Abstract: The Arizona Oil and Gas Commission in conjunction with the Arizona Geological Survey have collected a large amount of data for the oil and gas wells in the state. The data covers over 1000 wells that were drilled in the 1940s to the present. This data includes copies of permits, location information, scanned copies of well logs, and digitized versions of the well logs in .las file format. These files have been difficult to distribute efficiently because of an unfriendly user interface. This project gives the Arizona Geological Survey a way to distribute the oil and gas well data through a web application that uses as many of their existing services as possible. To create this map I used the Esri JavaScript API. In this application the users can select multiple wells by clicking and dragging over the well heads they want. This will then display the metadata in a grid along with hyperlinks to the available files for those wells. This data will be primarily used by companies involved with carbon sequestration or others seeking information for geological exploration.
Introduction

The Arizona Oil and Gas Conservation Commission (AZOGCC), which is administered by the Arizona Geological Survey (AZGS), regulates the drilling and production of all oil, gas, helium, carbon dioxide and geothermal resources in Arizona. One of the responsibilities of the AZOGCC is the distribution of well logs and permit files for wells that are three years or older. This project is to aid in the distribution of these files. This study will examine the problems with the old system used to distribute this data and the systems used by other organizations to distribute similar data. Then I will explain the process of a new application I developed for distributing this data.

The old web based system for delivering oil and gas well data was dilapidated and hard to use. The old web page used a text based system to find the data. This system was difficult to use because the system required the users to type in the exact name of the wells or it would not find them. Additionally, instructions for the system were not very easy to understand. The Oil and Gas Commissioner even expressed difficulty operating the system and he had been working with it since it was created.

The data delivered by the AZGS is largely used by organizations working on carbon capture and sequestration. The Rock Mountain Carbon Capture and Sequestration (RMCCS) team is the largest of these groups. The RMCCS is using this data to determine which geological formations have the greatest potential for carbon sequestration in the Rocky Mountain Region. Other users of this data are oil, gas and mining companies conducting exploration for future development of geologic resources.
Given the range of technical abilities of those using this application, the AZGS should have a web application that is flexible for user needs while providing valuable information.

**Methodology**

![Figure 1. USGS Core and Cuttings Well Catalog](image)

Web Application Review/Evaluation

Before prescribing recommendations for functionality and use of a revised AZGS system I reviewed and evaluated three comparable web applications hosted by other agencies. The USGS Core and Cuttings Well Catalog (United States Geological Survey 2013) (Figure 1) provided the best examples of the functions and tools that would serve the needs of the AZGS.
There are a few reasons for this, the first of which is because it uses the Esri JavaScript API. I liked the idea of using the Esri API and this was one of the first apps that I found that used the API. I had a hard time at first finding other pages that used the API outside of the Esri developer’s page. I wanted to see what others had done with it in real life scenarios.

This web application was also a great example for AZGS because the site delivered a similar type of data to their users. They used a map based web application to allow users to browse the wells that had data available to download.

The layout of the USGS page was clean and easy to understand. The map dominated the page and there was very little text and what was there was simple. This text included basic instructions on how to operate the page, names of legend items, visual extent and coordinates of the cursor on the map. The map was also simple in what was displayed starting with a basic base map of aerial photography. It showed the locations of all the wells and state boundaries at all scales. At larger scales more locational information is displayed including county boundaries, townships and sections.

Concerns about the USGS map include how the data was delivered. After selecting the well if the user wants to download the data they have to leave the app and go to another page. An improvement that could be integrated into the AZGS application is that the user is kept on the same page. The experience of switching between pages introduces additional wait time and can be very tedious.

