE SpeciesxData
(
 SppDataKey int NOT NULL
  CONSTRAINT FKSpeciesxDataSppDataKey FOREIGN KEY (SppDataKey) REFERENCES
SpeciesData(SppDataKey) ON UPDATE CASCADE ON DELETE CASCADE,
 SP_Code varchar(6) NOT NULL
  CONSTRAINT FKSpeciesxDataSP_Code FOREIGN KEY (SP_Code) REFERENCES
Species(SP_Code) ON UPDATE CASCADE ON DELETE CASCADE,
 CONSTRAINT PKSpeciesxData PRIMARY KEY (SppDataKey, SP_Code)
);

CREATE TABLE CollectxData
(
 SppDataKey int NOT NULL
  CONSTRAINT FKCollectxDataSppDataKey FOREIGN KEY (SppDataKey) REFERENCES
SpeciesData(SppDataKey) ON UPDATE CASCADE ON DELETE CASCADE,
 MethodKey int NOT NULL
  CONSTRAINT FKCollectxDataMethodKey FOREIGN KEY (MethodKey) REFERENCES
CollectMethod_lut(MethodKey) ON UPDATE CASCADE ON DELETE CASCADE,
 CONSTRAINT PKCollectxData PRIMARY KEY (SppDataKey, MethodKey)
);
```

# APPENDIX D: SAM WEB APPLICATION FLEX ACTIONSCRIPT AND MXML CODE

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
      xmlns:s="library://ns.adobe.com/flex/spark"
      xmlns:mx="library://ns.adobe.com/flex/mx"
      xmlns:esri="http://www.esri.com/2008/ags"
      xmlns:layers="com.esri.ags.layers.*"
      pageTitle="SAM Maps"
      width="99%"
      height="99%"
      verticalCenter="0">
 <fx:Style>
  @namespace esri "http://www.esri.com/2008/ags";
  esri|Navigation
  {
   top: 100;
   right: NaN;
   left: 30;
   chrome-color:#036665;
  }
 </fx:Style>
 <fx:Script>
  <![CDATA[
   import com.esri.ags.FeatureSet;
   import com.esri.ags.Units;
   import com.esri.ags.events.ExtentEvent;
   import com.esri.ags.events.MapEvent;
   import com.esri.ags.geometry.Extent;
   import com.esri.ags.geometry.MapPoint;
   import com.esri.ags.utils.GraphicUtil;
   import com.esri.ags.utils.WebMercatorUtil;
   import mx.controls.Alert;
   import mx.controls.Image;
   import mx.events.FlexEvent;
   import mx.printing.FlexPrintJob;
   import mx.printing.FlexPrintJobScaleType;
   import mx.rpc.AsyncResponder;
   import spark.events.IndexChangeEvent;
  //Extent Display Function
   protected function showExtentInGeographic(extent:Extent):String
   {
    const geoExtent:Extent =
WebMercatorUtil.webMercatorToGeographic(Map.extent) as Extent;
    // return geoExtent.toString() + ".." ;
    return " " + geoExtent.xmin.toFixed(6)
     + ", " + geoExtent.ymin.toFixed(6)
     + ", " + geoExtent.xmax.toFixed(6)
     + ", " + geoExtent.ymax.toFixed(6)
     + "   (wkid: " + geoExtent.spatialReference.wkid + ")";
   }
  //Colorado Extent Button Function
```

```
   protected function btnCO_clickHandler(event:MouseEvent):void
   {
    Map.extent = new Extent(-12749780, 4250816, -10772202, 5204751);
   }
  //Change Display and Extent of Selected Feature
   private function doQuery():void
   {
    // clear the graphics layer
     myGraphicsLayer.clear();
  queryTask.execute(query, new AsyncResponder(onResult, onFault));
    function onResult(featureSet:FeatureSet, token:Object = null):void
    {
     if (featureSet.features.length == 0)
     {
      Alert.show("No District found. Please try again.");
     }
     else
     {
      var graphicsExtent:Extent =
GraphicUtil.getGraphicsExtent(featureSet.features);
      if (graphicsExtent)
      {
       Map.extent = graphicsExtent.expand(1.4);
      }
     }
    }
    function onFault(info:Object, token:Object = null):void
    {
     Alert.show(info.toString());
    }
   }
   //Print Function
   private function doPrint(myFlexPrintJobScaleType:String):void
   {
    // Create an instance of the FlexPrintJob class.
    var myPrintJob:FlexPrintJob = new FlexPrintJob();
    // Start the print job.
    if (myPrintJob.start())
    {
     try
     {
      // hide the zoom slider so it won't be printed
      Map.zoomSliderVisible = false;
      btnCO.visible = false;
      bb.visible = false;
      // Add the panel to print.
      myPrintJob.addObject(pnlMap, myFlexPrintJobScaleType);
      // turn the zoom slider back on
      Map.zoomSliderVisible = true;
      btnCO.visible = true;
      bb.visible = true;
     }
     catch (error:Error)
     {
```

```
   Alert.show(error.toString());
  }
  // Send the job to the printer.
