



A Simplified Web Interface Manager for Building Information Model Management

Final Report of SWIM Phase 1



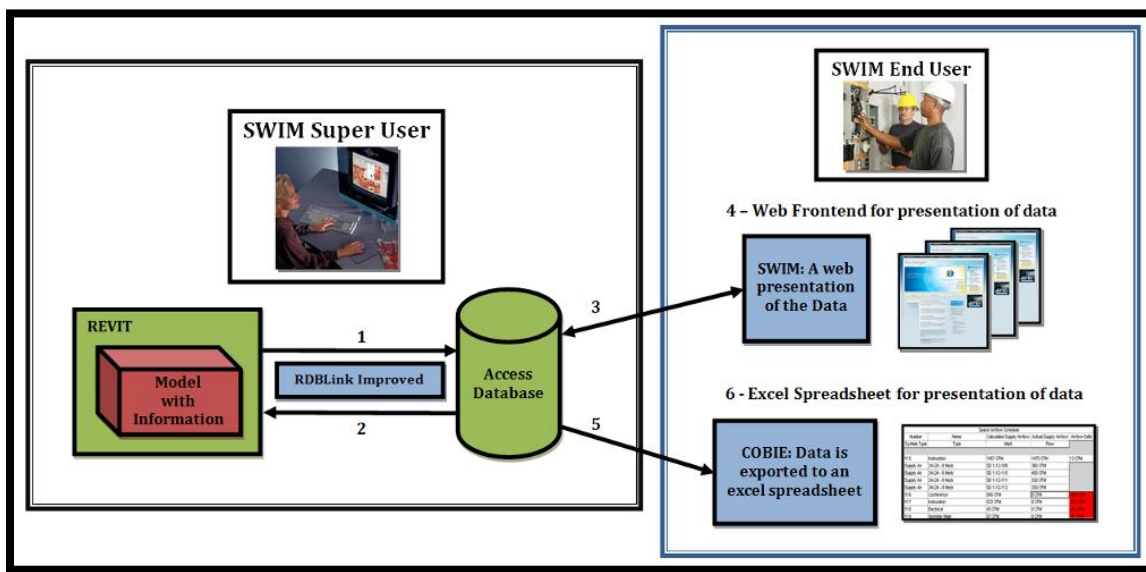
**Lacey Duckworth
Project Manager
University of Southern Mississippi
Lacey.Duckworth@eagles.usm.edu
601-266-4894**

Index

Executive Summary	1
Chapter 1: Project Overview	2
Chapter 2: Basic User Introduction to SWIM	3
Chapter 3: Setting Up and Running SWIM	4
Chapter 4: Full User Guide to SWIM	5
Appendix A: Super User Code	7
Appendix B: End User Code	8

Executive Summary

The University of Southern Mississippi team collaborated with Edward Huehl of the US Army Corps of Engineers to develop a simplified web interface for retrieving and modifying BIM information extracted from a REVIT model database. In developing this web interface, two types of users were identified, the SWIM Super User and the SWIM End User. The purpose of the Super User is to set up the web front end for the End User. Thus, the Super User needed access to the REVIT model, the ability to push the REVIT information into a database, the ability to create the web front end for the end user, and the ability to convert the Information in the database about a REVIT model into a COBie Spreadsheet format. The End User on the other hand only needed access to the home screen of the web front end so that they could preview information about the REVIT model, preview and manipulate this information, preview the COBie Spreadsheet, and save the changes back to the database. A diagram illustrating the roles of the Super and End User can be seen in the figure below.



The following is a summary of major implementation issues in the delivered product:

- The base software package needs to be downloaded and unzipped from [here](#). This file already contains a usable version of the SWIM front end connected to a database.
- Only the Internet Explorer web browser is supported.
- The model has to exist in REVIT 2010.
- RDB Link inside of REVIT has to be updated with the code provided in Chapter 3.
- Excel 2007 is has to be installed on the super and end user computer so the super user can export the REVIT model information into the COBie format and the end user can preview it form their machine.
- Access 2007 has to be installed on the super and end users machine in order to maintain the database.
- ODBC Connection is required for connecting to the database.
- .NET framework has to be installed on the super user machine which can be found [here](#).

SWIM was completed on time, November 12, 2010, and under budget. This final report, software and source code provided here was delivered to Edward Huell on November 12, 2010. The USM team members who helped complete the project with titles include the following:

- Lacey Duckworth, Project Manager
- Julie Alexander, Web Design and Database Code
- Andrew Cordar, COBie Formatter and Super User Developer
- Ila Higginbotham, Web Design and User Manual
- Adam Seyfarth, Database Code

Chapter 1 - Project Overview

Background

Construction Operations Building Information Exchange (COBie) is a rapidly evolving standard to capture information electronically during design and construction, to provide it to facility managers. COBie eliminates the current inefficient process of transferring massive amounts of paper documents (such as manuals, warranties and usage guides) to facility operators. COBIE eliminates the need for post-hoc as-built data capture and helps to reduce operational costs. .

This electronic document format is intended to smooth the closeout process by providing a repository for incremental data collection from the beginning of building design through the various phases of construction. This information is essential to support the operations, maintenance, and the management of the facilities assets by the owner and/or facility manager.

Following Construction Closeout the COBie document then forms a live data set for continued use by facility managers to find access and update relevant maintenance documents. One method for accumulating COBie data is to annotate Building Information Models (BIM) documents which encompass both the architecture of a structure but also the properties of each sub-system such as doors, heating and cooling units and fixed electronics such as projectors. BIM documents are an attractive target for accumulating COBie data but BIM software such as AutoDesk REVIT is complex and beyond the expertise of most construction and facility managers. This barrier to wider use of COBie can be overcome by proving simplified interfaces for non-expert users to search, access, modify and add data to the underlying BIM dataset.

Objectives

The *S.W.I.M. – A Simplified Web Interface Manager for Building Information Model Information Management – Phase 1* will provide ERDC CAD-BIM staff with a new standalone browser-based tool which will allow non-expert users to access and modify data previously added to BIM electronic documents. These changes can then be easily propagated back to the BIM document by re-importing the underlying BIM database.

This 3-month summer project (for 2010 fiscal year) will be a collaborative effort of The University of Southern Mississippi Schools of Computing and School of Construction with the U.S. Army Engineering Research and Development Center (ERDC) CAD/BIM Technology Center.

The project tasks will be completed by USM students with the guidance of two professors one from the School Computing and one from the School of Construction following the specifications to be provided by ERDC staff.

Task List

S.W.I.M. Phase 1 will use an incremental delivery strategy to rapidly progress from concept, to specification, to delivery of intermediate products to delivery of a final product. The following are the four sub-tasks to complete the project:

Sub-Task 1 - Test Cases and Software Requirements: In order to deliver a simplified web interface which meets the current USACE needs the USM faculty and students will work with ERDC staff to identify motivating examples. This interaction with ERDC staff and the repository of examples will help the

USM team to identify the specific functionality needed by the standalone web interface. In addition the development team will consider the specific security needs and information assurance environment in which the completed software will run.

Sub-Task 2- Detailed Module Specification: This Sub-task will focus on developing the detailed module specifications for the S.W.I.M. Phase 1 to meet the exact needs of the USACE. The specification will be structured using the following four elements:

1. Web Interface “Look and Feel” with proposed presentation technology,
2. Database Connectivity Layer with proposed connectivity technology,
3. UML Use Case diagrams showing all anticipated user interaction with the interface

Sub-Task 3 - Development and Testing of Intermediate Version of Module: The detailed specifications developed in Sub-Task 2 will be used as a guide to create a first working version S.W.I.M. phase 1 using .NET with the Microsoft Visual Studio development environment and MS ACCESS database examples drawn from the motivating examples collected in Sub-Task 1.

This intermediate version of the module will provide the most basic functionality described in the specification (such as search, edit, highlight changes, etc.) and will provide a working prototype of the more advanced features (such as history, change logs, query and replace, etc.). Sub-task 3 will conclude with a presentation by the project students delivered in Vicksburg at the CAD/BIM center. The purpose of this review meeting is to provide the USM team with feedback on the usability of the prototype web interface, adherence to specification, compliance with easy installation requirements and other features at an intermediate stage where fine tuning can still be easily accomplished.

Sub-Task 4 - Development and Delivery of Final Web Interface: With the feedback provided in Sub-task 3 the USM team will finalize the coding of the web interface and will then proceed with the delivery process as follows:

1. review of code for completeness of in-line documentation
2. development of command-line and usage summary document in Adobe Pro Acrobat Extended
3. delivery of product via mail, e-mail or ftp

availability via e-mail or by teleconference to provide live help as necessary to insure the successful handoff of the completed product.

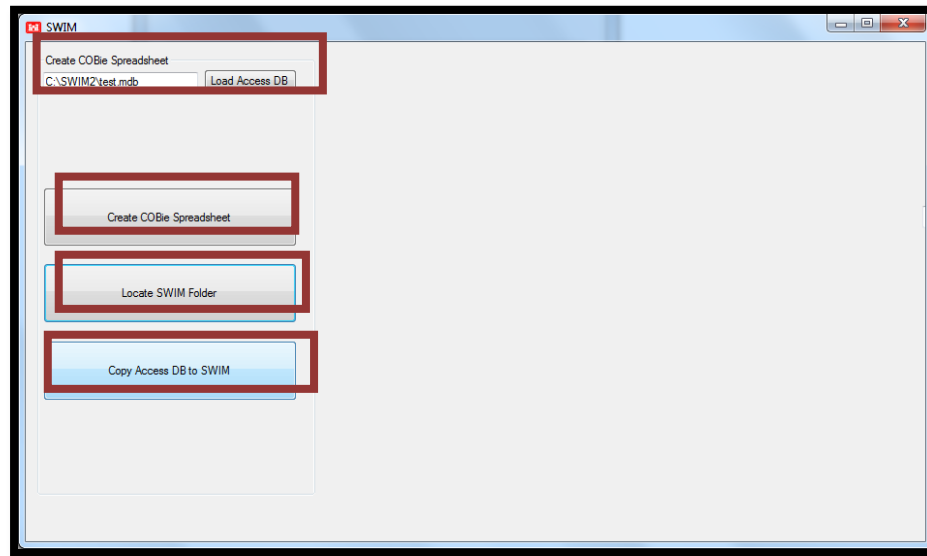
Summary of Deliverables

The USM team will deliver the following: (1) a web interface coded in .NET which connects to an exported REVIT database in MS ACCESS and provides necessary functionality for non-expert users to modify COBie data annotations, (2) a workflow description for combining the elements of the web interface and the exported REVIT database as a single executable unit which can be run with no additional modules and without install privileges on typical information assurance regime government or military computer systems, (3) help files which assist the target user in utilizing the delivered product, (4) assistance in using the new software for the first time, and (5) a final report summarizing all activities.

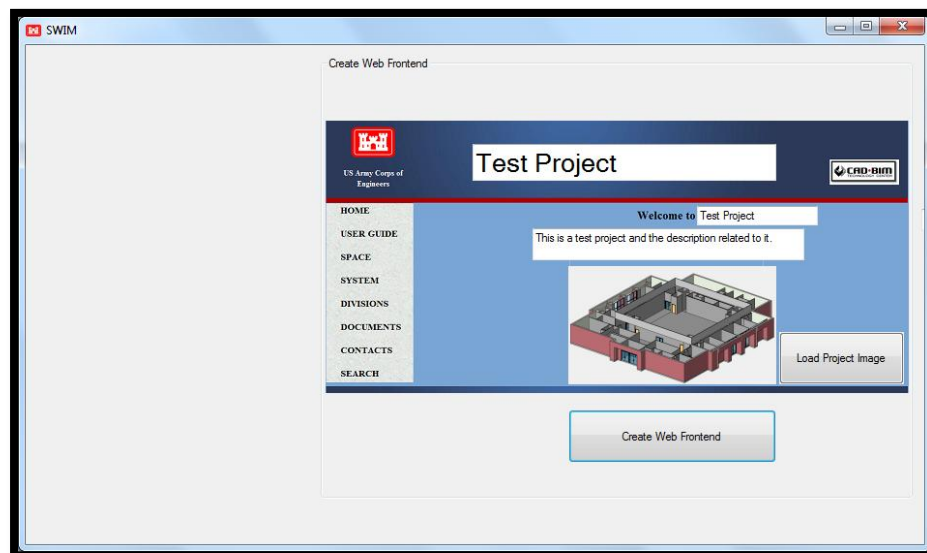
Chapter 2: Basic User Introduction to SWIM

There are two types of users that can interact with SWIM, the super user and the end user. The purpose of the super user is to transfer the information about all objects associated with a project in REVIT into a format that end users can easily find and manipulate the information. The format created by the super user is a web front end that can be created using the Super User Graphical User Interface (GUI).

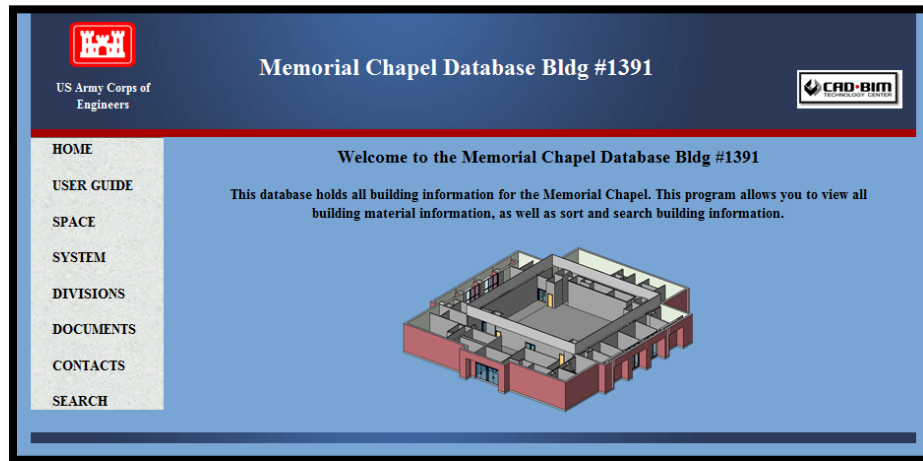
Below is the GUI that the super user interacts with in order to set up the web front end for the end user. This GUI is a step by step process as only the red boxes show up at once each step is complete. From top to bottom, the first box loads the access database file for creating the COBie Spreadsheet, the second button creates the COBie Spreadsheet, and the third allows the super user to locate the SWIM file.



that has the standard files for the web front end, and the fourth button copies the database specified in the first button to the SWIM folder and allows the super user to personalize the web front end by using another GUI below.



Once the super user has filled in this information and selected the “Create Web Frontend” button, the end user's web front-end has been created and looks like the figure seen below.



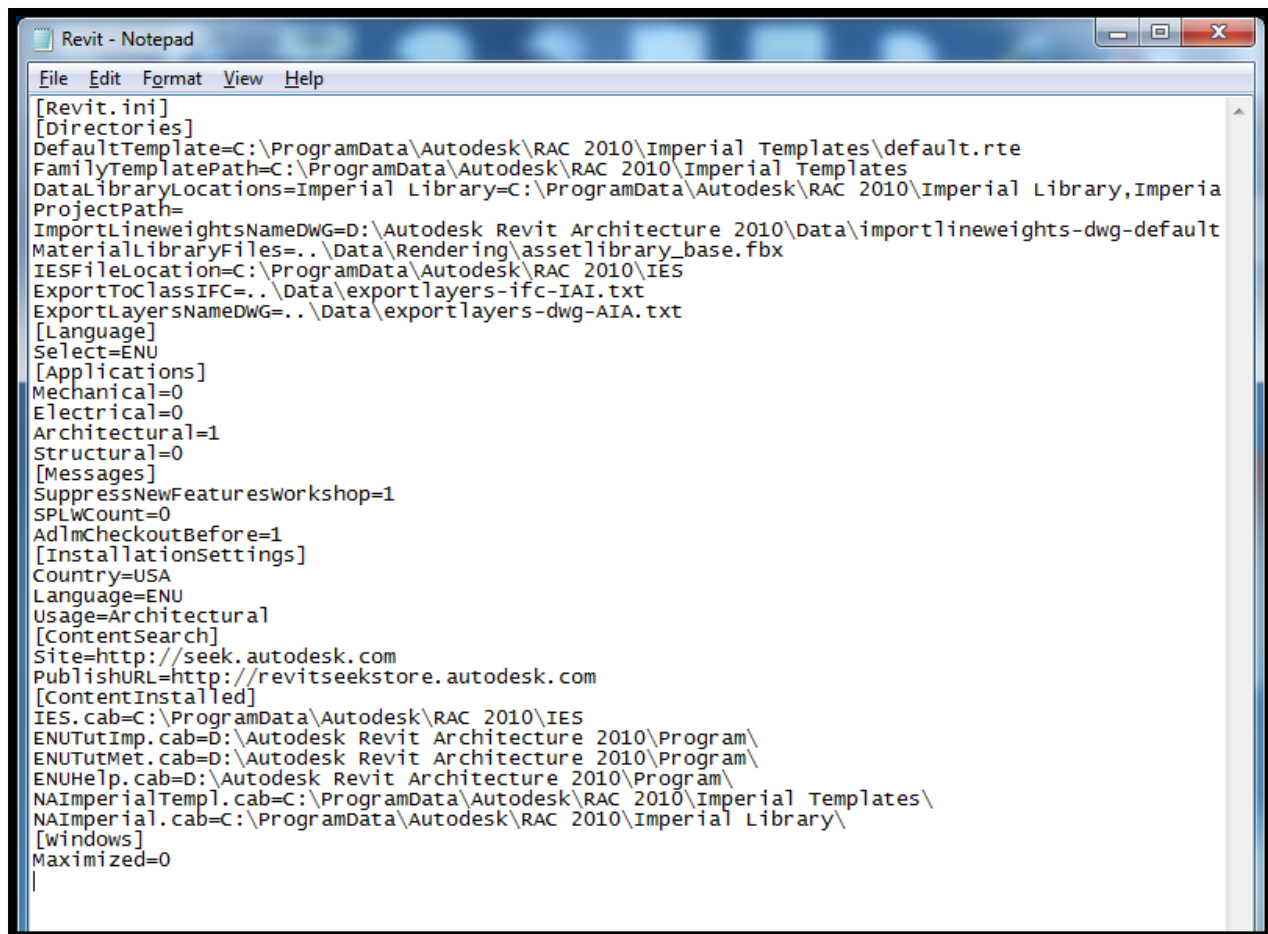
Chapter 3 provides a more detailed explanation of how to set up SWIM for both the super and end user.

Chapter 3: Setting Up and Running SWIM

Setting up SWIM only requires setting up the super user because setting this up develops the end user program. Before the setup for the super user can begin, there are several pieces of software that need to be installed on the computer. This software includes Internet Explorer, Microsoft Access, Microsoft ODBC, Microsoft Excel, REVIT 2010, and the .NET framework, which can be downloaded [here](#). Once this software is installed, the process is to fix the RDBLink exporter which handles exporting the database from REVIT, using the RDBLink to export the database, completing the super user GUI and running the SWIM program

Fix RDBLink

In order to install the RDBLink Add-in, the user first needs to locate the REVIT Application Folder and go to the Program folder. It will look something like “Autodesk REVIT Architecture 2010->Program”. Once inside this folder the Revit.ini file needs to be located and opened in notepad. The file should look similar to the figure below.



```
[Revit.ini]
[Directories]
DefaultTemplate=C:\ProgramData\Autodesk\RAC 2010\Imperial Templates\default.rte
FamilyTemplatePath=C:\ProgramData\Autodesk\RAC 2010\Imperial Templates
DataLibraryLocations=Imperial Library=C:\ProgramData\Autodesk\RAC 2010\Imperial Library,Imperia
ProjectPath=
ImportLineWeightsNamedDWG=D:\Autodesk Revit Architecture 2010\Data\importlineweights-dwg-default
MaterialLibraryFiles=..\Data\Rendering\assetlibrary_base.fbx
IESFileLocation=C:\ProgramData\Autodesk\RAC 2010\IES
ExportToClassIFC=..\Data\exportlayers-ifc-IAI.txt
ExportLayersNamedDWG=..\Data\exportlayers-dwg-AIA.txt
[Language]
Select=ENU
[Applications]
Mechanical=0
Electrical=0
Architectural=1
Structural=0
[Messages]
SuppressNewFeaturesWorkshop=1
SPLWCount=0
AdlmCheckoutBefore=1
[InstallationSettings]
Country=USA
Language=ENU
Usage=Architectural
[ContentSearch]
Site=http://seek.autodesk.com
PublishURL=http://revitseekstore.autodesk.com
[ContentInstalled]
IES.cab=C:\ProgramData\Autodesk\RAC 2010\IES
ENUTutImp.cab=D:\Autodesk Revit Architecture 2010\Program\
ENUTutMet.cab=D:\Autodesk Revit Architecture 2010\Program\
ENUHelp.cab=D:\Autodesk Revit Architecture 2010\Program\
NAImperialTemp.cab=C:\ProgramData\Autodesk\RAC 2010\Imperial Templates\
NAImperial.cab=C:\ProgramData\Autodesk\RAC 2010\Imperial Library\
[Windows]
Maximized=0
|
```

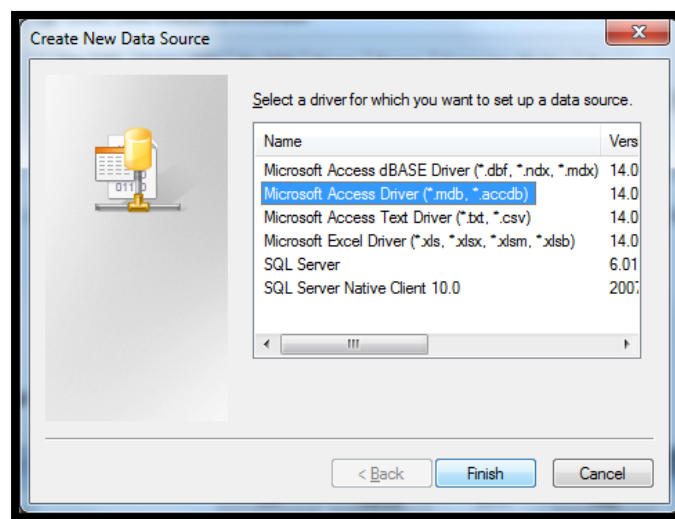
At the end of the file, the following information needs to be added:

```
[ExternalCommands]
ECCount=1
ECName1=RDBLink
ECClassName1=Revit.SDK.Samples.RDBLink.CS.Command
ECAssembly1=(Location of RDBLink.dll file)
```

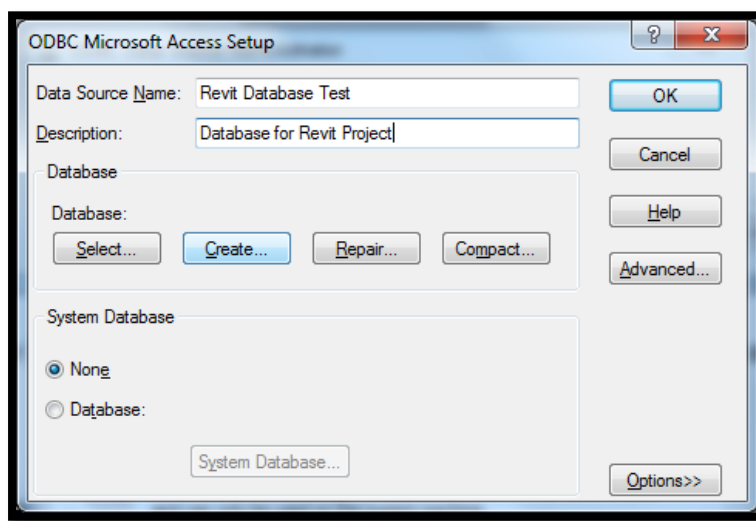
Once this information is added, the file should be saved and REVIT opened to confirm the add-in was added. To verify it was added, this tool should be located in the Add-Ins tab in REVIT.

Using RDBLink to Export the Information

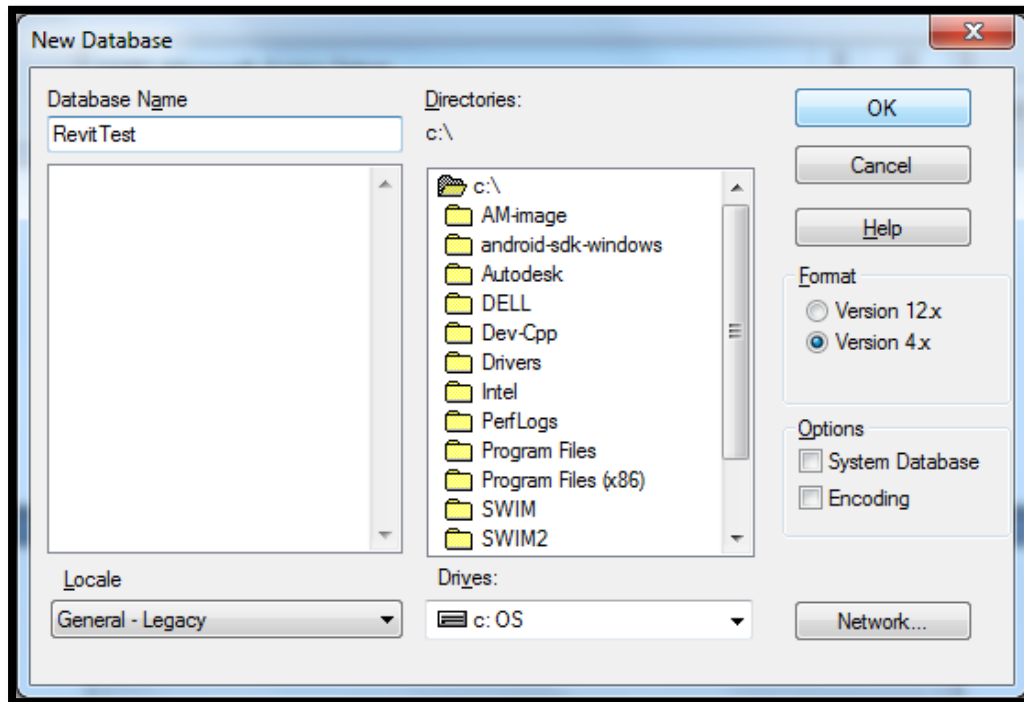
After the RDBLink is installed, the user needs to create a data source (“MS Access Database”) using “Data Sources (ODBC)” located in Control Panel -> System and Security -> Administrative Tools. The user should click “Add”, select “Microsoft Access Driver”, and click “Finish”.