A unique feature on the USGS application is the ability to switch between a text based search and a map based search. The USGS application uses a tab system so that the user does have to change pages if they want to use the other search function.
The second web application I evaluated was the Query Related Resources (Esri 2013) example on the ArcGIS developer’s page (Figure 2). This app is also used to share well log data. Good examples of functionality included the way the data for wells was on the same page as the map. This application used a widget called a data grid to organize and display the data. Issues with this page included the inability to select more than one well unless they were close together.
The last application that I evaluated was the Utah Gas web application (Figure 3). This application did not include any mapping features. It relied on a text query to locate and deliver the data. It did have a good way of displaying search results. This application used a grid to display the data but this grid had hyperlinks in it to download the individual logs associated with the wells.

**Application Requirements**

The criteria that the AZGS set for the application included the following. A simple to use interface, a map based search feature.
A simple user interface that any user could understand was the most important criteria for the application. This is because the interface of the old system was so difficult to use. The old system used a text based search method. If a user wanted to find anything they had to type in the name with quotes around it and the case and punctuation had to match the file names exactly. Also the user instructions for the tool were not clear.

The AZGS also wanted the new application to be map based. This would enable a user to easily see the location of the wells within the state and in relation to each other. This would also make searching for the data more visually pleasing.

In addition to the AZGS priorities, after reviewing other web applications I developed a list of additional expectations for the new application. JavaScript has become a common scripting language for GIS on the internet. Given the functions of the web app, using JavaScript in the new AZGS application seem like a feasible challenge, especially since I had never used it before in GIS. I have previous experience with other programing languages, so I was familiar with some of the basics of JavaScript.

The next criteria that I set for myself was to use the ArcGIS JavaScript API when building the application. An Application Program Interface (API) is “a set of commands, functions, and protocols which programmers can use when building software for a specific operating system. The API allows programmers to use predefined functions to interact with the operating system, instead of writing them from scratch. (Tech Terms 2013)” There are other APIs for creating mapping applications on the internet and these include Leaflet, Google Maps and OpenLayers. I wanted to use the ArcGIS API because there is a large community of developers and lots of
documentation. I considered using Leaflet but I found the documentation confusing and hard to find. For this reason I decided to continue using the ArcGIS API.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Reason</th>
<th>Priority Set by</th>
<th>Priority set by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple interface</td>
<td>The application should be easy to operate for users of all skill levels.</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Map Based Interface</td>
<td>Allows users to see location of wells and their relation to each other.</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>JavaScript</td>
<td>Easily adaptable and is expected to be supported by all major web browsers for foreseeable future.</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Display Basic Well Data</td>
<td>Well data will assist in locating the desired files.</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Utah Oil and Gas Well Log Search

Data

The application was developed to distribute the oil and gas well data in Arizona. This well data, i.e. logs and permits, range in age from 1903 for the earliest permit and 1948 for the earliest log to the present day. A majority of the data that is being distributed by the application is ten years or older. The well header feature class that is used to distribute this data is updated when a permit is issued by the AZOGCC.

Well Headers

The well header feature class is a collection of all the known well heads in Arizona. This includes geothermal, water, oil, gas, potash and test wells. This data comes from a variety of
sources including the Arizona Department of Water Resources, the Arizona Department of Environmental Quality, and the Arizona Geological Survey. Because the data comes from these different sources the quality varies greatly. The application uses only the points that represent oil and gas wells. To accomplish this I applied a filter to the data so that the wells that are classified as oil or gas are included in the map. This way I did not have to consider the quality of all the well heads in the feature class.

Figure 4. Well headers in northeast corner of Arizona
The quality of the oil and gas wells also varies because of the age of the wells. The newer wells in the feature class have the greatest location accuracy because they were collected with GPS or total stations. Their locations can also be verified with aerial photography. The accuracy of the older wells may be lower because many of them were drilled before GPS was invented. The well sites may have been reclaimed or overgrown so that verifying their locations with GPS or aerial photography is difficult or impossible. For those wells without exact latitude or longitude the location was estimated using the Public Land Survey System (PLSS) system and the legal descriptions provided in the permits issued by the AZOGCC.