  myPrintJob.send();
 }
}


protected function ddlSpecies_changeHandler(event:IndexChangeEvent):void
{
 if (ddlSpecies.selectedIndex > -1)
 {
  pnlMap.title = "SAM Map - " + ddlSpecies.selectedItem;
  lblTrans.visible = true;
  hsTrans.visible = true;
  hsTrans.value = 0.7;
  switch (ddlSpecies.selectedIndex)
  {
   case 0:
    myLegend.layers = [abertssquirrel,areadistricts]
   break
   case 1:
    myLegend.layers = [baldeagle,areadistricts]
   break
   case 2:
    myLegend.layers = [bighornsummer,areadistricts]
    break
   case 3:
    myLegend.layers = [bighornwinter,areadistricts]
    break
   case 4:
    myLegend.layers = [blackbear,areadistricts]
    break
   case 5:
    myLegend.layers = [btpd,areadistricts]
    break
   case 6:
    myLegend.layers = [bwquail,areadistricts]
    break
   case 7:
    myLegend.layers = [cstgrouse,areadistricts]
    break
   case 8:
    myLegend.layers = [elksummer,areadistricts]
    break
   case 9:
    myLegend.layers = [elkwinter,areadistricts]
    break
   case 10:
    myLegend.layers = [geese,areadistricts]
    break
   case 11:
    myLegend.layers = [gbheron,areadistricts]
    break
```

```
case 12:
 myLegend.layers = [gpchicken,areadistricts]
 break
case 13:
 myLegend.layers = [grsagegrouse,areadistricts]
 break
case 14:
 myLegend.layers = [gunnpdog,areadistricts]
 break
case 15:
 myLegend.layers = [gunnsagegrouse,areadistricts]
 break
case 16:
 myLegend.layers = [kitfox,areadistricts]
 break
case 17:
 myLegend.layers = [leasttern,areadistricts]
 break
case 18:
 myLegend.layers = [lpchicken,areadistricts]
 break
case 19:
 myLegend.layers = [massasauga,areadistricts]
 break
case 20:
 myLegend.layers = [moose,areadistricts]
 break
case 21:
 myLegend.layers = [mtngoat,areadistricts]
 break
case 22:
 myLegend.layers = [mtnlion,areadistricts]
 break
case 23:
 myLegend.layers = [mdsummer,areadistricts]
 break
case 24:
 myLegend.layers = [mdwinter,areadistricts]
 break
case 25:
 myLegend.layers = [osprey,areadistricts]
 break
case 26:
 myLegend.layers = [peregrine,areadistricts]
 break
case 27:
 myLegend.layers = [pipingplover,areadistricts]
 break
case 28:
 myLegend.layers = [pstgrouse,areadistricts]
 break
case 29:
 myLegend.layers = [pronghornsummer,areadistricts]
 break
```

```
    case 30:
     myLegend.layers = [pronghornwinter,areadistricts]
     break
    case 31:
     myLegend.layers = [pheasant,areadistricts]
     break
    case 32:
     myLegend.layers = [riverotter,areadistricts]
     break
    case 33:
     myLegend.layers = [scaledquail,areadistricts]
     break
    case 34:
     myLegend.layers = [swiftfox,areadistricts]
     break
    case 35:
     myLegend.layers = [thlizard,areadistricts]
     break
    case 36:
     myLegend.layers = [turkey,areadistricts]
     break
    case 37:
     myLegend.layers = [pelican,areadistricts]
     break
    case 38:
     myLegend.layers = [wtdeer,areadistricts]
     break
    case 39:
     myLegend.layers = [wtpdog,areadistricts]
     break
    case 40:
     myLegend.layers = [ptarmigan,areadistricts]
     break
    default:
     myLegend.layers = [areadistricts]

   }
  }
  else
  {
   pnlMap.title = "SAM Map";
   lblTrans.visible = false;
   hsTrans.visible = false;
   myLegend.layers = [areadistricts];
  }
 }
protected function application1_initializeHandler(event:FlexEvent):void
 {
   /* myLegend.layers = [ legendLayer ];  */
 }
 protected function btnClear_clickHandler(event:MouseEvent):void
 {
  ddlSpecies.selectedIndex = -1;
  ddlDistrict.selectedIndex = -1;
```

```
   pnlMap.title = "SAM Map";
   myGraphicsLayer.clear();
   lblTrans.visible = false;
   hsTrans.visible = false;
   myLegend.layers = [areadistricts]
  }
  /* [Bindable]
  private var sliderValue:Number = hsTrans.value/10; */
  protected function tbLegend_clickHandler(event:MouseEvent):void
  {
   // TODO Auto-generated method stub
  }
  protected function GMUbuttonGroup_changeHandler(event:Event):void
 {
   switch (GMUbuttonGroup.selectedValue)
  {
   case "Big Game GMU's":
    bggmu.visible = true
    bhsgmu.visible = false
    mtngoatgmu.visible = false
    break
   case "Bighorn Sheep GMU's":
    bhsgmu.visible = true
    bggmu.visible = false
    mtngoatgmu.visible = false
    break
   case "Mountain Goat GMU's":
    mtngoatgmu.visible = true
    bggmu.visible = false
    bhsgmu.visible = false
    break
   default:
    bggmu.visible = false
    bhsgmu.visible = false
    mtngoatgmu.visible = false
  }
 }
  protected function cbGMU_changeHandler(event:Event):void
  {
   if (cbGMU.selected==true)
   {
    GMUbuttonGroup.enabled=true
   }
   else
   {
    GMUbuttonGroup.selection=null
    GMUbuttonGroup.enabled=false
    bggmu.visible = false
    bhsgmu.visible = false
    mtngoatgmu.visible = false
   }
  }
 ]]>
</fx:Script>
```

```
 <fx:Declarations>
  <esri:SimpleFillSymbol id="sfs"
          alpha="0.0"
          color="0xFF0000"
          outline="{sls}"/>
  <esri:SimpleLineSymbol id="sls"
          color="0xE600A8"
          width="3"/>
  <esri:QueryTask id="queryTask"

 url="http://dowftcgis/ArcGIS/rest/services/Base/AreaDistrictSAM/MapServer/7"
      useAMF="false"/>
  <esri:Query id="query"
     outSpatialReference="{Map.spatialReference}"
     returnGeometry="true"
     text="{ddlDistrict.selectedItem}"
     where="DISTRICTID = {ddlDistrict.selectedItem}">
  </esri:Query>
  <s:RadioButtonGroup id="GMUbuttonGroup"