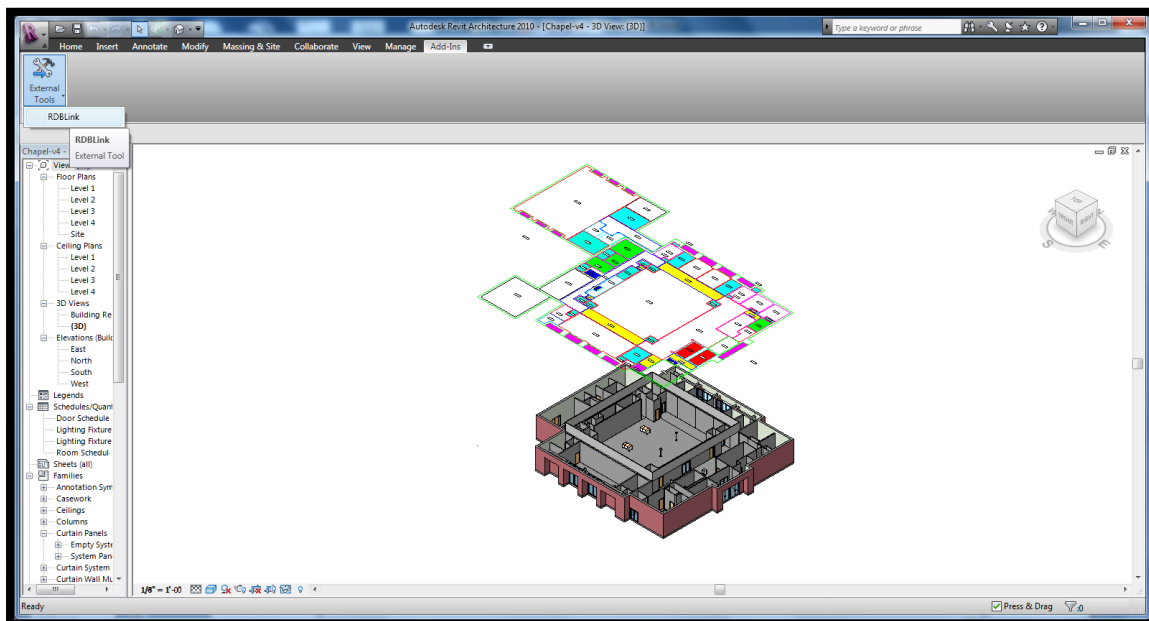
Next, the user will give the Data Source a name and description, and click “Create”



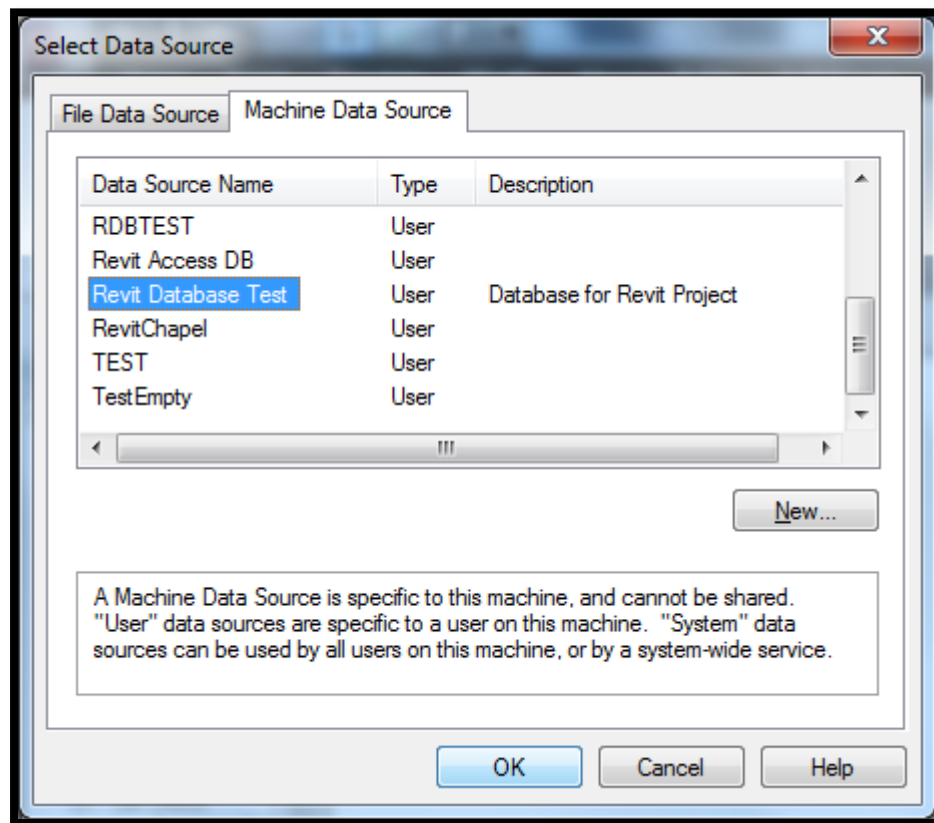
Then, the user will specify a name and location for the file and click “Ok”



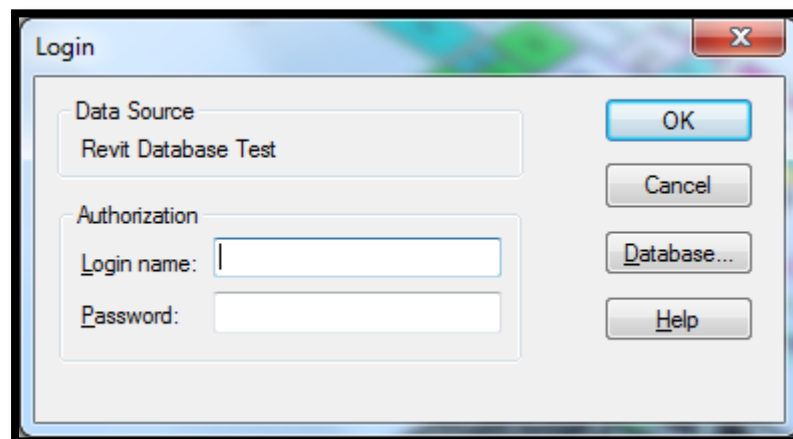
The next step is to open REVIT and load a REVIT project file that is desired to be exported to a database. In order to do this, the user will click “Add-Ins” -> “External Tools” -> “RDBLink”



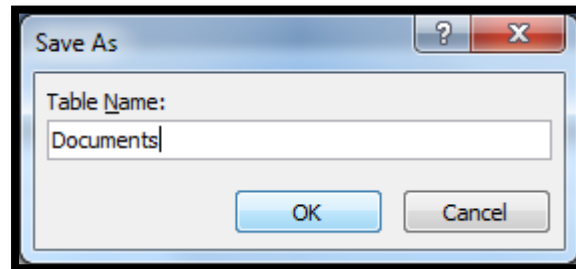
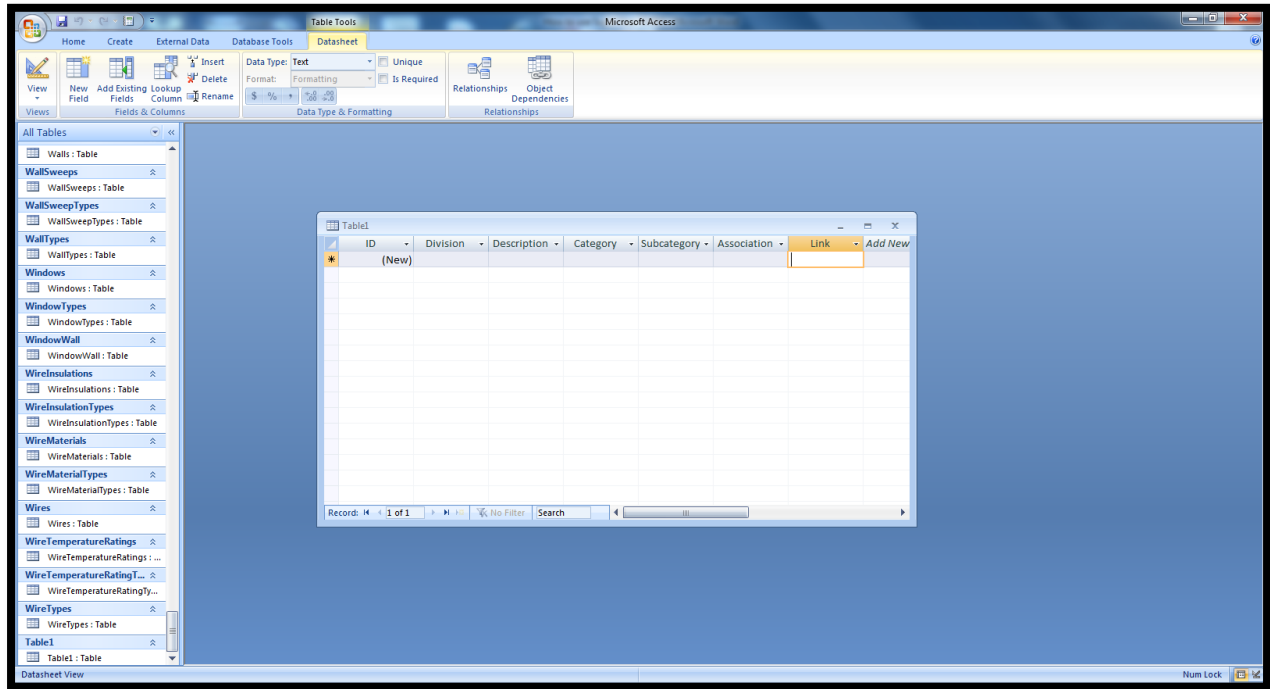
Then they should click “Export into ODBC Database”, click the “Machine Data Source” tab, and select the database you created in the previous steps then click “Ok”.



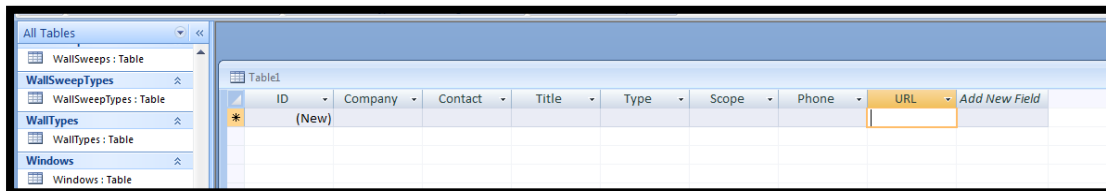
If the database is password protected, the user will enter the password, otherwise they will click “Ok” in the next screen. The REVIT Project will export and the database will successfully be populated.



There is some information that cannot be exported from the project, so the user will have to open the newly created Access Database and create a new table called “Documents” with the fields seen in the figure below and save the file as “Documents”.



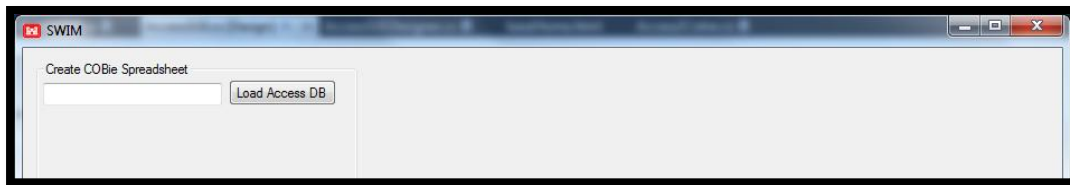
Because Contacts is not within the REVIT model, a contacts table will have to be created with the fields as seen in the figure below and saved as “Contacts”.



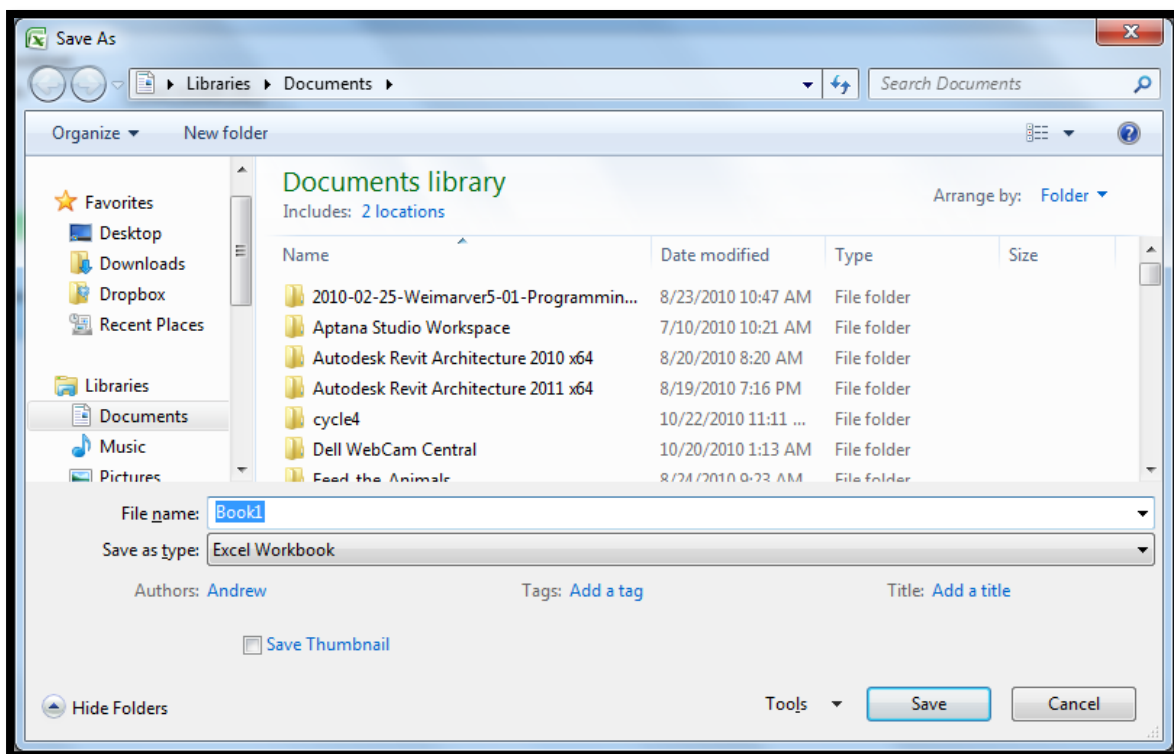
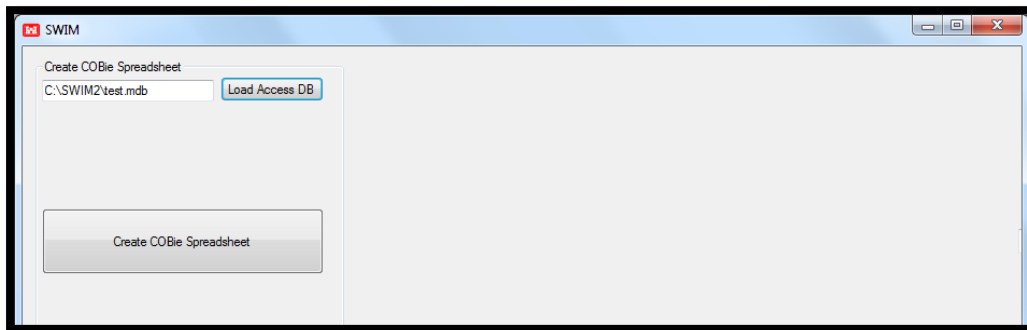
Completing Super User GUI

Once the database has been created, the Super user will open SWIM.exe located in the Super User GUI folder in order to set up the COBie Spreadsheet and web front end for the end user. To do this,

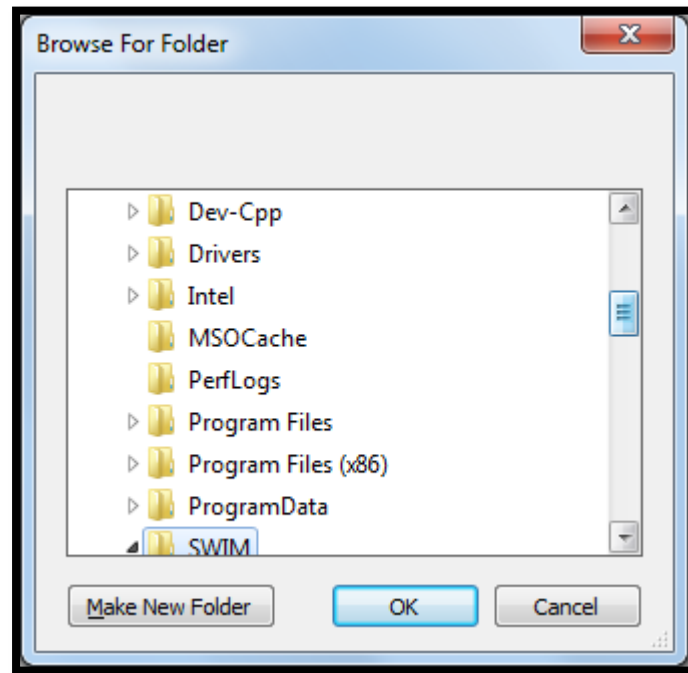
the super user will first locate the Access Database used for the project by clicking on Load Access DB and selecting the path of the database.



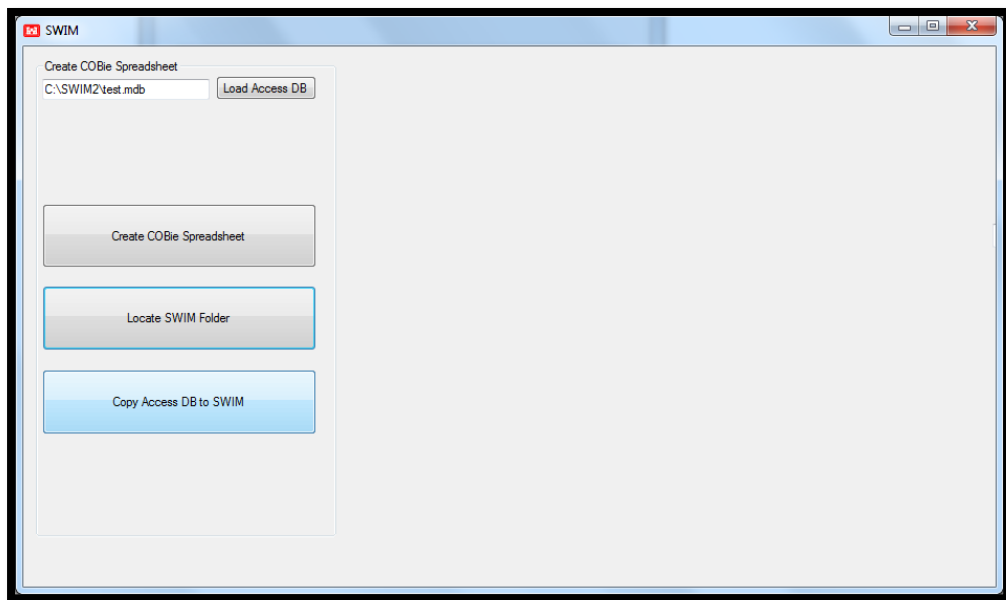
By clicking “Create COBie Spreadsheet” and providing a name and location to store the COBie spreadsheet the file is created using the database selected using the “Load Access DB” button.



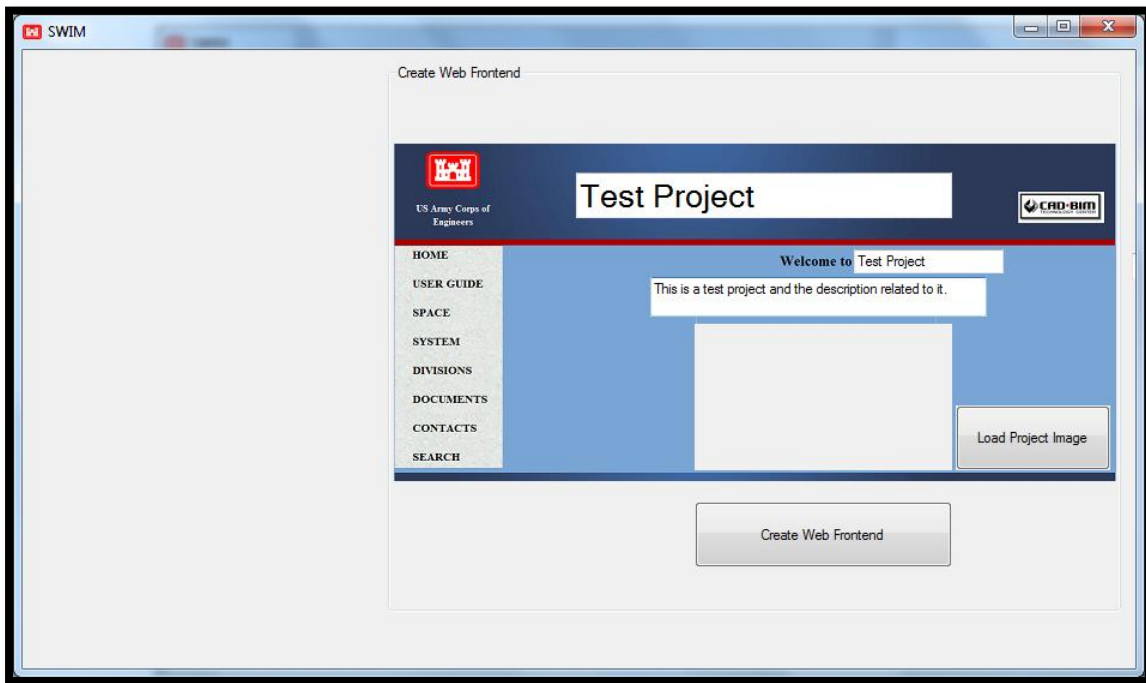
This will allow the user to locate the existing SWIM folder which contains the standard html files.



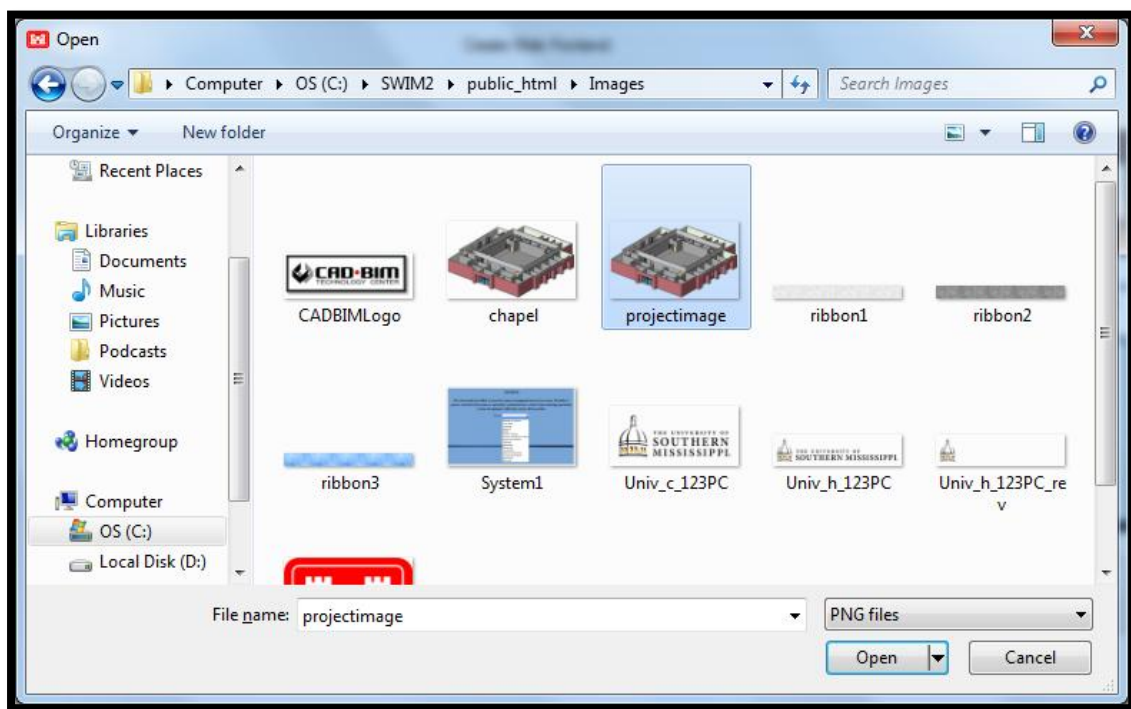
After selecting the SWIM folder and choosing OK, the user needs to click on the “Copy Access DB to SWIM” button



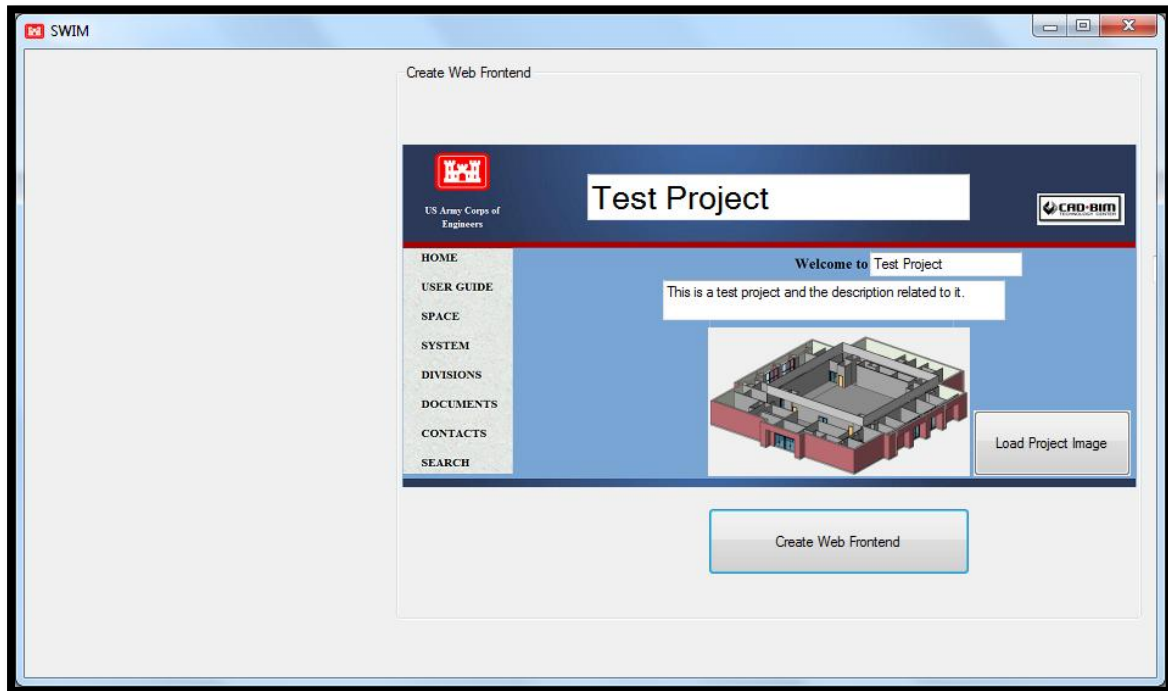
The end user can now fill out what they wish the front welcome screen for the project to look like.



The user can change the title of the welcome screen, the welcome message, as well as add an image of the project under consideration by using the “Load Project Image” button. Selecting this button will allow them to choose an image file and add to the welcome screen.



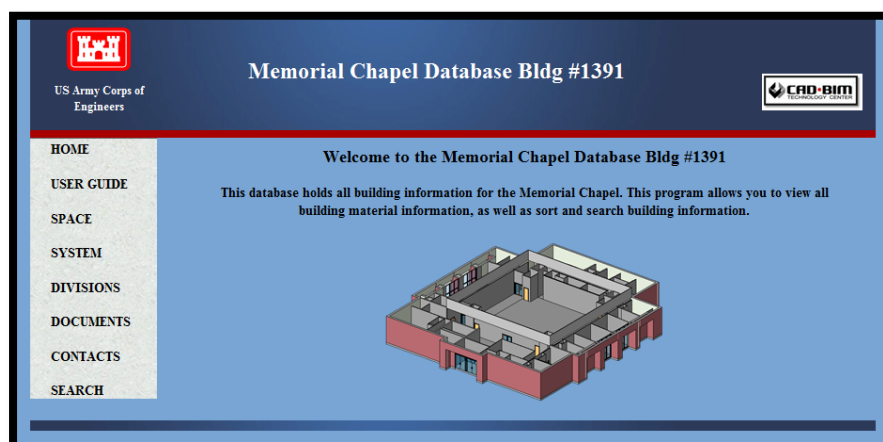
By selecting the “Create Web Frontend” the super user has chosen to publish the web frontend welcome screen for the end user.



Now, the super user has linked the information from a REVIT Model to a web front end they can either open the “Home.html” file in the SWIM directory using Internet Explorer if they wish to preview the web front end or they can burn the SWIM folder to a CD in order to deliver to the end user to preview and manipulate. In order to show what can be created, the following is an example of the SWIM front end for the end user that is connected to a memorial chapel database.

End User Web Front End

Open the SWIM folder and then the Home.html file using Internet Explorer. A banner will drop down from the top of the window asking to allow ActiveX controls to run. Right click the banner and click “Allow content...”. A security Warning box will pop up asking if you are sure you want to let this file run active content. Click “Yes”. A window, like the one seen below will show.



SPACE

Click the SPACE tab on the left hand menu. In the description box, type “desk” (Denoted by the white arrow). In the Space drop down menu select “147 Worship Center123456” (Denoted by the black arrow) and click submit.

SPACE

This contains the information about spaces that are available in the Memorial Chapel database. You can choose to search for all of the psaces in this building or search for a particular room by entering the room number.

Description:

Space: <All>

←

←

You will now see the table below.

Show 5 entries

Search:

Division	System	Description	Room	Id
12	Furnishings	Desk	147 WORSHIP CENTER123456	185811
12	Furnishings	Desk	147 WORSHIP CENTER123456	187475

Showing 1 to 2 of 2 entries

First Previous 1 Next Last

From here you are able to adjust the number of entries show (Denoted by the blue box), search all results present in the table (Denoted by the green box), and move from page to page of results (Denoted by the orange box).

Under the “Description” column click the first link Desk. This takes you to an information page with all the information stored on that particular desk. See Image Below.

Item: Changes here will affect this item only.

Field Current Value Modify

Id: 185811

System: Furnishings

Division: 12

Room: 147 WORSHIP CENTER123456

Type: Changes here will affect all items of this type.

Field Current Value Modify

Id: 131610

EquipmentDescription: Desk

EquipmentType: 60" x 30"

Manufacturer: Elkay

ModelNumber: S4S354d5

ModelDescription: Elkay Desk

Associated Documents

Show 5 entries

Search:

ID	Division	Category	Subcategory	Association
4	12	Building Systems and Acoustics - Cabinetry and Casework	Shop	Building-Systems-and-Acoustics-Cabinetry-and-Casework.pdf

Showing 1 to 1 of 1 entries

This information includes all product information, all associated documents, and all associated contacts. Here you can modify the room, Equipment Description, Equipment Type, Manufacturer, Model Number, and Model Description by typing the changes in the corresponding text fields/dropdown menu and clicking “Save Changes” (Denoted by the green box). To view the document associated with this product, click the link in the Link column of the Associated Documents table (Denoted by the blue box). To view the website of the associated contact, click the link under Website under the Associated Contact table (Denoted by the red box). You can also click on the manufacturer, the Contact Name, and the Associated Document Description (Denoted by the orange boxes). Clicking on any of these links brings you to a page with additional information about the manufacturer, contact, and document respectively. See Image Below.

Field	Current Value	Modify
Id:	4	
Division:	12	
Description:	Building Systems and Acoustics - Cabinetry and Casework	
Category:	Submittals	
Subcategory:	Shop Drawings	
Association:	CDC Warehouse	
Link:	Building-Systems-and-Acoustics---Cabinetry-and-Casework.pdf	
		Save Changes

From this page you can edit the additional information by filling in the proper text field and clicking “Save Changes” (Denoted by green box).

SYSTEM

Click the SYSTEM tab on the left hand menu. In the description box, type “desk” (Denoted by the white arrow). In the System drop down menu select “Furnishings” (Denoted by the black arrow) and click submit.

SYSTEM

This site provides the ability to search for a piece of equipment based on its system. The ability is given to search for all systems or a particular system based on a criteria. Upon selecting a particular system, all equipment within that system will be provided.

Description:

System:

This is the resulting table.

Show

5

 entries

Search:

Division	System	Description	Room	Id
12	Furnishings	Desk	147 WORSHIP CENTER123456	185811
12	Furnishings	Desk	104 ADMIN.	185853
12	Furnishings	Desk	134 ACTIVITY ROOM	185965
12	Furnishings	Desk	147 WORSHIP CENTER123456	187475

Showing 1 to 4 of 4 entries

First

Previous

1

Next

Last

The resulting table has the same functionality as mentioned earlier.

DIVISION

Click the DIVISION tab on the left hand menu. In the description box, type “desk” (Denoted by the white arrow). In the Division drop down menu select “12” (Denoted by the black arrow) and click submit.

DIVISION

This site provides the ability to search for a piece of equipment based on its division. The ability is given to search for all divisions or a particular division based on a criteria. Upon selecting a particular division, all equipment within that division will be provided.