The well header feature class is in the World Geodetic System 1984 revision (WGS 84). The feature class is in WGS 84 because it is being used for many projects within the AZGS. This data is also served externally to the National Geothermal Data System (NGDS) to aid in the development of geothermal energy within the United States. Part of the data standards for the NGDS is that the data be served in the WGS 84 projection (USGIN 2013). I chose to stay with this projection to reduce problems that could arise with other projects that use this data. Another reason I stayed with this projection was that the AZGS prefers to keep data duplication to a minimum within their servers and I was unable to convince the database administrator that there was a need to have a duplicate of the data in another projection. If a duplicated service were created there was no guarantee that it would be kept up to date. The existing well header service is updated regularly by a script that checks a spreadsheet that is updated whenever the Oil and Gas Commissioner issues a new well permit.
Figure 5. RelatedResource field in the Wellheader service
This application only required a small amount of manipulation to prepare the feature class for deployment. Within the feature class there is a column labeled RelatedResources (Figure 5). The data type for this column is text. The fields in this column contain hyperlinks and their titles. If there is more than one hyperlink for the well they will be delineated by a pipe “|”. These links connect to data on other servers that is related to the well. To make the application function I had to update this related resource field with the links to the well logs and permit files on the AZGS servers. I also had to make sure the server had the most recent files for each well.

**PLSS Data**

The PLSS data used in the application comes from a map service that the AZGS employs in multiple web mapping applications (Figures 6 and 7). This includes the Earth Fissure Map and the Association of American State Geologist (AASG) Data Tracking Map. The service contains both sections and the townships and ranges within Arizona. The service is set up with scale ranges and as a tiled map service so that the data does not overwhelm the map. There are twenty levels within the service to allow for smooth transitions between the different scales.
Figure 6. Sections

Figure 7. Townships and Ranges
Methodology

Code

In terms of process, function is primary then appearance is addressed. Geographic data can be published in many ways on the internet. Some of these consist of open source code and the others are open APIs. Open source software is software that is used, changed and shared by anyone (Open Source Initiative 2013). Leaflet and Mapbook are examples of open source web mapping application. Open APIs are APIs that are free to use by anyone but users cannot change the source code. ArcGIS JavaScript API and ArcGIS Viewer for Flex are both open APIs.

The open API ArcGIS JavaScript was used in this project primarily for simplicity. The primary reason I choose to use the API was simplicity. It can be highly customized and complex if a user is experienced. It can also be simple and easy to understand if the user has a limited experience with writing computer code unlike some programming languages. JavaScript is expected to be supported for years because of its widespread use on the internet. It does not require additional plug-ins to run so it can be used in all of the major browsers. The following sections discuss how JavaScript is used in the major parts of the new AZGS application.

DataGrid

The most important piece of code in the application besides the map is the DataGrid. The DataGrid is a Dijit, a type of widget built into the ArcGIS JavaScript API. DataGrid is a way to display tabular data within a webpage. The grid allows users to sort the data alphabetically or
numerically by any column in the table. The DataGrid can be defined within HTML or JavaScript but is populated with JavaScript.

```
| API No | State Parent No | Operator | County | Township | Section | Depth | Formation TD | Well Holder | Date
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>02/01/2005</td>
<td>AZ000010469</td>
<td>Apache</td>
<td>Yavapai</td>
<td>190</td>
<td>31</td>
<td>3550</td>
<td>Petroleum granite</td>
<td>2010-01-01</td>
<td>02/01/2005</td>
</tr>
<tr>
<td>02/01/2005</td>
<td>AZ000010469</td>
<td>Apache</td>
<td>Yavapai</td>
<td>190</td>
<td>31</td>
<td>3550</td>
<td>Petroleum granite</td>
<td>2010-01-01</td>
<td>02/01/2005</td>
</tr>
</tbody>
</table>
```

**Figure 8. DataGrid populated with Well Header data**

Within the revised application the grid is populated by reading the wellheader table. The updateGrid function within my JavaScript file is programmed to read the following fields: APINo, OtherID, WellName, County, Twp, Rge, Section_, DrillerTotalDepth, FormationTD, RelatedResource. Each of these fields is assigned to fill the corresponding column with the exception of the RelatedResource field. The RelatedResource field contains file names and URLs for the files on the AZGS servers. When this field is read, the code searches for “|” and “,” and then it divides the data based on these (Figure 5). The pipe divides the field based on the resource and the comma divides the URL and its associated name. The file extension on the end of the URL is what determines which column the data will be placed in.