   change="GMUbuttonGroup_changeHandler(event)"
   enabled="false"/>
 </fx:Declarations>
 <s:BorderContainer width="100%" height="84" borderVisible="false"
backgroundImage="@Embed('../Images/stripe.jpg')"
backgroundImageFillMode="repeat" horizontalCenter="0">
  <s:BitmapImage source="@Embed('../Images/cpawtext.gif')" right="0"/>
  <s:BitmapImage source="@Embed('../Images/cpawlogo.gif')" left="0" />
 </s:BorderContainer>
 <s:BorderContainer x="0" y="86" width="100%" height="25"
backgroundColor="#036665" backgroundAlpha="0.85" borderVisible="false">
  <s:Label x="5" text="WildlifeGIS Species Activity Web Mapping Application"
fontSize="18" color="#FAF5F5" fontWeight="bold" fontFamily="Georgia"
verticalCenter="0"/>
  <s:Button click="doPrint(FlexPrintJobScaleType.SHOW_ALL)" right="10"
verticalCenter="0" label="Print" id="btnPrint" width="50"/>
  <s:ToggleButton id="tbLegend" right="70" label="Legend" verticalCenter="0"
width="70"/>
  <mx:LinkButton id="lnkDefs" label="Definitions" click="navigateToURL(new
URLRequest('../Documents/SAMDefinitions.pdf'),'quote')"
   verticalCenter="0" right="150" color="#FCF9F9" fontSize="14"/>
  <mx:LinkButton id="lnkSched" label="Data Schedule" click="navigateToURL(new
URLRequest('../Documents/SAM_MapSchedule2010.pdf'),'quote')"
   verticalCenter="0" right="{lnkDefs.width + 150}" color="#FCF9F9"
fontSize="14"/>
 </s:BorderContainer>
 <s:BorderContainer x="0" y="112" width="100%" height="30"
contentBackgroundColor="#036665" backgroundColor="#036665"
backgroundAlpha="0.7" borderVisible="false">
  <s:Label x="10" verticalCenter="0" text="Turn On A Layer:" color="#FCF9F9"
fontSize="14"/>
  <s:DropDownList id="ddlSpecies" x="122" verticalCenter="0" width="175"
prompt="--Choose A Species--"  contentBackgroundColor="#CCCCCC"
change="ddlSpecies_changeHandler(event)">
   <s:ArrayList>
```

```
    <fx:String>Abert's Squirrel</fx:String>
    <fx:String>Bald Eagle</fx:String>
    <fx:String>Bighorn Sheep - Summer Activities</fx:String>
    <fx:String>Bighorn Sheep - Winter Activities</fx:String>
    <fx:String>Black Bear</fx:String>
    <fx:String>Black-tailed Prairie Dog</fx:String>
    <fx:String>Bobwhite Quail</fx:String>
    <fx:String>Columbian Sharp-tailed Grouse</fx:String>
    <fx:String>Elk - Summer Activities</fx:String>
    <fx:String>Elk - Winter Activities</fx:String>
    <fx:String>Geese</fx:String>
    <fx:String>Great Blue Heron</fx:String>
    <fx:String>Greater Prairie Chicken</fx:String>
    <fx:String>Greater Sage Grouse</fx:String>
    <fx:String>Gunnison's Prairie Dog</fx:String>
    <fx:String>Gunnison's Sage Grouse</fx:String>
    <fx:String>Kit Fox</fx:String>
    <fx:String>Least Tern</fx:String>
    <fx:String>Lesser Prairie Chicken</fx:String>
    <fx:String>Massasauga</fx:String>
    <fx:String>Moose</fx:String>
    <fx:String>Mountain Goat</fx:String>
    <fx:String>Mountain Lion</fx:String>
    <fx:String>Mule Deer - Summer Activities</fx:String>
    <fx:String>Mule Deer - Winter Activities</fx:String>
    <fx:String>Osprey</fx:String>
    <fx:String>Peregrine Falcon</fx:String>
    <fx:String>Piping Plover</fx:String>
    <fx:String>Plains Sharp-tailed Grouse</fx:String>
    <fx:String>Pronghorn - Summer Activities</fx:String>
    <fx:String>Pronghorn - Winter Activities</fx:String>
    <fx:String>Ring-necked Pheasant</fx:String>
    <fx:String>River Otter</fx:String>
    <fx:String>Scaled Quail</fx:String>
    <fx:String>Swift Fox</fx:String>
    <fx:String>Texas-horned Lizard</fx:String>
    <fx:String>Turkey</fx:String>
    <fx:String>White Pelican</fx:String>
    <fx:String>White-tailed Deer</fx:String>
    <fx:String>White-tailed Prairie Dog</fx:String>
    <fx:String>White-tailed Ptarmigan</fx:String>
   </s:ArrayList>
  </s:DropDownList>
  <s:Label x="350" verticalCenter="0" text="Zoom To:" color="#FCF9F9"
fontSize="14"/>
  <s:DropDownList id="ddlDistrict" x="415" verticalCenter="0" width="150"
prompt="--Choose A District--" contentBackgroundColor="#CCCCCC"
change="doQuery()">
   <s:ArrayList>
    <fx:String>121</fx:String>
    <fx:String>122</fx:String>
    <fx:String>123</fx:String>
    <fx:String>124</fx:String>
    <fx:String>125</fx:String>
```

```
<fx:String>127</fx:String>
<fx:String>128</fx:String>
<fx:String>141</fx:String>
<fx:String>142</fx:String>
<fx:String>143</fx:String>
<fx:String>144</fx:String>
<fx:String>145</fx:String>
<fx:String>146</fx:String>
<fx:String>181</fx:String>
<fx:String>182</fx:String>
<fx:String>183</fx:String>
<fx:String>184</fx:String>
<fx:String>185</fx:String>
<fx:String>186</fx:String>
<fx:String>187</fx:String>
<fx:String>221</fx:String>
<fx:String>222</fx:String>
<fx:String>223</fx:String>
<fx:String>224</fx:String>
<fx:String>225</fx:String>
<fx:String>226</fx:String>
<fx:String>227</fx:String>
<fx:String>228</fx:String>
<fx:String>241</fx:String>
<fx:String>242</fx:String>
<fx:String>243</fx:String>
<fx:String>244</fx:String>
<fx:String>245</fx:String>
<fx:String>246</fx:String>
<fx:String>247</fx:String>
<fx:String>261</fx:String>
<fx:String>262</fx:String>
<fx:String>263</fx:String>
<fx:String>264</fx:String>
<fx:String>265</fx:String>
<fx:String>266</fx:String>
<fx:String>281</fx:String>
<fx:String>282</fx:String>
<fx:String>283</fx:String>
<fx:String>284</fx:String>
<fx:String>285</fx:String>
<fx:String>286</fx:String>
<fx:String>287</fx:String>
<fx:String>288</fx:String>
<fx:String>321</fx:String>