Description:

Division:

<All>

Submit

Below is an example of the resulting table which has the same functionality as mentioned earlier.

Show 5 entries				Search:	
Division	System	Description	Room	Id	
12	Furnishings	Desk	147 WORSHIP CENTER123456	185811	
12	Furnishings	Desk	104 ADMIN.	185853	
12	Furnishings	Desk	134 ACTIVITY ROOM	185965	
12	Furnishings	Desk	147 WORSHIP CENTER123456	187475	
Showing 1 to 4 of 4 entries				First	Previous 1 Next Last

DOCUMENTS

Click the DOCUMENTS tab on the left hand menu. In the text box type “Building Systems and Acoustics” (Denoted by the white arrow). In the drop down menu select Submittals (Denoted by the black arrow) and click Search.

DOCUMENTS

This site provides a list of all documents available for equipment within the Memorial Chapel.
The ability exists to search for all documents or a particular document based on the document type.

Search Criteria

And/Or -Select Category-

[Upload A Document](#)

This is the resulting table.

ID	Division	Description	Category	Subcategory	Association	Link
4	12	Building Systems and Acoustics - Cabinetry and Casework	Submittals	Shop Drawings	CDC Warehouse	Building-Systems-and-Acoustics---Cabinetry-and-Casework.pdf

Showing 1 to 1 of 1 entries

[First](#) [Previous](#) [1](#) [Next](#) [Last](#)

The resulting table has the same functionality as mention earlier.

Click the DOCUMENTS tab on the left hand menu. Click the link titled “Upload A Document” in the bottom left hand corner. This takes you to an UPLOAD DOCUMENTS page. See Image Below.

UPLOAD DOCUMENTS

This page allows you to upload a new document to the database.
Please fill in all fields.

Choose file:

Set file information:

Description:

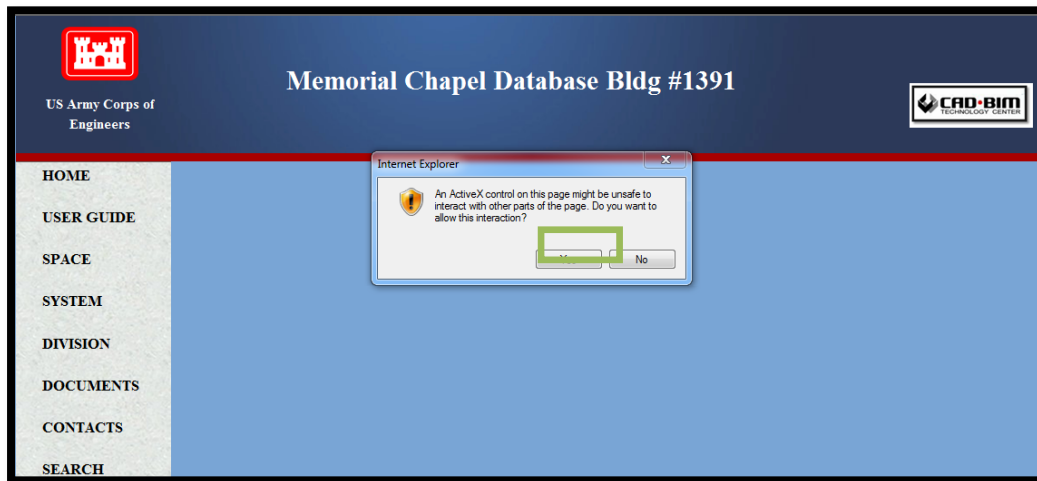
Association:

-Select Category-

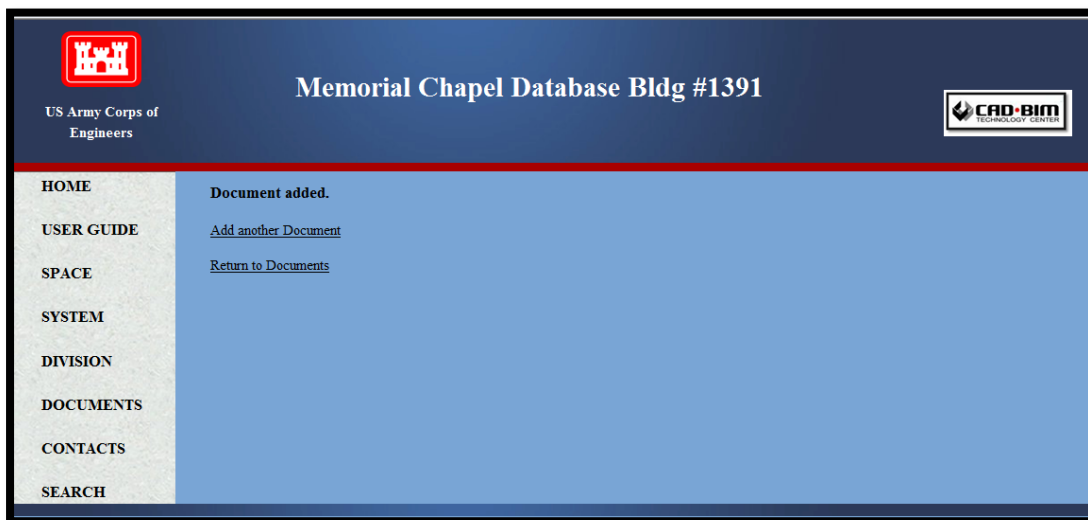
-Select Subcategory-

-Select Division-

Click the button titled “Browse...” and select a document to upload (Denoted by the black arrow). In the description text box type “Sample Document 1” (Denoted by the yellow arrow). In the Association text box type “Sample Warehouse” (Denoted by the red arrow). In the Category drop down menu select “Submittals” (Denoted by the purple arrow). In the Subcategory drop down menu select “Parts List” (Denoted by the orange arrow). In the Division drop down menu select “12” (Denoted by the green arrow). Click the “Add Document” button (Denoted by the blue arrow). A pop up message will appear. Click “Yes”. See Image Below.



A new screen will appear with a message saying your document has been added to the database. You will now have the option to return to the Documents page or add another document. See Image Below.



If you return to view all the documents in the database you will now see your added document. See Image Below.

Show entries Search:

ID	Division	Description	Category	Subcategory	Association	Link
1	null	1 Cover Sheet (Public Works Business Center)	CAD Drawings	As Built / Record Drawings	CDC Warehouse	1-Cover-Sheet-Public-Works-Business-Center.dgn
4	12	Building Systems and Acoustics - Cabinetry and Casework	Submittals	Shop Drawings	CDC Warehouse	Building-Systems-and-Acoustics---Cabinetry-and-Casework.pdf
5	13	Test	CAD Drawings	As Built / Record Drawings	asdf	16200---Copper-Cable-Test-CA31-1726-1750.pdf
6	8	fff	Reports	None	fff	16700---Fluke-Test-D1-D19-V1-V19.pdf
10	12	Sample Document 1	Submittals	Parts List	Sample Warehouse	bike.o3dtgz

Showing 1 to 5 of 5 entries First Previous 1 Next Last

CONTACTS

Click the CONTACTS tab on the left hand menu. In the text box type “Bob” (Denoted by the white arrow). In the Type drop down menu select “General” (Denoted by the black arrow). Click Search.

CONTACTS

This site provides the ability to search for contacts associated with the memorial chapel. The ability to search for all contacts or a specific contact exists based on the type of contact exists.

Search Criteria

Search by Name

-OR- Search by Type...

This is the resulting table.

Show entries Search:

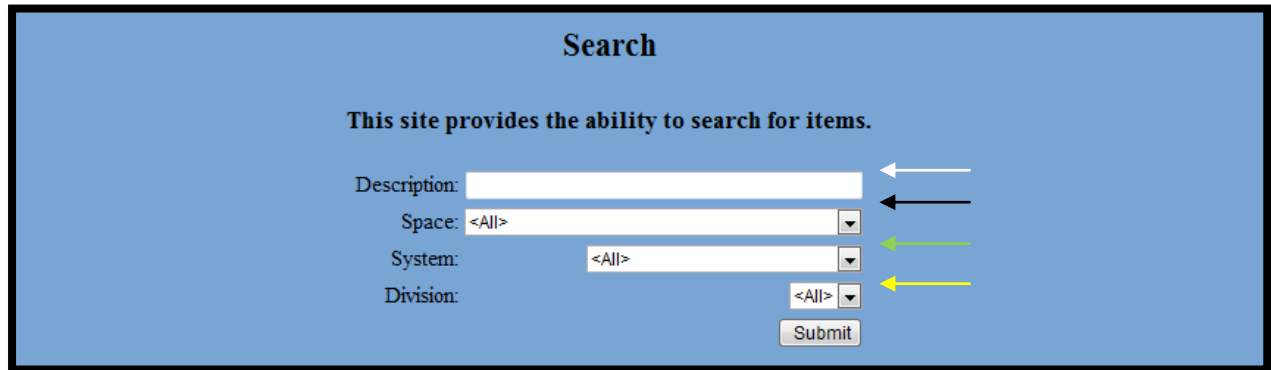
Contact	Company	Title	Type	Scope	Phone	Website	ID
Dylan, Bob	Phase Tech	Singer/Songwriter	General	Interiors	null	www.website.com	14

Showing 1 to 1 of 1 entries First Previous 1 Next Last

The resulting table has the same functionality as mentioned earlier.

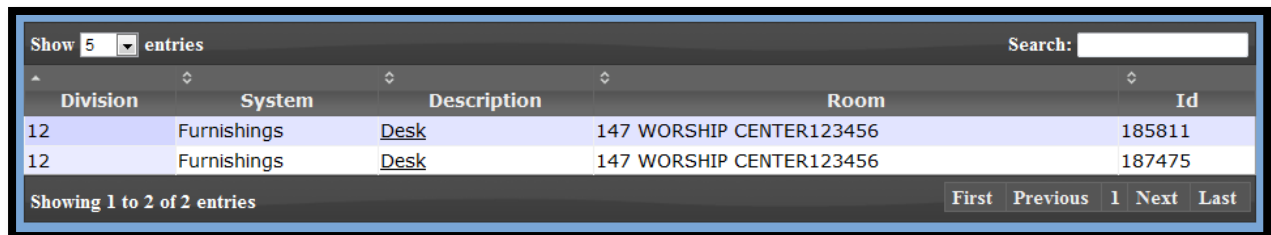
Search Button

Click the SEARCH button on the left hand menu. In the Description text box type “Desk” (Denoted by the white arrow). In the space drop down menu select “147 WORSHIP CENTER123456” (Denoted by the black arrow). In the System drop down menu select “Furnishings” (Denoted by the green arrow). In the Division drop down menu select “12” (Denoted by the yellow arrow). Click the Submit button.



The screenshot shows a search form titled "Search" on a blue background. Below the title is the text "This site provides the ability to search for items." The form contains four input fields with labels to their left: "Description:", "Space:", "System:", and "Division:". Each field has a corresponding colored arrow pointing to it from the right: a white arrow for Description, a black arrow for Space, a green arrow for System, and a yellow arrow for Division. The Space, System, and Division fields are dropdown menus, all currently showing "<All>". A "Submit" button is located at the bottom right of the form.

This is the resulting table.



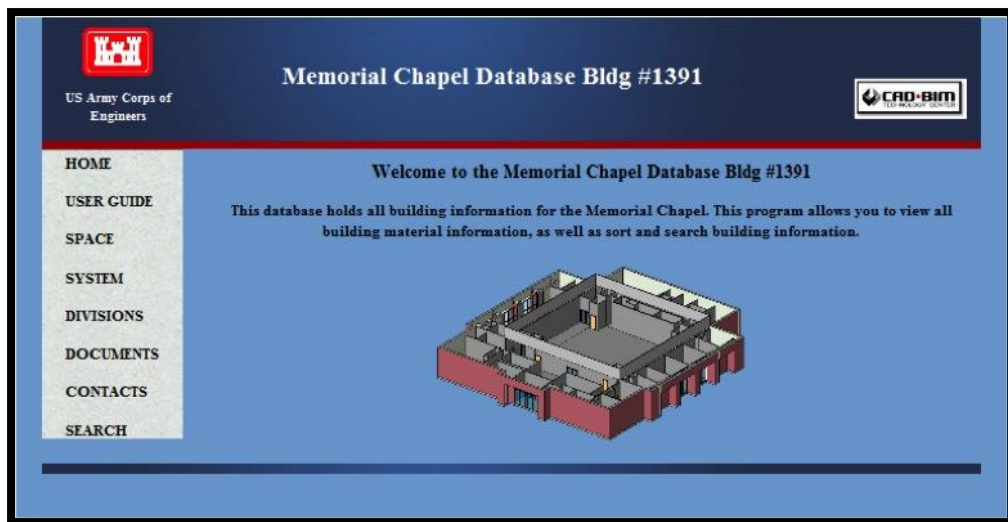
The screenshot shows a table with 5 columns: Division, System, Description, Room, and Id. The table has 2 rows of data. Above the table is a "Show 5 entries" dropdown and a "Search:" text box. Below the table is a pagination bar with "Showing 1 to 2 of 2 entries" and buttons for "First", "Previous", "1", "Next", and "Last".

Division	System	Description	Room	Id
12	Furnishings	Desk	147 WORSHIP CENTER123456	185811
12	Furnishings	Desk	147 WORSHIP CENTER123456	187475

The resulting table has the same functionality as mentioned earlier.

Chapter 4: Complete User Guide for SWIM

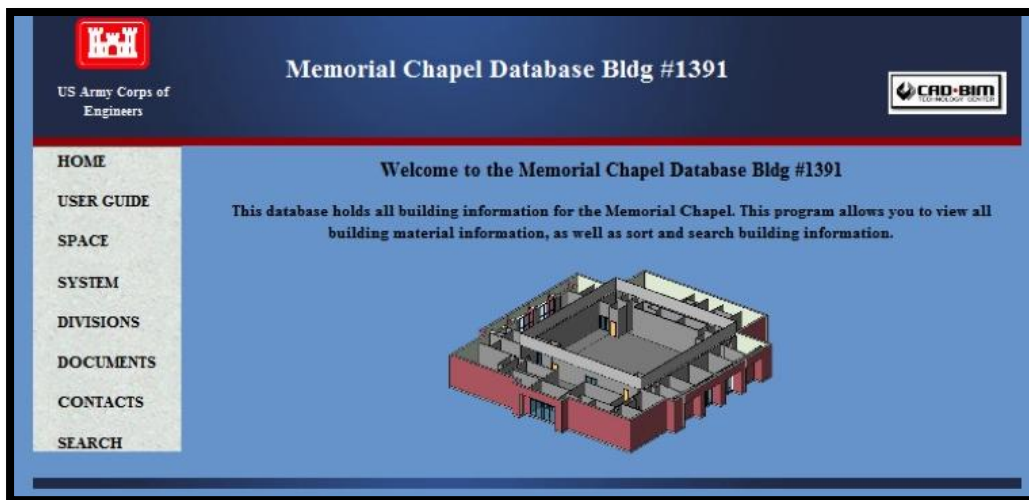
While all of the features of the super user are explained in Chapter 3: Setting Up and running SWIM, there is more information about the super user that was discussed in that chapter. This chapter discusses how to access the information while interacting with the end users web front end. In essence, looking at the welcome page of the end user, the information provided below discusses all of the features provided in the left navigation buttons (Home, space, system, divisions, documents, contacts, and search) as well as other features that exist outside the normal button operations. The only button not discussed is the User Guide button as this is a link to the user manual while in the web interface.



Home Button

HOME allows the user to return to the home page at any point in a search. This screen shows an image of the Project's building and shares an introduction with the user.

After clicking the HOME tab the user will be brought to this screen:



Space Button

SPACE allows the user to search for all the spaces in the Project or search for a particular room by entering the room number.

After clicking on the SPACE tab the user will be brought to this screen:

First, to search all spaces the user will leave all fields blank and click submit.

In order to see more than 5 entries, the user can choose how many entries they want to view at a time by clicking on the drop down box in the upper right hand corner of the results table.
This is shown below:

Show **100** entries Search:

Division	System	Description	Room	Id
7	Thermal & Moisture Protection	Basic Roof	ROOF	167482
7	Thermal & Moisture Protection	Basic Roof	ROOF	172232
7	Thermal & Moisture Protection	Basic Roof	ROOF	172283
7	Thermal & Moisture Protection	Basic Roof	ROOF	172381
7	Thermal & Moisture Protection	Basic Roof	ROOF	172593
10	Specialties	Fire Alarm - Wall	146 MEN	191519
12	Furnishings	Base Cabinet-Built In	102 NARTHEX / RECEPTION	188955
12	Furnishings	Desk	147 WORSHIP CENTER123456	185811
12	Furnishings	Desk	104 ADMIN.	185853
12	Furnishings	Desk	134 ACTIVITY ROOM	185965
12	Furnishings	Desk	147 WORSHIP CENTER123456	187475
12	Furnishings	Bed-Bunk	140 NURSERY	190833
12	Furnishings	Bed-Bunk	108 CHAPLIN'S OFFICE NO. 2	190863
16	Electrical	Floor Lamp - Standup	134 ACTIVITY ROOM	186568
16	Electrical	Floor Lamp - Standup	133 CORR	187289
16	Electrical	Floor Lamp - Standup	147 WORSHIP CENTER123456	187443
16	Electrical	Floor Lamp - Torchiere	147 WORSHIP CENTER123456	192223

Showing 1 to 17 of 17 entries First Previous 1 Next Last

Second, the user can search for a specific space by selecting it in the Space drop down menu.
Example: The user would choose Room 147, which is the Worship Center:

Description:

Space: 147 WORSHIP CENTER123456

Everything that is in Room 147 appears as the search result:

Show **5** entries Search:

Division	System	Description	Room	Id
12	Furnishings	Desk	147 WORSHIP CENTER123456	185811
12	Furnishings	Desk	147 WORSHIP CENTER123456	187475
16	Electrical	Floor Lamp - Standup	147 WORSHIP CENTER123456	187443
16	Electrical	Floor Lamp - Torchiere	147 WORSHIP CENTER123456	192223

Showing 1 to 4 of 4 entries First Previous 1 Next Last

Third, the user can search by description by typing in the text box.
Example: The user would type in desk:

Description:

Space: <All>

All items with desk in the description will appear in the search results:

Show 5 entries		Search:		
Division	System	Description	Room	Id
12	Furnishings	Desk	147 WORSHIP CENTER123456	185811
12	Furnishings	Desk	104 ADMIN.	185853
12	Furnishings	Desk	134 ACTIVITY ROOM	185965
12	Furnishings	Desk	147 WORSHIP CENTER123456	187475
Showing 1 to 4 of 4 entries				First Previous 1 Next Last

System Button

SYSTEM provides the ability to search for a piece of equipment based on its system. The ability is given to search for all systems or a particular system based on a criteria. Upon selecting a particular system, all equipment within that system will be provided.

After clicking on the SYSTEM tab the user will be brought to this screen:



US Army Corps of
Engineers

Memorial Chapel Database Bldg #1391



[HOME](#)
[USER GUIDE](#)
[SPACE](#)
[SYSTEM](#)
[DIVISIONS](#)
[DOCUMENTS](#)
[CONTACTS](#)
[SEARCH](#)

SYSTEM

This site provides the ability to search for a piece of equipment based on its system. The ability is given to search for all systems or a particular system based on a criteria. Upon selecting a particular system, all equipment within that system will be provided.

Description:

System:

First, the user can search all systems in the database by leaving all fields blank and clicking submit.

If the user wants to see more than 5 entries at a time, they can change the number of results shown by selecting the drop down box from the top right corner of the results table:

Show 5 entries					Search:
Division	System	Description	Room	Id	
7	Thermal & Moisture Protection	Basic Roof	ROOF	167482	
7	Thermal & Moisture Protection	Basic Roof	ROOF	172232	
7	Thermal & Moisture Protection	Basic Roof	ROOF	172283	
7	Thermal & Moisture Protection	Basic Roof	ROOF	172381	
7	Thermal & Moisture Protection	Basic Roof	ROOF	172593	
Showing 1 to 5 of 17 entries					First Previous 1 2 3 4 Next Last

The database will then expand to look like this:

Show 100 entries					Search:
Division	System	Description	Room	Id	
7	Thermal & Moisture Protection	Basic Roof	ROOF	167482	
7	Thermal & Moisture Protection	Basic Roof	ROOF	172232	
7	Thermal & Moisture Protection	Basic Roof	ROOF	172283	
7	Thermal & Moisture Protection	Basic Roof	ROOF	172381	
7	Thermal & Moisture Protection	Basic Roof	ROOF	172593	
10	Specialties	Fire Alarm - Wall	146 MEN	191519	
12	Furnishings	Base Cabinet-Built In	102 NARTHEX / RECEPTION	188955	
12	Furnishings	Desk	147 WORSHIP CENTER123456	185811	
12	Furnishings	Desk	104 ADMIN.	185853	
12	Furnishings	Desk	134 ACTIVITY ROOM	185965	
12	Furnishings	Desk	147 WORSHIP CENTER123456	187475	
12	Furnishings	Bed-Bunk	140 NURSERY	190833	
12	Furnishings	Bed-Bunk	108 CHAPLIN'S OFFICE NO. 2	190863	
16	Electrical	Floor Lamp - Standup	134 ACTIVITY ROOM	186568	
16	Electrical	Floor Lamp - Standup	133 CORR	187289	
16	Electrical	Floor Lamp - Standup	147 WORSHIP CENTER123456	187443	
16	Electrical	Floor Lamp - Torchiere	147 WORSHIP CENTER123456	192223	
Showing 1 to 17 of 17 entries					First Previous 1 Next Last

Second, the user can search for a specific system by selecting it from the System drop down menu. Example: The user would choose Thermal & Moisture Protection:

Description:

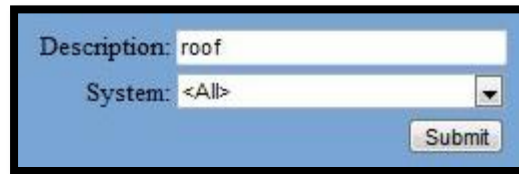
System:

Submit

All items in the Thermal & Moisture Protection system appear in the search results:

Show 5 entries					Search:
"Division (level?)"	System	Description	Room	Id	
7	Thermal & Moisture Protection	Basic Roof	ROOF	167482	
7	Thermal & Moisture Protection	Basic Roof	ROOF	172232	
7	Thermal & Moisture Protection	Basic Roof	ROOF	172283	
7	Thermal & Moisture Protection	Basic Roof	ROOF	172381	
7	Thermal & Moisture Protection	Basic Roof	ROOF	172593	
Showing 1 to 5 of 5 entries					First Previous 1 Next Last

Third, the user can search by description by typing in the text box.
Example: The user would type in roof:



Description:

System:

Submit

All items with the word roof in their description appear in the search results:

Show entries

Search:

Division	System	Description	Room	Id
7	Thermal & Moisture Protection	Basic Roof	ROOF	167482
7	Thermal & Moisture Protection	Basic Roof	ROOF	172232
7	Thermal & Moisture Protection	Basic Roof	ROOF	172283
7	Thermal & Moisture Protection	Basic Roof	ROOF	172381
7	Thermal & Moisture Protection	Basic Roof	ROOF	172593

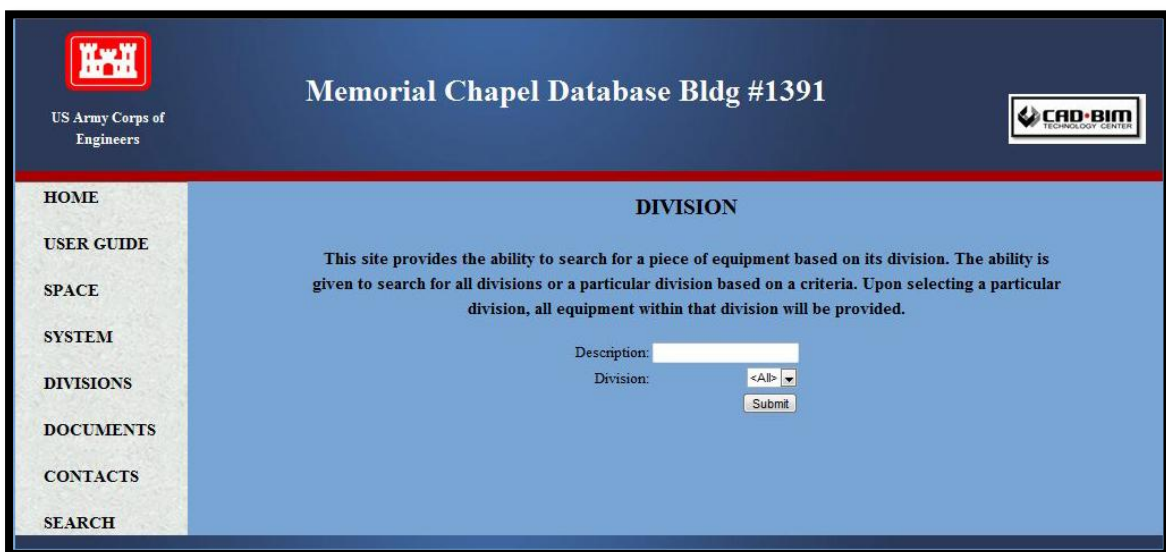
Showing 1 to 5 of 5 entries

First Previous 1 Next Last

Division Button

DIVISION provides the ability to search for a piece of equipment based on its division. The user is able to search for all divisions or a particular division. When the user selects a particular division, all equipment within that division will be provided.

After selecting the DIVISION tab from the menu, the user will see this screen:



US Army Corps of Engineers

Memorial Chapel Database Bldg #1391

CAD-BIM TECHNOLOGY CENTER

HOME

USER GUIDE

SPACE

SYSTEM

DIVISIONS

DOCUMENTS

CONTACTS

SEARCH

DIVISION

This site provides the ability to search for a piece of equipment based on its division. The ability is given to search for all divisions or a particular division based on a criteria. Upon selecting a particular division, all equipment within that division will be provided.

Description:

Division:

Submit

First, the user can search all divisions in the database by leaving all fields blank and clicking submit.

Show 100 entries

Search:

Division	System	Description	Room	Id
7	Thermal & Moisture Protection	<u>Basic Roof</u>	ROOF	167482
7	Thermal & Moisture Protection	<u>Basic Roof</u>	ROOF	172232
7	Thermal & Moisture Protection	<u>Basic Roof</u>	ROOF	172283
7	Thermal & Moisture Protection	<u>Basic Roof</u>	ROOF	172381
7	Thermal & Moisture Protection	<u>Basic Roof</u>	ROOF	172593
10	Specialties	<u>Fire Alarm - Wall</u>	146 MEN	191519
12	Furnishings	<u>Base Cabinet-Built In</u>	102 NARTHEX / RECEPTION	188955
12	Furnishings	<u>Desk</u>	147 WORSHIP CENTER123456	185811
12	Furnishings	<u>Desk</u>	104 ADMIN.	185853
12	Furnishings	<u>Desk</u>	134 ACTIVITY ROOM	185965
12	Furnishings	<u>Desk</u>	147 WORSHIP CENTER123456	187475
12	Furnishings	<u>Bed-Bunk</u>	140 NURSERY	190833
12	Furnishings	<u>Bed-Bunk</u>	108 CHAPLIN'S OFFICE NO. 2	190863
16	Electrical	<u>Floor Lamp - Standup</u>	134 ACTIVITY ROOM	186568
16	Electrical	<u>Floor Lamp - Standup</u>	133 CORR	187289
16	Electrical	<u>Floor Lamp - Standup</u>	147 WORSHIP CENTER123456	187443
16	Electrical	<u>Floor Lamp - Torchiera</u>	147 WORSHIP CENTER123456	192223

Showing 1 to 17 of 17 entries

First Previous 1 Next Last

Second, the user can search for all items in a specific division by selecting a specific division and leaving all other fields blank.

Example: The user would select option 16 from the Division drop down menu:

Description:

Division: 16

The results would be all items with the division 16:

Show 5 entries

Search:

Division	System	Description	Room	Id
16	Electrical	<u>Floor Lamp - Standup</u>	134 ACTIVITY ROOM	186568
16	Electrical	<u>Floor Lamp - Standup</u>	133 CORR	187289
16	Electrical	<u>Floor Lamp - Standup</u>	147 WORSHIP CENTER123456	187443
16	Electrical	<u>Floor Lamp - Torchiera</u>	147 WORSHIP CENTER123456	192223

Showing 1 to 4 of 4 entries

First Previous 1 Next Last

Lastly, the user can search for an item by typing it in the description text box.

Example: The user would type floor in the description text box.

Description: floor

Division: <All>

All items with floor in the description will appear in the search results:

Show 5 entries		Search:		
Division	System	Description	Room	Id
16	Electrical	Floor Lamp - Standup	134 ACTIVITY ROOM	186568
16	Electrical	Floor Lamp - Standup	133 CORR	187289
16	Electrical	Floor Lamp - Standup	147 WORSHIP CENTER123456	187443
16	Electrical	Floor Lamp - Torchiera	147 WORSHIP CENTER123456	192223
Showing 1 to 4 of 4 entries				First Previous 1 Next Last

Documents Tab

DOCUMENTS provide a list of all documents available for equipment within the Project. The user has the ability to search for all documents or a particular document based on the document type.

After clicking on the DOCUMENTS tab the user will see this screen:


Memorial Chapel Database Bldg #1391


HOME
USER GUIDE
SPACE
SYSTEM
DIVISION
DOCUMENTS
CONTACTS
SEARCH

DOCUMENTS

This site provides a list of all documents available for equipment within the Memorial Chapel. The ability exists to search for all documents or a particular document based on the document type.

