

# **The Natural Resource Decision Support System (NRDSS): Creation of a 9.2 ArcGIS Server Website and a SQL Server 2005 ArcSDE 9.2 Database for the Minnesota Board of Water and Soil Resources (BWSR)**

Barbara Featherly<sup>1,2</sup>

<sup>1</sup>*Department of Resource Analysis, Saint Mary's University of Minnesota, Winona, MN 55987*

<sup>2</sup>*GeoSpatial Services, Saint Mary's University of Minnesota, Winona, MN 55987*

**Keywords:** GIS, SSURGO, ArcGIS Server 9.2, ArcSDE 9.2, Microsoft Windows Applications, Website, SQL Server, ASP.net 2.0, VB.net, Soils Analysis, Soil Suitability

## **Abstract**

In the fall of 2007, GeoSpatial Services (GSS) accepted a contract from the Minnesota Board of Water and Soil Resources (BWSR) to develop a 9.2 ArcGIS Server application. This application was to be a web mapping site that would support Soils Survey Geographic Database (SSURGO) soils data interaction and allow for querying of a subset of the SSURGO dataset for the State of Minnesota. In order to make management of the site possible, a number of applications and scripts were created. This paper describes the functionalities and tools designed and built in the project.

## **Introduction**

The purpose of this research project was to create an application to aid in the distribution of soil information to users across Minnesota using ESRI's ArcServer 9.2 .NET SDK. This project was labeled the Natural Resource Decision Support System (NRDSS) and was developed in conjunction with needs identified by the Minnesota Board of Water and Soil Resources (BWSR). In the past, there were earlier versions of this application in ArcIMS 9.1 that had been approved by Minnesota Board of Water and Soil Resources. The ArcGIS Server version of this project application was to recreate the functionality of the former site.

As such, this paper focuses on the development functionality per infrastructure requirements. The application had four infrastructure requirements. The first requirement involved saving database information using SQL Server 2005 with the

exception of soils shapefiles that were packaged with the Soils Survey Geographic Database (SSURGO). These files were packaged separately and saved in a predefined location on the hard drive. The second requirement stated the application needed to work in Internet Explorer Browser 6.0 or higher. The third requirement stated the system and network administrators would need to have access to the web server and the database server boxes. The final requirement was Microsoft Windows Operating Systems (OS) must be used to manage the basic web application tools and startup functions. This included adding the SSURGO shapefile to the map.

There were several components and applications required in order to make the final application manageable and efficient. The ArcGIS Server web mapping application was the main website. This allows users of the internet to view, query, download, and print maps of soils information by predefined districts as well

as other background information to support other natural resource management decision-making processes. To support the website and the database, several XP/Server 2003 windows applications were also created: Clean Files, Clean Up Tables, and Update Soils. Both clean up tools delete temporary files and tables created by the website in response to user requests. The update soils tool is a service that runs on a local computer and it allows administrators to reload SSURGO soils tables for each map unit.

Since the database uses only select fields and tables from the entire SSURGO database, it was necessary to create a method of updating data where the user would not require any knowledge of Access or SQL Server 2005. More about the data structure will be discussed in the section, Windows Applications subsection Update Soils. There are two types of soil queries that run in the application: selection from a query tool and predefined queries. In order to manage the predefined queries, the Query Administration website was created. The website allows administrators to create, modify and delete queries that run against the soils database.

## **Database**

Two databases were created using SQL Server 2005. The first database was dedicated to the SSURGO database files and all related tables. The second database was dedicated to all other data using SDE 9.2 spatial or tabular data including temporary tables. The data were split into two groups: SSURGO data and other data. The following sections explain the distribution and source of all data utilized in the main website and the administration website.

### ***SSURGO data***

The SSURGO databases are extensive tables possessing the ability to contain attributes for a myriad of soil properties reaching more than 500 different attributes. The reason for the number of attributes is because SSURGO databases are designed to hold information of all soil structures that may be present in land within the United States. Each set of tables is packaged in an Access database with an associated shapefile of the study area. NRCS distributes this data to the public on a website and is aggregated together by map areas. The map areas roughly equate to counties in Minnesota with some larger counties subdivided into districts.

The SSURGO database contains many attributes that are relevant to specific soils types. For example there are places in the SSURGO database designated for the definition of soil permafrost characteristics, these attributes were not included because there are no soils in Minnesota with permafrost. In order to limit the database size and increase application's performance, the SSURGO database was simplified to include only those attributes identified by BWSR to be relevant to Minnesota soil structures.

In addition to retaining only a portion of the attributes, the database was restructured such that every resulting table in the database contained a one-to-one relationship with all the others to allow for a direct linkage to information in the soil shapefiles.

Attribute values were retained and only a few additional fields were generated to be used for table joining purposes. The database can be updated automatically using the soils update tool. In turn, this updates a single map district published by National Resources Conservation Service (NRCS), resulting in an up-to-date soils database for an area. Refer to the section on the soils update tool for more information. Refer to

Appendix A for a breakdown of the attributes from the SSURGO database. Refer to Appendix B for a complete breakdown of the data restructuring. The application that restructures the data assumes that each map unit has a maximum of six components and six horizons per component. When a database had more than six of either a component or a horizon than the tool retains the six with the highest percentage of the total parent unit.

### ***Other Data***

There are several other tables contained in the soils database. One table is an imported metadata table created by NRCS and distributed with the SSURGO databases. This table is used to inform users about the various fields while they are either creating a query in the main website or creating a predefined query in the Canned Query Administration website. Another table holds the user information for the Canned Query Administration website. For each predefined query created, a SQL Server procedure is produced in the database and can then be run by users of the main website.

Another table in the soil database retains information about the predefined queries used to populate tools in the main site and makes the procedures more easily accessible. In a folder on the server that runs the website, all of the shapefiles are saved to a local disk. The final tables contain column information for fields not defined by SSURGO and another table for information describing the SSURGO shapefiles. This includes a listing of the soil district and availability status. Not all map areas were available when this application was created.

The second database was Spatially Enabled using ESRI ArcSDE 9.2 SP5 and contains all of the vector data needed by the site. This includes a 30 meter digital elevation model for the entire state. The

main site also includes a link to the Limick Aerial Photography service. This link connects the website to a web media service controlled by a third party. This service is provided free of charge and requires that the main website have access to the internet outside the network although it can be configured to run through a proxy server.