<fx:String>322</fx:String>
<fx:String>323</fx:String>
<fx:String>324</fx:String>
<fx:String>325</fx:String>
<fx:String>326</fx:String>
<fx:String>327</fx:String>
<fx:String>328</fx:String>
<fx:String>329</fx:String>
<fx:String>330</fx:String>
```

```
<fx:String>341</fx:String>
<fx:String>342</fx:String>
<fx:String>343</fx:String>
<fx:String>344</fx:String>
<fx:String>345</fx:String>
<fx:String>347</fx:String>
<fx:String>348</fx:String>
<fx:String>361</fx:String>
<fx:String>362</fx:String>
<fx:String>363</fx:String>
<fx:String>364</fx:String>
<fx:String>365</fx:String>
<fx:String>366</fx:String>
<fx:String>367</fx:String>
<fx:String>368</fx:String>
<fx:String>381</fx:String>
<fx:String>382</fx:String>
<fx:String>383</fx:String>
<fx:String>384</fx:String>
<fx:String>385</fx:String>
<fx:String>386</fx:String>
<fx:String>387</fx:String>
<fx:String>421</fx:String>
<fx:String>422</fx:String>
<fx:String>423</fx:String>
<fx:String>424</fx:String>
<fx:String>425</fx:String>
<fx:String>426</fx:String>
<fx:String>427</fx:String>
<fx:String>428</fx:String>
<fx:String>429</fx:String>
<fx:String>442</fx:String>
<fx:String>443</fx:String>
<fx:String>444</fx:String>
<fx:String>445</fx:String>
<fx:String>446</fx:String>
<fx:String>447</fx:String>
<fx:String>448</fx:String>
<fx:String>449</fx:String>
<fx:String>462</fx:String>
<fx:String>463</fx:String>
<fx:String>464</fx:String>
<fx:String>465</fx:String>
<fx:String>466</fx:String>
<fx:String>467</fx:String>
<fx:String>468</fx:String>
<fx:String>482</fx:String>
<fx:String>483</fx:String>
<fx:String>484</fx:String>
<fx:String>485</fx:String>
<fx:String>486</fx:String>
<fx:String>488</fx:String>
<fx:String>491</fx:String>
<fx:String>492</fx:String>
```

```
    <fx:String>493</fx:String>
    <fx:String>494</fx:String>
    <fx:String>495</fx:String>
    <fx:String>496</fx:String>
    <fx:String>521</fx:String>
    <fx:String>522</fx:String>
    <fx:String>523</fx:String>
    <fx:String>524</fx:String>
    <fx:String>525</fx:String>
    <fx:String>526</fx:String>
    <fx:String>527</fx:String>
    <fx:String>541</fx:String>
    <fx:String>542</fx:String>
    <fx:String>543</fx:String>
    <fx:String>544</fx:String>
    <fx:String>545</fx:String>
    <fx:String>546</fx:String>
    <fx:String>547</fx:String>
    <fx:String>548</fx:String>
    <fx:String>551</fx:String>
   </s:ArrayList>
  </s:DropDownList>
  <s:Button id="btnClear" label="Clear" click="btnClear_clickHandler(event)"
x="600" verticalCenter="0" width="60"/>
  <s:Label id="lblTrans" x="700" verticalCenter="0" text="Adjust
{ddlSpecies.selectedItem} Layer Transparency:" color="#FCF9F9"
     fontSize="14" visible="false"/>
  <s:HSlider id="hsTrans" x="{lblTrans.width + 715}" verticalCenter="0"
minimum="0.4" maximum="0.7" snapInterval="0.01" value="0.7"
       showDataTip="false" visible="false"/>
 </s:BorderContainer>
 <s:Panel
  id="pnlMap"
  title="SAM Map"
  height="100%" y="140" width="100%" chromeColor="#FFFACC"
fontFamily="Georgia" fontSize="14">
  <esri:Map id="Map" logoVisible="false" height="100%" width="100%" y="0">
   <esri:lods>
    <esri:LOD resolution="156543.033928" scale="591657527.591555"/>
    <esri:LOD resolution="78271.5169639999" scale="295828763.795777"/>
    <esri:LOD resolution="39135.7584820001" scale="147914381.897889"/>
    <esri:LOD resolution="19567.8792409999" scale="73957190.948944"/>
    <esri:LOD resolution="9783.93962049996" scale="36978595.474472"/>
    <esri:LOD resolution="4891.96981024998" scale="18489297.737236"/>
    <esri:LOD resolution="2445.98490512499" scale="9244648.868618"/>
    <esri:LOD resolution="1222.99245256249" scale="4622324.434309"/>
    <esri:LOD resolution="611.49622628138" scale="2311162.217155"/>
    <esri:LOD resolution="305.748113140558" scale="1155581.108577"/>
    <esri:LOD resolution="152.874056570411" scale="577790.554289"/>
    <esri:LOD resolution="76.4370282850732" scale="288895.277144"/>
    <esri:LOD resolution="38.2185141425366" scale="144447.638572"/>
    <esri:LOD resolution="19.1092570712683" scale="72223.819286"/>
    <esri:LOD resolution="9.55462853563415" scale="36111.909643"/>
    <esri:LOD resolution="4.77731426794937" scale="18055.954822"/>
   </esri:lods>
   <esri:extent>
    <esri:Extent id="cOExtent" xmin="-12749780" ymin="4250816" xmax="-
10772202" ymax="5204751">
     <esri:SpatialReference wkid="102100"/>
     <!-- same as tiled map service below -->
    </esri:Extent>
   </esri:extent>
   <esri:VETiledLayer
    key="AhqAyChDsB-DoqT-N9Lj6VzWco8NkVqFbaNwhJTU2eT43lxUDvfkhf9NMlDaohK8"
    mapStyle="road"
    visible="{bb.selectedIndex==0}" />
   <esri:ArcGISTiledMapServiceLayer

 url="http://server.arcgisonline.com/ArcGIS/rest/services/USA_Topo_Maps/MapSer
ver"
    visible="{bb.selectedIndex == 1}"/>
   <esri:VETiledLayer
    key="AhqAyChDsB-DoqT-N9Lj6VzWco8NkVqFbaNwhJTU2eT43lxUDvfkhf9NMlDaohK8"
    mapStyle="aerialWithLabels"
    visible="{bb.selectedIndex == 2}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="areadistricts"
    name="CPW Administrative Boundaries"

 url="http://dowftcgis/ArcGIS/rest/services/Base/AreaDistrictSAM/MapServer"/>
   <esri:ArcGISTiledMapServiceLayer
    id="comap"
    name="COMaP Land Ownership"
    url="http://dowftcgis/ArcGIS/rest/services/Base/COMaP/MapServer"
    alpha="0.7"
    visible="{cbCOMaP.selected == true}" />
   <esri:ArcGISDynamicMapServiceLayer
    id="bggmu"
    name="Big Game GMU's"