When the data is placed in the DataGrid it is transformed. The file name becomes a hyperlink by using the “href=” command to associate the name with the given URL. The Well Folder column is populated with the URLs that end with the “.pdf” extension. Scanned Well Log is populated with “.tif” extensions and LAS Data is populated with data that ends in the “.las” extension. If none of these are present for an individual record the cell is left blank.
**InfoWindow**

The InfoWindow is another important piece of code that allows users to find the desired data (Figure 9). The InfoWindow is a widget that is based primarily in a HTML popup that usually contains the attributes of the feature that is clicked on by the user. These attributes can be modified with functions within the window to meet the needs of the application. For example the data can be used to fill a pie chart to show comparisons between the different features. The window can also be used to show images such as photos taken of the feature on the ground. Only one InfoWindow can be displayed at a time within the application. In the revised AZGS application the window is used as a way for users to easily select an individual well and examine its attributes.
For this application I displayed the same data in the popup that was displayed in the DataGrid. I used similar functions to the updateGrid function used to populate the popup. There are three functions that read the RelatedResources field and create the hyperlinks. Each type of file required a separate function to read it and turn it into a functioning hyperlink. The remaining data can be read directly by the InfoWindow widget.
Select

Selecting the wells is the last of the major pieces of code in the application (Figures 10 and 11). This is done with the initSelectToolBar function. The function uses two classes to accomplish the selection. Draw and Query. Draw allows the users to draw a box around the desired wells by clicking and dragging. Once the box is drawn, the second class, Query, is activated. The Query class uses the geometry created by the Draw class to complete the selection. Querying the table for those features that were selected then activates the updateGrid function that populates the DataGrid.

**Map Search**: Press the Select Wells button and then draw a box around the area of interest.

**LAS Download**: Clicking on the LAS data opens the LAS data as a text file, use the "Save Page As" function to save the file as a .las file.

**Overview Map**: Click the arrow in the upper right corner to activate the map.

**Map Navigation**: Use the mouse wheel to zoom in and out. Click and hold to pan.

**Caution**: Chrome, Safari, Firefox IE9 or greater are required to properly view this page.

**Select Wells**  **Clear Selection**

*Figure 10. Selection menu*
Look and Feel

The AZOGCC web site that will host the revised AZGS application has a tan and brown theme with gradients that fill the background. They achieved this with an open source content management system called Drupal (Buytaert 2013). Drupal gives developers the ability to create a framework to build a website around. The AZGS also used Drupal to create the navigation bar and background that is uniform across the website.

I considered using the Drupal template for the application but it presents limitations in the design. The template creates a narrow column in the center of the page that contains all the content, the remainder of the page is filled with a tan to white gradient. The unused gradient can
be resized without losing any of the content. The AZGS used this template to easily compensate for varying screen sizes. This is a common practice that I have seen on other websites and content management software and it works well if the content is mostly text.

To maintain consistency between the AZOGCC web site and the revised application, a similar color scheme was maintained. The legend has a darker tan color similar to the tan color on the top of the gradient. The navigation bar is dark brown, similar to the Drupal navigation bar. The gradient in the revised application was added for subtle variety and starts at the original brown and it ends at a slightly lighter brown.