### **Windows Applications**

These applications are used for maintenance purposes. The files are saved on the local disk. Both cleaning applications are executable files that are run by the Task management options of a Windows OS and were scheduled to run once every hour. The Update Soils application is a windows service that actively watches a folder on a local drive. Whenever the contents change (file is copied to folder), the system triggers for the executable to run.

### ***Clean Files***

This executable was made using Visual Studio 2005 windows form template using VB as the language and the .net framework. The ArcServer output folder is defined by the administrator of ArcGIS Server and can be found on the web server. This location is used by all the tools within the application to create temporary objects and map images.

Every time the executable is run, the application checks the ArcServer output folder and looks for all files with the following extensions: csv, png, tiff, jpg, pdf and bmp, that are older than a half hour and deletes them. These are temporary files created by various tools that run on the website. When it is finished searching, the files in the main directory and the table\_export sub directory, the application stops running.

### ***Clean Tables***

This executable was made using Visual Studio 2005 windows form template using VB as the language and the .net framework. Every time the executable is run, this application creates a connection to the spatially enabled database described in the Other Data section above. It then searches the database for all tables starting with Query\_ or Temp\_ and tests their created date. All tables older than a half hour are deleted from the database. When searching is finished, the application closes its connection to the database and stops running.

### *Update Soils*

This executable was made using Visual Studio 2005 windows service template using VB as the language and the .net framework. The service can be installed on any windows computer in the same network as the database server. The service is installed on a computer. Whenever it is running, it monitors the "C:\update\_soils\" directory of the computer. Whenever a file is copied to the directory, the service runs an executable that verifies that the file is an Access database and that it is finished loading.

In order to extract the map unit from the Access database, the application establishes a connection with the SQL Server Soils database and begins extracting the needed tables for the database. To save room, the application only creates tables that are required to copy the data. For example, if none of the map units in the access database have more than 3 components, then tables for the other three components (4-6) and horizons (19-36) are not created. When all the rows for the needed fields are copied from the Access database, the application closes its connection with the SQL server database and deletes the access database from the directory.

The application searches the

directory for more access databases and follows the same protocol for each database. If an error occurs during the processing of a database, the database is not deleted but moved to the errors folder along with a log file with the same name as the database file and describing the error that is preventing importation. When all of the access databases found in the directory are processed, the executable file stops running and the service starts monitoring the directory. Whenever the service is stopped, the monitoring of the directory stops and the application is no longer triggered.

### **Website Applications**

There were two websites created for this application. Both websites were developed to run in IIS 6.0 using Windows Server 2003. Any errors that occur in the sites are saved either in a log file on the local drive or contained within the IIS standard log depending on the source of the error.

The first website discussed is the Administration site for the Canned Queries. This site is a vb.net ASPX web application that utilizes form authentication and a user table to secure the application against unauthorized use. The purpose of the site is to create, modify or delete TSQL procedures that are run in the second application. Whatever changes or queries are created, the mapping website responds to these changes in real-time as long as the user clicks the finish button for each tool.

The second application is a vb.net ASPX ArcServer 9.2 mapping web application called NRDSS. This is the main site developed. The tools in the application can be split into these categories: Navigation or Zoom tools, Soils Queries, and the Soil export tools.

### *Administration of Canned Queries*

First, the user must login to the application (Figure 1). Once the user has logged in, they must choose the tool desired for use. Using frames the website shows a menu that remains on the left hand side of the client's screen (Figure 2). The right hand side of the screen changes depending on the tool selected by the user.

The Help link opens a PDF version of the website's user manual. To logout of the site and close the browser window, the user must click the Exit Site link. To open the main website, the user clicks the Return to Mapping Site link.

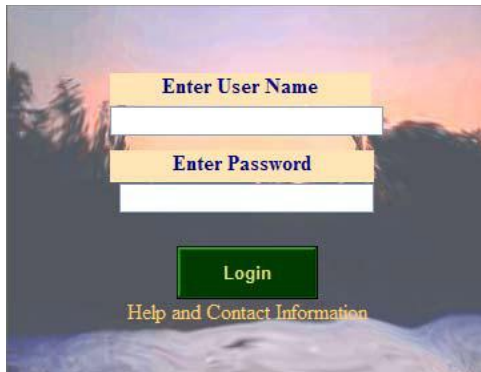


Figure 1. Administration Website Login Screen.

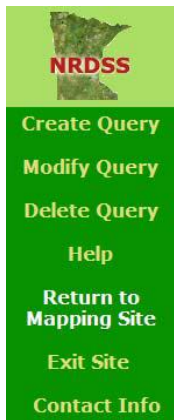


Figure 2. Administration Menu.

The Contact Information link aids the user in reporting errors and issues experienced while running the site. When the user wishes to run the three main procedure tools in the application (Create, Modify and Delete Query), they click the

matching link.

### Create Query

There are several steps to creating a query. To see images depicting these steps refer to Appendix C where the steps are numbered in order of use. This was created as a step by step process. The user creates the query and then enters each restriction one by one, finally triggering the creation of a SQL Server procedure.

The first step allows user to give the query a name and a label. The name can not be more than 100 characters and can not have spaces or under scores in it. The label will be the name that will be visible in all dropdown menus like the Existing Queries drop down. Neither the name nor the label can be the same as those already in existence (Figure 3).

<b>Name</b>	<b>Query Label</b>
<input type="text" value="Test"/>	<input type="text" value="Test 11/09/2007"/>
<b>Start Creation</b>	<b>Query Description</b>
	<input type="text" value="Running for help document creation"/>

Figure 3. Create Name, Label and Description.

After clicking the start button more options appear. There are three drop down menus for each of the three different kinds of soil items: Mapunit, Component, and Horizon. The user chooses a variable and clicks the add button under the variable (Figure 4).

When the user wishes to get more information about the variables in the drop downs, they choose a variable and click the Current Variable Information button. A description of the variable will appear below the form the user is working on. The first line shows the name of the variable, a colon and the variables data type. The second line holds the unit of measurement that the value it may hold. The last paragraph describes what the variable represents. When finished looking at the description, the user can click

the Close Description button and the area will turn blank (Appendix C).



Figure 4. Menus for Soil Data.