    url="http://dowftcgis/ArcGIS/rest/services/Base/GMU/MapServer"
    visible="false"/>
   <esri:ArcGISDynamicMapServiceLayer
    id="bhsgmu"
    name="Bighorn Sheep GMU's"
    url="http://dowftcgis/ArcGIS/rest/services/Base/BHSGMU/MapServer"
    visible="false"/>
   <esri:ArcGISDynamicMapServiceLayer
    id="mtngoatgmu"
    name="Mountain Goat GMU's"
    url="http://dowftcgis/ArcGIS/rest/services/Base/MTGGMU/MapServer"
    visible="false"/>
   <esri:ArcGISTiledMapServiceLayer
    id="abertssquirrel"
    name="Abert's Squirrel"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/AbertsSquirrel/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==0}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="baldeagle"
    name="Bald Eagle"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/BaldEagle/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==1}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="bighornsummer"
    name="Bighorn Sheep - Summer"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/Bighorn-Summer/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==2}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="bighornwinter"
    name="Bighorn Sheep - Winter"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/Bighorn-Winter/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==3}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="blackbear"
    name="Black Bear"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/BlackBear/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==4}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="btpd"
    name="Black-tailed Prairie Dog"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/BTPD/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==5}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="bwquail"
    name="Bobwhite Quail"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/BWQuail/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==6}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="cstgrouse"
    name="Columbian Sharp-tailed Grouse"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/CSTGrouse/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==7}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="elksummer"
    name="Elk - Summer"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/Elk-Summer/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==8}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="elkwinter"
    name="Elk - Winter"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/Elk-Winter/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==9}"/>
   <esri:ArcGISTiledMapServiceLayer
```

```
 id="geese"
 name="Geese"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/Geese/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==10}"/>
<esri:ArcGISTiledMapServiceLayer
 id="gbheron"
 name="Great Blue Heron"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/GBHeron/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==11}"/>
<esri:ArcGISTiledMapServiceLayer
 id="gpchicken"
 name="Greater Prairie Chicken"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/GPChicken/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==12}"/>
<esri:ArcGISTiledMapServiceLayer
 id="grsagegrouse"
 name="Greater Sage Grouse"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/GrSageGrouse/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==13}"/>
<esri:ArcGISTiledMapServiceLayer
 id="gunnpdog"
 name="Gunnison's Prairie Dog"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/GunnPDog/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==14}"/>
<esri:ArcGISTiledMapServiceLayer
 id="gunnsagegrouse"
 name="Gunnison's Sage Grouse"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/GunnSageGrouse/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==15}"/>
<esri:ArcGISTiledMapServiceLayer
 id="kitfox"
 name="Kit Fox"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/KitFox/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==16}"/>
<esri:ArcGISTiledMapServiceLayer
 id="leasttern"
 name="Least Tern"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/LeastTern/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==17}"/>
<esri:ArcGISTiledMapServiceLayer
 id="lpchicken"
 name="Lesser Prairie Chicken"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/LPChicken/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==18}"/>
<esri:ArcGISTiledMapServiceLayer
```

```
 id="massasauga"
 name="Massasauga"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/Massasauga/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==19}"/>
<esri:ArcGISTiledMapServiceLayer
 id="moose"
 name="Moose"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/Moose/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==20}"/>
<esri:ArcGISTiledMapServiceLayer
 id="mtngoat"
 name="Mountain Goat"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/MtnGoat/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==21}"/>
<esri:ArcGISTiledMapServiceLayer
 id="mtnlion"
 name="Mountain Lion"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/MtnLion/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==22}"/>
<esri:ArcGISTiledMapServiceLayer
 id="mdsummer"
 name="Mule Deer - Summer"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/MuleDeer-Summer/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==23}"/>