**Symbolization**

The symbolization of the wellheads and the PLSS data within the map was kept simple in the revised application. The wellheader symbol is a small black circle that is centered over the well. This symbol is the same one used for the original data created for the NGDS. For the PLSS data, the symbol was kept to a simple grey outline for the both the township and range and the sections. The grey was a neutral color chosen during the development of the other parts of the application. The intent was to replace the grey with another color when I was closer to finishing the remainder of the project. When I was close to finishing the application I felt that the color worked well with the color scheme found in the other parts of the application. There is a scale range applied to the PLSS data so that sections do not appear at the same time as townships and ranges. The township and range data is displayed when the map is between the 1:144,000 and 1:288,000. The sections appear between the scale ranges of 1:18,000 and 1:72,000. These scale ranges were picked to minimize the clutter on the screen at any scale.
Base Map

The base map selected for the application was the ArcGIS Online World Topographic (Figure 12). This was done for two reasons. First was to reduce the number of services needed to be maintained on the AZGS servers. At least once a year the AZGS takes an inventory of the data on their servers and if they cannot find anyplace the data is being used then they remove it. The idea is to make the application relevant for as long as possible and using this base map would help increase the lifespan of the application.

Figure 12. ArcGIS Online World Topographic Base Map
Esri has ten basic base maps available for public use. The World Imagery map created an overwhelming feeling, given the data I was trying to display. It also lacked any locational information like road names or county boundaries that a user who may be unfamiliar with Arizona would need. For these reasons the topographic map seemed most logical to use. It provides the user with basic terrain data and has labeled roads, state and county boundaries. This base map is also updated regularly by Esri so the AZGS will not need someone to maintain it.

**Development**

Most of the project was focused on the development of the JavaScript, HTML and the CSS files. The JavaScript file is the largest of the three files in this application. This file does most of the work. All of the functions needed to run the application are stored in this file. This includes the updateGrid function, scale bar, selection functionality, and the legend. The services that populate the application, wellheaders, PLSS and the base map, are accessed and managed here.

The HTML file creates the interface that connects the user to the JavaScript and CSS files. HTML relies on the use of tags to organize them. In smaller applications the JavaScript code can be entered directly in the HTML file using the `<script>` tag. With larger applications this can make organization and development difficult. In this application and other large applications the HTML simply references the JavaScript file. The HTML file also controls the layout of the different parts of the application including the placement of the map and the DataGrid.
The CSS file is the smallest of the files in the application. This file controls all of the styling within the application. This includes things like the fonts and colors. The navigation bar in the application relies the most on the CSS file. Like the JavaScript file, all the information within this file can be included in the HTML file. This is accomplished using the <style> tag. The last steps in creating the application were spent editing this file in an effort to match the other pages in the website.

**Results**

The result of this project was the creation of the Oil and Gas Viewer application on the Arizona Oil and Gas Conservation Commissions website. The application allows users to search a map of Arizona for oil and gas wells. Once they select the desired wells they are given a list of all the permit and well log files that are available for download. The navigation bar at the top of the screen contains links to the homepages of the AZGS and the Oil and Gas Conservation Commission and a text based search page that accesses the same database. This text search is still under development by the AZGS. The left side of the screen contains the instructions for the application, the button that activates the selection feature, and the legend for the map. The bottom portion of the screen contains the DataGrid where all of the information for the selected wells and the hyperlinks to the associated files.
Figure 13. Application with Select function results
Figure 14. Application with InfoWindow function results
Conclusion

The purpose of the project was to create an application to deliver oil and gas well data to the public. The end result is a functioning application called the Oil and Gas Viewer. This application is currently being hosted on the Arizona Oil and Gas Conservation Commission’s website. It has been in operation since the beginning of October 2013 and has been viewed over 400 times to date. I attempted to make the code simple so that other web developers could easily maintain it. I learned that developing an application, even a simple one, takes more than one person. I spent a large amount of time getting help from the other web developers in the office and on the Esri forums. The application would have taken twice as long to finish if I did not have the help of my colleagues. If the AZGS plans to extend the lifespan of the application I would add a few more features to assist the users. I would add a Zoom To Selected function in the DataGrid. I would also like to add a dropdown menu that would let the users zoom to the desired township and range.
References


Division of Oil, Gas and Mining - Department of Natural Resources. Oil & Gas Well Log Search. 2013.  


—. Query RelatedRecords Example. 2013.  