Search Criteria

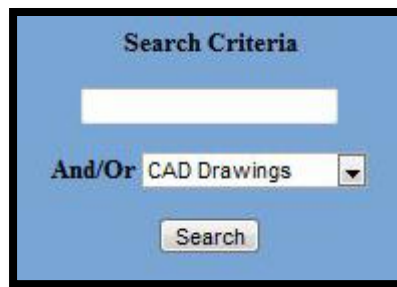
And/Or -Select Category-

First, the user can search all the documents in the database by leaving all fields blank and clicking the submit button:

Show 5 entries					Search:	
ID	Division	Description	Category	Subcategory	Association	Link
1	null	1 Cover Sheet (Public Works Business Center)	CAD Drawings	As Built / Record Drawings	CDC Warehouse	1-Cover-Sheet-Public-Works-Business-Center.dgn
4	12	Building Systems and Acoustics - Cabinetry and Casework	Submittals	Shop Drawings	CDC Warehouse	Building-Systems-and-Acoustics---Cabinetry-and-Casework.pdf
Showing 1 to 2 of 2 entries					First Previous 1 Next Last	

Second, the user can search for all documents in a category by selecting a value from the drop down menu and leaving the text box blank.

Example: The user would search for CAD Drawings:



A blue dialog box titled "Search Criteria". It contains a text input field, a label "And/Or" followed by a dropdown menu showing "CAD Drawings", and a "Search" button.

Every document that has a category of CAD Drawings appears in the search results:



Search results table showing 1 entry. The table has columns: ID, Division, Description, Category, Subcategory, Association, and Link. The entry is for a cover sheet for Public Works Business Center.

ID	Division	Description	Category	Subcategory	Association	Link
1	null	1 Cover Sheet (Public Works Business Center)	CAD Drawings	As Built / Record Drawings	CDC Warehouse	1-Cover-Sheet-Public-Works-Business-Center.dgn

Showing 1 to 1 of 1 entries

Lastly, the user can search for a specific document by typing in the document description. Example: The user enters the word cover:



A blue dialog box titled "Search Criteria". The text input field contains the word "cover". The "And/Or" dropdown menu is set to "-Select Category-". The "Search" button is visible.

Every document that has the word cover in its description appears in the search results:



Search results table showing 1 entry. The table has columns: ID, Division, Description, Category, Subcategory, Association, and Link. The entry is for a cover sheet for Public Works Business Center.

ID	Division	Description	Category	Subcategory	Association	Link
1	null	1 Cover Sheet (Public Works Business Center)	CAD Drawings	As Built / Record Drawings	CDC Warehouse	1-Cover-Sheet-Public-Works-Business-Center.dgn

Showing 1 to 1 of 1 entries

Contacts Button

CONTACTS provide the ability to search for contact personnel associated with the Project. SWIM provides the ability to search for all contacts or contacts of a certain type.

After clicking on the CONTACTS tab in the menu the user will see this screen:

First, the user can search all contacts by leaving all fields blank and click the submit button:

Show entries Search:

Company	Contact	Title	Type	Scope	Phone	Website	ID
Circle C Construction	Kissinhuq, Amanada	null	Subcontractor	null	601-867-5309	null	15
Elkay	Jones, Tim	null	Manufacturer	Doors & Windows	555-555-5557	www.Elkay.com	7
NuTone	Smith, Mike	CEO	Supplier	none	888-888-8887	none	9
Phase Tech	Dylan, Bob	Singer/Songwriter	General	Interiors	null	www.website.com	14
SquareD	Clooney, George	Actor	General	Fire Protection	800-800-8008	www.clooneyMania.com	11
State Commercial	Mr. Ed	null	Subcontractor	Plumbing	555-555-5555	www.statecommercial.com	12
Trane Commercial Systems	null	null	Manufacturer	HVAC	none	www.trane.com	10
USA Rental	Smith, Jane	Agent	null	Plumbing	null	null	16
Vanguard Alarm Services	null	null	Supplier	Fire Protection	null	null	13

Showing 1 to 9 of 9 entries First Previous 1 Next Last

Second, the user can search for a contact by their type by selecting the type under the Type drop down. Example: The user would choose General:

Every contact that is under the type General appears in the search results:

Show 5 entries		Search:					
Type	Company	Contact	Title	Scope	Phone	Website	ID
General	SquareD	Clooney, George	Actor	Fire Protection	800-800-8008	www.clooneyMania.com	11
General	Phase Tech	Dylan, Bob	Singer/Songwriter	Interiors	null	www.website.com	14
Showing 1 to 2 of 2 entries						First	Previous 1 Next Last

Lastly, the user would search for the contact's name by typing it in the text box.
Example: The user would type in the name Bob:

Search Criteria

Search by Name

-OR- Search by Type...

Every contact that has the name Bob appears in the search results:

Show 5 entries		Search:					
Contact	Company	Title	Type	Scope	Phone	Website	ID
Dylan, Bob	Phase Tech	Singer/Songwriter	General	Interiors	null	www.website.com	14
Showing 1 to 1 of 1 entries						First	Previous 1 Next Last

Search Button

SEARCH provides the ability to search for items in the Project.

After clicking the SEARCH tab in the menu the user will see this page:



US Army Corps of
Engineers

Memorial Chapel Database Bldg #1391



[HOME](#)
[USER GUIDE](#)
[SPACE](#)
[SYSTEM](#)
[DIVISIONS](#)
[DOCUMENTS](#)
[CONTACTS](#)
[SEARCH](#)

Search

This site provides the ability to search for items.

Description:
Space:
System:
Division:

First, the user can search all entries in the database by leaving all fields blank and clicking Submit. The result would be a table with all items in the Project database:

Show entries
Search:

Division	System	Description	Room	Id
7	Thermal & Moisture Protection	Basic Roof	ROOF	167482
7	Thermal & Moisture Protection	Basic Roof	ROOF	172232
7	Thermal & Moisture Protection	Basic Roof	ROOF	172283
7	Thermal & Moisture Protection	Basic Roof	ROOF	172381
7	Thermal & Moisture Protection	Basic Roof	ROOF	172593
10	Specialties	Fire Alarm - Wall	146 MEN	191519
12	Furnishings	Base Cabinet-Built In	102 NARTHEX / RECEPTION	188955
12	Furnishings	Desk	147 WORSHIP CENTER123456	185811
12	Furnishings	Desk	104 ADMIN.	185853
12	Furnishings	Desk	134 ACTIVITY ROOM	185965
12	Furnishings	Desk	147 WORSHIP CENTER123456	187475
12	Furnishings	Bed-Bunk	140 NURSERY	190833
12	Furnishings	Bed-Bunk	108 CHAPLIN'S OFFICE NO. 2	190863
16	Electrical	Floor Lamp - Standup	134 ACTIVITY ROOM	186568
16	Electrical	Floor Lamp - Standup	133 CORR	187289
16	Electrical	Floor Lamp - Standup	147 WORSHIP CENTER123456	187443
16	Electrical	Floor Lamp - Torchiere	147 WORSHIP CENTER123456	192223

Showing 1 to 17 of 17 entries

Second, the user can search just the description for an item by typing in the Description text box and leaving all other fields blank.

Example: The user would type in floor as the description:

Description:

Space:

System:

Division:

All items containing the word floor in the description appear in the search results:

Show entries

Search:

Division	System	Description	Room	Id
16	Electrical	Floor Lamp - Standup	134 ACTIVITY ROOM	186568
16	Electrical	Floor Lamp - Standup	133 CORR	187289
16	Electrical	Floor Lamp - Standup	147 WORSHIP CENTER123456	187443
16	Electrical	Floor Lamp - Torchiere	147 WORSHIP CENTER123456	192223

Showing 1 to 4 of 4 entries

First Previous 1 Next Last

Third, the user can search by space only by selecting a value in the Space drop down and leaving all other fields blank.

Example: The user would choose Rm 147:

Description:

Space:

System:

Division:

Every item in Room 147 appears in the search results:

Show entries

Search:

Division	System	Description	Room	Id
12	Furnishings	Desk	147 WORSHIP CENTER123456	185811
12	Furnishings	Desk	147 WORSHIP CENTER123456	187475
16	Electrical	Floor Lamp - Standup	147 WORSHIP CENTER123456	187443
16	Electrical	Floor Lamp - Torchiere	147 WORSHIP CENTER123456	192223

Showing 1 to 4 of 4 entries

First Previous 1 Next Last

Fourth, the user can search a certain system by selecting a value in the System drop down and leaving all other fields blank.

Example: The user would choose Thermal & Moisture Protection:

Description:

Space:

System:

Division:

All items in the Project under the system Thermal & Moisture Protection appear in the search results:

Show entries

Search:

"Division (level?)"	System	Description	Room	Id
7	Thermal & Moisture Protection	Basic Roof	ROOF	167482
7	Thermal & Moisture Protection	Basic Roof	ROOF	172232
7	Thermal & Moisture Protection	Basic Roof	ROOF	172283
7	Thermal & Moisture Protection	Basic Roof	ROOF	172381
7	Thermal & Moisture Protection	Basic Roof	ROOF	172593

Showing 1 to 5 of 5 entries

First Previous 1 Next Last

Fifth, the user can search the division by selecting a value from the Division drop down menu and leaving all other fields blank. Example: The user would choose division 16:

Description:

Space:

System:

Division:

All items in the Project under division 16 appear in the search results:

Show entries

Search:

Division	System	Description	Room	Id
16	Electrical	Floor Lamp - Standup	134 ACTIVITY ROOM	186568
16	Electrical	Floor Lamp - Standup	133 CORR	187289
16	Electrical	Floor Lamp - Standup	147 WORSHIP CENTER123456	187443
16	Electrical	Floor Lamp - Torchiera	147 WORSHIP CENTER123456	192223

Showing 1 to 4 of 4 entries

First Previous 1 Next Last

Lastly, the user can limit their search to what they are looking for specifically by choosing different combinations of description, space, system, and division.

Example: The user would choose the word floor, in Room 147, dealing with the electrical system, and division 16:

Description:

Space:

System:

Division:

The search results will show all items matching this search description:

Show entries

Search:

Division	System	Description	Room	Id
16	Electrical	Floor Lamp - Standup	147 WORSHIP CENTER123456	187443
16	Electrical	Floor Lamp - Torchiere	147 WORSHIP CENTER123456	192223

Showing 1 to 2 of 2 entries

First Previous 1 Next Last

Other Features

Features outside of the information provided from the links to the buttons include, search, sort, select number of entries to show, edit data, and add documents.

If the user wishes to sort the data, they click on the column heading to select the column and then click again to order the values. The figure below arranges the ids into increasing order.

Show entries

Search:

Division	System	Description	Room	Id
12	Furnishings	Desk	147 WORSHIP CENTER123456	185811
12	Furnishings	Desk	147 WORSHIP CENTER123456	187475
16	Electrical	Floor Lamp - Standup	147 WORSHIP CENTER123456	187443
16	Electrical	Floor Lamp - Torchiere	147 WORSHIP CENTER123456	192223

Showing 1 to 4 of 4 entries

First Previous 1 Next Last

Show entries

Search:

Division	System	Description	Room	Id
16	Electrical	Floor Lamp - Torchiere	147 WORSHIP CENTER123456	192223
12	Furnishings	Desk	147 WORSHIP CENTER123456	187475
16	Electrical	Floor Lamp - Standup	147 WORSHIP CENTER123456	187443
12	Furnishings	Desk	147 WORSHIP CENTER123456	185811

Showing 1 to 4 of 4 entries

First Previous 1 Next Last

SWIM provides the ability to adjust the number of entries shown (Denoted by the blue box), search all results present in the table (Denoted by the green box), and move from page to page of results (Denoted by the orange box).

Show 5 entries					Search:	
Division	System	Description	Room		Id	
12	Furnishings	Desk	147 WORSHIP CENTER	123456	185811	
12	Furnishings	Desk	147 WORSHIP CENTER	123456	187475	
Showing 1 to 2 of 2 entries					First	Previous 1 Next Last

SWIM also provides the ability to edit and save data back to the project's database. In order to manipulate the data, the user fills in the information in the specified text box and submits the changes by selecting the "Save Changes" button.

Field	Current Value	Modify
Id:	4	
Division:	12	
Description:	Building Systems and Acoustics - Cabinetry and Casework	
Category:	Submittals	
Subcategory:	Shop Drawings	
Association:	CDC Warehouse	
Link:	Building-Systems-and-Acoustics---Cabinetry-and-Casework.pdf	
		Save Changes

Appendix A – Super User Code

Program.cs

The following is the entry program for the Super User GUI.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace Access_To_Cobie
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new AccessGUI());
        }
    }
}
```

Access2Cobie.cs

The following is code for converting an Access Database to a COBie Spreadsheet.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Access = Microsoft.Office.Interop.Access;
using Excel = Microsoft.Office.Interop.Excel;
using dao;

namespace Access_To_Cobie
{
    class Access2Cobie
    {
        private string fileName;
        private Excel.Application excelApp;
        private dao.Database db;
        private dao.Workspace ws;
        public Access2Cobie(string fn, Excel.Application app)
        {
            fileName = fn;
            excelApp = app;
            DBEngine dbEng = new DBEngine();
            ws = dbEng.Workspaces[0];
        }
    }
}
```

```

    db = ws.OpenDatabase(fileName);
}

~Access2Cobie()
{
}

public void createSpaceSheet(Excel.Workbook wkb2)
{
    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "Space";
    Excel.Range range = wks2.get_Range("A1", Type.Missing);
    range.Value = "Name";

    range = wks2.get_Range("B1", Type.Missing);
    range.Value = "CreatedBy";
    range = wks2.get_Range("C1", Type.Missing);
    range.Value = "CreatedOn";
    range = wks2.get_Range("D1", Type.Missing);
    range.Value = "Category";
    range = wks2.get_Range("E1", Type.Missing);
    range.Value = "FloorName";
    range = wks2.get_Range("F1", Type.Missing);
    range.Value = "Description";
    range = wks2.get_Range("G1", Type.Missing);
    range.Value = "ExtSystem";
    range = wks2.get_Range("H1", Type.Missing);
    range.Value = "ExtObject";
    range = wks2.get_Range("I1", Type.Missing);
    range.Value = "ExtIdentifier";
    range = wks2.get_Range("J1", Type.Missing);
    range.Value = "RoomTag";
    range = wks2.get_Range("K1", Type.Missing);
    range.Value = "UsableHeight";
    range = wks2.get_Range("L1", Type.Missing);
    range.Value = "GrossArea";
    range = wks2.get_Range("M1", Type.Missing);
    range.Value = "NetArea";
    range = wks2.get_Range("A1", "M1");
    range.Interior.ColorIndex = 15;
    range.Orientation = Excel.XlOrientation.xlUpward;
    range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
    range.Font.Bold = true;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;
    string sql = "Select Levels.Name,Rooms.Name,Rooms.Number,Rooms.Area From Levels,
    Rooms Where Levels.Id = Rooms.Level;";
    //dao.Database db = accessApp.CurrentDb();
    dao.Recordset rs = db.OpenRecordset(sql, Type.Missing, Type.Missing, Type.Missing);
    int i = 0;
    while (rs.EOF != true)
    {
        range = wks2.Range["A" + (i + 2), Type.Missing];
        range.Value = rs.Fields["Number"].Value;
    }
}

```

```

range = wks2.Range["B" + (i + 2), Type.Missing];
range.Value = "";
range = wks2.Range["C" + (i + 2), Type.Missing];
range.Value = System.DateTime.Now.ToString();
range = wks2.Range["D" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["E" + (i + 2), Type.Missing];
range.Value = rs.Fields["Levels.Name"].Value;
range = wks2.Range["F" + (i + 2), Type.Missing];
range.Value = rs.Fields["Rooms.Name"].Value;
range = wks2.Range["G" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["H" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["I" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["J" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["K" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["L" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["M" + (i + 2), Type.Missing];
range.Value = "n/a";
rs.MoveNext();
i++;
}

range = wks2.Range["A2:A" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ",F2:F" +
excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 6;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

range = wks2.Range["C2:D2", "C" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":D" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 6;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

range = wks2.Range["B2:B" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ",E2:E" +
excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 40;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

range = wks2.Range["G2:I2", "G" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ":I"
+ excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 39;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

range = wks2.Range["J2:M2", "J" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":M" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 35;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

```

```

range = wks2.Range["A2:A" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":M2,M2:M" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Columns.AutoFit();
rs.Close();
System.Runtime.InteropServices.Marshal.ReleaseComObject(rs);
rs = null;
GC.Collect();
}

public void createFacilitySheet(Excel.Workbook wkb2)
{
    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "Facility";
    Excel.Range range = wks2.Range["A1"];
    range.Value = "Name";
    range = wks2.Range["B1"];
    range.Value = "CreatedBy";
    range = wks2.Range["C1"];
    range.Value = "CreatedOn";
    range = wks2.Range["D1"];
    range.Value = "Category";
    range = wks2.Range["E1"];
    range.Value = "ProjectName";
    range = wks2.Range["F1"];
    range.Value = "SiteName";
    range = wks2.Range["G1"];
    range.Value = "LinearUnits";
    range = wks2.Range["H1"];
    range.Value = "AreaUnits";
    range = wks2.Range["I1"];
    range.Value = "VolumeUnits";
    range = wks2.Range["J1"];
    range.Value = "CurrencyUnits";
    range = wks2.Range["K1"];
    range.Value = "AreaMeasurement";
    range = wks2.Range["L1"];
    range.Value = "ExternalSystem";
    range = wks2.Range["M1"];
    range.Value = "ExternalProjectObject";
    range = wks2.Range["N1"];
    range.Value = "ExternalProjectIdentifier";
    range = wks2.Range["O1"];
    range.Value = "ExternalSiteObject";
    range = wks2.Range["P1"];
    range.Value = "ExternalSiteIdentifier";
    range = wks2.Range["Q1"];
    range.Value = "ExternalFacilityObject";
    range = wks2.Range["R1"];
    range.Value = "ExternalFacilityIdentifier";
    range = wks2.Range["S1"];
    range.Value = "Description";
    range = wks2.Range["T1"];
    range.Value = "ProjectDescription";
}

```

```

range = wks2.Range["U1"];
range.Value = "SiteDescription";
range = wks2.Range["V1"];
range.Value = "Phase";
string sql = "Select ProjectName From ProjectInformation";
//dao.Database db = accessApp.CurrentDb();
dao.Recordset rs = db.OpenRecordset(sql, Type.Missing, Type.Missing, Type.Missing);
int i = 0;
while (!rs.EOF)
{
    range = wks2.Range["A" + (i + 2), Type.Missing];
    range.Value = rs.Fields["ProjectName"].Value;

    range = wks2.Range["E" + (i + 2), Type.Missing];
    range.Value = rs.Fields["ProjectName"].Value;
    i++;
    rs.MoveNext();
}

range = wks2.get_Range("A1", "V1");
range.Interior.ColorIndex = 15;
range.Orientation = Excel.XlOrientation.xlUpward;
range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
range.Font.Bold = true;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
range = wks2.Range["A2:A" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 6;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
range = wks2.Range["C2:K2", "C" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":K" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 6;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
range = wks2.Range["B2:B" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 40;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
range = wks2.Range["L2:R2", "L" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":R" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 39;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
range = wks2.Range["S2:V2", "S" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":V" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 35;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
range = wks2.Range["A2:A" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":V2,V2:V" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Columns.AutoFit();
rs.Close();
System.Runtime.InteropServices.Marshal.ReleaseComObject(rs);
rs = null;
GC.Collect();
}

public void AutoFitColumn(Excel.Worksheet ws, int col)
{

```

```

        ((Excel.Range)ws.Cells[1, col]).EntireColumn.AutoFit();
        ((Excel.Range)ws.Cells[1, col]).EntireRow.AutoFit();
    }

```

```

public void createFloorSheet(Excel.Workbook wkb2)
{
    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "Floor";

    Excel.Range range = wks2.Range["A1"];
    range.Value = "Name";

    range = wks2.Range["B1"];
    range.Value = "CreatedBy";
    range = wks2.Range["C1"];
    range.Value = "CreatedOn";
    range = wks2.Range["D1"];
    range.Value = "Category";
    range = wks2.Range["E1"];
    range.Value = "ExtSystem";
    range = wks2.Range["F1"];
    range.Value = "ExtObject";
    range = wks2.Range["G1"];
    range.Value = "ExtIdentifier";
    range = wks2.Range["H1"];
    range.Value = "Description";
    range = wks2.Range["I1"];
    range.Value = "Elevation";
    range = wks2.Range["J1"];
    range.Value = "Height";
    string sql = "Select Name, Elevation From Levels";
    //dao.Database db = accessApp.CurrentDb();
    dao.Recordset rs = db.OpenRecordset(sql);
    int i = 0;
    while (!rs.EOF)
    {
        range = wks2.Range["A" + (i + 2), Type.Missing];
        range.Value = rs.Fields["Name"].Value;
        range = wks2.Range["B" + (i + 2), Type.Missing];
        range.Value = "n/a";
        range = wks2.Range["C" + (i + 2), Type.Missing];
        range.Value = System.DateTime.Now.ToString();
        range = wks2.Range["D" + (i + 2), Type.Missing];
        range.Value = "n/a";
        range = wks2.Range["E" + (i + 2), Type.Missing];
        range.Value = "n/a";
        range = wks2.Range["F" + (i + 2), Type.Missing];
        range.Value = "n/a";
        range = wks2.Range["G" + (i + 2), Type.Missing];
        range.Value = "n/a";
        range = wks2.Range["H" + (i + 2), Type.Missing];
        range.Value = rs.Fields["Name"].Value;
        range = wks2.Range["I" + (i + 2), Type.Missing];
        range.Value = rs.Fields["Elevation"].Value;
    }
}

```

```

        range = wks2.Range["J" + (i + 2), Type.Missing];
        range.Value = "n/a";
        rs.MoveNext();
        i++;
    }

    range = wks2.Range["A2:A" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
    range.Interior.ColorIndex = 6;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;
    range = wks2.Range["C2:D2", "C" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
        ":D" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
    range.Interior.ColorIndex = 6;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;
    range = wks2.Range["B2:B" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
    range.Interior.ColorIndex = 40;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;
    range = wks2.Range["E2:G2", "E" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
        ":G" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
    range.Interior.ColorIndex = 39;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;
    range = wks2.Range["H2:J2", "H" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
        ":J" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
    range.Interior.ColorIndex = 35;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;
    range = wks2.get_Range("A1", "J1");
    range.Interior.ColorIndex = 15;
    range.Orientation = Excel.XlOrientation.xlUpward;
    range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
    range.Font.Bold = true;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;

    range = wks2.Range["A2:A" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ":J2,J2:J"
        + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
    range.Columns.AutoFit();

    rs.Close();
    System.Runtime.InteropServices.Marshal.ReleaseComObject(rs);
    rs = null;
    GC.Collect();
}

public void createZoneSheet(Excel.Workbook wkb2)
{
    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "Zone";

    Excel.Range range = wks2.Range["A1"];
    range.Value = "Name";
    range = wks2.Range["B1"];
    range.Value = "CreatedBy";
    range = wks2.Range["C1"];
    range.Value = "CreatedOn";
}

```

```

range = wks2.Range["D1"];
range.Value = "Category";
range = wks2.Range["E1"];
range.Value = "SpaceName";
range = wks2.Range["F1"];
range.Value = "ExtSystem";
range = wks2.Range["G1"];
range.Value = "ExtObject";
range = wks2.Range["H1"];
range.Value = "ExtIdentifier";
range = wks2.Range["I1"];
range.Value = "Description";
range = wks2.get_Range("A1", "I1");
range.Interior.ColorIndex = 15;
range.Orientation = Excel.XlOrientation.xlUpward;
range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
range.Font.Bold = true;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

```

```

    GC.Collect();
}

```

```

public void createTypeSheet(Excel.Workbook wkb2)
{

```

```

    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "Type";

```

```

    Excel.Range range = wks2.Range["A1"];
    range.Value = "Name";
    range = wks2.Range["B1"];
    range.Value = "CreatedBy";
    range = wks2.Range["C1"];
    range.Value = "CreatedOn";
    range = wks2.Range["D1"];
    range.Value = "Category";
    range = wks2.Range["E1"];
    range.Value = "Description";
    range = wks2.Range["F1"];
    range.Value = "AssetType";
    range = wks2.Range["G1"];
    range.Value = "Manufacturer";
    range = wks2.Range["H1"];
    range.Value = "ModelNumber";
    range = wks2.Range["I1"];
    range.Value = "WarrantyGuarantorParts";
    range = wks2.Range["J1"];
    range.Value = "WarrantyDurationParts";
    range = wks2.Range["K1"];
    range.Value = "WarrantyGuarantorLabor";
    range = wks2.Range["L1"];
    range.Value = "WarrantyDurationLabor";
    range = wks2.Range["M1"];
    range.Value = "WarrantyDurationUnit";

```

```

range = wks2.Range["N1"];
range.Value = "ExtSystem";
range = wks2.Range["O1"];
range.Value = "ExtObject";
range = wks2.Range["P1"];
range.Value = "ExtIdentifier";
range = wks2.Range["Q1"];
range.Value = "ReplacementCost";
range = wks2.Range["R1"];
range.Value = "ExpectedLife";
range = wks2.Range["S1"];
range.Value = "DurationUnit";
range = wks2.Range["T1"];
range.Value = "WarrantyDescription";
string[] array = {"Casework", "ElectricalEquipment",
    "ElectricalFixtures", "Furniture", "Doors", "Windows", "FurnitureSystems", "GenericModels", "LightingFixtures", "MechanicalEquipment", "PlumbingFixtures", "SpecialtyEquipment", "Ducts", "Duct Fittings", "TelephoneDevices", "NurseCallDevices", "SecurityDevices", "CommunicationDevices", "DataDevices", "FireAlarmDevices", "LightingDevices", "Sprinklers"};
string sql;
//dao.Database db = accessApp.CurrentDb();
dao.Recordset rs = db.OpenRecordset("Select * from Doors");
int i = 0;
string helper = "";
foreach (string type in array)
{
    string test = type[type.Length - 1].ToString();
    if (test == "s")
    {
        helper = type.Substring(0, type.Length - 1);
        sql = "Select TypeName,FamilyName From " + type + ", " + helper + "Types Where " + type + ".TypeId = " + helper + "Types.Id Order By " + type + ".TypeId ASC";
    }
    else
    {
        sql = "Select TypeName,FamilyName From " + type + ", " + type + "Types Where " + type + ".TypeId = " + type + "Types.Id Order By " + type + ".TypeId ASC";
        rs = db.OpenRecordset(sql, Type.Missing, Type.Missing, Type.Missing);

        while (!rs.EOF)
        {
            range = wks2.Range["A" + (i + 2), Type.Missing];
            range.Value = rs.Fields["TypeName"].Value;

            range = wks2.Range["B" + (i + 2), Type.Missing];
            range.Value = "n/a";

            range = wks2.Range["C" + (i + 2), Type.Missing];
            range.Value = System.DateTime.Now.ToString();

            range = wks2.Range["D" + (i + 2), Type.Missing];
            range.Value = "n/a";

            range = wks2.Range["E" + (i + 2), Type.Missing];

```

```

range.Value = rs.Fields["TypeName"].Value;
range = wks2.Range["F" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["G" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["H" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["I" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["J" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["K" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["L" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["M" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["N" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["O" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["P" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["Q" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["R" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["S" + (i + 2), Type.Missing];
range.Value = "n/a";
range = wks2.Range["T" + (i + 2), Type.Missing];
range.Value = rs.Fields["FamilyName"].Value;
test = rs.Fields["TypeName"].Value;
rs.MoveNext();
while (!rs.EOF && test == rs.Fields["TypeName"].Value)
{
    rs.MoveNext();
}
i++;
}

```

```

range = wks2.Range["A2:A" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 6;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

```

```

range = wks2.Range["C2:F2", "C" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":F" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 6;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

```

```

range = wks2.Range["H2:M2", "H" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":M" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 6;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

```

```

        range = wks2.Range["B2:B" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
"G2:G" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
        range.Interior.ColorIndex = 40;
        range.Borders.Weight = Excel.XlBorderWeight.xlThin;

        range = wks2.Range["N2:P2", "N" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":P" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
        range.Interior.ColorIndex = 39;
        range.Borders.Weight = Excel.XlBorderWeight.xlThin;

        range = wks2.Range["Q2:T2", "Q" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":T" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
        range.Interior.ColorIndex = 35;
        range.Borders.Weight = Excel.XlBorderWeight.xlThin;
    }

    range = wks2.get_Range("A1", "T1");
    range.Interior.ColorIndex = 15;
    range.Orientation = Excel.XlOrientation.xlUpward;
    range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
    range.Font.Bold = true;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;

    range = wks2.Range["A2:A" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":T2,T2:T" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
    range.Columns.AutoFit();
    //AutoFitColumn(wks2, 100);

    rs.Close();
    System.Runtime.InteropServices.Marshal.ReleaseComObject(rs);
    rs = null;
    GC.Collect();
}

public void createSystemSheet(Excel.Workbook wkb2)
{
    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "System";
    Excel.Range range = wks2.Range["A1"];
    range.Value = "Name";
    range = wks2.Range["B1"];
    range.Value = "CreatedBy";
    range = wks2.Range["C1"];
    range.Value = "CreatedOn";
    range = wks2.Range["D1"];
    range.Value = "Category";
    range = wks2.Range["E1"];
    range.Value = "ComponentName";
    range = wks2.Range["F1"];
    range.Value = "ExtSystem";
}

```

```

range = wks2.Range["G1"];
range.Value = "ExtObject";
range = wks2.Range["H1"];
range.Value = "ExtIdentifier";
range = wks2.Range["I1"];
range.Value = "Description";
range = wks2.get_Range("A1", "L1");
range.Interior.ColorIndex = 15;
range.Orientation = Excel.XlOrientation.xlUpward;
range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
range.Font.Bold = true;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
GC.Collect();
}