Clicking one of the Add buttons causes the Choose Operation buttons to be activated. The name of the variable is added to the variable restrictions list and the add buttons disappear. These buttons represent the different types of comparison operations that can be done. If the variable has a domain then only the = and Not buttons are active. The following is a description of the operations. See Figure 5 for an image of the menu: = is the variable equal to entry, < is the variable less than the entry, > is the variable greater than the entry, Not is the variable not equal to the entered value, and Like is used for textual fields and determines if the entry has the same value as the variable. The Like operation is the only field that the wildcard character % can be used. Refer to Table 1 for examples of all the different data entry approaches.



Figure 5. Variable Operation Menu.

Finally, the user must add a label so that users of the main website will be able to differentiate this restriction from others (Figure 6).

Also note that several restrictions can have the same label so that once the query is run, it will evaluate those restrictions as a single restriction. For example, there were two restrictions (brockdepmin > 200) and (brockdepmin < 0)

with the same label. The predefined query would give those features (mukeys) with a value less than 0 and more than 200 the same label.

Table 1. Examples showing Restriction values.

Numeric or Textual Entries	<p>Enter a value for comparison</p> <input type="text" value="100"/> <p>Enter Value</p>
Wild Card Entries	<p>Enter a value for comparison</p> <input type="text" value="Jim%"/> <p>Enter Value</p>
Domain Entries	<p>Enter a value for comparison</p> <input type="text" value="Choose Domain ..."/> <p>Enter Value</p>

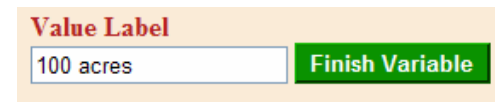


Figure 6. Value Label Menu.

If a user wanted to add another restriction, they would click on the Add Restrictions Button and follow the same procedure illustrated previously. Once all the labels and all of their restrictions are added, then the user fills out the Query description field. If the user clicks the Delete Button, the variable restriction that is selected in the list is deleted. Variable restrictions must be deleted at one time (Figure 7).

When run from the main website, the procedure will evaluate a feature until it finds the first label that applies to it. It prioritizes from the top to the bottom of list. If a user wishes to change the priority of a label, they use the arrow keys (Figure 8).

Variables can only be moved a single location at a time. The steps a user would follow would be as follows:

- 1) Select an entry from the variable restrictions list.
- 2) Click the up arrow to increase the labels priority or click the down arrow to decrease the labels priority.

Once all the restrictions and the Query Descriptions have been entered, the user is to click the Finish button. An alert

will inform the user when the procedure has been created. The user tests a new query by opening the NRDSS site. For an image, refer to Create Query step 4 of Appendix C.

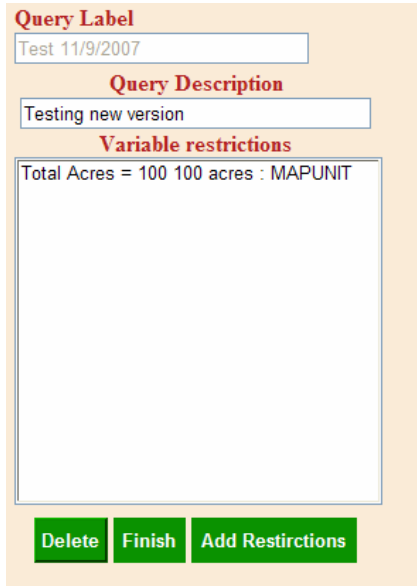


Figure 7. Example of the Menu for the Final Steps of the Create Query Tool.

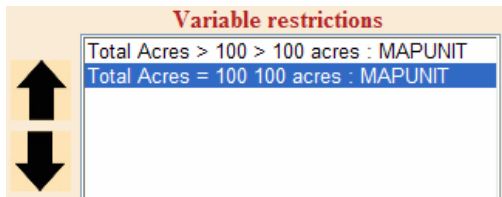


Figure 8. Example of the Menu for the Priority.

### Modify Query

In order to start modifying a query, the first thing the user needs to do is choose a query to change. Once a query has been chosen and the appropriate button clicked, all information is loaded into the form about that query including the list of restrictions. The user follows the same work flow and uses the same tools as the Build Query Tool. The Table Description can be modified in order to fit the changes to the query. The name of the query can not be changed.

### Delete Query

The user chooses the query to delete and click's the Delete Selected. The procedure is deleted from of the database and the information saved for this query in the query table is removed (Figure 9).

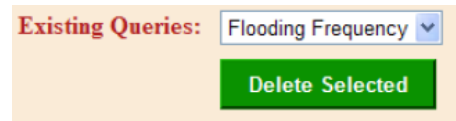


Figure 9. Delete Query Menu.

### *NRDSS (ArcServer 9.2 web mapping Application)*

Refer to Appendix C for an image of the NRDSS mapping site. The site has a lot of different functionalities and each tool can be opened by clicking on the link with the same name as the tool. Tools sit in windows on the left hand side of the screen next to the map.

### Navigation or Zoom Tools

#### *Zoom to Soil District*

A user selects a district from the list and the map zooms to the boundary of that district (Figure 10).



Figure 10. Zoom to District Menu.

This is an Active Server Page (ASP) drop down list that is populated from the Soils\_Districts table using a SQL Query that runs at page load. When a user chooses a district, a JavaScript triggers the AJAX procedure that runs the code that zooms to the extent of the feature in the District SDE layer with the given name. As soon as the user selects a district, the map automatically triggers the zoom. This is done to reduce the number of mouse clicks necessary to

complete the zoom process.

### *Choose Quad*

The user selects a name of a county (Top), then selects the name of a 2400k Quad (Bottom) and the map zooms to the boundary of the selected quad. By triggering the change automatically, the number of mouse clicks needed is reduced (Figure 11). This was created using two ASP drop down list objects. When the user changes the district, it triggers a SQL query of the Quad look up table in the database and populates all quads in or near that county in alphabetic order. When a user changes the name of the Quad, it triggers the code that zooms to the extent of the feature with the given name in the Quads SDE layer.

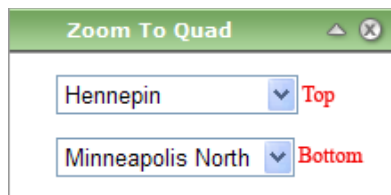


Figure 11. Zoom to Quad Menu.

### *Enter Scale*

The user enters a map scale in the ASP textbox. After clicking go (html button), a JavaScript function that starts an AJAX procedure that will change the map's scale to the scale entered in the text box (Figure 12).