<esri:ArcGISTiledMapServiceLayer
 id="mdwinter"
 name="Mule Deer - Winter"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/MuleDeer-Winter/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==24}"/>
<esri:ArcGISTiledMapServiceLayer
 id="osprey"
 name="Osprey"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/Osprey/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==25}"/>
<esri:ArcGISTiledMapServiceLayer
 id="peregrine"
 name="Peregrine Falcon"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/Peregrine/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==26}"/>
<esri:ArcGISTiledMapServiceLayer
 id="pipingplover"
 name="Piping Plover"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/PipingPlover/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==27}"/>
<esri:ArcGISTiledMapServiceLayer
```

```
 id="pstgrouse"
 name="Plains Sharp-tailed Grouse"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/PSTGrouse/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==28}"/>
<esri:ArcGISTiledMapServiceLayer
 id="pronghornsummer"
 name="Pronghorn - Summer"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/Pronghorn-Summer/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==29}"/>
<esri:ArcGISTiledMapServiceLayer
 id="pronghornwinter"
 name="Pronghorn - Winter"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/Pronghorn-Winter/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==30}"/>
<esri:ArcGISTiledMapServiceLayer
 id="pheasant"
 name="Ring-necked Pheasant"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/Pheasant/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==31}"/>
<esri:ArcGISTiledMapServiceLayer
 id="riverotter"
 name="River Otter"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/RiverOtter/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==32}"/>
<esri:ArcGISTiledMapServiceLayer
 id="scaledquail"
 name="Scaled Quail"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/ScaledQuail/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==33}"/>
<esri:ArcGISTiledMapServiceLayer
 id="swiftfox"
 name="Swift Fox"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/SwiftFox/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==34}"/>
<esri:ArcGISTiledMapServiceLayer
 id="thlizard"
 name="Texas-horned Lizard"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/THLizard/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==35}"/>
<esri:ArcGISTiledMapServiceLayer
 id="turkey"
 name="Turkey"
 url="http://dowftcgis/ArcGIS/rest/services/SAM/Turkey/MapServer"
 alpha="{hsTrans.value}"
 visible="{ddlSpecies.selectedIndex ==36}"/>
<esri:ArcGISTiledMapServiceLayer
```

```
    id="pelican"
    name="White Pelican"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/Pelican/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==37}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="wtdeer"
    name="White-tailed Deer"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/WTDeer/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==38}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="wtpdog"
    name="White-tailed Prairie Dog"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/WTPDog/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==39}"/>
   <esri:ArcGISTiledMapServiceLayer
    id="ptarmigan"
    name="White-tailed Ptarmigan"
    url="http://dowftcgis/ArcGIS/rest/services/SAM/Ptarmigan/MapServer"
    alpha="{hsTrans.value}"
    visible="{ddlSpecies.selectedIndex ==40}"/>
   <!--<esri:ArcGISDynamicMapServiceLayer

 url="http://dowgistest/ArcGIS/rest/services/EditTests/RaptorTest/MapServer"
    id="legendLayer"/>-->
   <esri:GraphicsLayer id="myGraphicsLayer"
        graphicProvider="{queryTask.executeLastResult.features}"
        symbol="{sfs}"/>
  </esri:Map>
  <s:ToggleButton id="tbMoreData" y="10" label="More Data" width="90"
fontSize="12" chromeColor="#036665" fontWeight="bold" color="#FAF4F4"
x="10"/>
  <s:BorderContainer id="bcMoreData" width="200" height="135" y="10"
borderVisible="true" left="99" backgroundColor="#B6C2B8"
backgroundAlpha="0.80" visible="{tbMoreData.selected == true}"
cornerRadius="2">
   <s:VGroup paddingTop="5" paddingBottom="5" width="100%" gap="0">
    <s:BorderContainer width="100%" height="25" borderVisible="false"
backgroundColor="#036665" backgroundAlpha="0.2">
     <s:CheckBox
      id="cbCOMaP" label="COMaP Land Ownership"
      left="10"
      verticalCenter="0">
     </s:CheckBox>
    </s:BorderContainer>
    <s:BorderContainer width="100%" height="25" borderVisible="false"
backgroundAlpha="0.0">
     <s:CheckBox
      id="cbGMU" label="Game Management Units"
      left="10"
      verticalCenter="0"
      change="cbGMU_changeHandler(event)">
     </s:CheckBox>
    </s:BorderContainer>
    <s:BorderContainer width="100%" height="25" borderVisible="false"
backgroundColor="#036665" backgroundAlpha="0.2">
     <s:RadioButton
      value="Big Game GMU's" label="Big Game GMU's"
      groupName="GMUbuttonGroup"
      left="20"
      verticalCenter="0">
     </s:RadioButton>
    </s:BorderContainer>
    <s:BorderContainer width="100%" height="25" borderVisible="false"
backgroundAlpha="0.0">
     <s:RadioButton
      value="Bighorn Sheep GMU's" label="Bighorn Sheep GMU's"
      groupName="GMUbuttonGroup"
      left="20"
      verticalCenter="0">
     </s:RadioButton>
    </s:BorderContainer>