```

```

public void createSpareSheet(Excel.Workbook wkb2)
{
    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "Spare";

    Excel.Range range = wks2.Range["A1"];
    range.Value = "Name";
    range = wks2.Range["B1"];
    range.Value = "CreatedBy";
    range = wks2.Range["C1"];
    range.Value = "CreatedOn";
    range = wks2.Range["D1"];
    range.Value = "Category";
    range = wks2.Range["E1"];
    range.Value = "TypeName";
    range = wks2.Range["F1"];
    range.Value = "Suppliers";
    range = wks2.Range["G1"];
    range.Value = "ExtSystem";
    range = wks2.Range["H1"];
    range.Value = "ExtObject";
    range = wks2.Range["I1"];
    range.Value = "ExtIdentifier";
    range = wks2.Range["J1"];
    range.Value = "Description";
    range = wks2.Range["K1"];
    range.Value = "SetNumber";
    range = wks2.Range["L1"];
    range.Value = "PartNumber";
    range = wks2.get_Range("A1", "L1");
    range.Interior.ColorIndex = 15;
    range.Orientation = Excel.XlOrientation.xlUpward;
    range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
    range.Font.Bold = true;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;
    GC.Collect();
}

```

```

public void createJobSheet(Excel.Workbook wkb2)
{
    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "Job";
    Excel.Range range = wks2.Range["A1"];
    range.Value = "Name";
    range = wks2.Range["B1"];
    range.Value = "CreatedBy";
    range = wks2.Range["C1"];
    range.Value = "CreatedOn";
    range = wks2.Range["D1"];
    range.Value = "Category";
    range = wks2.Range["E1"];
    range.Value = "Status";
    range = wks2.Range["F1"];
    range.Value = "TypeName";
    range = wks2.Range["G1"];
    range.Value = "Description";
    range = wks2.Range["H1"];
    range.Value = "Duration";
    range = wks2.Range["I1"];
    range.Value = "DurationUnit";
    range = wks2.Range["J1"];
    range.Value = "Start";
    range = wks2.Range["K1"];
    range.Value = "TaskStartUnit";
    range = wks2.Range["L1"];
    range.Value = "Frequency";
    range = wks2.Range["M1"];
    range.Value = "FrequencyUnit";
    range = wks2.Range["N1"];
    range.Value = "ExtSystem";
    range = wks2.Range["O1"];
    range.Value = "ExtObject";
    range = wks2.Range["P1"];
    range.Value = "ExtIdentifier";
    range = wks2.Range["Q1"];
    range.Value = "TaskNumber";
    range = wks2.Range["R1"];
    range.Value = "Priors";
    range = wks2.Range["S1"];
    range.Value = "ResourceNames";
    range = wks2.get_Range("A1", "S1");
    range.Interior.ColorIndex = 15;
    range.Orientation = Excel.XlOrientation.xlUpward;
    range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
    range.Font.Bold = true;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;
    GC.Collect();
}

public void createResourceSheet(Excel.Workbook wkb2)
{
    Excel.Worksheet wks2 = wkb2.Worksheets.Add();

```

```

wks2.Name = "Resource";
Excel.Range range = wks2.Range["A1"];
range.Value = "Name";
range = wks2.Range["B1"];
range.Value = "CreatedBy";
range = wks2.Range["C1"];
range.Value = "CreatedOn";
range = wks2.Range["D1"];
range.Value = "Category";
range = wks2.Range["E1"];
range.Value = "ExtSystem";
range = wks2.Range["F1"];
range.Value = "ExtObject";
range = wks2.Range["G1"];
range.Value = "ExtIdentifier";
range = wks2.Range["H1"];
range.Value = "Description";
range = wks2.get_Range("A1", "H1");
range.Interior.ColorIndex = 15;
range.Orientation = Excel.XlOrientation.xlUpward;
range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
range.Font.Bold = true;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
GC.Collect();
}
public void createDocumentSheet(Excel.Workbook wkb2)
{
    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "Document";
    Excel.Range range = wks2.Range["A1"];
    range.Value = "Name";
    range = wks2.Range["B1"];
    range.Value = "CreatedBy";
    range = wks2.Range["C1"];
    range.Value = "CreatedOn";
    range = wks2.Range["D1"];
    range.Value = "Category";
    range = wks2.Range["E1"];
    range.Value = "ApprovalBy";
    range = wks2.Range["F1"];
    range.Value = "Stage";
    range = wks2.Range["G1"];
    range.Value = "SheetName";
    range = wks2.Range["H1"];
    range.Value = "RowName";
    range = wks2.Range["I1"];
    range.Value = "Directory";
    range = wks2.Range["J1"];
    range.Value = "File";
    range = wks2.Range["K1"];
    range.Value = "ExtSystem";
    range = wks2.Range["L1"];
    range.Value = "ExtObject";
    range = wks2.Range["M1"];

```

```

range.Value = "ExtIdentifier";
range = wks2.Range["N1"];
range.Value = "Description";
range = wks2.Range["O1"];
range.Value = "Reference";
range = wks2.get_Range("A1", "O1");
range.Interior.ColorIndex = 15;
range.Orientation = Excel.XlOrientation.xlUpward;
range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
range.Font.Bold = true;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
GC.Collect();
}

```

```

public void createAttributeSheet(Excel.Workbook wkb2)
{
    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "Attribute";

    Excel.Range range = wks2.Range["A1"];
    range.Value = "Name";
    range = wks2.Range["B1"];
    range.Value = "CreatedBy";
    range = wks2.Range["C1"];
    range.Value = "CreatedOn";
    range = wks2.Range["D1"];
    range.Value = "Category";
    range = wks2.Range["E1"];
    range.Value = "SheetName";
    range = wks2.Range["F1"];
    range.Value = "RowName";
    range = wks2.Range["G1"];
    range.Value = "Value";
    range = wks2.Range["H1"];
    range.Value = "Unit";
    range = wks2.Range["I1"];
    range.Value = "ExtSystem";
    range = wks2.Range["J1"];
    range.Value = "ExtObject";
    range = wks2.Range["K1"];
    range.Value = "ExtIdentifier";
    range = wks2.Range["L1"];
    range.Value = "Description";
    range = wks2.get_Range("A1", "L1");
    range.Interior.ColorIndex = 15;
    range.Orientation = Excel.XlOrientation.xlUpward;
    range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
    range.Font.Bold = true;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;
    GC.Collect();
}

```

```

public void createCoordinateSheet(Excel.Workbook wkb2)
{

```

```

Excel.Worksheet wks2 = wkb2.Worksheets.Add();
wks2.Name = "Coordinate";
Excel.Range range = wks2.Range["A1"];
range.Value = "Name";
range = wks2.Range["B1"];
range.Value = "CreatedBy";
range = wks2.Range["C1"];
range.Value = "CreatedOn";
range = wks2.Range["D1"];
range.Value = "Category";
range = wks2.Range["E1"];
range.Value = "SheetName";
range = wks2.Range["F1"];
range.Value = "RowName";
range = wks2.Range["G1"];
range.Value = "CoordinateXAxis";
range = wks2.Range["H1"];
range.Value = "CoordinateYAxis";
range = wks2.Range["I1"];
range.Value = "CoordinateZAxis";
range = wks2.Range["J1"];
range.Value = "ExtSystem";
range = wks2.Range["K1"];
range.Value = "ExtObject";
range = wks2.Range["L1"];
range.Value = "ExtIdentifier";
range = wks2.get_Range("A1", "L1");
range.Interior.ColorIndex = 15;
range.Orientation = Excel.XlOrientation.xlUpward;
range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
range.Font.Bold = true;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
GC.Collect();
}

```

```

public void createConnectionSheet(Excel.Workbook wkb2)
{
    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "Connection";
    Excel.Range range = wks2.Range["A1"];
    range.Value = "Name";
    range = wks2.Range["B1"];
    range.Value = "CreatedBy";
    range = wks2.Range["C1"];
    range.Value = "CreatedOn";
    range = wks2.Range["D1"];
    range.Value = "ComponentName1";
    range = wks2.Range["E1"];
    range.Value = "ComponentName2";
    range = wks2.Range["F1"];
    range.Value = "ComponentType";
    range = wks2.Range["G1"];
    range.Value = "ExtSystem";
}

```

```

range = wks2.Range["H1"];
range.Value = "ExtObject";
range = wks2.Range["I1"];
range.Value = "ExtIdentifier";
range = wks2.Range["J1"];
range.Value = "Description";
range = wks2.get_Range("A1", "J1");
range.Interior.ColorIndex = 15;
range.Orientation = Excel.XlOrientation.xlUpward;
range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
range.Font.Bold = true;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
GC.Collect();
}

```

```

public void createIssueSheet(Excel.Workbook wkb2)
{

```

```

    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "Issue";
    Excel.Range range = wks2.Range["A1"];
    range.Value = "Name";
    range = wks2.Range["B1"];
    range.Value = "CreatedBy";
    range = wks2.Range["C1"];
    range.Value = "CreatedOn";
    range = wks2.Range["D1"];
    range.Value = "Type";
    range = wks2.Range["E1"];
    range.Value = "Risk";
    range = wks2.Range["F1"];
    range.Value = "Chance";
    range = wks2.Range["G1"];
    range.Value = "Impact";
    range = wks2.Range["H1"];
    range.Value = "SheetName1";
    range = wks2.Range["I1"];
    range.Value = "RowName1";
    range = wks2.Range["J1"];
    range.Value = "SheetName2";
    range = wks2.Range["K1"];
    range.Value = "RowName2";
    range = wks2.Range["L1"];
    range.Value = "Description";
    range = wks2.Range["M1"];
    range.Value = "Owner";
    range = wks2.Range["N1"];
    range.Value = "Mitigation";
    range = wks2.Range["O1"];
    range.Value = "ExtSystem";
    range = wks2.Range["P1"];
    range.Value = "ExtObject";
    range = wks2.Range["Q1"];
    range.Value = "ExtIdentifier";

```

```

range = wks2.get_Range("A1", "Q1");
range.Interior.ColorIndex = 15;
range.Orientation = Excel.XlOrientation.xlUpward;
range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
range.Font.Bold = true;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
GC.Collect();
}

public void createComponentSheet(Excel.Workbook wkb)
{
    int numOfRooms = 0;
    Excel.Worksheet wks2 = wkb.Worksheets.Add();
    wks2.Name = "Component";

    Excel.Range range = wks2.Range["A1"];
    range.Value = "Name";
    range = wks2.Range["B1"];
    range.Value = "CreatedBy";
    range = wks2.Range["C1"];
    range.Value = "CreatedOn";
    range = wks2.Range["D1"];
    range.Value = "TypeName";
    range = wks2.Range["E1"];
    range.Value = "Space";
    range = wks2.Range["F1"];
    range.Value = "Description";
    range = wks2.Range["G1"];
    range.Value = "ExtSystem";
    range = wks2.Range["H1"];
    range.Value = "ExtObject";
    range = wks2.Range["I1"];
    range.Value = "ExtIdentifier";
    range = wks2.Range["J1"];
    range.Value = "SerialNumber";
    range = wks2.Range["K1"];
    range.Value = "InstallationDate";
    range = wks2.Range["L1"];
    range.Value = "WarrantyStartDate";
    range = wks2.Range["M1"];
    range.Value = "TagNumber";
    range = wks2.Range["N1"];
    range.Value = "BarCode";
    range = wks2.Range["O1"];
    range.Value = "AssetIdentifier";
    range = wks2.get_Range("A1", "O1");
    range.Interior.ColorIndex = 15;
    range.Orientation = Excel.XlOrientation.xlUpward;
    range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
    range.Font.Bold = true;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;
}

```

```

//string sql = "Select * From CobieDoorComponent ORDER BY Id Asc";
string sql = "SELECT DoorTypes.FamilyName, DoorTypes.TypeName, Doors.Id,
Rooms.Number From Doors, Rooms, RoomFromToAssociations, DoorTypes Where Doors.Id =
RoomFromToAssociations.Id AND RoomFromToAssociations.FromRoom = Rooms.Id AND
Doors.TypeId = DoorTypes.Id Order By Doors.Id ASC;"
+ "UNION SELECT DoorTypes.FamilyName, DoorTypes.TypeName,
Doors.Id,Rooms.Number From Doors, Rooms, RoomFromToAssociations, DoorTypes Where Doors.Id =
RoomFromToAssociations.Id AND RoomFromToAssociations.ToRoom = Rooms.Id AND Doors.TypeId
= DoorTypes.Id Order By Doors.Id ASC;";

//dao.Database db = accessApp.CurrentDb();
dao.Recordset rs = db.OpenRecordset(sql,Type.Missing, Type.Missing,Type.Missing);
int rc = rs.RecordCount;
int i;
for (i = 0; i < rs.RecordCount; i++)
{
    int id = rs.Fields["Id"].Value;
    rs.Close();
    rs = db.OpenRecordset("SELECT DoorTypes.FamilyName, DoorTypes.TypeName,
Doors.Id,Rooms.Number From Doors, Rooms, RoomFromToAssociations, DoorTypes Where Doors.Id =
RoomFromToAssociations.Id AND RoomFromToAssociations.FromRoom = Rooms.Id AND
Doors.TypeId = DoorTypes.Id AND Doors.Id = " + id + " Order By Doors.Id ASC;"
+ "UNION SELECT DoorTypes.FamilyName, DoorTypes.TypeName,Rooms.Number From
Doors, Rooms, RoomFromToAssociations, DoorTypes Where Doors.Id = RoomFromToAssociations.Id
AND RoomFromToAssociations.ToRoom = Rooms.Id AND Doors.TypeId = DoorTypes.Id AND
Doors.Id = " + id + " Order By Doors.Id ASC;" );

    range = wks2.Range["A" + (i + 2), Type.Missing];
    range.Value = rs.Fields["FamilyName"].Value + ":";
    range.Value += rs.Fields["TypeName"].Value;

    range = wks2.Range["B" + (i + 2), Type.Missing];
    // if (rs.Fields["CreatedBy"].Value == null)
    range.Value = "N/A";
    //else
    // range.Value = rs.Fields["CreatedBy"].Value;

    range = wks2.Range["C" + (i + 2), Type.Missing];
    //if (rs.Fields["CreateDate"].Value == null)
    range.Value = "N/A";
    //else
    // range.Value = rs.Fields["CreateDate"].Value;

    range = wks2.Range["D" + (i + 2), Type.Missing];
    range.Value = rs.Fields["TypeName"].Value;

    range = wks2.Range["E" + (i + 2), Type.Missing];
    for (int a = 0; a < rs.RecordCount; a++)
    {
        range.Value += rs.Fields["Number"].Value + ",";
        rs.MoveNext();
    }
    rs.MovePrevious();
}

```

```

range = wks2.Range["F" + (i + 2), Type.Missing];
range.Value = "N/A";
range = wks2.Range["G" + (i + 2), Type.Missing];
range.Value = ""; //rs.Fields["ExtSystem"].Value;
range = wks2.Range["H" + (i + 2), Type.Missing];
range.Value = ""; //rs.Fields["ExtObject"].Value;
range = wks2.Range["I" + (i + 2), Type.Missing];
range.Value = rs.Fields["Id"].Value;
range = wks2.Range["J" + (i + 2), Type.Missing];
range.Value = "N/A";
range = wks2.Range["K" + (i + 2), Type.Missing];
range.Value = "N/A";
range = wks2.Range["L" + (i + 2), Type.Missing];
range.Value = "N/A";
range = wks2.Range["M" + (i + 2), Type.Missing];
range.Value = "N/A";
range = wks2.Range["N" + (i + 2), Type.Missing];
//range.Value = rs.Fields["BarCode"].Value;
range.Value = "N/A";
range = wks2.Range["O" + (i + 2), Type.Missing];
range.Value = "";
numOfRooms += rs.RecordCount;
rs = db.OpenRecordset(sql, Type.Missing, Type.Missing, Type.Missing);
for (int a = 0; a < numOfRooms; a++)
{
    rs.MoveNext();
}
if (numOfRooms == 92)
    break;
}
// rs.Close();
string helper = "";
string[] array = {"Casework", "ElectricalEquipment",
"ElectricalFixtures", "Furniture", "FurnitureSystems", "GenericModels", "LightingFixtures", "MechanicalEq
uipment", "PlumbingFixtures", "SpecialtyEquipment", "Ducts", "DuctFittings", "TelephoneDevices", "Nurse
CallDevices", "SecurityDevices", "CommunicationDevices", "DataDevices", "FireAlarmDevices", "Lighting
Devices", "Sprinklers"};
foreach (string table in array)
{
    rs.Close();
    string test = table[table.Length - 1].ToString();
    if (test == "s")
    {
        helper = table.Substring(0, table.Length - 1);
        sql = "SELECT " + helper + "Types.FamilyName, " + helper + "Types.TypeName, " + table
+ ".Id, Rooms.Number From Rooms, RoomAssociations, " + table + ", " + helper + "Types Where " +
table + ".Id = RoomAssociations.Id AND RoomAssociations.RoomId = Rooms.Id AND " +
        helper + "Types.Id = " + table + ".TypeId";
    }
    else

```

```

sql = "SELECT " + table + "Types.FamilyName, " + table + "Types.TypeName, " + table +
".Id, Rooms.Number From Rooms, RoomAssociations, " + table + ", " + table + "Types Where " + table +
".Id = RoomAssociations.Id AND RoomAssociations.RoomId = Rooms.Id AND " +
table + "Types.Id = " + table + ".TypeId";

```

```

rs = db.OpenRecordset(sql, Type.Missing, Type.Missing, Type.Missing);
while (!rs.EOF)
{
    range = wks2.Range["A" + (i + 2), Type.Missing];
    range.Value = rs.Fields["FamilyName"].Value + ":";
    range.Value += rs.Fields["TypeName"].Value + ":" + rs.Fields["TypeName"].Value;

    range = wks2.Range["B" + (i + 2), Type.Missing];
    //if (rs.Fields["CreatedBy"].Value == null)
        range.Value = "N/A";
    //else
        // range.Value = rs.Fields["CreatedBy"].Value;
    range = wks2.Range["C" + (i + 2), Type.Missing];
    //if (rs.Fields["CreatedDate"].Value == null)
        range.Value = "N/A";
    // else
        // range.Value = rs.Fields["CreatedDate"].Value;
    range = wks2.Range["D" + (i + 2), Type.Missing];
    range.Value = rs.Fields["TypeName"].Value;
    range = wks2.Range["E" + (i + 2), Type.Missing];
    range.Value += rs.Fields["Number"].Value;
    range = wks2.Range["F" + (i + 2), Type.Missing];
    range.Value = "N/A";
    range = wks2.Range["G" + (i + 2), Type.Missing];
    range.Value = "";
    // range.Value = rs.Fields["ExtSystem"].Value;
    range = wks2.Range["H" + (i + 2), Type.Missing];
    range.Value = "";
    //range.Value = rs.Fields["ExtObject"].Value;
    range = wks2.Range["I" + (i + 2), Type.Missing];
    range.Value = rs.Fields["Id"].Value;
    // range.Value = rs.Fields["ExtIdentifier"].Value;
    range = wks2.Range["J" + (i + 2), Type.Missing];
    range.Value = "N/A";
    range = wks2.Range["K" + (i + 2), Type.Missing];
    range.Value = "N/A";
    range = wks2.Range["L" + (i + 2), Type.Missing];
    range.Value = "N/A";
    range = wks2.Range["M" + (i + 2), Type.Missing];
    range.Value = "N/A";
    range = wks2.Range["N" + (i + 2), Type.Missing];
    range.Value = "";
    // range.Value = rs.Fields["BarCode"].Value;
    range = wks2.Range["O" + (i + 2), Type.Missing];
    range.Value = "N/A";
    i++;
    rs.MoveNext();
}

```

```

    }

    range = wks2.Range["A2:A" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ",C2:C"
+ excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ",F2:F" +
excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
    range.Interior.ColorIndex = 6;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;

    range = wks2.Range["G2:I2", "G" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ":I"
+ excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
    range.Interior.ColorIndex = 39;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;

    range = wks2.Range["B2:B" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ",D2:D"
+ excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ",E2:E" +
excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
    range.Interior.ColorIndex = 40;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;

    range = wks2.Range["J2:O2", "J" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ":O"
+ excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
    range.Interior.ColorIndex = 35;
    range.Borders.Weight = Excel.XlBorderWeight.xlThin;

    range = wks2.Range["A2:A" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":O2,O2:O" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
    range.Columns.AutoFit();

    rs.Close();
    System.Runtime.InteropServices.Marshal.ReleaseComObject(rs);
    rs = null;
    GC.Collect();
}

public void createContactSheet(Excel.Workbook wkb2)
{
    Excel.Worksheet wks2 = wkb2.Worksheets.Add();
    wks2.Name = "Contact";
    Excel.Range range = wks2.get_Range("A1", Type.Missing);
    range.Value = "Email";
    range = wks2.get_Range("B1", Type.Missing);
    range.Value = "CreatedBy";
    range = wks2.get_Range("C1", Type.Missing);
    range.Value = "CreatedOn";
    range = wks2.get_Range("D1", Type.Missing);
    range.Value = "Category";
    range = wks2.get_Range("E1", Type.Missing);
    range.Value = "Company";
    range = wks2.get_Range("F1", Type.Missing);
    range.Value = "Phone";
    range = wks2.get_Range("G1", Type.Missing);
    range.Value = "ExtSystem";
    range = wks2.get_Range("H1", Type.Missing);

```

```

range.Value = "ExtObject";
range = wks2.get_Range("I1", Type.Missing);
range.Value = "ExtIdentifier";
range = wks2.get_Range("J1", Type.Missing);
range.Value = "Department";
range = wks2.get_Range("K1", Type.Missing);
range.Value = "OrganizationCode";
range = wks2.get_Range("L1", Type.Missing);
range.Value = "GivenName";
range = wks2.get_Range("M1", Type.Missing);
range.Value = "FamilyName";
range = wks2.get_Range("N1", Type.Missing);
range.Value = "Street";
range = wks2.get_Range("O1", Type.Missing);
range.Value = "PostalBox";
range = wks2.get_Range("P1", Type.Missing);
range.Value = "Town";
range = wks2.get_Range("Q1", Type.Missing);
range.Value = "StateRegion";
range = wks2.get_Range("R1", Type.Missing);
range.Value = "PostalCode";
range = wks2.get_Range("S1", Type.Missing);
range.Value = "Country";
range = wks2.get_Range("A1", "S1");
range.Interior.ColorIndex = 15;
range.Orientation = Excel.XlOrientation.xlUpward;
range.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;
range.Font.Bold = true;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;
range = wks2.get_Range("A2", Type.Missing);
range = range.get_Resize(2, 19);
string sql = "SELECT Type, Company, Phone, Contact From Contacts";
//dao.Database db = accessApp.CurrentDb();
dao.Recordset rs = db.OpenRecordset(sql, Type.Missing, Type.Missing, Type.Missing);
// int rc = rs.RecordCount;
int i = 0;;
while (!rs.EOF)
{
    range = wks2.Range["D" + (i + 2), Type.Missing];
    range.Value = rs.Fields["Type"].Value;
    range = wks2.Range["E" + (i + 2), Type.Missing];
    range.Value = rs.Fields["Company"].Value;
    range = wks2.Range["F" + (i + 2), Type.Missing];
    range.Value = rs.Fields["Phone"].Value;
    range = wks2.Range["L" + (i + 2), Type.Missing];
    object name = rs.Fields["Contact"].Value;
    int d = name.ToString().IndexOf(',');
    if (d != -1)
        range.Value = name.ToString().Substring(d+1);
    else
        range.Value = rs.Fields["Contact"];

    range = wks2.Range["M" + (i + 2), Type.Missing];
    if (d != -1)

```

```

        range.Value = name.ToString().Substring(0, d);
    else
        range.Value = "";
    rs.MoveNext();
    i++;
}
rs.Close();
/* double[,] saRet = new double[2, 19];
for (long iRow = 0; iRow < 2; iRow++)
{
    for (long iCol = 0; iCol < 19; iCol++)
    {
        saRet[iRow, iCol] = iRow * iCol;
    }
}

*/

//range.set_Value(Type.Missing, saRet);
range = wks2.Range["A2:A" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ",C2:C"
+ excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ":F2,F2:F" +
excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 6;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

range = wks2.Range["B2:B" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 40;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

range = wks2.Range["G2:G" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ":I2,I2:I"
+ excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 39;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

range = wks2.Range["J2:J" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count + ":S2,S2:S"
+ excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Interior.ColorIndex = 35;
range.Borders.Weight = Excel.XlBorderWeight.xlThin;

range = wks2.Range["A2:A" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count +
":S2,S2:S" + excelApp.ActiveCell.CurrentRegion.Cells.Rows.Count];
range.Columns.AutoFit();

System.Runtime.InteropServices.Marshal.ReleaseComObject(wks2);
System.Runtime.InteropServices.Marshal.ReleaseComObject(range);
System.Runtime.InteropServices.Marshal.ReleaseComObject(rs);
rs = null;
wks2 = null;
range = null;

}

}

}

```

AccessGUI.cs

The following contains all of the GUI control actions (button clicks, text entry, etc)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;
using System.IO;

namespace Access_To_Cobie
{
    public partial class AccessGUI : Form
    {
        Excel.Application excelApp;
        Access2Cobie bridge;
        String swimDir;
        String accessDir;

        public AccessGUI()
        {
            InitializeComponent();
        }

        private void createCobie_Click(object sender, EventArgs e)
        {
            if (textBox1.Text == "")
            {
                MessageBox.Show("You must select an access database");
                return;
            }
            progressBar1.Visible = true;
            excelApp = new Excel.Application();
            bridge = new Access2Cobie(@textBox1.Text, excelApp);
            // excelApp.Visible = true;

            Excel.Workbook wkb = excelApp.Workbooks.Add();
            bridge.createIssueSheet(wkb);
            progressBar1.Increment(6);
            bridge.createConnectionSheet(wkb);
            progressBar1.Increment(6);
            bridge.createCoordinateSheet(wkb);
            progressBar1.Increment(6);
            bridge.createAttributeSheet(wkb);
            progressBar1.Increment(6);
            bridge.createDocumentSheet(wkb);
            progressBar1.Increment(6);
            bridge.createJobSheet(wkb);
        }
    }
}
```

```

        progressBar1.Increment(6);
        bridge.createResourceSheet(wkb);
        progressBar1.Increment(6);
        bridge.createSpareSheet(wkb);
        progressBar1.Increment(6);
        bridge.createSystemSheet(wkb);
        progressBar1.Increment(6);
        bridge.createComponentSheet(wkb);
        progressBar1.Increment(6);
        bridge.createTypeSheet(wkb);
        progressBar1.Increment(6);
        bridge.createZoneSheet(wkb);
        progressBar1.Increment(6);
        bridge.createSpaceSheet(wkb);
        progressBar1.Increment(6);
        bridge.createFloorSheet(wkb);
        progressBar1.Increment(6);
        bridge.createFacilitySheet(wkb);
        progressBar1.Increment(6);
        bridge.createContactSheet(wkb);
        progressBar1.Increment(6);
        wkb.SaveCopyAs(Excel.XlFileFormat.xlExcel8);
        wkb.Close();
        System.Runtime.InteropServices.Marshal.ReleaseComObject(wkb);
        wkb = null;
        GC.Collect();
        excelApp.Quit();
        System.Runtime.InteropServices.Marshal.ReleaseComObject(excelApp);
        excelApp = null;
        MessageBox.Show("The spreadsheet was successfully created!");
        button3.Visible = true;
        progressBar1.Visible = false;
    }

    private void loadDB_Click(object sender, EventArgs e)
    {
        if (openFileDialog1.ShowDialog() == DialogResult.OK)
        {
            textBox1.Text = openFileDialog1.FileName;
            accessDir = textBox1.Text;
        }
        if (textBox1.Text == String.Empty)
        {
            MessageBox.Show("You didn't select a file");
        }
        else
        {
            createCobie.Visible = true;
        }
    }

    private void AccessGUI_Load(object sender, EventArgs e)
    {

```

```

    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (swimDir == null)
        {
            MessageBox.Show("You have not selected the SWIM Directory");
            return;
        }
        if (openFileDialog2.ShowDialog() == DialogResult.OK)
        {
            pictureBox3.Load(openFileDialog2.FileName);
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        String[] files =
{"ContactInfo", "Contacts", "Contacts2", "Division", "DocumentInfo", "Documents", "Documents2", "Documents3", "EqResults", "EquipmentInfo", "Home", "Search", "Space", "System", "uploadDocument", "uploadDocumentResults", "User Guide"};
        if (swimDir == null)
        {
            MessageBox.Show("You have not selected the SWIM Directory");
            return;
        }
        StringBuilder finalFile;
        string temp;
        string fileExtension;
        string[] file;
        foreach (String s in files)
        {
            temp = "";
            fileExtension = "";
            finalFile = new StringBuilder();
            file = File.ReadAllLines(swimDir + "\\baseFiles\\base" + s + ".html");

            int i = 0;
            foreach (string line in file)
            {
                if (line.Contains("Project Name Here"))
                {
                    if (i > 0)
                        temp = line.Replace("Project Name Here", textBox2.Text);
                    else
                        temp = line.Replace("Project Name Here", textBox7.Text);

                    finalFile.Append(temp + "\r\n");

                    i++;

                    continue;
                }
            }
        }
    }

```

```

else if (line.Contains("Subproject Name Here"))
{
    temp = line.Replace("Subproject Name Here", textBox5.Text);
    finalFile.Append(temp + "\r\n");
    continue;
}
if (line.Contains("defaultimage.png"))
{
    fileExtension = System.IO.Path.GetExtension(pictureBox3.ImageLocation);
    File.Copy(pictureBox3.ImageLocation, swimDir + "\\Images\\projectimage" +
fileExtension, true);
    //File.Copy(pictureBox3.ImageLocation, "../SWIM/public_html/Images");
    temp = line.Replace("defaultimage.png", "projectimage" + fileExtension);
    finalFile.Append(temp + "\r\n");
    continue;
}

    finalFile.Append(line + "\r\n");
}

File.WriteAllText(swimDir + "\\\" + s + ".html", finalFile.ToString());
}
MessageBox.Show("Web Frontend Created");
}

private void button3_Click(object sender, EventArgs e)
{
    if (folderBrowserDialog1.ShowDialog() == DialogResult.OK)
    {
        swimDir = folderBrowserDialog1.SelectedPath;
        MessageBox.Show("SWIM Directory is " + swimDir);
    }
    else
    {
        MessageBox.Show("You did not select a SWIM directory");
        return;
    }

    button4.Visible = true;
}

private void button4_Click(object sender, EventArgs e)
{
    if (swimDir == null)
    {
        MessageBox.Show("You have not selected the SWIM Directory");
        return;
    }

    if (accessDir == null)
    {
        MessageBox.Show("You have not selected the Access DB");
        return;
    }
}

```

```

        File.Copy(accessDir, swimDir + "\\test.mdb", true);

        MessageBox.Show("File Copied Successfully");
        groupBox2.Visible = true;
        groupBox1.Visible = false;
    }

}
}

```

SymbolList.cs

The following file was altered in RDB Link (PopulateDbRow) so that it would not overwrite content that was added to the Access Database.