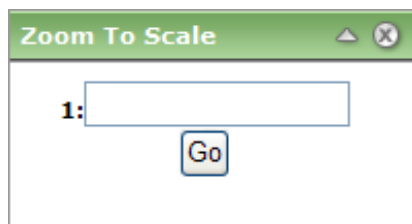


Figure 12. Zoom to Scale Menu.

### *Enter Township or Section*

When the user enters a Township, Range and Section (not required for tool to run),

the map changes and zooms to the boundary of the entered Township and range (Figure 13).

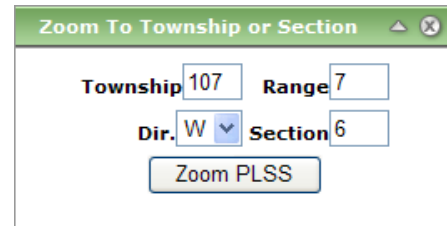


Figure 13. Zoom to Township or Section Menu.

The Township, Range and Section entries are ASP text boxes. Direction (Dir) is an ASP drop down list. When the user presses the Zoom PLSS button, it triggers a function that queries either the Township or the Section SDE feature layer (depending on entry) and zooms the map to the extent of the feature that represents the Township, Range and optional Section values. If the search does not find a feature the application displays an alert to the user informing them that their entry did not return a result from the database.

### *Run Soil Queries*

#### *Map Builder (Create Selection Query)*

This tool allows users to search soil attributes for soil features in the map. When the query is complete, it selects those soil features visible in the map that match the query the user created. They are shown in yellow and the description is added to the Table of Contents of the map. Refer to the Map Builder section of Appendix C for more information on how this tool works from a user perspective. The results of the query are added to the Soils Export Tools list. This will export a shapefile with the spatial results that can be downloaded by the user.

#### *Canned Queries*



The Canned Query Tool allows users to run predefined procedures created by the Query Administration website. Refer to the Canned Query section of Appendix C for more information on how this tool works from a user perspective. The tool takes the name of the procedure that the user selects and runs the TSQL procedure from the database. The procedure will run against the tables representing the Soils data that has been added to the map. The tool can only function on a single map district at a time. The results of the query are added to the map using a predefined coloration ramp (same for all queries run from the canned query drop down list). The layers are added to the TOC and a Query Results box is added to the export tools Extract Window (Figure 14). Only one Query Result is visible at a time. The user can choose to extract the results as a shapefile that is imported into a zip file.

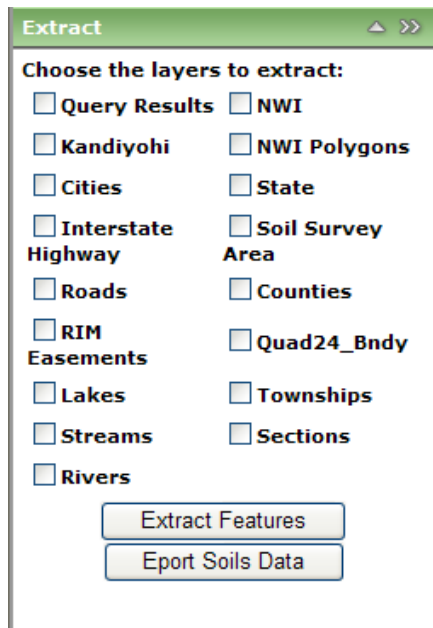


Figure 14. Extract Tools Menu.

### Soils Export Tools

When the user chooses a soil map area to be added to the map, a check box is added to the Extract Tool list with the name of the map area as a label. The user then checks the

box and clicks the Extract Features HTML button. The system then exports a shapefile of the features visible in the map that is in the layer. When the process is complete, the shapefile is added to a zip file and a link to that file is added to the results window. The user can download the file and extract it to a folder. The shapefile can then be used in ArcMap.

There are two other types of soil extracts that can be performed: Extract Soils Tables and Extract Query Results (Figure 15). Extract Soils Tables allow users to extract all the soil tables in the database related to the current soil map survey. The second tool allows for the extraction of query results. The tool exports a shapefile of the results in the map and the related table that holds the results information including labels. See section Extract Query Results for more information.

### *Extract Soils Tables*

When the user clicks the Export Soils Data html button, a JavaScript function configures the AJAX procedure that starts the export process. The system exports every table entry for all the features visible in the current map of the added soils map area. Each table is saved as a csv file delimited by a semicolon. Once all the tables are created they are zipped into a uniquely named zip file and a link is added to the results table.

Once the tables are downloaded, they can be extracted to a folder and opened using Excel or any other database system compatible with delimited files.

### *Extract Query Results (canned or map builder)*

Refer to Appendix C for a more detailed description of the workings of this tool.

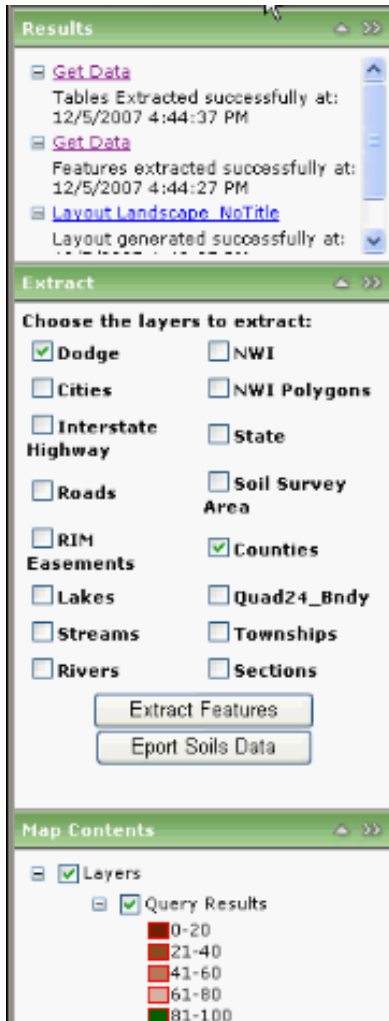


Figure 15. Extract soils tables and Extract Features results along with the extract tool and the Query Results Layer example.

## Conclusion

This paper explains the functionalities of an end product. The creation of the final product required large amounts of data and many applications. During the process of creating the individual pieces of the application the author developed experience with the language (vb.net) in the setting of the web development process, including best practices. The design required database functionality to join the data with spatial information.

In addition, time was spent learning SQL Server 2005 to create the necessary database tools required. The integration of other applications and creation of the Query Builder made the application user friendly. It also made it possible for internal BWSR support of the application by someone who did not know how to create procedures in a SQL Server database.