    <s:BorderContainer width="100%" height="25" borderVisible="false"
backgroundColor="#036665" backgroundAlpha="0.2">
     <s:RadioButton
      value="Mountain Goat GMU's" label="Mountain Goat GMU's"
      groupName="GMUbuttonGroup"
      left="20"
      verticalCenter="0">
     </s:RadioButton>
    </s:BorderContainer>
   </s:VGroup>
  </s:BorderContainer>
  <s:ButtonBar id="bb"
      right="10" y="10"
      requireSelection="true" fontSize="12" chromeColor="#036665"
fontWeight="bold" color="#FAF4F4">
   <s:dataProvider>
    <s:ArrayList>
     <fx:String>Streets</fx:String>
     <fx:String>U.S. Topo</fx:String>
     <fx:String>Imagery</fx:String>
    </s:ArrayList>
   </s:dataProvider>
  </s:ButtonBar>
  <s:BorderContainer id="bcLegend" width="275" height="100%" y="50"
borderVisible="true" right="10" backgroundColor="#B6C2B8"
backgroundAlpha="0.80" visible="{tbLegend.selected == true}"
cornerRadius="5">
    <!--backgroundImage="@Embed('../Images/SAMLegend.jpg')"-->
   <esri:Legend id="myLegend"
       width="100%" height="100%" left="10" top="10"
       layers="{[areadistricts]}"
       map="{Map}"
       respectCurrentMapScale="true"
       />
  </s:BorderContainer>
 </s:Panel>
 <s:Button x="25" y="250" label="CO" width="33" id="btnCO" enabled="true"
click="btnCO_clickHandler(event)" fontSize="8" height="18" cornerRadius="5"
toolTip="Zoom to Colorado" chromeColor="#036665" color="#FBF5F5"
fontWeight="normal"/>
</s:Application>
```

## APPENDIX E: PYTHON CODE TO ADD FIELDS

```
#-----------------------------------------------------------------------------------------------------------
# ADD AUDIT FIELDS TO ALL SAM FEATURE CLASSES
# CREATED BY ROBERT SACCO CPW
#-----------------------------------------------------------------------------------------------------------
# IMPORT SYSTEM MODULES
IMPORT SYS, STRING, OS, ARCPY
FROM ARCPY IMPORT ENV
#SET THE LICENSE
ARCPY.SETPRODUCT("ARCINFO")
#SET ENVIRONMENT
ENV.OVERWRITEOUTPUT = TRUE
ENV.WORKSPACE = "DATABASE CONNECTIONS\SAM@DPWGISADMIN@DEFAULT.SDE"
WORKSPACE = ENV.WORKSPACE
FDS = ARCPY.LISTDATASETS("*", "FEATURE")
FOR FD IN FDS:
   PRINT FD
   FCS = ARCPY.LISTFEATURECLASSES("*", "ALL", FD)
   FOR FC IN FCS:
      FLD = ARCPY.LISTFIELDS(FC, "*UPDATE*", "ALL")
      IF LEN(FLD) > 0:
         PRINT "DID NOT ADD AUDIT FIELDS TO: " + FC
      ELSE:
         ARCPY.ADDFIELD_MANAGEMENT(FC, "ATTLASTUPDATEBY", "TEXT", "", "", 15, "ATTRIBUTE
LAST UPDATED BY")
         ARCPY.ADDFIELD_MANAGEMENT(FC, "SHPLASTUPDATEBY", "TEXT", "", "", 15, "SHAPE LAST
UPDATED BY")
         ARCPY.ADDFIELD_MANAGEMENT(FC, "ATTLASTUPDATEDATE", "DATE", "", "", "", "DATE
ATTRIBUTE LAST UPDATED")
         ARCPY.ADDFIELD_MANAGEMENT(FC, "SHPLASTUPDATEDATE", "DATE", "", "", "", "DATE
SHAPE LAST UPDATED")
         PRINT "ADDED AUDIT FIELDS TO: " + FC
      DESC = ARCPY.DESCRIBE(FC)
      IF DESC.SHAPETYPE == "POLYGON":
         DTFLD = ARCPY.LISTFIELDS(FC, "DISTRIBUTIONTYPE", "ALL")
         IF LEN(DTFLD) > 0:
            PRINT "DID NOT ADD DISTRIBUTION FIELDS TO: " + FC
         ELSE:
            ARCPY.ADDFIELD_MANAGEMENT(FC, "DISTRIBUTIONTYPE", "TEXT", "", "", 25, "RANGE
DISTRIBUTION TYPE", "", "", "DISTRIBUTIONTYPES")
            ARCPY.ASSIGNDEFAULTTOFIELD_MANAGEMENT(FC, "DISTRIBUTIONTYPE", "KNOWN")
            PRINT "ADDED DISTRIBUTION TYPE FIELD TO: " + FC
PRINT "SCRIPT COMPLETED"
```

# APPENDIX F: PYTHON CODE TO CHECK OUT DATA

```
#------------------------------------------------------------------------------------------------------------
# CHECKOUT ALL SPECIES FROM SAM SDE DATABASE TO SPECIES PERSONAL GEODATABASES
#------------------------------------------------------------------------------------------------------------
# IMPORT SYSTEM MODULES
IMPORT SYS, STRING, OS, ARCPY
FROM ARCPY IMPORT ENV
#SET THE LICENSE
ARCPY.SETPRODUCT("ARCINFO")
#LOAD TOOLBOXES
ENV.OVERWRITEOUTPUT = TRUE
#DELETE EXISTING PERSONAL GEODATABASES
ENV.WORKSPACE = R"G:\PROJECTS\SAM\DATA\SAMSESSIONDBS\SDECHECKOUT"
CHECKOUTDBS = ARCPY.LISTWORKSPACES("*","ACCESS")
FOR CHECKOUTDB IN CHECKOUTDBS:
   ARCPY.DELETE_MANAGEMENT(CHECKOUTDB,"WORKSPACE")
   PRINT "DELETED " + CHECKOUTDB
PRINT "FINISHED DELETING EXISTING DATABASES IN FOLDER."
#CREATE NEW GEODATABASES AND CHECKOUT SAM INDIVIDUAL SPECIES FEATURE DATASETS
FROM SAMEDIT VERSION
ENV.WORKSPACE = "DATABASE CONNECTIONS\SAM@DPWGISADMIN@SAMEDIT.SDE"
WORKSPACE = ENV.WORKSPACE
CHECKOUTFDS = ARCPY.LISTDATASETS("*","FEATURE")
FOR CHECKOUTFD IN CHECKOUTFDS:
   SAMFD = CHECKOUTFD[8:100]
ARCPY.CREATEPERSONALGDB_MANAGEMENT(R"G:\PROJECTS\SAM\DATA\SAMSESSIONDBS\SDE
CHECKOUT",SAMFD + ".MDB","10.0")
   PRINT "CREATED NEW PERSONAL GDB FOR " + SAMFD
   ARCPY.CREATEREPLICA_MANAGEMENT(CHECKOUTFD,"CHECK_OUT",
"G:\\PROJECTS\\SAM\\DATA\\SAMSESSIONDBS\\SDECHECKOUT\\" + SAMFD + ".MDB",SAMFD +
"CHECKOUT","FULL")
   PRINT "CHECKED OUT " + SAMFD
   ARCPY.ALTERVERSION_MANAGEMENT(WORKSPACE,SAMFD + "CHECKOUT","#","SAM
CHECKOUT VERSION","PRIVATE")
ARCPY.ALTERVERSION_MANAGEMENT(WORKSPACE,"SAMEDIT","#","#","PROTECTED")
PRINT "SCRIPT COMPLETED"
```

# APPENDIX G: VISUAL STUDIO ESRI C# ADD-IN CODE

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using ESRI.ArcGIS.Editor;
using System.Security.Principal;
using System.Windows.Forms;
using ESRI.ArcGIS.ArcMap;
using ESRI.ArcGIS.Geodatabase;
namespace SAMAudit
{
    /// <summary>
    /// AuditFeaturesExtension class implementing custom ESRI Editor Extension
functionalities.