```

//
// (C) Copyright 2003-2008 by Autodesk, Inc.
//
// Permission to use, copy, modify, and distribute this software in
// object code form for any purpose and without fee is hereby granted,
// provided that the above copyright notice appears in all copies and
// that both that copyright notice and the limited warranty and
// restricted rights notice below appear in all supporting
// documentation.
//
// AUTODESK PROVIDES THIS PROGRAM "AS IS" AND WITH ALL FAULTS.
// AUTODESK SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTY OF
// MERCHANTABILITY OR FITNESS FOR A PARTICULAR USE. AUTODESK, INC.
// DOES NOT WARRANT THAT THE OPERATION OF THE PROGRAM WILL BE
// UNINTERRUPTED OR ERROR FREE.
//
// Use, duplication, or disclosure by the U.S. Government is subject to
// restrictions set forth in FAR 52.227-19 (Commercial Computer
// Software - Restricted Rights) and DFAR 252.227-7013(c)(1)(ii)
// (Rights in Technical Data and Computer Software), as applicable.

// SymbolList.cs

using System;
using System.Collections.Generic;
using System.Text;
using Autodesk.Revit;
using System.Data;
using Autodesk.Revit.Parameters;

namespace Revit.SDK.Samples.RDBLink.CS
{
    /// <summary>
    /// Provides export and import functions for Revit symbols.
    /// </summary>
    public class SymbolList : ElementList
    {
        #region Fields

```

```

/// <summary>
/// Used for the name suffix of type name when create a new type if
/// the type name is duplicated
/// </summary>
int m_numberSuffix = 2;
#endregion

#region Properties
/// <summary>
/// whether user can create a new record in a table
/// </summary>
public override bool SupportCreate
{
    get
    {
        if (Elements.IsEmpty)
        {
            return false;
        }
        else
        {
            return true;
        }
    }
}
#endregion

#region Methods
/// <summary>
/// Insert data of an element to a DataRow.
/// </summary>
/// <param name="element">element to retrieve data.</param>
/// <param name="row">DataRow to populate.</param>
protected override void PopulateDbRow(Element element, DataRow row)
{
    foreach (ColumnInfo colInfo in TableInfo)
    {
        // if it is a built-in parameter, using BuiltInParameter to
        // get data
        if (colInfo.BuiltInParameter != BuiltInParameter.INVALID)
        {
            BuiltInParameter bip = colInfo.BuiltInParameter;
            Parameter parameter = element.get_Parameter(bip);
            if (parameter != null)
            {
                object value = GetParameterDdValue(parameter);
                if (value.ToString() != row[colInfo.Name].ToString() && row[colInfo.Name].ToString()
!= "")
                {
                    continue;
                }
                row[colInfo.Name] = GetParameterDdValue(parameter);
            }
        }
        // if it is not a built-in parameter, using definition name

```

```

        // to get data
        else
        {

            Parameter param = GetParameterByDefinitionName(element, colInfo.Name);
            object value = GetParameterDdValue(param);

            if (value.ToString() != row[colInfo.Name].ToString() && row[colInfo.Name].ToString() !=
""")
                continue;
            else
                row[colInfo.Name] = GetParameterDdValue(param);
        }
    }
}

/// <summary>
/// Import data from dataTable
/// </summary>
/// <param name="dataTable">dataTable imported</param>
/// <returns>ErrorTable which preserves the importing results</returns>
public override ErrorTable Import(DataTable dataTable)
{
    // create error data for table
    ErrorTable et = new ErrorTable(TableInfo.TableId, TableInfo.Name, dataTable);
    // iterate each row
    foreach (DataRow row in dataTable.Rows)
    {
        // skip if it is deleted
        if (row.RowState == DataRowState.Deleted) continue;
        // find the corresponding element in Revit
        Element elem = FindCorrespondingObject(row) as Element;

        ErrorRow er = null;
        // no corresponding element found
        if (elem == null)
        {
            // symbol tables support create elements
            elem = CreateNewElement(row);
            // create failed, we need to delete the row
            if (elem == null)
            {
                // create error data for row
                er = new ErrorRow(row);
                // set its state to deleted
                er.State = DataRowState.Deleted;
                // flag it with HasChange to be true
                er.HasChange = true;
                // flag each cell to Equals
                for (int colIndex = 0; colIndex < dataTable.Columns.Count; colIndex++)
                {
                    er.Cells.Add(new ErrorCell(dataTable.Columns[colIndex].ColumnName,
UpdateResult.Equals));
                }
            }
        }
    }
}

```

```

    }
    // if create successfully
    else
    {
        // update the new element
        er = UpdateOneElement(elem, row);
        // flag its state to new added
        er.State = DataRowState.Added;
        // flag it with HasChange to be true
        er.HasChange = true;
    }
}
// we found the corresponding element
else
{
    // update the new element
    er = UpdateOneElement(elem, row);
}

// add row error data to table error data
et.Rows.Add(er);
if (!et.HasChange && er.HasChange)
{
    // mark its change state
    et.HasChange = true;
}
}

return et;
}

/// <summary>
/// Create an element in Revit using the information stored in a DataRow
/// </summary>
/// <param name="row">DataRow stores the creation information</param>
/// <returns>An element if create successfully, otherwise null</returns>
protected virtual Element CreateNewElement(DataRow row)
{
    try
    {
        // get "FaimlyName" and "TypeName" and get their column name
        string colFamilyName = ColumnRes("ColN_BP_ALL_MODEL_FAMILY_NAME");
        string colTypeName = ColumnRes("ColN_BP_ALL_MODEL_TYPE_NAME");
        string familyName = row[colFamilyName].ToString();
        string typeName = row[colTypeName].ToString();

        // make sure family name and type name not be null or Empty.
        if (familyName == null || typeName == null ||
            familyName == string.Empty || typeName == string.Empty)
            return null;

        // following code is to make sure family name and type name is unique
        string tmpTypeName = typeName;
        while(true)

```

```

{
    // replace "" to "" to avoid sql keywords
    string sqlTypeName = typeName.Replace("", "");
    string sqlFamilyName = familyName.Replace("", "");

    // if a duplicated row exists, give another new name for type
    if (row.Table.Select(string.Format("[{0}] = '{1}' and [{2}] = '{3}'",
        colFamilyName, sqlFamilyName, colTypeName, sqlTypeName)).Length > 1)
    {
        typeName = tmpTypeName + " " + m_numberSuffix;
        row[colTypeName] = typeName;
        ++m_numberSuffix;
    }
    else
        break;
}

// find a symbol to duplicate
Parameter familyNameParam = null;
Element rightSymbol = null;
foreach (Element elem in this.Elements)
{
    if (rightSymbol == null) rightSymbol = elem;

    familyNameParam =
elem.get_Parameter(BuiltInParameter.ALL_MODEL_FAMILY_NAME);
    if (familyNameParam == null) continue;
    if (familyNameParam.AsString().Equals(familyName))
    {
        rightSymbol = elem;
        break;
    }
}

// if can't find any Symbol to duplicate, null will be returned.
if (rightSymbol == null) return null;

Symbol symbolUsetoDuplicate = rightSymbol as Symbol;

// duplicate the symbol with a new name
Element duplicatedSymbol = symbolUsetoDuplicate.Duplicate(typeName);
this.Elements.Insert(duplicatedSymbol);

// update its new id to database
row[ColumnRes(TableInfo.PrimaryKeys[0])] = duplicatedSymbol.Id.Value;
return duplicatedSymbol;
}
catch
{
    // if data table haven't FamilyName column or the TypeName is repeated,
    // duplication will fail. so null will be returned.
    return null;
}
}

```

```

/// <summary>
/// Update an element with a DataRow
/// </summary>
/// <param name="elem">element to update</param>
/// <param name="row">row used to update</param>
/// <returns>ErrorRow which records the update results</returns>
private ErrorRow UpdateOneElement(Element elem, DataRow row)
{
    ErrorRow er = new ErrorRow(row);

    Parameter param = null;
    object paramValue = null;
    foreach (ColumnInfo colInfo in this.TableInfo)
    {
        // get parameter

        // for built-in parameter
        if (colInfo.BuiltInParameter != BuiltInParameter.INVALID)
        {
            param = elem.get_Parameter(colInfo.BuiltInParameter);
        }
        // for shared or project parameter
        else
        {
            param = GetParameterByDefinitionName(elem, colInfo.Name);
        }

        paramValue = row[colInfo.Name];

        // we can't get "Assembly Codes", deal with it separately
        UpdateResult ur = (colInfo.BuiltInParameter == BuiltInParameter.UNIFORMAT_CODE ?
            UpdateResult.AssemblyCode :
            // update the parameter with the cell value
            UpdateParameter(param, paramValue));
        // create cell error data
        ErrorCell ec = new ErrorCell(colInfo.Name, ur);
        er.Cells.Add(ec);

        // set HasChange property to row error data
        switch (ur)
        {
            case UpdateResult.Unknown:
            case UpdateResult.Success:
            case UpdateResult.Failed:
            case UpdateResult.Exception:
            case UpdateResult.ParameterNull:
            case UpdateResult.ReadOnlyFailed:
                er.HasChange = true;
                break;
            case UpdateResult.AssemblyCode:
            case UpdateResult.Equals:

        default:
    }
}

```

```

        break;
    }
}

return er;
}

/// <summary>
/// Update a designated parameter with some value
/// </summary>
/// <param name="param">Parameter to update</param>
/// <param name="paramValue">value used to update</param>
/// <returns>updated result</returns>
private UpdateResult UpdateParameter(Parameter param, object paramValue)
{
    try
    {
        //whether parameter is null
        bool parameterIsNull = (param == null);
        //whether value is null or DBNull
        bool valueIsNull = IsNullOrDBNull(paramValue);

        //parameter is null
        if (parameterIsNull)
        {
            if (valueIsNull) return UpdateResult.Equals;
            else return UpdateResult.ParameterNull;
        }
        //parameter is not null
        else
        {
            //whether parameter's value is empty
            bool parameterValueIsEmpty = IsParameterValueEmpty(param);
            //whether value is empty
            bool valueIsEmpty = IsValueEmpty(param.StorageType, paramValue);

            //parameter's value is empty
            if (parameterValueIsEmpty)
            {
                if (valueIsNull || valueIsEmpty) return UpdateResult.Equals;

                //parameter is read-only
                if (param.IsReadOnly) return UpdateResult.ReadOnlyFailed;

                //value is not empty, set the parameter with the value
                bool res1 = SetParameter(param, paramValue);
                return res1 ? UpdateResult.Success : UpdateResult.Failed;
            }
            //parameter's value is not empty
            else
            {
                //value is null or empty
                if (valueIsNull || valueIsEmpty)
                {

```

```

        if (param.IsReadOnly) return UpdateResult.ReadOnlyFailed;

        // value is empty or null, set the parameter value to empty
        bool res1 = SetParameterValueToEmpty(param);
        return res1 ? UpdateResult.Success : UpdateResult.Failed;
    }
    //value is not null or empty and parameter's value is not empty
    else
    {
        //verify whether value equals to parameter's value
        bool valueEquals = false;
        switch (param.StorageType)
        {
            case StorageType.Double:
                double doubleValue = param.AsDouble();
                try
                {
                    double unitValue = Unit.CovertFromAPI(param.DisplayUnitType,
doubleValue);

                    valueEquals = unitValue.Equals((Double)paramValue);
                }
                catch
                {
                    valueEquals = doubleValue.Equals((Double)paramValue);
                }
                break;
            case StorageType.ElementId:
                valueEquals = param.AsElementId().Value.Equals((int)paramValue);
                break;
            case StorageType.Integer:
                valueEquals = param.AsInteger().Equals((int)paramValue);
                break;
            case StorageType.None:
                string value = param.AsValueString();
                if (value == null)
                    valueEquals = (paramValue == null || paramValue == DBNull.Value) ? true :
false;
                else
                    valueEquals = value.Equals(paramValue.ToString());
                break;
            case StorageType.String:
                string value2 = param.AsString();
                if (value2 == null)
                    valueEquals = (paramValue == null || paramValue == DBNull.Value) ? true :
false;
                else
                    valueEquals = value2.Equals(paramValue.ToString());
                break;
            default:
                break;
        }

        // if equals return Equals
        if (valueEquals) return UpdateResult.Equals;
    }

```



```

    {
        case StorageType.Double:
            return parameter.AsDouble() == paraDoubleNull;
        case StorageType.ElementId:
            return (parameter.AsElementId().Value == paraElementIdValueNull);
        case StorageType.Integer:
            return (parameter.AsInteger() == paraIntegerNull);
        case StorageType.None:
            string val = parameter.AsValueString();
            return (string.IsNullOrEmpty(val));
        case StorageType.String:
            val = parameter.AsString();
            return (string.IsNullOrEmpty(val));
        default:
            return false;
    }
}

```

```

/// <summary>
/// Set parameter to some specific empty value
/// </summary>
/// <param name="param">Parameter to set</param>
/// <returns>true if set successfully otherwise false</returns>
private bool SetParameterValueToEmpty(Parameter param)
{

```

```

    switch (param.StorageType)
    {
        case StorageType.Double:
            return param.Set(paraDoubleNull);
        case StorageType.ElementId:
            ElementId eid;
            eid.Value = paraElementIdValueNull;
            return param.Set(ref eid);
        case StorageType.Integer:
            return param.Set(paraIntegerNull);
        case StorageType.None:
            return param.Set(paraStringNull);
        case StorageType.String:
            return param.Set(paraStringNull);
        default:
            return false;
    }
}

```

```

/// <summary>
/// Set parameter to a designated value
/// </summary>
/// <param name="parameter">Parameter to set</param>
/// <param name="value">value to set to</param>
/// <returns>true if set successfully otherwise false</returns>
private bool SetParameter(Parameter parameter, object value)
{
    switch (parameter.StorageType)
    {

```

```

        case StorageType.Double:
            try
            {
                Double apiValue = Unit.CovertToAPI((double)value, parameter.DisplayUnitType);
                return parameter.Set(apiValue);
            }
            catch
            {
                return parameter.Set((Double)value);
            }
        case StorageType.ElementId:
            ElementId eId = new ElementId();
            eId.Value = (int)value;
            return parameter.Set(ref eId);
        case StorageType.Integer:
            return parameter.Set((int)value);
        case StorageType.None:
            return parameter.SetValueString(value.ToString());
        case StorageType.String:
            return parameter.Set(value.ToString());
        default:
            return false;
    }
}

/// <summary>
/// Whether the parameter value is null or DBNull
/// </summary>
/// <param name="paramValue">Parameter value</param>
/// <returns>true if yes otherwise false</returns>
private bool IsNullOrDBNull(object paramValue)
{
    return paramValue == null || paramValue == DBNull.Value;
}

/// <summary>
/// Verify whether an element belongs to a specific table.
/// </summary>
/// <param name="element">element to verify.</param>
/// <returns>true if the element belongs to the table, otherwise false.</returns>
protected override bool CanAppend(Element element)
{
    BuiltInCategory bic = GetCategoryId(element);
    ElementType et = element is Symbol ? ElementType.SYMBOL : ElementType.INSTANCE;
    return TableInfo.TableId.Equals("TabN_" + bic.ToString() + "_" + et.ToString());
}
#endregion
};
}

```

Appendix B – End User Code

HTML Files

All of these files are the html code for the different pages in the site.

ContactInfo.html

View an individual contact record in detail, optionally editing it.

```
<!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>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>COBie SWIM</title>
  <link rel="icon" type="image/ico" href="Images/castle.ico">
  <style type="text/css" media="screen">
    @import "css/reset.css";
    @import "css/demo_table_jui.css";
    @import "css/ui-darkness/jquery-ui-1.8.5.custom.css";
    @import "css/ui-darkness/images";
  </style>

  <script type="text/javascript" src="js/accessdb.js"></script>
  <script type="text/javascript" src="js/jquery.js"></script>
  <script type="text/javascript" src="js/jquery.dataTables.js"></script>
  <script type="text/javascript" src="js/table_setup.js"></script>
  <script type="text/javascript" src="js/queryparam.js"></script>
  <script type="text/javascript" src="js/finddb.js"></script>
  <script type="text/javascript" src="js/lists.js"></script>

  <!-- TemplateBeginEditable name="doctitle" -->
  <title>Untitled Document</title>
  <!-- TemplateEndEditable -->
  <!-- TemplateBeginEditable name="head" -->
  <!-- TemplateEndEditable -->
  <link href="style.css" rel="stylesheet" type="text/css" />
</head>

<body>
  <div class="container">
    <div class="header">
      <div align="center">
        <table width="100%" border="0">
          <tr>
            <td width="16%" height="125" >
              <p align="center">
                
              </p>
              <h4 align="center">US Army Corps of Engineers</h4>
            </td>
            <td width="64%">
              <div align="left">
```

```

        <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
        </div>
    </td>

    <td width="20%">
        <p align="center">&nbsp;</p>
        <p align="right">
            
        </p>
    </td>
</tr>
</table>

</div>
</div>
<div class="break"></div>
<div class="sidebar1">
<ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISIONS</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
</ul>
<!-- end .sidebar1 --></div>
<div class="content">
<div id="con">
    <script type="text/javascript" src="js/ContactInfo.js"></script>
</div>
<!-- end .content --></div>
<div class="footer">
<!-- end .footer --></div>
<!-- end .container --></div>
</body>
</html>

```

Contacts.html

Provides the contacts page with allows individuals to search for contacts.

```

<!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>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>COBie SWIM</title>
    <link rel="icon" type="image/ico" href="Images/castle.ico">
    <script type="text/javascript" src="js/accessdb.js"></script>
    <script type="text/javascript" src="js/jquery.js"></script>
    <script type="text/javascript" src="js/jquery.dataTables.js"></script>
    <script type="text/javascript" src="js/queryparam.js"></script>
    <script type="text/javascript" src="js/table_setup.js"></script>

    <script TYPE="text/javascript">
    <!--
    function dropdown(mySel)
    {
        var myWin, myVal;
        myVal = mySel.options[mySel.selectedIndex].value;
        if(myVal)
        {
            if(mySel.form.target)myWin = parent[mySel.form.target];
            else myWin = window;
            if (! myWin) return true;
            myWin.location = myVal;
        }
        return false;
    }
    //-->
</script>

    <!-- TemplateBeginEditable name="doctitle" -->
    <title>Untitled Document</title>
    <!-- TemplateEndEditable -->
    <!-- TemplateBeginEditable name="head" -->
    <!-- TemplateEndEditable -->
    <link href="style.css" rel="stylesheet" type="text/css" /></head>

<body>

    <div class="container">
    <div class="header">
    <div align="center">
    <table width="100%" border="0">
        <tr>
            <td width="16%" height="125" ><p align="center">
                </p>
                <h4 align="center">US Army Corps of Engineers</h4>
            </td>
            <td width="64%">
                <div align="left">

```

```

        <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
    </div>
</td>

<td width="20%"><p align="center">&nbsp;</p>
    <p align="right">
        
    </p>
</td>
</tr>
</table>
</div>
</div>
<div class="break"></div>
<div class="sidebar1">
<ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISIONS</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
</ul>

<!-- end .sidebar1 --></div>

<div class="content">
<h2 align="center">CONTACTS</h2>
<h4 align="center">This site provides the ability to search for contacts associated with the
memorial chapel. The ability to search for all contacts or a specific contact exists based on the
type of contact exists.</h4>

<!-- <td width="34%"><div align="center"><form method="link" action="Contacts2.html">
<input type="submit" value="Search All Contacts"></form></div></td> -->

<td width="66%"><div align="center">
    <p><h2>Search Criteria</h2></p>
    <p><b>Search by Name</b></p>

    <form name="myform" action="Contacts2.html" method="get">
        <ul><p>
            <input type="text" name="name" onkeydown="if(event.keyCode ==13)
            document.getElementById('btnSearch').click()"></p>
            <p><b>-OR- </b>
            <select name="dropDown">
                <option value="" selected>Search by Type...</option>
                <option value="Manufacturer">Manufacturer</option>
                <option value="Supplier">Supplier</option>
                <option value="General">General</option>
                <option value="Subcontractor">Subcontractor</option></select> </p>
        </ul>
    </form>
</div>
</td>
</tr>
</table>
</div>
</div>

```

```
        <p><input type="submit" id="btnSearch" value="Search"
        href="Contacts2.html"></p>
    </form>
</div>
</td>
<p align="center">&nbsp;</p>
<div align="center"></div></div>
<div class="footer">
<!-- end .footer --></div>
<!-- end .container --></div>
</body>
</html>
```

Contacts2.html

The page that shows the results of a search for a contact.

```

<!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>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>COBie SWIM</title>
  <link rel="icon" type="image/ico" href="Images/castle.ico">
  <style type="text/css" media="screen">
    @import "css/reset.css";
    @import "css/demo_table_jui.css";
    @import "css/ui-darkness/jquery-ui-1.8.5.custom.css";
    @import "css/ui-darkness/images";
  </style>
  <script type="text/javascript" src="js/accessdb.js"></script>
  <script type="text/javascript" src="js/jquery.js"></script>
  <script type="text/javascript" src="js/jquery.dataTables.js"></script>
  <script type="text/javascript" src="js/queryparam.js"></script>
  <script type="text/javascript" src="js/finddb.js"></script>
  <script type="text/javascript" src="js/table_setup.js"></script>

  <!-- TemplateBeginEditable name="doctitle" -->
  <title>Untitled Document</title>
  <!-- TemplateEndEditable -->
  <!-- TemplateBeginEditable name="head" -->
  <!-- TemplateEndEditable -->
  <link href="style.css" rel="stylesheet" type="text/css" />

</head>

<body>

  <div class="container">
    <div class="header">
      <div align="center">
        <table width="100%" border="0">
          <tr>
            <td width="16%" height="125" >
              <p align="center">
                
              </p>
              <h4 align="center">US Army Corps of Engineers</h4>
            </td>
            <td width="64%">
              <div align="left">
                <h1 align="center">Memorial Chapel Database Bldg
                #1391</h1>
              </div>
            </td>
          </tr>
        </table>
      </div>
    </div>
  </div>

```

```

        <td width="20%"><p align="center">&nbsp;</p>
            <p align="right">
                
            </p>
        </td>
    </tr>
</table>

</div>
</div>
<div class="break"></div>
<div class="sidebar1">
<ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISIONS</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
</ul>
<!-- end .sidebar1 --></div>

<div class="content">
<h2 align="center">CONTACTS</h2>
h4 align="center">
A list of all contacts for the memorial chapel is provided sorted by.</h4>
<script type="text/javascript" src="js/allContacts_table.js"></script>

<!-- end .content --></div>
<div class="footer">
<!-- end .footer --></div>
<!-- end .container --></div>
</body>
</html>

```

Division.html

Search for items by any combination of description and division.

```

<!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>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>COBie SWIM</title>
  <link rel="icon" type="image/ico" href="Images/castle.ico">
  <script type="text/javascript" src="js/accessdb.js"></script>
  <script type="text/javascript" src="js/jquery.js"></script>
  <script type="text/javascript" src="js/jquery.dataTables.js"></script>
  <script type="text/javascript" src="js/queryparam.js"></script>
  <script type="text/javascript" src="js/table_setup.js"></script>
  <script type="text/javascript" src="js/findddb.js"></script>
  <script type="text/javascript" src="js/table_setup.js"></script>
  <script type="text/javascript" src="js/lists.js"></script>
  <script type="text/javascript" src="js/equipment/search_functions.js"></script>
  <script TYPE="text/javascript">
    <!--
    function dropdown(mySel)
    {
      var myWin, myVal;
      myVal = mySel.options[mySel.selectedIndex].value;
      if(myVal)
      {
        if(mySel.form.target)myWin = parent[mySel.form.target];
        else myWin = window;
        if (! myWin) return true;
        myWin.location = myVal;
      }
      return false;
    }
    //-->
  </script>
  <!-- TemplateBeginEditable name="doctitle" -->
  <title>Untitled Document</title>
  <!-- TemplateEndEditable -->
  <!-- TemplateBeginEditable name="head" -->
  <!-- TemplateEndEditable -->
  <link href="style.css" rel="stylesheet" type="text/css" />
</head>
<body>
  <div class="container">
    <div class="header">
      <div align="center">
        <table width="100%" border="0">
          <tr>
            <td width="16%" height="125" >
              <p align="center">
                
              </p>
              <h4 align="center">US Army Corps of Engineers</h4>
            </td>
          </tr>
        </table>
      </div>
    </div>
  </div>

```

```

        </td>

        <td width="64%">
            <div align="left">
                <h1 align="center">Memorial Chapel Database Bldg
                #1391</h1>
            </div>
        </td>

        <td width="20%">
            <p align="center">&nbsp;</p>
            <p align="right">
                
            </p>
        </td>
    </tr>
</table>
</div>
</div>
<div class="break"></div>
<div class="sidebar1">
<ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISIONS</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
</ul>
<!-- end .sidebar1 --></div>
<div class="content">
<h2 align=center>
<p>DIVISION</p></h2>
<h3 align=center>
<p>This site provides the ability to search for a piece of equipment based on its division.
The ability is <br /> given to search for all divisions or a particular division based on a
criteria. Upon selecting a particular<br />division, all equipment within that division will
be provided.</h3></p>
<div align=center>
<!--<script type="text/javascript" src="js/equipment/division.js"></script>-->
<script type="text/javascript">
    search_init();
    search_description();
    search_division();
    search_end();
</script>
</div>
<!-- end .content -->
</div>
<div class="footer">
!-- end .footer --></div>

```

```
        <!-- end .container --></div>
</body>
</html>
```

DocumentInfo.html

View an individual document record in detail, optionally editing it.

```

<!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>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>COBie SWIM</title>
  <link rel="icon" type="image/ico" href="Images/castle.ico">
  <style type="text/css" media="screen">
    @import "css/reset.css";
    @import "css/demo_table_jui.css";
    @import "css/ui-darkness/jquery-ui-1.8.5.custom.css";
    @import "css/ui-darkness/images";
  </style>

  <script type="text/javascript" src="js/accessdb.js"></script>
  <script type="text/javascript" src="js/jquery.js"></script>
  <script type="text/javascript" src="js/jquery.dataTables.js"></script>
  <script type="text/javascript" src="js/jqueryparam.js"></script>
  <script type="text/javascript" src="js/table_setup.js"></script>
  <script type="text/javascript" src="js/finddb.js"></script>
  <script type="text/javascript" src="js/lists.js"></script>

  <!-- TemplateBeginEditable name="doctitle" -->
  <title>Untitled Document</title>
  <!-- TemplateEndEditable -->
  <!-- TemplateBeginEditable name="head" -->
  <!-- TemplateEndEditable -->
  <link href="style.css" rel="stylesheet" type="text/css" />
</head>

<body>

  <div class="container">
  <div class="header">
  <div align="center">
    <table width="100%" border="0">
      <tr>
        <td width="16%" height="125" ><p align="center">
          </p>
          <h4 align="center">US Army Corps of Engineers</h4>
          </p>
        </td>
        <td width="64%">
          <div align="left">
            <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
          </div>
        </td>
        <td width="20%"><p align="center">&nbsp;</p>
          <p align="right">
            </p>
        </td>
      </tr>
    </table>
  </div>
  </div>
  </div>

```

```

</tr>
</table>
</div></div>
<div class="break"></div>
<div class="sidebar1">
<ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISIONS</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
</ul>
<!-- end .sidebar1 --></div>

<div class="content">
<div id="con">
    <script type="text/javascript" src="js/DocumentInfo.js"></script>
</div>
<!-- end .content --></div>
<div class="footer">
<!-- end .footer --></div>
<!-- end .container --></div>

</body>
</html>

```

Documents.html
Search for documents.