## Acknowledgements

I would like to thank Lane Urtel for help. This project would have been much more difficult without his experience and support. I would also like to thank the experts that answer ESRI support forum questions. Without all the anonymous information and experiences shared on those many pages, this application would have taken much longer to develop and the final product would not have been as robust.

Appendix A. Soils Database Data Dictionary.

**LEGEND - 1 Table**

Field Name	Brief description	Instances per Record	General Data Type	SSURGO Table
Lkey	Legend Key	1	Text (varchar)	Legend
Areaacres	Area Acres	1	Number	Legend
Areaname	Area Name	1	Text (varchar)	Legend
Areasymbol	Area Symbol	1	Text (varchar)	Legend
Areatypename	Area Type Name	1	Text (varchar)	Legend
Cordate	Correlation Date	1	Date/Time	Legend
Legendcertstat	Legend Certification Status	1	Text (varchar)	Legend
Legenddesc	Legend Description	1	Text (varchar)	Legend
Legendsuituse	Geographic Applicability	1	Text (varchar)	Legend
Mlraoffice	MLRA Office	1	Text (varchar)	Legend
mouagencyresp	MOU Agency Responsible	1	Text (varchar)	Legend
Projectscale	Project Scale	1	Number	Legend
Ssastatus	Survey Status	1	Text (varchar)	Legend
ssurgoarchived	SSURGO Archived	1	Date/Time	Legend
isComp1_3	Saves Yes/No if table exists	1	Boolean	Created by update service
isComp1_3	Saves Yes/No if table exists	1	Boolean	Created by update service
isHoz01_6	Saves Yes/No if table exists	1	Boolean	Created by update service
isHoz07_12	Saves Yes/No if table exists	1	Boolean	Created by update service
isHoz13_18	Saves Yes/No if table exists	1	Boolean	Created by update service
isHoz19_24	Saves Yes/No if table exists	1	Boolean	Created by update service
isHoz25_30	Saves Yes/No if table exists	1	Boolean	Created by update service
isHoz31_36	Saves Yes/No if table exists	1	Boolean	Created by update service
<b>Culmns per table:</b>		<b>22</b>		

**MAPUNIT – 1 Table**

Field Name	Brief description	Instances per Record	General Data Type	SSURGO Table
Mukey	Map Unit Key	1	Text (varchar)	mapunit
Lkey	Legend Key	1	Text (varchar)	mapunit
Muacres	Map Unit Acres	1	Number	mapunit
Muname	Map Unit Name	1	Text (varchar)	mapunit
Musym	Map Unit Symbol	1	Text (varchar)	mapunit
aws0150wta	Available Water Storage 0-150 cm Weighted Average	1	Decimal	muaggatt
aws025wta	Available Water Storage 0-25 cm Weighted Average	1	Decimal	muaggatt
Brockdepmin	Bedrock Depth - Minimum	1	Decimal	muaggatt
Drclassdcd	Drainage Class - Dominant Condition	1	Text (varchar)	muaggatt
Drclasswetest	Drainage Class – Wetttest	1	Text (varchar)	muaggatt
Engslgcd	ENG - Sewage Lagoons - Dominant Condition	1	Text (varchar)	muaggatt
Engslgcdp	ENG - Sewage Lagoons - Dominant Component	1	Text (varchar)	muaggatt

Engstafded	ENG - Septic Tank Absorption Fields - Dominant Condition	1	Text (varchar)	muaggatt
Engstafll	ENG - Septic Tank Absorption Fields - Least Limiting	1	Text (varchar)	muaggatt
Engstafml	ENG - Septic Tank Absorption Fields - Most Limiting	1	Text (varchar)	muaggatt
Flodfreqdcd	Flooding Frequency - Dominant Condition	1	Text (varchar)	muaggatt
flodfreqdmax	Flooding Frequency - Maximum Condition	1	Text (varchar)	muaggatt
Hydgrpdc	Hydrologic Group - Dominant Conditions	1	Text (varchar)	muaggatt
Pondfreqprs	Ponding Frequency - Presence	1	Text (varchar)	muaggatt
slopegradcp	Slope Gradient - Dominant Component	1	Integer	muaggatt
slopegradwta	Slope Gradient - Weighted Average	1	Integer	muaggatt
wtdpannmin	Water Table Depth - Annual - Minimum	1	Text (varchar)	muaggatt
wtdpaprunmin	Water Table Depth - April - June - Minimum	1	Integer	muaggatt
Hydclprs	Hydric Classification - Presence	1	Text (varchar)	muaggatt
	<b>Culums per table:</b>	<b>24</b>		

#### COMPONENTS - 2 Tables

Field Name	Brief description	Instances per Record	General Data Type	SSURGO Table
Mukey	Map Unit Key	2	Text (varchar)	Mapunit
Cokey	Component Key	6	Text (varchar)	Component
Erokind	Erosion Indicator Kind - Listed by Component	6	Text (varchar)	Coerosionacc
Rvindicator	RV indicator - Listed by Component	6	Text (varchar)	Coerosionacc
hydriccriterion	Hydric Criteria - Listed by Component	6	Text (varchar)	Cohydriccriteria
Compkind	Component Kind	6	Text (varchar)	Component
Compname	Component Name	6	Text (varchar)	Component
compct_r	Percent MapUnit	6	Integer	Component
Corsteel	Risk for Steel Corrosion	6	Text (varchar)	Component
Drainagecl	Drainage Class	6	Text (varchar)	Component
Frostact	Potential for frost action	6	Text (varchar)	Component
Geomdesc	Geomorphic Description	6	Text (varchar)	Component
Hydgrp	Hydrological Group	6	Text (varchar)	Component
Hydricon	Hydric Condition	6	Text (varchar)	Component
Hydricrating	Component Hydric Rating	0	Text (varchar)	Component
Localphase	Local Phase	6	Text (varchar)	Component
map_r	Average amount of precipitation	6	Decimal	Component
Runoff	Run Off Class	6	Text (varchar)	Component
slope_r	Slope Gradient	6	Integer	Component
slopelenusle_r	Slope Length	6	Decimal	Component

Taxclname	Taxonomic Class Name	6	Text (varchar)	Component
<b>Columns per table:</b>		<b>58</b>		