    /// </summary>
    public class AuditFeaturesExtension : ESRI.ArcGIS.Desktop.AddIns.Extension
    {
        public AuditFeaturesExtension()
        {
        }
        protected override void OnStartup()
        {
            Events.OnStartEditing += new
IEditEvents_OnStartEditingEventHandler(Events_OnStartEditing);
            Events.OnStopEditing += new
IEditEvents_OnStopEditingEventHandler(Events_OnStopEditing);
        }
        public string getUser()
        {
            string user = WindowsIdentity.GetCurrent().Name.Substring(10);
            string first = user.Substring(0, 1).ToUpper();
            string last = user.Substring(user.Length - 1, 1).ToUpper();
            string middle = user.Substring(1, user.Length - 2);
            string editUser = first + middle + last;
            return editUser;
        }
        public DateTime getDate()
        {
            DateTime editDate = DateTime.Now;
            return editDate;
        }
        void Events_OnStartEditing()
        {
            if
(ArcMap.Editor.EditWorkspace.PathName.Contains("C:\\SAM\\Database\\SAMDBS"))
            {
                Events.OnCreateFeature += new
IEditEvents_OnCreateFeatureEventHandler(Events_OnCreateFeature);
                Events.OnChangeFeature += new
IEditEvents_OnChangeFeatureEventHandler(Events_OnChangeFeature);
                //Events.OnDeleteFeature += new
IEditEvents_OnDeleteFeatureEventHandler(Events_OnDeleteFeature);
```

```
             }
        }
        void Events_OnStopEditing(bool Save)
        {
        }
        void Events_OnCreateFeature(ESRI.ArcGIS.Geodatabase.IObject obj)
        {
            IFeature inFeature = (IFeature)obj;
            string editUser = getUser();
            DateTime editDate = getDate();
            int attBy = inFeature.Fields.FindField("AttLastUpdateBy");
            inFeature.set_Value(attBy, editUser);
            int attDate = inFeature.Fields.FindField("AttLastUpdateDate");
            inFeature.set_Value(attDate, editDate);
            int shpBy = inFeature.Fields.FindField("ShpLastUpdateBy");
            inFeature.set_Value(shpBy, editUser);
            int shpDate = inFeature.Fields.FindField("ShpLastUpdateDate");
            inFeature.set_Value(shpDate, editDate);
        }
        void Events_OnChangeFeature(ESRI.ArcGIS.Geodatabase.IObject obj)
        {
            IFeature inFeature = (IFeature)obj;
            IRowChanges rowChanges = (IRowChanges)obj;
            IFeatureChanges featureChanges = (IFeatureChanges)obj;
            string editUser = getUser();
            DateTime editDate = getDate();
            for (int i = 0; i < inFeature.Fields.FieldCount; i++)
            {
                if (rowChanges.ValueChanged[i])
                {
                    if ((i != inFeature.Fields.FindField("ShpLastUpdateBy")) && (i != 
inFeature.Fields.FindField("ShpLastUpdateDate")) && (i != 
inFeature.Fields.FindField("Shape")) && (i != inFeature.Fields.FindField("Shape_Area")) 
&& (i != inFeature.Fields.FindField("Shape_Length")))
                    {
                        int attBy = inFeature.Fields.FindField("AttLastUpdateBy");
                        inFeature.set_Value(attBy, editUser);
                        int attDate = inFeature.Fields.FindField("AttLastUpdateDate");
                        inFeature.set_Value(attDate, editDate);
                    }
                }
            }
            if (featureChanges != null && featureChanges.ShapeChanged)
            {
                int shpBy = inFeature.Fields.FindField("ShpLastUpdateBy");
                inFeature.set_Value(shpBy, editUser);
                int shpDate = inFeature.Fields.FindField("ShpLastUpdateDate");
                inFeature.set_Value(shpDate, editDate);
            }
        }
        protected override void OnShutdown()
        {
        }
        #region Editor Events
        #region Shortcut properties to the various editor event interfaces
        private IEditEvents_Event Events
        {
            get { return ArcMap.Editor as IEditEvents_Event; }
        }
        private IEditEvents2_Event Events2
        {
            get { return ArcMap.Editor as IEditEvents2_Event; }
        }
        private IEditEvents3_Event Events3
        {
            get { return ArcMap.Editor as IEditEvents3_Event; }
        }
        private IEditEvents4_Event Events4
        {
            get { return ArcMap.Editor as IEditEvents4_Event; }
        }
        #endregion
        void WireEditorEvents()
        {
            //
            //  TODO: Sample code demonstrating editor event wiring
            //
            Events.OnCurrentTaskChanged += delegate
            {
                if (ArcMap.Editor.CurrentTask != null)
                    System.Diagnostics.Debug.WriteLine(ArcMap.Editor.CurrentTask.Name);
            };
            Events2.BeforeStopEditing += delegate(bool save) { OnBeforeStopEditing(save);
};
        }
        void OnBeforeStopEditing(bool save)
        {
        }
        #endregion
    }

}
```

## APPENDIX H: PYTHON CODE TO CHECK IN DATA

```
#------------------------------------------------------------------------------------------------------------
# CHECKIN ALL SPECIES FROM PERSONAL GEODATABASES TO SAM SDE DATABASE
#------------------------------------------------------------------------------------------------------------
# IMPORT SYSTEM MODULES
IMPORT SYS, STRING, OS, ARCPY
FROM ARCPY IMPORT ENV
#SET THE LICENSE
ARCPY.SETPRODUCT("ARCINFO")
#LOAD TOOLBOXES
ARCPY.OVERWRITEOUTPUT = TRUE
#CHECKIN SAM INDIVIDUAL SPECIES FEATURE DATASETS FROM ACCESS DATABASES TO
SAMEDIT VERSION
ENV.WORKSPACE = "DATABASE CONNECTIONS\SAM@DPWGISADMIN@SAMEDIT.SDE"
WORKSPACE = ENV.WORKSPACE
CHECKOUTFDS = ARCPY.LISTDATASETS("*","FEATURE")
FOR CHECKOUTFD IN CHECKOUTFDS:
    SAMFD = CHECKOUTFD[8:100]
ARCPY.SYNCHRONIZECHANGES_MANAGEMENT("G:\\PROJECTS\\SAM\\DATA\\SAMSESSIONDBS\
\SDECHECKOUT\\" + SAMFD + ".MDB",SAMFD +
"CHECKOUT",WORKSPACE,"FROM_GEODATABASE1_TO_2","IN_FAVOR_OF_GDB1"
,"BY_OBJECT","RECONCILE")
    PRINT "CHECKED IN " + SAMFD
ARCPY.ALTERVERSION_MANAGEMENT(WORKSPACE,"SAMEDIT","#","#","PUBLIC")
PRINT "SCRIPT COMPLETED"
```