```

<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>COBie SWIM</title>
<link rel="icon" type="image/ico" href="Images/castle.ico">
<script type="text/javascript" src="js/accessdb.js"></script>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery.dataTables.js"></script>
<script type="text/javascript" src="js/queryparam.js"></script>
<script type="text/javascript" src="js/table_setup.js"></script>
<script type="text/javascript" src="js/finddb.js"></script>
<script type="text/javascript" src="js/documents.js"></script>
<script TYPE="text/javascript">
    <!--
        function dropdown(mySel)
        {
            var myWin, myVal;
            myVal = mySel.options[mySel.selectedIndex].value;
            if(myVal)
            {
                if(mySel.form.target)myWin = parent[mySel.form.target];
                else myWin = window;
                if (! myWin) return true;
                myWin.location = myVal;
            }
            return false;
        }
    //-->
</script>

<!-- TemplateBeginEditable name="doctitle" -->
<title>Untitled Document</title>
<!-- TemplateEndEditable -->
<!-- TemplateBeginEditable name="head" -->
<!-- TemplateEndEditable -->
<link href="style.css" rel="stylesheet" type="text/css" /></head>

<body>

    <div class="container">
    <div class="header">
    <div align="center">
    <table width="100%" border="0">
        <tr>
            <td width="16%" height="125" >
                <p align="center">
                    </p>
                    <h4 align="center">US Army Corps of Engineers</h4>
                </td>
            <td width="64%">

```

```

        <div align="left">
        <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
        </div>
    </td>
    <td width="20%"><p align="center">&nbsp;</p>
        <p align="right"></p>
    </td>
</tr>
</table>

</div>
</div>
<div class="break"></div>
<div class="sidebar1">
<ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISION</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
</ul>
<!-- end .sidebar1 --></div>

<div class="content">

<h2 align=center>
<p>DOCUMENTS</p></h2>
<h3 align=center>
<p>This site provides a list of all documents available for equipment within the Memorial
Chapel. <br />
The ability exists to search for all documents or a particular
document based on the document type.</p>
</h3>
<center>
<p><b>Search Criteria</b></p>
<form name = "sForm" action = "Documents3.html">
    <p><input type="text" name="Search" value=""> </p>
    <p><b>And/Or</b>
    <select name="Category">
        <option value="Select Category">-Select Category-</option>
        <option value="CAD Drawings">CAD Drawings</option>
        <option value="Submittals">Submittals</option>
        <option value="Correspondence">Correspondence</option>
        <option value="Drawings">Drawings</option>
        <option value="Equipment Manuals">Equipment Manuals</option>
        <option value="Reports">Reports</option>
        <option value="Submittals">Submittals</option>
    </select></p>
    <p><input type="submit" value="Search">
</form></center>

```

```

        <div align=right><a href ="uploadDocument.html">Upload A Document</a></div>
    <h3 align=center>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    </h3>
    <!-- end .content -->
    </div>
    <div class="footer">
    <!-- end .footer --></div>
    <!-- end .container --></div>
</body>
</html>
```

Documents3.html

Show results for search for documents.

```

<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>COBie SWIM</title>
<link rel="icon" type="image/ico" href="Images/castle.ico">
<style type="text/css" media="screen">
    @import "css/reset.css";
    @import "css/demo_table_jui.css";
    @import "css/ui-darkness/jquery-ui-1.8.5.custom.css";
    @import "css/ui-darkness/images";
</style>
<script type="text/javascript" src="js/accessdb.js"></script>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery.dataTables.js"></script>
<script type="text/javascript" src="js/queryparam.js"></script>
<script type="text/javascript" src="js/table_setup.js"></script>
<script type="text/javascript" src="js/findddb.js"></script>
<script type="text/javascript" src="js/documents.js"></script>
<script TYPE="text/javascript">
<!--
    function dropdown(mySel)
    {
        var myWin, myVal;
        myVal = mySel.options[mySel.selectedIndex].value;
        if(myVal)
        {
            if(mySel.form.target)myWin = parent[mySel.form.target];
            else myWin = window;
            if (! myWin) return true;
            myWin.location = myVal;
        }
        return false;
    }
-->
</script>

<!-- TemplateBeginEditable name="doctitle" -->
<title>Untitled Document</title>
<!-- TemplateEndEditable -->
<!-- TemplateBeginEditable name="head" -->
<!-- TemplateEndEditable -->
<link href="style.css" rel="stylesheet" type="text/css" /></head>

<body>

    <div class="container">
    <div class="header">
    <div align="center">
    <table width="100%" border="0">
        <tr>

```

```

        <td width="16%" height="125" >
            <p align="center">
                </p>
                <h4 align="center">US Army Corps of Engineers</h4>
            </td>
        <td width="64%">
            <div align="left">
                <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
            </div>
        </td>
        <td width="20%">
            <p align="center">&nbsp;</p>
            <p align="right">
                </p>
            </td>
    </tr>
</table>
</div>
</div>
<div class="break"></div>
<div class="sidebar1">
<ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISIONS</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
</ul>
<!-- end .sidebar1 --></div>
<div class="content">
<!--
    <ul id="tabmenu">
        <li><a class="active" href="#">Tab 1</a></li>
        <li><a class="" href="#">Tab 2</a></li>
        <li><a class="" href="#">Tab 3</a></li>
    </ul>
-->
<!-- end .content -->
<!--<h2 align="center"><p>ROOM 117</h2></p>
<h3 align="center"><p>This provides a list of all CAD drawings in the building with links to
each of the particular documents.
</h3>
<h4 align="center">Documents:
<select> </select></h4></p>-->
<script type="text/javascript">
    var criteria = "";
    var category = "";
    criteria = GetParam("Search");
    category = GetParam("Category");
    searchDocuments(criteria,category);
</script>

```

```
</div>
<div class="footer">
<!-- end .footer --></div>
<!-- end .container --></div>
</body>
</html>
```

EqResults.html

Show results for search for items.

```

<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>COBie SWIM</title>
<link rel="icon" type="image/ico" href="Images/castle.ico">
<style type="text/css" media="screen">
    @import "css/reset.css";
    @import "css/demo_table_jui.css";
    @import "css/ui-darkness/jquery-ui-1.8.5.custom.css";
    @import "css/ui-darkness/images";
</style>
<script type="text/javascript" src="js/accessdb.js"></script>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery.dataTables.js"></script>
<script type="text/javascript" src="js/queryparam.js"></script>
<script type="text/javascript" src="js/finddb.js"></script>
<script type="text/javascript" src="js/table_setup.js"></script>
<script type="text/javascript" src="js/lists.js"></script>
<script TYPE="text/javascript">
    <!--
    function dropdown(mySel)
    {
        var myWin, myVal;
        myVal = mySel.options[mySel.selectedIndex].value;
        if(myVal)
        {
            if(mySel.form.target)myWin = parent[mySel.form.target];
            else myWin = window;
            if (! myWin) return true;
            myWin.location = myVal;
        }
        return false;
    }
    //-->
</script>

<!-- TemplateBeginEditable name="doctitle" -->
<title>Untitled Document</title>
<!-- TemplateEndEditable -->
<!-- TemplateBeginEditable name="head" -->
<!-- TemplateEndEditable -->
<link href="style.css" rel="stylesheet" type="text/css" /></head>

<body>

    <div class="container">
    <div class="header">
    <div align="center">
    <table width="100%" border="0">
        <tr>

```

```

        <td width="16%" height="125" ><p align="center">
            </p>
            <h4 align="center">US Army Corps of Engineers</h4>
            </p>
        </td>
        <td width="64%">
            <div align="left">
                <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
            </div>
        </td>
        <td width="20%"><p align="center">&nbsp;</p>
            <p align="right">
                </p>
        </td>
    </tr>
</table>
</div>
</div>
<div class="break"></div>
<div class="sidebar1">
<ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISIONS</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
</ul>
<!-- end .sidebar1 --></div>
<div class="content">
<!--
<ul id="tabmenu">
    <li><a class="active" href="#">Tab 1</a></li>
    <li><a class="" href="#">Tab 2</a></li>
    <li><a class="" href="#">Tab 3</a></li>
</ul>
-->
<div>
<h3>Results</h3>
<script type="text/javascript" src="js/equipment/results.js"></script>
</div>
<!-- end .content --></div>
<div class="footer">
<!-- end .footer --></div>
<!-- end .container --></div>
</body>
</html>

```

EquipmentInfo.html

View an individual item record in detail, optionally editing it.

```

<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>COBie SWIM</title>
<link rel="icon" type="image/ico" href="Images/castle.ico">
<style type="text/css" media="screen">
    @import "css/reset.css";
    @import "css/demo_table_jui.css";
    @import "css/ui-darkness/jquery-ui-1.8.5.custom.css";
    @import "css/ui-darkness/images";
</style>
<script type="text/javascript" src="js/accessdb.js"></script>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery.dataTables.js"></script>
<script type="text/javascript" src="js/queryparam.js"></script>
<script type="text/javascript" src="js/table_setup.js"></script>
<script type="text/javascript" src="js/finddb.js"></script>
<script type="text/javascript" src="js/lists.js"></script>

<!-- TemplateBeginEditable name="doctitle" -->
<title>Untitled Document</title>
<!-- TemplateEndEditable -->
<!-- TemplateBeginEditable name="head" -->
<!-- TemplateEndEditable -->
<link href="style.css" rel="stylesheet" type="text/css" /></head>

<body>

    <div class="container">
    <div class="header">
    <div align="center">
    <table width="100%" border="0">
        <tr>
            <td width="16%" height="125" ><p align="center">
                </p>
                <h4 align="center">US Army Corps of Engineers</h4>
            </td>
            <td width="64%">
                <div align="left">
                <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
                </div>
            </td>
            <td width="20%"><p align="center">&nbsp;</p>
                <p align="right">
                </p></td>
        </tr>
    </table>
    </div>
    </div>
    </div>

```

```

<div class="break"></div>
<div class="sidebar1">
  <ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISIONS</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
  </ul>
  <!-- end .sidebar1 --></div>
  <div class="content">
    <div id="con">
      <script type="text/javascript" src="js/EquipmentInfo_table.js"></script>
    </div>
  <!-- end .content --></div>
  <div class="footer">
    <!-- end .footer --></div>
  <!-- end .container --></div>
</body>
</html>

```

Home.html

Show the welcome page.

```

<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>COBie SWIM</title>
<link rel="icon" type="image/ico" href="Images/castle.ico">
<script type="text/javascript" src="js/accessdb.js"></script>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery.dataTables.js"></script>
<script type="text/javascript" src="js/queryparam.js"></script>
<script type="text/javascript" src="js/table_setup.js"></script>
<script TYPE="text/javascript">
<!--
function dropdown(mySel)
{
    var myWin, myVal;
    myVal = mySel.options[mySel.selectedIndex].value;
    if(myVal)
    {
        if(mySel.form.target)myWin = parent[mySel.form.target];
        else myWin = window;
        if (! myWin) return true;
        myWin.location = myVal;
    }
    return false;
}
//-->
</script>
<!-- TemplateBeginEditable name="doctitle" -->
<title>Untitled Document</title>
<!-- TemplateEndEditable -->
<!-- TemplateBeginEditable name="head" -->
<!-- TemplateEndEditable -->
<link href="style.css" rel="stylesheet" type="text/css" /></head>

<body>
<div class="container">
<div class="header">
<div align="center">
<table width="100%" border="0">
    <tr>
        <td width="16%" height="125" ><p align="center">
            </p>
            <h4 align="center">US Army Corps of Engineers</h4>
        </td>
        <td width="64%">
            <div align="left">
                <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
            </div>
        </td>
        <td width="20%"><p align="center">&nbsp;</p>

```

```

                <p align="right">
                 </p>
            </td>
        </tr>
    </table>
</div>
</div>
<div class="break"></div>
<div class="sidebar1">
<ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISIONS</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
</ul>
<!-- end .sidebar1 --></div>
<div class="content">
<h2 align="center">
<!--
<ul id="tabmenu">
    <li><a class="active" href="#">Tab 1</a></li>
    <li><a class="" href="#">Tab 2</a></li>
    <li><a class="" href="#">Tab 3</a></li>
</ul>
-->
<!-- end .content -->Welcome to the Memorial Chapel Database Bldg #1391</h2>
<h3 align="center">This database holds all building information for the Memorial Chapel. This
program allows you to view all <br> building material information, as well as sort and search
building information.</h3>
<p align="center">
</p>
</div>
<div class="footer">
<!-- end .footer --></div>
<!-- end .container --></div>
</body>
</html>

```

Search.html

Search the items by any combination of description, division, space and system.

```

<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>COBie SWIM</title>
<link rel="icon" type="image/ico" href="Images/castle.ico">
<script type="text/javascript" src="js/accessdb.js"></script>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery.dataTables.js"></script>
<script type="text/javascript" src="js/queryparam.js"></script>
<script type="text/javascript" src="js/table_setup.js"></script>
<script type="text/javascript" src="js/finddb.js"></script>
<script type="text/javascript" src="js/table_setup.js"></script>
<script type="text/javascript" src="js/lists.js"></script>
<script type="text/javascript" src="js/equipment/search_functions.js"></script>
<script TYPE="text/javascript">
<!--
        function dropdown(mySel)
        {
            var myWin, myVal;
            myVal = mySel.options[mySel.selectedIndex].value;
            if(myVal)
            {
                if(mySel.form.target)myWin = parent[mySel.form.target];
                else myWin = window;
                if (! myWin) return true;
                myWin.location = myVal;
            }
            return false;
        }
//-->
</script>

<!-- TemplateBeginEditable name="doctitle" -->
<title>Untitled Document</title>
<!-- TemplateEndEditable -->
<!-- TemplateBeginEditable name="head" -->
<!-- TemplateEndEditable -->
<link href="style.css" rel="stylesheet" type="text/css" /></head>

<body>

    <div class="container">
    <div class="header">
    <div align="center">
    <table width="100%" border="0">
        <tr>
            <td width="16%" height="125" ><p align="center">
                </p>
                <h4 align="center">US Army Corps of Engineers</h4>
            </td>
        </tr>
    </table>
    </div>
    </div>
    </div>
    </div>

```

```

        </td>
        <td width="64%">
            <div align="left">
                <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
            </div>
        </td>
        <td width="20%"><p align="center">&nbsp;</p>
            <p align="right">
                </p></td>
    </tr>
</table>
</div>
</div>
<div class="break"></div>
<div class="sidebar1">
<ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISIONS</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
</ul>
<!-- end .sidebar1 --></div>
<div class="content">
<h2 align="center">
<p>Search</p></h2>
<h3 align="center">
<p>This site provides the ability to search for items.</p>
</h3>
<div align="center">
<!--<script type="text/javascript" src="js/equipment/search.js"></script>-->
<script type="text/javascript">
    search_init();
    search_description();
    search_room();
    search_system();
    search_division();
    search_end();
</script>
</div>
<!-- end .content -->
</div>
<div class="footer">
<!-- end .footer --></div>
<!-- end .container --></div>
</body>
</html>

```

Space.html

Search the items by any combination of description and space.

```

<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>COBie SWIM</title>
<link rel="icon" type="image/ico" href="Images/castle.ico">
<script type="text/javascript" src="js/accessdb.js"></script>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery.dataTables.js"></script>
<script type="text/javascript" src="js/table_setup.js"></script>
<script type="text/javascript" src="js/queryparam.js"></script>
<script type="text/javascript" src="js/finddb.js"></script>
<script type="text/javascript" src="js/table_setup.js"></script>
<script type="text/javascript" src="js/lists.js"></script>
<script type="text/javascript" src="js/equipment/search_functions.js"></script>
<script TYPE="text/javascript">
<!--
        function dropdown(mySel)
        {
            var myWin, myVal;
            myVal = mySel.options[mySel.selectedIndex].value;
            if(myVal)
            {
                if(mySel.form.target)myWin = parent[mySel.form.target];
                else myWin = window;
                if (! myWin) return true;
                myWin.location = myVal;
            }
            return false;
        }
    //-->
</script>

<!-- TemplateBeginEditable name="doctitle" -->
<title>Untitled Document</title>
<!-- TemplateEndEditable -->
<!-- TemplateBeginEditable name="head" -->
<!-- TemplateEndEditable -->
<link href="style.css" rel="stylesheet" type="text/css" /></head>

<body>

    <div class="container">
    <div class="header">
    <div align="center">
    <table width="100%" border="0">
        <tr>
            <td width="16%" height="125" ><p align="center">
                </p>
                <h4 align="center">US Army Corps of Engineers</h4>
            </td>

```

```

        <td width="64%">
            <div align="left">
                <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
            </div>
        </td>
        <td width="20%"><p align="center">&nbsp;</p>
            <p align="right">
                </p>
        </td>
    </tr>
</table>
</div>
</div>
<div class="break"></div>
<div class="sidebar1">
    <ul class="nav">
        <h3><a href="Home.html">HOME</a></h3>
        <h3><a href="User Guide.html">USER GUIDE</a></h3>
        <h3><a href="Space.html">SPACE<br></a></h3>
        <h3><a href="System.html">SYSTEM</a></h3>
        <h3><a href="Division.html">DIVISIONS</a></h3>
        <h3><a href="Documents.html">DOCUMENTS</a></h3>
        <h3><a href="Contacts.html">CONTACTS</a></h3>
        <h3><a href="Search.html">SEARCH</a></h3>
    </ul>
<!-- end .sidebar1 --></div>

<div class="content">

    <h2 align=center>
    <p>SPACE</p></h2>
    <h3 align=center>
    <p>This contains the information about spaces that are available in the Memorial Chapel
    database. You can<br />choose to search for all of the psaces in this building or search for a
    particular room by entering the room number.</p> </h3>
    <div align=center>
    <!--<script type="text/javascript" src="js/equipment/space.js"></script>-->
    <script type="text/javascript">
        search_init();
        search_description();
        search_room();
        search_end();
    </script>
    </div>
    <!-- end .content -->
</div>
<div class="footer">
    <!-- end .footer --></div>
<!-- end .container --></div>
</body>
</html>

```

System.html

Search the items by any combination of description and system.

```
<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>COBie SWIM</title>
<link rel="icon" type="image/ico" href="Images/castle.ico">
<script type="text/javascript" src="js/accessdb.js"></script>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery.dataTables.js"></script>
<script type="text/javascript" src="js/table_setup.js"></script>
<script type="text/javascript" src="js/queryparam.js"></script>
<script type="text/javascript" src="js/finddb.js"></script>
<script type="text/javascript" src="js/table_setup.js"></script>
<script type="text/javascript" src="js/lists.js"></script>
<script type="text/javascript" src="js/equipment/search_functions.js"></script>
<script TYPE="text/javascript">
<!--
    function dropdown(mySel)
    {
        var myWin, myVal;
        myVal = mySel.options[mySel.selectedIndex].value;
        if(myVal)
        {
            if(mySel.form.target)myWin = parent[mySel.form.target];
            else myWin = window;
            if (! myWin) return true;
            myWin.location = myVal;
        }
        return false;
    }
//-->
</script>

<!-- TemplateBeginEditable name="doctitle" -->
<title>Untitled Document</title>
<!-- TemplateEndEditable -->
<!-- TemplateBeginEditable name="head" -->
<!-- TemplateEndEditable -->
<link href="style.css" rel="stylesheet" type="text/css" /></head>

<body>

    <div class="container">
    <div class="header">
    <div align="center">
    <table width="100%" border="0">
        <tr>
            <td width="16%" height="125" >
                <p align="center">
                    </p>
```

```

        <h4 align="center">US Army Corps of Engineers</h4>
    </td>
    <td width="64%">
        <div align="left">
            <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
        </div>
    </td>
    <td width="20%"><p align="center">&nbsp;</p>
        <p align="right">
            </p>
        </td>
    </tr>
</table>
</div>
</div>
<div class="break"></div>
<div class="sidebar1">
    <ul class="nav">
        <h3><a href="Home.html">HOME</a></h3>
        <h3><a href="User Guide.html">USER GUIDE</a></h3>
        <h3><a href="Space.html">SPACE<br></a></h3>
        <h3><a href="System.html">SYSTEM</a></h3>
        <h3><a href="Division.html">DIVISIONS</a></h3>
        <h3><a href="Documents.html">DOCUMENTS</a></h3>
        <h3><a href="Contacts.html">CONTACTS</a></h3>
        <h3><a href="Search.html">SEARCH</a></h3>
    </ul>
    <!-- end .sidebar1 --></div>
    <div class="content">
        <h2 align=center>
        <p>SYSTEM</p></h2>
        <h3 align=center>
        <p>This site provides the ability to search for a piece of equipment based on its system. The
        ability is <br /> given to search for all systems or a particular system based on a criteria. Upon
        selecting a particular <br /> system, all equipment within that system will be provided.</p>
        </h3>
        <div align=center>
            <!--<script type="text/javascript" src="js/equipment/system.js"></script>-->
            <script type="text/javascript">
                search_init();
                search_description();
                search_system();
                search_end();
            </script>
        </div>
    </div>
    <!-- end .content -->
    </div>
    <div class="footer">
        <!-- end .footer --></div>
    <!-- end .container --></div>
</body>
</html>

```

Add a record to the list of documents.

112

```

        </td>
        <td width="64%">
            <div align="left">
                <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
            </div>
        </td>
        <td width="20%">
            <p align="center">&nbsp;</p>
            <p align="right">
                
            </p>
        </td>
    </tr>
</table>
</div>
</div>
<div class="break"></div>
<div class="sidebar1">
<ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISION</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
</ul>
<!-- end .sidebar1 --></div>

<div class="content">

<h2 align=center>
<p>UPLOAD DOCUMENTS</p></h2>
<h3 align=center>
<p>This page allows you to upload a new document to the database.</p>
<p>Please fill in all fields.</p>
</h3>
<table>
    <tr>
        <td>
            <FORM name="drop_list" id="drop_list" method="get" action
            ="uploadDocumentResults.html"></td>
            <tr>
                <td><b>Choose file:</b>
                <p><input type="file" name="FileName"></p>
            </td>
            </tr>
            <tr>
                <td>
                <b>Set file information:</b>
            </td>
            </tr>
        </tr>
    </table>

```

```

<tr>
  <td>
    <p>Description:
    <input type="text" name="Description"/>
    </p>
  </td>
</tr>
<tr>
  <td>
    <p>Association:
    <input type="text" name="Association" /></p>
  </td>
</tr>
<tr>
  <td>
    <p>
    <SELECT id ="Category" NAME="Category"
    onChange="SelectSubCat();" >
    <Option value="">-Select Category-</option>
    </SELECT></p>
  </td>
</tr>
<tr>
  <td>
    <p>
    <SELECT id="SubCat" NAME="SubCat">
    <Option value="">-Select Subcategory-</option>
    </SELECT></p>
  </td>
</tr>
<tr>
  <td>
    <p>
    <select name="Division">
    <option value="Select Division">-Select Division-
    </option>
    <option value="null">Division Unknown</option>
    <option>1</option><option>2</option>
    <option>3</option> <option>4</option>
    <option>5</option><option>6</option>
    <option>7</option><option>8</option>
    <option>9</option><option>10</option>
    <option>11</option><option>12</option>
    <option>13</option><option>14</option>
    <option>15</option><option>16</option>
    </select></p>
  </td>
</tr>
<tr>
  <td>
    <input type="submit" value="Add Document">
  </td>
</tr>
</FORM>

```

```

                                </tr>
                            </table>

        <!-- end .content -->
    </div>
    <div class="footer">
        <!-- end .footer --></div>
    <!-- end .container --></div>

</body>
</html>
```

Show the result of an attempt to add a document record.

116

```

        <td width="64%">
            <div align="left">
                <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
            </div>
        </td>
        <td width="20%"><p align="center">&nbsp;</p>
            <p align="right">
                
            </p>
        </td>
    </tr>
</table>
</div>
</div>
<div class="break"></div>
<div class="sidebar1">
<ul class="nav">
    <h3><a href="Home.html">HOME</a></h3>
    <h3><a href="User Guide.html">USER GUIDE</a></h3>
    <h3><a href="Space.html">SPACE<br></a></h3>
    <h3><a href="System.html">SYSTEM</a></h3>
    <h3><a href="Division.html">DIVISION</a></h3>
    <h3><a href="Documents.html">DOCUMENTS</a></h3>
    <h3><a href="Contacts.html">CONTACTS</a></h3>
    <h3><a href="Search.html">SEARCH</a></h3>
</ul>
<!-- end .sidebar1 --></div>

<div class="content">
    <script type="text/javascript" src="js/uploadDocument.js"></script>
    <p><p><a href="uploadDocument.html">Add another Document</a>
    <p><a href="Documents.html">Return to Documents</a>
<!-- end .content -->
</div>
<div class="footer">
<!-- end .footer --></div>
<!-- end .container --></div>
</body>
</html>

```

User Guide.html

Links to PDF files which describe how to use the site.

```
<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>COBie SWIM</title>
<link href="style.css" rel="stylesheet" type="text/css" /></head>

<body>

  <div class="container">
    <div class="header">
      <div align="center">
        <table width="100%" border="0">
          <tr>
            <td width="16%" height="125" ><p align="center">
              </p>
              <h4 align="center">US Army Corps of Engineers</h4>
            </td>
            <td width="64%">
              <div align="left">
                <h1 align="center">Memorial Chapel Database Bldg #1391</h1>
              </div>
            </td>
            <td width="20%"><p align="center">&nbsp;</p>
              <p align="right">
                </p>
            </td>
          </tr>
        </table>
      </div>
    </div>
  </div>
  <div class="break"></div>
  <div class="sidebar1">
    <ul class="nav">
      <h3><a href="Home.html">HOME</a></h3>
      <h3><a href="User Guide.html">USER GUIDE</a></h3>
      <h3><a href="Space.html">SPACE<br></a></h3>
      <h3><a href="System.html">SYSTEM</a></h3>
      <h3><a href="Division.html">DIVISIONS</a></h3>
      <h3><a href="Documents.html">DOCUMENTS</a></h3>
      <h3><a href="Contacts.html">CONTACTS</a></h3>
      <h3><a href="Search.html">SEARCH</a></h3>
    </ul>
    <!-- end .sidebar1 --></div>
  <div class="content">
    <h2 align="center">
    <h2 align="center">Welcome to the User Guide for the Memorial Chapel Database Bldg
    #1391</h2>
```

```

<h3 align="left">This is a step by step guide including how to start the software, what each
button provides, how to run a search for each button, and what results can be expected based on a
search.</h3><p align="left">Here is the links to the PDF files to the sections the user needs help
on:</p>
<menu>
<p align="left">
<b><li type="circle"><a href="PDF/How to Start.pdf">HOW TO START</a></a><br />
<b><li type="circle"><a href="PDF/HOME TAB.pdf">HOME</a></a><br />
<b><li type="circle"><a href="PDF/USER GUIDE TAB.pdf">USER GUIDE</a></a><br />
<b><li type="circle"><a href="PDF/SPACE TAB.pdf">SPACE</a></a><br />
<b><li type="circle"><a href="PDF/SYSTEM TAB.pdf">SYSTEM</a></a><br />
<b><li type="circle"><a href="PDF/DIVISION TAB.pdf">DIVISIONS</a></a><br />
<b><li type="circle"><a href="PDF/DOCUMENTS TAB.pdf">DOCUMENT</a>S</a><br />
<b><li type="circle"><a href="PDF/CONTACTS TAB.pdf">CONTACTS</a></a><br />
<b><li type="circle"><a href="PDF/SEARCH TAB.pdf">SEARCH</a></a><br />
<b><li type="circle"><a href="PDF/OTHER FEATURES.pdf">OTHER FEATURES</a></a>
</p></li></menu>
<div class="footer">
  <!-- end .footer --></div>
<!-- end .container --></div>
</body>
</html>

```

style.css

Styling information for all of the webpages.

```
body {
    font: 100%/1.4 "Times New Roman", Times, serif;
    background: #79A5D5;
    margin: 0;
    padding: 0;
}

/* ~~ Element/tag selectors ~~ */
ul, ol, dl { /* Due to variations between browsers, it's best practices to zero padding and margin
on lists. For consistency, you can either specify the amounts you want here, or on the list items
(LI, DT, DD) they contain. Remember that what you do here will cascade to the .nav list unless
you write a more specific selector. */
    padding: 0;
    margin: 0;
}

h1, h2, h3, h4, h5, h6, p {
    margin-top: 0; /* removing the top margin gets around an issue where margins can escape from
their containing div. The remaining bottom margin will hold it away from any elements that
follow. */
    padding-right: 15px;
    padding-left: 15px; /* adding the padding to the sides of the elements within the divs,
instead of the divs themselves, gets rid of any box model math. A nested div with side padding
can also be used as an alternate method. */
}

a img { /* this selector removes the default blue border displayed in some browsers around an
image when it is surrounded by a link */
    border: none;
}

/* ~~ Styling for your site's links must remain in this order - including the group of selectors that create
the hover effect. ~~ */
a:link {
    text-decoration: underline; /* unless you style your links to look extremely unique, it's best to
provide underlines for quick visual identification */
    color: #000;
}
a:visited {
    text-decoration: underline;
    color: #000;
}
a:hover, a:active, a:focus { /* this group of selectors will give a keyboard navigator the same hover
experience as the person using a mouse. */
    text-decoration: none;
    color: #000;
}

/* ~~this fixed width container surrounds the other divs~~ */
.container {
    width: 1200px;
```

```

        height: 100%;
        background: #79A5D5;
        margin: 0 auto; /* the auto value on the sides, coupled with the width, centers the layout */
    }

```

/* ~~ the header is not given a width. It will extend the full width of your layout. It contains an image placeholder that should be replaced with your own linked logo ~~ */

```

.header {
    background-image:url('Images/SB2.jpg');
    color:#FFF;
    width: 100%;
    height: 149px;
    padding-top: 10px;
}

```

/* ~~ These are the columns for the layout. ~~

1) Padding is only placed on the top and/or bottom of the divs. The elements within these divs have padding on their sides. This saves you from any "box model math". Keep in mind, if you add any side padding or border to the div itself, it will be added to the width you define to create the **total** width. You may also choose to remove the padding on the element in the div and place a second div within it with no width and the padding necessary for your design. You may also choose to remove the padding on the element in the div and place a second div within it with no width and the padding necessary for your design.

2) No margin has been given to the columns since they are all floated. If you must add margin, avoid placing it on the side you're floating toward (for example: a right margin on a div set to float right). Many times, padding can be used instead. For divs where this rule must be broken, you should add a "display:inline" declaration to the div's rule to tame a bug where some versions of Internet Explorer double the margin.

3) Since classes can be used multiple times in a document (and an element can also have multiple classes applied), the columns have been assigned class names instead of IDs. For example, two sidebar divs could be stacked if necessary. These can very easily be changed to IDs if that's your preference, as long as you'll only be using them once per document.

4) If you prefer your nav on the right instead of the left, simply float these columns the opposite direction (all right instead of all left) and they'll render in reverse order. There's no need to move the divs around in the HTML source.