### CHORIZON - Split into 6 tables

Field Name	Brief description	Instances per Record	General Data Type	SSURGO Table
Mukey	Map Unit Key - in every table	6	Text (varchar)	Mapunit
Cokey	Component Key - in every table	6	Text (varchar)	Component
aashind_r	AASHTO group index	30	Integer	Chorizon
awc_r	Available Water Capacity - Representative Value	30	Decimal	Chorizon
Chkey	Horizon Key - matches all lower records to component key	30	Text (varchar)	Chorizon
claytotal_r	Total amount of clay present	30	Decimal	Chorizon
ec_r	Electrical conductivity	30	Decimal	Chorizon
cec7_r	Cation Exchange Capacity - Representative Value	30	Decimal	Chorizon
frag3to10	Rock Fragments 3-10 - Representative Value	30	Decimal	Chorizon
fraggt10_r	Rock Fragments >10 - Representative Value	30	Decimal	Chorizon
hzdepb_r	Horizon Bottom Depth - Representative Value	30	Integer	Chorizon
hzdept_r	Horizon Top Depth - Representative Value	30	Integer	Chorizon
Hzname	Official Designation	30	Text (varchar)	Chorizon
ksat_h	Ksat - High Value	30	Decimal	Chorizon
ksat_l	Ksat - Low Value	30	Decimal	Chorizon
ksat_r	Ksat - Representative Value	30	Decimal	Chorizon
lep_r	Linear Extensibility -High Value	30	Decimal	Chorizon
ll_r	Liquid Limit - Representative Value	30	Decimal	Chorizon
om_r	Organic Matter - Representative Value	30	Decimal	Chorizon
ph1to1h2o_r	PH Soil Reaction - Representative Value	30	Decimal	Chorizon
pi_r	Plasticity Index - Representative Value	30	Decimal	Chorizon
sandtotal_r	Total amount of sand present	30	Decimal	Chorizon
silttotal_r	Total amount of silt present	30	Decimal	Chorizon
Texdesc	Horizon Texture Descriptions	30	Text (varchar)	Chtexturegrp
Texture	Horizon Texture Modifier and Class Concatenation	30	Text (varchar)	Chtexturegrp
Unifiedcl	Horizon Unified Classes	30	Text (varchar)	Chunified
<b>Columns per table:</b>		<b>122</b>		
<b>All soils data stats</b>				
<b>Maximum Number of Tables:</b>		<b>10</b>		
<b>Maximum Number of Columns:</b>		<b>998</b>		

### COLUMNINFO - 1 TABLE

Field Name	Brief description	Instances per Record	General Data Type	SSURGO Table
Tabphynname	Table Physical Name	1	Text (varchar)	Mdstattabcols
Colphynname	Column Physical Name	1	Text (varchar)	Mdstattabcols
Collogname	Cloumn Logical Name	1	Text (varchar)	Mdstattabcols
Collabel	Column Label	1	Text (varchar)	Mdstattabcols
Uom	Unit of Measurement	1	Text (varchar)	Mdstattabcols
Coldesc	Column Description	1	Text (varchar)	Mdstattabcols
Colprecision	Column Precision	1	Integer	Mdstattabcols

Columns per table: 7

**STOREDQUERY - 1 TABLE** (Table created by Stored Query Administrator )

Field Name	Brief description	Instances per Record	General Data Type	SSURGO Table
Sqkey	StoredQuery-key	1	Text(varchar)	N/A
Islinked	Boolean to determine if query is stand alone or part of a model.	1	Bit	N/A
linkChild	reference to next query in the model - link to physical	1	Text(varchar)	N/A
linkParent	reference to last query in the model - link to physical	1	Text(varchar)	N/A
linkOrder	What order the sequence should run (e.g. 3 would mean that the query should be run third)	1	Integer	N/A
sqname	Stored Query physical Name - link to physical location	1	Text(varchar)	N/A
<b>Columns per table:</b>		<b>6</b>		

**Soils\_Districts** (Copied from the soil districts shapefile)

Field Name	Brief description	Instances per Record	General Data Type	SSURGO Table
OBJECTID	Unique Identifier	1	Integer	Shapefile
SASTATCODE	Code that represents the data's availability 1 = Tabular Only 2 = Tabular and Spatial 0 = Unpublished	1	Integer	Shapefile
NAME	District Name Label	1	Text(varchar)	Shapefile
STATUS	Holds the download status of the feature 1st digit = STATATCODE 2nd digit = 1(Downloaded) or 0(Not Available)	1	Integer	Shapefile
PI	PI availability 1 = Yes 0 = No	1	Integer	Shapefile
<b>Columns per table:</b>		<b>5</b>		

**Mdstatdomdet** (Copied from Dodge SSURGO soils database - could be copied from any of the databases)

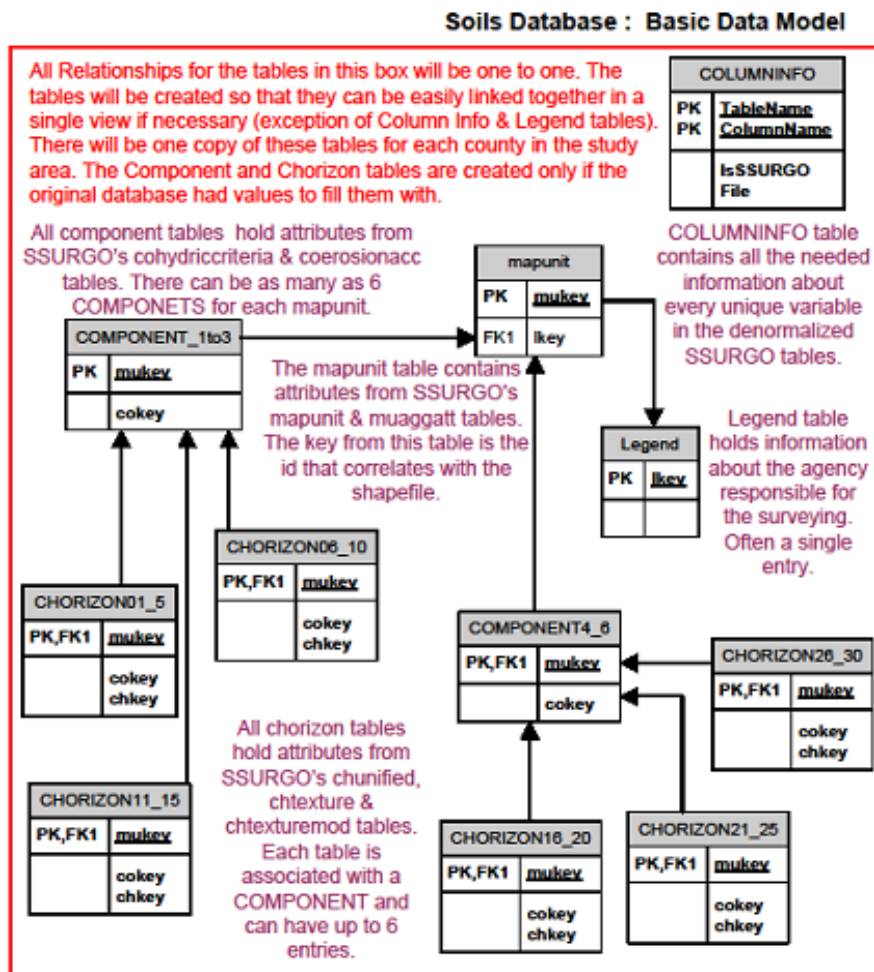
Field Name	Brief description	Instances per Record	General Data Type	SSURGO Table
Domainname		1	Text(varchar)	Dodge County - mdstatdomdet
Choice	Choice	1	Text(varchar)	Dodge County - mdstatdomdet
choicesequence	Choice Sequence	1	Integer	Dodge County - mdstatdomdet
Choicedesc	Choice Description	1	Text(varchar)	Dodge County - mdstatdomdet
Choceobsolete	Lists if domain no longer recorded	1	Boolean	Dodge County - mdstatdomdet
<b>Columns per table:</b>		<b>5</b>		

**STORED\_QUERY\_LOG** (Stand Alone table created to store password/users for Admin site)

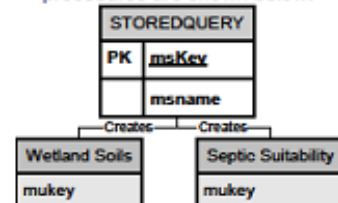
Field Name	Brief description	Instances per Record	General Data Type	SSURGO Table
IDField	Unique Identifier	1	Integer	Data Entry

User_Name	Login Name	1	Text(varchar)	Data Entry
User_Pass	Login Password	1	Text(varchar)	Data Entry
Can_Manage	Field set if user can delete a query	1	Text(varchar)	Data Entry
<b>Columns per table:</b>		<b>4</b>		

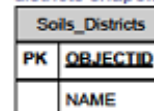
Appendix B. Soils Database Data Model.



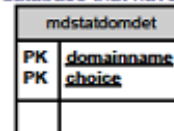
STOREDQUERY is a lookup table that holds information about queries (procedures) created using the AdminNRDSS application. Example procedures are shown below.



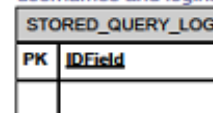
Soil\_Districts is a lookup table that holds information about the soils districts and is a copy of the attribute table of the soils districts shapefile.



mdstatdomdet is a lookup table from the Dodge County SSURGO database. This table lists the domains for all columns in the database that have them.



STORED\_QUERY\_LOG is a look up table that holds the AdminQuery apps usernames and logins.



Appendix C. Web Application Images.

**Extract Query Results**

Tool allows users to export Query Results as a zipped shapefile.

Extract Features

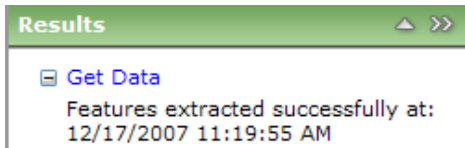
Click this box to retrieve features data.

Warnings: You must have a soils layer added to the map. Even if soils layer not visible it will extract data for soils features in the map. This action can take a while so be careful before resubmitting request.

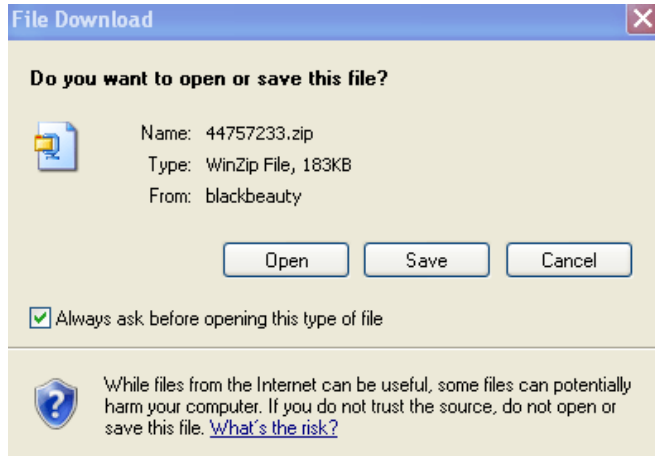
Wait until file link is shown in **Results** section of the Console.

Click on the Get Data link.

Click on save and follow the wizard instructions.

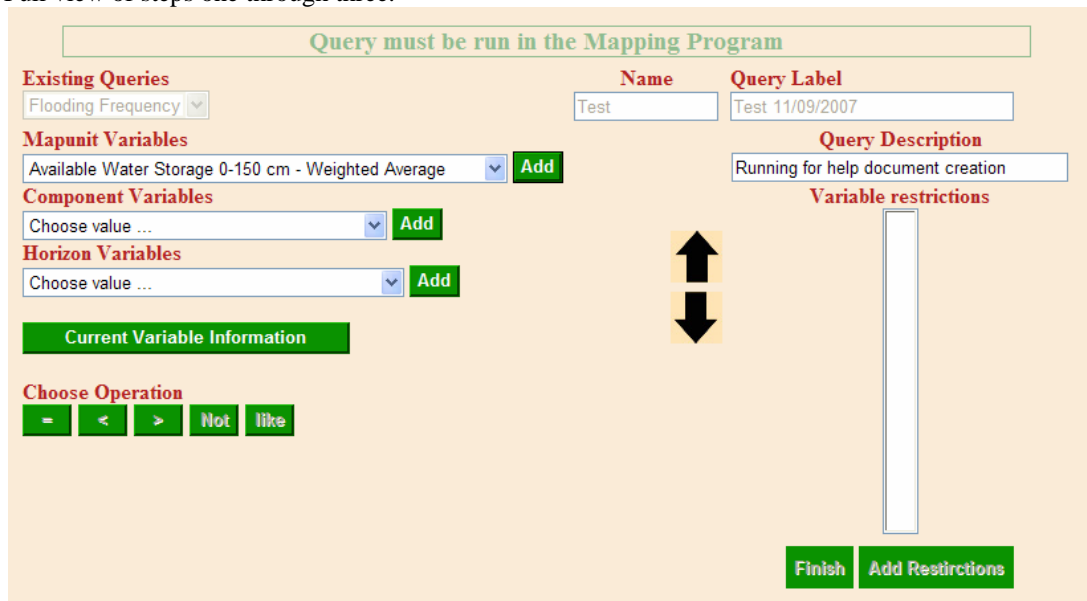


In order to view the query result in ArcMap you will need to extract them to a folder first. The Query Results Table is saved using a ; as the column delimiter. Open the file in Excel and export the table as a dbf4 document. Open ArcMap and add the Query Results shapefile and the Query Results dbf4 file. Create a spatial join between the two Mukey (shapefile) and MKEY (table). Export the shapefile from ArcMap and it will include all the information from both the Table and the shapefile.



### Create Query

Full view of steps one through three.





The button Current Variable Information returns a description of a variable includes the data type and unit of measurement.

**Available Water Storage 0-150 cm - Weighted Average : Numeric**

Unit of Measurement : centimeters

Available water storage (AWS). The volume of water that the soil, to a depth of 150 centimeters, can store that is available to plants. It is reported as the weighted average of all components in the map unit, and is expressed as centimeters of water. AWS is calculated from AWC (available water capacity) which is commonly estimated as the difference between the water contents at 1/10 or 1/3 bar (field capacity) and 15 bars (permanent wilting point) tension, and adjusted for salinity and fragments.

[Close Description](#)

## Map Builder

Allows user to search soils attributes for soils features in the map.

Window opens when [Desc ?](#) Clicked

**Current Unit Type**  
Choose Table... ▾

Choose Unit Type:  
MapUnit  
Component  
Horizon

**Current Variable Name**  
Choose Variable ... ▾

Choose variable that you wish to restrict the results by.

[Desc ?](#)

Numbers in image shown above

1. Variable's full name
2. Variable Data Type
3. Unit of Measurement
4. Variable Description

[Next Step](#)

Click **Next Step** to enter value or restriction.

The Variables data type will determine how the variable's restriction will be entered.

Entry Type #1

Entry Type #2

Entry Type #3

**Restriction Entered**

MAPUNIT		slopegradwta		0 to 3
1		2		3

1. Variable is **Textual**  
 Enter Text  
 Click **Next Step**
2. Variable is **Numeric**  
 Enter lowest value in **From** box  
 Enter highest value in **To** box  
 Click **Next Step**
3. Variable has a **Domain**  
 Choose domain value  
 Click **Next Step**

If you have more than one domain in the results list the results returned will contain features with each domain entered.

Click **Next step** to add the restrictions to the list.  
 Click **Go Back** to choose a different variable.

The restrictions once added have format:

1. Unit Type restriction from
2. Short name if variable
3. Variable restriction entered

Click **Run Query** to get results.  
 Click **Clear List** to start a new set of Restrictions.

### Canned Query

Tool allows users to run a predefined Query on the soils features in the map.

**Run Canned Query** ?

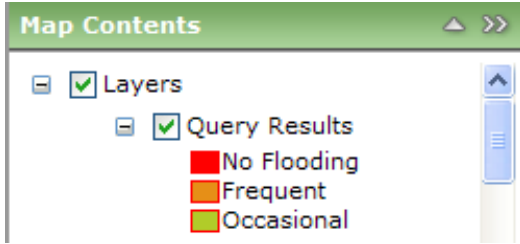
**Choose Query**

\*Verify soils layer has been added and visible prior to using tool

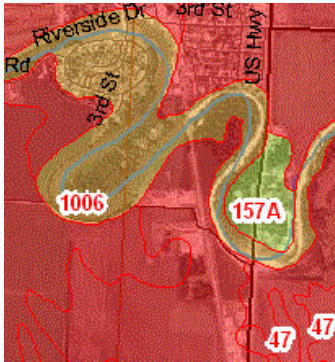
**Choose Query**

Choose a query from the list  
 Click Run Query

**Example:** If you choose Flooding Frequency



**Map Contents:** Shows which colors have what label.



**Map:** Each feature is colored according to the Label from the **Map Contents** that it corresponds to.

**Choose Query**

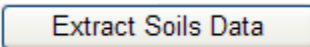
Hydric Report

**If you choose Hydric Report:** A new window will open showing the report for the entire soil survey area currently added to the map.

You may need to have pop up blocker off to view the document.

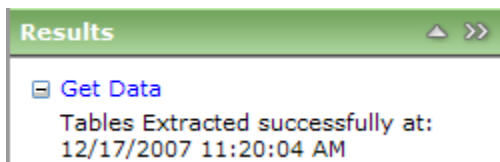
### Extract Soils Tables

Tool allows users to export a version of the soils information that has been added to the map. The tables are exported out as CSV and can be imported into most database systems.



Click this box to retrieve soils data for the soils features in the map.

Warnings: You must have a soils layer added to the map. Even if soils layer not visible it will extract data for soils features in the map. This action can take a while so be careful before resubmitting request.



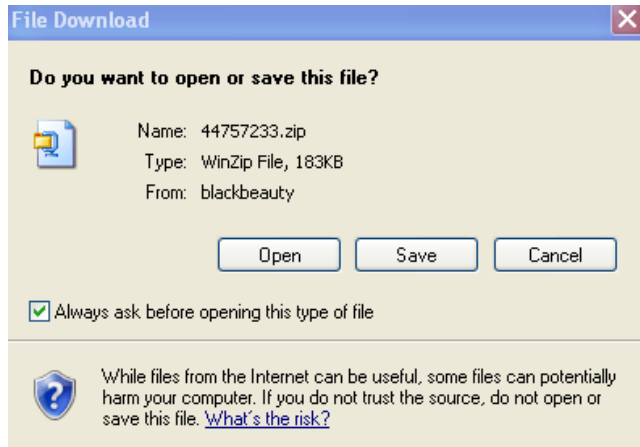
Wait until file link is shown in Results section of the Console.

Click on the Get Data link

Click on save and follow the wizard instructions.

In order to view the files in excel you will need to extract them to a folder first.

Each file is saved with using a ; as the column delimiter.



## Main Website