```

*/
.sidebar1 {
    float: left;
    width: 15%;
    height: 370px;
    background-image:url('Images/grey.jpg');
    text-color:#FFF;
    line-height: 125%;
}
.content {
    padding: 10px 0;
    width: 83%;
    float: right;
}

```

```

.content2 {
    padding: 10px 0;
    width: 83%;
    float: center;
}
.break {
    padding: 5px 0;
    background: #A80000;
    position: relative; /* this gives IE6 hasLayout to properly clear */
    clear: both; /* this clear property forces the .container to understand where the columns end and
contain them */
}

/* ~~ This grouped selector gives the lists in the .content area space ~~ */
.content ul, .content ol {
    padding: 0 15px 15px 40px; /* this padding mirrors the right padding in the headings and
paragraph rule above. Padding was placed on the bottom for space between other elements on the
lists and on the left to create the indention. These may be adjusted as you wish. */
}

/* ~~ The navigation list styles (can be removed if you choose to use a premade flyout menu like Spry)
~~ */
ul.nav {
    list-style: none; /* this removes the list marker */
    border-top: 1px solid #666; /* this creates the top border for the links - all others are
placed using a bottom border on the LI */
    margin-bottom: 0px; /* this creates the space between the navigation on the content below */
    color: #000;
}
ul.nav li {
    border-bottom: 10px solid #666; /* this creates the button separation */
}
ul.nav a, ul.nav a:visited {
    /* grouping these selectors makes sure that your links retain their button look even after being
visited */
    padding: 5px 5px 5px 15px;
    display: block;
    /* this gives the link block properties causing it to fill the whole LI containing it. This causes the
entire area to react to a mouse click. */
    width: 160px; /*this width makes the entire button clickable for IE6. If you don't need to support
IE6, it can be removed. Calculate the proper width by subtracting the padding on this link from
the width of your sidebar container. */
    text-decoration: none;
    color: #000;
    /*background: #666;*/
}
ul.nav a:hover, ul.nav a:active, ul.nav a:focus { /* this changes the background and text color for both
mouse and keyboard navigators */
    background: #666;
    color: #FFF;
}

/* ~~ The footer ~~ */

```

```

.footer {
    padding: 7px 0;
    background-image:url('Images/SB2.jpg');
    position: relative; /* this gives IE6 hasLayout to properly clear */
    clear: both; /* this clear property forces the .container to understand where the columns end and
contain them */
}

/* ~~ miscellaneous float/clear classes ~~ */
.fltrt {
    /* this class can be used to float an element right in your page. The floated element must precede
the element it should be next to on the page. */
    float: right;
    margin-left: 8px;
}
.flflt {
    /* this class can be used to float an element left in your page. The floated element must precede
the element it should be next to on the page. */
    float: left;
    margin-right: 8px;
}
.clearfloat {
    /* this class can be placed on a <br /> or empty div as the final element following the last floated
div (within the #container) if the #footer is removed or taken out of the #container */
    clear:both;
    height:0;
    font-size: 1px;
    line-height: 0px;
}
/* ~~ The Tab Menu ~~ */
pre {text-indent: 30px}

#tabmenu {
    color: #000;
    border-bottom: 2px solid black;
    margin: 12px 0px 0px 0px;
    padding: 0px;
    z-index: 1;
    padding-left: 10px;
}
#tabmenu li {
    display: inline;
    overflow: hidden;
    list-style-type: none;
}
#tabmenu a, a.active {
    color: #000;
    background: #CCC;
    font: bold 1em;
    border: 2px solid #000;
    padding: 2px 5px 0px 5px;
    margin: 0;
    text-decoration: none;
}

```

```

#tabmenu a.active {
    color: #000;
    background: #FFF;
    border-bottom: 3px solid #FFF;
}
#tabmenu a:hover {
    color: #000;
    background: #FFF;
}
#tabmenu a:visited {
    color: #000;
}
#tabmenu a.active:hover {
    background: #FFF;
    color: #000;
}
#con {
    text-align: left;
    background: #FFF;
    /*height: 300px;*/
    padding: 10px;
    border: 15px;
    border-top: 2px solid black;
    border-bottom: 3px solid #000;
    border-left: 3px solid #000;
    border-right: 3px solid #000;
    z-index: 2;
    overflow: auto;
}
/*
#con a {
    text-decoration: underline;
    color: #000;
}
#con a:hover {
    background: #fff;
    color: #000;
}
*/

/* ~~ The SELECT Dropdown Menu ~~ */
select.limited-width {
    width: 140px;
    position: static;
}

select.expanded-width {
    width: auto;
    position: absolute;
}

#info{
    width: 100%;

```

```
}  
  
#eqresults {  
    width: 100%;  
}  
  
#eqinforesults {  
    width: 100%;  
}  
#conresults {  
    width: 100%;  
}  
  
#eqform td {  
    padding: 2px 10px;  
}
```

reset.css

Styling information so that the tables can be used with the Data Tables theme.

```
/*
 * Reset Tables - Andrew Cordar
 */

#info {font-family: Verdana; font-size:15px; border-collapse:collapse; border-spacing:0;}
#eqresults {font-family: Verdana; font-size:15px; border-collapse:collapse; border-spacing:0;}
#eqinforesults {font-family: Verdana; font-size:15px; border-collapse:collapse; border-spacing:0;}
#conresults {font-family: Verdana; font-size:15px; border-collapse:collapse; border-spacing:0;}
#docinfo {font-family: Verdana; font-size:15px; border-collapse:collapse; border-spacing:0;}
```

JavaScript files

allContacts_table.js

Generate results for a search for contacts.

```
//tableName = queryParams("table");
var myDB = finddb();

var SQLs = [];
var searchName = queryParams("name");
var dropDown = queryParams("dropDown");
var manufacturer = queryParams("manufacturer");

var SQL = "select distinct Company as Company, " +
    "Contact as Contact, " +
    "Title as Title, " +
    "Type as Type, " +
    "Scope as Scope, " +
    "Phone as Phone, " +
    "URL as Website, " +
    "ID " +
    "from Contacts";

var SQL_Name_Type = "select Contact as Contact, " +
    "Company as Company, " +
    "Title as Title, " +
    "Type as Type, " +
    "Scope as Scope, " +
    "Phone as Phone, " +
    "URL as Website, " +
    "ID " +
    "from Contacts where Contact LIKE \"%\" + cleanString(searchName) + \"%\""+
    " and Type = \"\" + cleanString(dropDown) + \"\"";

var SQL_Name = "select Contact as Contact, " +
    "Company as Company, " +
    "Title as Title, " +
    "Type as Type, " +
    "Scope as Scope, " +
    "Phone as Phone, " +
```

```

        "URL as Website, " +
        "ID " +
        "from Contacts where Contact LIKE " + "\"" + cleanString(searchName) + "%\"";

var SQL_Manu = "select Type as Type, " +
    "Company as Company, " +
    "Contact as Contact, " +
    "Title as Title, " +
    "Scope as Scope, " +
    "Phone as Phone, " +
    "URL as Website, " +
    "ID " +
    "from Contacts where Type='Manufacturer'";

var SQL_Supp = "select Type as Type, " +
    "Company as Company, " +
    "Contact as Contact, " +
    "Title as Title, " +
    "Scope as Scope, " +
    "Phone as Phone, " +
    "URL as Website, " +
    "ID " +
    "from Contacts where Type='Supplier'";

var SQL_Gen = "select Type as Type, " +
    "Company as Company, " +
    "Contact as Contact, " +
    "Title as Title, " +
    "Scope as Scope, " +
    "Phone as Phone, " +
    "URL as Website, " +
    "ID " +
    "from Contacts where Type='General'";

var SQL_Sub = "select Type as Type, " +
    "Company as Company, " +
    "Contact as Contact, " +
    "Title as Title, " +
    "Scope as Scope, " +
    "Phone as Phone, " +
    "URL as Website, " +
    "ID " +
    "from Contacts where Type='Subcontractor'";

if(!searchName){
    if(dropDown == "Manufacturer")
        SQLs.push(SQL_Manu);
    else if(dropDown == "Supplier")
        SQLs.push(SQL_Supp);
    else if(dropDown == "General")
        SQLs.push(SQL_Gen);
}

```

```

        else if(dropDown == "Subcontractor")
            SQLs.push(SQL_Sub);
        else
            SQLs.push(SQL);
    }
    else{
        if(!dropDown)
            SQLs.push(SQL_Name);
        else
            SQLs.push(SQL_Name_Type);
    }

    // Only one can be pushed, so add the manufacturer clause to end of first.
    if (manufacturer) {
        manu_clause = "Company like '"+cleanString(manufacturer)+"'";
        if (SQLs[0] == SQL) {
            SQLs[0] += " where " + manu_clause;
        } else {
            SQLs[0] += " and " + manu_clause;
        }
    }
}

var BigSQL = SQLs.join(" union all ") + ";";

var resultSet = myDB.query(BigSQL, {table:{stringOut:true, id:"conresults"}});
if (resultSet) {
    document.write(resultSet);
} else {
    alert("Your search returned no results");
    history.back();
}
ContactInfo.js
Generate results for a request for an individual contact record.

var myDB = finddb();
var id = queryParam("id");

var SQLs = [];

var SQL = "select distinct Company as Company, " +
    "Contact as Contact, " +
    "Title as Title, " +
    "Type as Type, " +
    "Scope as Scope, " +
    "Phone as Phone, " +
    "URL as Website " +
    "from Contacts where ID = " + cleanString(id);
SQLs.push(SQL);
var BigSQL = SQLs.join(" union all ") + ";";

function save() {
    var donesomething = false;
    var val;
    var fields = [ "Company",

```

```

        "Contact",
        "Title",
        "Type",
        "Scope",
        "Phone" ];

for (var i=0; i < fields.length; i++) {
    val = document.getElementsByName(fields[i])[0].value;
    if (val && val != r[fields[i]]) {
        SQL = "update Contacts set "+fields[i]+" = \"\" + val + "\" \" +
            "where ID = " + id + ";";
        myDB.query(SQL);
        donesomething = true;
    }
}

// Website
val = document.getElementsByName("Website")[0].value;
if (val && val != r["Website"]) {
    SQL = "update Contacts set URL = \"\" + val + "\" \" +
        "where ID = " + id + ";";
    myDB.query(SQL);
    donesomething = true;
}

if (donesomething) {
    location.reload();
}
}

var resultSet = myDB.query(BigSQL);
if (resultSet) {
    r = resultSet[0];

    document.write("<table border=\"0\" id=\"eqform\">");

    // Item
    document.write('<tr><td><b>Field</b></th><td><b>Current
Value</b></th><td><b>Modify</b></th></tr>');
    document.write("<tr><td>Id:</td><td>"+id+"</td><td>/</tr>");
    document.write("<tr><td>Company:</td><td>"+r["Company"]+"</td>");
    document.write("<tr><td><input style=\"width:97%\" name=\"Company\"></td></tr>");
    document.write("<tr><td>Contact:</td><td>"+r["Contact"]+"</td>");
    document.write("<tr><td><input style=\"width:97%\" name=\"Contact\"></td></tr>");
    document.write("<tr><td>Title:</td><td>"+r["Title"]+"</td>");
    document.write("<tr><td><input style=\"width:97%\" name=\"Title\"></td></tr>");
    document.write("<tr><td>Type:</td><td>"+r["Type"]+"</td>");
    document.write("<tr><td><input style=\"width:97%\" name=\"Type\"></td></tr>");
    document.write("<tr><td>Scope:</td><td>"+r["Scope"]+"</td>");
    document.write("<tr><td><input style=\"width:97%\" name=\"Scope\"></td></tr>");
    document.write("<tr><td>Phone:</td><td>"+r["Phone"]+"</td>");
    document.write("<tr><td><input style=\"width:97%\" name=\"Phone\"></td></tr>");
    document.write("<tr><td>Website:</td><td><a
href=\"http://\"+r[\"Website\"]+\"\">"+r["Website"]+"</a></td>");

```

```
document.write("<td><input style=\"width:97%\" name=\"Website\"></td></tr>");
document.write("<tr><td /><td /><td><button type=\"button\" onclick=\"save();\">Save
Changes</button></td></tr>");
document.write("</table>");
} else {
    alert("Sorry, there is no contact information for this Company");
    history.back();
}
```

DocumentDropDown.js

Handle dynamic filling of the drop down selectors in uploadDocument.html.

```
function fillCategory(){
    // this function is used to fill the category list on load
    var cat = document.getElementById("Category");
    addOption(cat, "CAD Drawings", "CAD Drawings", "");
    addOption(cat, "Submittals", "Submittals", "");
    addOption(cat, "Correspondence", "Correspondence", "");
    addOption(cat, "Drawings", "Drawings", "");
    addOption(cat, "Equipment Manuals", "Equipment Manuals", "");
    addOption(cat, "Reports", "Reports", "");
}

function SelectSubCat(){
    // ON selection of category this function will work
    var cat = document.getElementById("Category");
    var SubCat = document.getElementById("SubCat");
    removeAllOptions(SubCat);
    addOption(SubCat, "", "-Select Subcategory-", "");

    if(cat.value == 'CAD Drawings'){
        addOption(SubCat, "As Built / Record Drawings", "As Built / Record Drawings");
    }
    if(cat.value == 'Submittals'){
        addOption(SubCat, "Parts List", "Parts List");
        addOption(SubCat, "Product Data", "Product Data");
        addOption(SubCat, "Shop Drawings", "Shop Drawings", "");
    }
    if(cat.value == 'Correspondence'){
        addOption(SubCat, "None", "None");
    }
    if(cat.value == 'Drawings'){
        addOption(SubCat, "As Built / Record Drawings", "As Built / Record Drawings");
        addOption(SubCat, "Construction", "Construction");
    }
    if(cat.value == 'Equipment Manuals'){
        addOption(SubCat, "Consolidated Manuals", "Consolidated Manuals");
        addOption(SubCat, "Installation Manuals", "Installation Manuals");
        addOption(SubCat, "Maintenance Manuals", "Maintenance Manuals");
        addOption(SubCat, "Manufacturer Warranties", "Manufacturer Warranties");
    }
    if(cat.value == 'Reports'){
        addOption(SubCat, "None", "None");
    }
}

////////////////////////////////

function removeAllOptions(selectbox)
{
    var i;
    for(i=selectbox.options.length-1;i>=0;i--)
```

```

    {
        //selectbox.options.remove(i);
        selectbox.remove(i);
    }
}

function addOption(selectbox, value, text )
{
    var optn = document.createElement("OPTION");
    optn.text = text;
    optn.value = value;

    selectbox.options.add(optn);
}

/*
var category = queryParam("CreateCategory");
var category = cleanString(category);

document.write("<p><select name=\"Subcategory\">");
document.write("<option value=\"Select Subcategory\">-Select SubCategory-</option>");

if(category == "CreateCADDrawings")
    document.write("<option value=\"As Built / Record Drawings\">As Built / Record
Drawings</option>");
else if(category == "CreateSubmittals"){
    document.write("<option value=\"Parts List\">Parts List</option>");
    document.write("<option value=\"Product Data\">Product Data</option>");
    document.write("<option value=\"Shop Drawings\">Shop Drawings</option>");
}
else if(category == "CreateCorrespondence")
    document.write("<option value=\"None\">None</option>");
else if(category == "CreateDrawings"){
    document.write("<option value=\"As Built / Record Drawings\">As Built / Record
Drawings</option>");
    document.write("<option value=\"Construction\">Construction</option>");
}
else if(category == "CreateEquipmentManuals"){
    document.write("<option value=\"Consolidated Manuals\">Consolidated Manuals</option>");
    document.write("<option value=\"Installation Manuals\">Installation Manuals</option>");
    document.write("<option value=\"Maintenance Manuals\">Maintenance Manuals</option>");
    document.write("<option value=\"Manufacturer Warranties\">manufacturer Warranties</option>");
}
else if(category == "CreateReports")
    document.write("<option value=\"None\">None</option>");

    document.write("</select>");*/
DocumentInfo.js
Generate results for a request for an individual document record.

var myDB = finddb();
var id = queryParam("id");

```

```

var SQL = "select * from Documents where ID = " + cleanString(id);

function save() {
    var donesomething = false;
    var val;

    for (var i in r) {
        if (i == "ID") continue;
        val = document.getElementsByName(i)[0].value;
        if (val && val != r[i]) {
            if (i == "Division") {
                val = parseInt(val);
                insertSQL = "update Documents set "+i+" = "+val+" " +
                    "where ID = " + id + " ";
            } else {
                insertSQL = "update Documents set "+i+" = '"+val+"' " +
                    "where ID = " + id + " ";
            }
            myDB.query(insertSQL);
            donesomething = true;
        }
    }

    if (donesomething) {
        location.reload();
    }
}

var resultSet = myDB.query(SQL);
if (resultSet) {
    r = resultSet[0];
    //      var myStr="This is a test!";
    //      document.write(addDash(myStr));
    document.write("<table border=\"0\" id=\"eqform\">");

    // Item
    document.write('<tr><td><b>Field</b></th><td><b>Current Value</b></th><td><b>Modify</b></th></tr>');
    for (var i in r) {
        if (i == "ID") {
            document.write("<tr><td>Id:</td><td>"+id+"</td><td /></tr>");
        } else if (i == "Link") {
            document.write("<tr><td>"+i+":</td><td><a href=\"StoredFiles/\">");
            for (var i in r) {
                if (i == "Category") {
                    document.write(addDash(r[i])+"/");
                }
                if (i == "Subcategory") {
                    document.write(addDash(r[i])+"/");
                }
            }
            document.write(r[i]+"\">"+r[i]+"</a></td>");
            document.write("<td><input style=\"width:97%\" name=\""+i+"\"></td></tr>");
            //alert("asdf");
        }
    }
}

```

```

        }
        else{
            document.write("<tr><td>"+i+":</td><td>"+r[i]+"</td>");
            document.write("<td><input style=\"width:97%\" name=\""+i+"\"></td></tr>");
        }
    }

    document.write("<tr><td /><td /><td><button type=\"button\" onclick=\"save();\">Save
Changes</button></td></tr>");
    document.write("</table>");
} else {
    alert("Your search returned no results.");
    history.back();
}
documents.js
Generate results for a search for documents.

var myDB = finddb();

function findAllDocs() {
    var sql = "Select * From Documents";

    var result = myDB.query(sql,{table:{stringOut:true, id:"info"}});
    if(result)
        document.write(result);
}

function GetParam(name) {
    var start = location.search.indexOf("?"+name+"=");
    if(start < 0) start = location.search.indexOf("&"+name+"=");
    if(start < 0) return "";

    start += name.length+2;
    var end = location.search.indexOf("&",start)-1;
    if(end<0) end = location.search.length;
    var result= "";
    for(var i = start; i <= end; i++) {
        var c = location.search.charAt(i);
        result = result+(c=='?'?'':c);
    }
    return unescape(result);
}

function searchDocuments(criteria,category) {
    if(category == "Select Category")
        var sql = "Select * From Documents Where Description LIKE '%" +criteria + "%'";
    else
        var sql = "Select * From Documents Where Description LIKE '%" + criteria + "%' AND Category =
'" + category + "'";

    var result = myDB.query(sql,{table:{stringOut:true, id:"docinfo"}});

    if(result) {

```

```
        document.write(result);  
    } else {  
        alert("No Results!");  
        history.back();  
    }  
}
```

EquipmentInfo_table.js

Generate results for a request for an individual item request.

```
var id = queryParams("id");

var myDB = finddb();

var things = table_list();
var systems = system_list();

var SQLs = [];
var r;

function save() {
    var donesomething = false;
    var val;

    // Room
    var room = document.getElementById("room_dd");
    val = room.options[room.selectedIndex].value;
    if (val && val != r["RoomId"]) {
        SQL = "update RoomAssociations set RoomId = " + val +
            " where Id = " + r["ItemId"] + " and RoomId = " + r["RoomId"] + ";";
        myDB.query(SQL);
        donesomething = true;
    }

    // EquipmentDescription
    val = document.getElementsByName("EquipmentDescription")[0].value;
    if (val && val != r["EquipmentDescription"]) {
        SQL = "update "+r["TypeTable"]+" set FamilyName = \"\" + val + "\" \" +
            "where Id = " + r["TypeId"] + ";";
        myDB.query(SQL);
        donesomething = true;
    }

    // EquipmentType
    val = document.getElementsByName("EquipmentType")[0].value;
    if (val && val != r["EquipmentType"]) {
        SQL = "update "+r["TypeTable"]+" set TypeName = \"\" + val + "\" \" +
            "where Id = " + r["TypeId"] + ";";
        myDB.query(SQL);
        donesomething = true;
    }

    // Manufacturer
    val = document.getElementsByName("Manufacturer")[0].value;
    if (val && val != r["Manufacturer"]) {
        SQL = "update "+r["TypeTable"]+" set Manufacturer = \"\" + val + "\" \" +
            "where Id = " + r["TypeId"] + ";";
        myDB.query(SQL);
        donesomething = true;
    }
}
```

```

// ModelNumber
val = document.getElementsByName("ModelNumber")[0].value;
if (val && val != r["ModelNumber"]) {
    SQL = "update "+r["TypeTable"]+" set Model = \"\" + val + "\" \" +
        "where Id = \" + r["TypeId"] + "\";";
    myDB.query(SQL);
    donesomething = true;
}

// ModelDescription
val = document.getElementsByName("ModelDescription")[0].value;
if (val && val != r["ModelDescription"]) {
    SQL = "update "+r["TypeTable"]+" set Description = \"\" + val + "\" \" +
        "where Id = \" + r["TypeId"] + "\";";
    myDB.query(SQL);
    donesomething = true;
}

if (donesomething) {
    location.reload();
}
}

for (var i in things) {
    var m = things[i][0];
    var t = things[i][1];
    var d = things[i][2];
    var s = systems[d-1];

    var SQL = "select "+t+".FamilyName as EquipmentDescription, " +
        t+".TypeName as EquipmentType, " +
        t+".Manufacturer as Manufacturer, " +
        t+".Model as ModelNumber, " +
        t+".Description as ModelDescription, " +
        t+".Id as TypeId, " +
        m+".Id as ItemId, " +
        "Rooms.Id as RoomId, " +
        "\""+s+"\" as System, " +
        d+" as Division, " +
        //"\""+m+"\" as ItemTable, " +
        "\""+t+"\" as TypeTable, " +
        "Rooms.Number + \" \" + Rooms.Name as Room " +
        "from "+m+", "+t+", RoomAssociations, Rooms " +
        "where "+m+".TypeId = "+t+".Id " +
        "and "+m+".Id = " + cleanString(id) +
        " and "+m+".Id = RoomAssociations.Id " +
        "and RoomAssociations.RoomId = Rooms.Id";
    SQLs.push(SQL);
}

if (SQLs.length > 0) {
    var BigSQL = SQLs.join(" union all ") + ";";
    var resultSet = myDB.query(BigSQL);

```

```

if (resultSet) {
    r = resultSet[0];

    document.write("<table border=\"0\" id=\"eqform\">");

    // Item
    document.write("<tr><td colspan=\"3\"><big><strong>Item:</strong> Changes here will affect this
item only.</big></td></tr>");
    document.write("<tr><td><b>Field</b></th><td><b>Current
Value</b></th><td><b>Modify</b></th></tr>");
    document.write("<tr><td>Id:</td><td>"+r["ItemId"]+"</td><td /></tr>");
    document.write("<tr><td>System:</td><td>"+r["System"]+"</td><td /></tr>");
    document.write("<tr><td>Division:</td><td>"+r["Division"]+"</td><td /></tr>");
    document.write("<tr><td>Room:</td><td>"+r["Room"]+"</td><td>");
    document.write("<td>");
    room_dropdown("", "tableid");
    document.write("</td></tr>");

    // Type
    document.write("<tr><td colspan=\"3\"><big><strong>Type:</strong> Changes here will affect all
items of this type.</big></td></tr>");
    document.write("<tr><td><b>Field</b></th><td><b>Current
Value</b></th><td><b>Modify</b></th></tr>");

    document.write("<tr><td>Id:</td><td>"+r["TypeId"]+"</td><td /></tr>");
    document.write("<tr><td>EquipmentDescription:</td><td>"+r["EquipmentDescription"]+"</td>");
    document.write("<td><input style=\"width:97%\" name=\"EquipmentDescription\"></td></tr>");
    document.write("<tr><td>EquipmentType:</td><td>"+r["EquipmentType"]+"</td>");
    document.write("<td><input style=\"width:97%\" name=\"EquipmentType\"></td></tr>");
    document.write("<tr><td>Manufacturer:</td><td><a
href=\"Contacts2.html?manufacturer="+r["Manufacturer"]+"\">"+r["Manufacturer"]+"</a></td>");
    document.write("<td><input style=\"width:97%\" name=\"Manufacturer\"></td></tr>");
    document.write("<tr><td>ModelNumber:</td><td>"+r["ModelNumber"]+"</td>");
    document.write("<td><input style=\"width:97%\" name=\"ModelNumber\"></td></tr>");
    document.write("<tr><td>ModelDescription:</td><td>"+r["ModelDescription"]+"</td>");
    document.write("<td><input style=\"width:97%\" name=\"ModelDescription\"></td></tr>");

    document.write("<tr><td /><td /><td><button type=\"button\" onclick=\"save();\">Save
Changes</button></td></tr>");
    document.write("</table>");

    // Associated documents
    var docSQL = "select Documents.* from " + r["TypeTable"] +
        ", TypesToDocuments, Documents where " +
        r["TypeTable"]+".ID = TypesToDocuments.TypeID and " +
        "TypesToDocuments.DocumentId = Documents.Id;";
    var resultSet = myDB.query(docSQL, {table:{stringOut:true, id:"docinfo"}});
    if (resultSet) {
        document.write("<h3>Associated Documents</h3>");
        document.write(resultSet);
    }

    // Associated contacts

```

```

        var contactSQL = "select distinct Company, Contact, Title, Type, Scope, Phone, Contacts.URL as
Website, Contacts.ID from " + r["TypeTable"] +
        ", TypesToContacts, Contacts where " +
        r["TypeTable"]+".ID = TypesToContacts.TypeID and " +
        "TypesToContacts.ContactId = Contacts.Id;";
        var resultSet = myDB.query(contactSQL, {table:{stringOut:true, id:"conresults"}});
        if (resultSet) {
            document.write("<h3>Associated Contacts</h3>");
            document.write(resultSet);
        }
    } else {
        alert("Search returned no results.");
        history.back();
    }
} else {
    alert("Search returned no results.");
    history.back();
}
}

```

finddb.js

Helper function for other JavaScript files to connect to the Access database.

```
function finddb() {
    // This should work with any partition now, although I feel concerned
    // it will catch the : in file:// ...
    var beginning = location.pathname.indexOf(":") - 1;
    var end = location.pathname.lastIndexOf("/");
    var path = location.pathname.substring(beginning, end+1);
    path = path.replace(/%20/g, " ");
    path = path.replace(/\\/g, "\\");
    var myDB = new ACCESSdb(path+"test.mdb", {showErrors:true});
    return myDB;
}
```

lists.js

Helper functions to contain lists (such as the relevant tables in the database, the systems and their corresponding division number) to try to avoid duplication of data.

```
function table_list() {
    return [ ["Casework", "CaseworkTypes", 12],
        ["ElectricalEquipment", "ElectricalEquipmentTypes", 16],
        ["ElectricalFixtures", "ElectricalFixtureTypes", 16],
        ["Furniture", "FurnitureTypes", 12],
        ["FurnitureSystems", "FurnitureSystemTypes", 12],
        ["GenericModels", "GenericModelTypes", 1],
        ["LightingFixtures", "LightingFixtureTypes", 16],
        ["MechanicalEquipment", "MechanicalEquipmentTypes", 15],
        ["PlumbingFixtures", "PlumbingFixtureTypes", 15],
        ["SpecialtyEquipment", "SpecialtyEquipmentTypes", 10],
        //["DuctTerminal", "DuctTerminalTypes"],
        ["TelephoneDevices", "TelephoneDeviceTypes", 11],
        ["NurseCallDevices", "NurseCallDeviceTypes", 11],
        ["SecurityDevices", "SecurityDeviceTypes", 11],
        ["CommunicationDevices", "CommunicationDeviceTypes", 11],
        ["DataDevices", "DataDeviceTypes", 11],
        ["FireAlarmDevices", "FireAlarmDeviceTypes", 13],
        ["LightingDevices", "LightingDeviceTypes", 13],
        ["Sprinklers", "SprinklerTypes", 13] ];
}
```

```
function system_list() {
    return ["General Information",
        "Civil Work",
        "Concrete",
        "Masonry",
        "Metals",
        "Wood & Plastics",
        "Thermal & Moisture Protection",
        "Doors and Windows",
        "Finishes",
        "Specialties",
        "Equipment",
        "Furnishings",
```

```

        "Special Construction",
        "Conveying Systems",
        "Mechanical",
        "Electrical"];
    }

function room_dropdown(empty_text, value_style) {
    document.write("<select id=\"room_dd\" name=\"room\">");
    document.write("<option value=\"\">"+empty_text+"</option>");
    var SQL = "SELECT DISTINCT Number, Name, Id FROM ROOMS;";
    var results = myDB.query(SQL);
    for (var i in results) {
        if (value_style == "tableid")
            document.write("<option value=\""+results[i]["Id"]+"\">");
        else
            document.write("<option value=\""+results[i]["Number"]+"\">");
        var str=results[i]["Number"] + " " + results[i]["Name"];
        document.write(str);
        document.write("</option>");
    }
    document.write("</select>");
}

```

queryparam.js

Helper functions to parse parameters sent to the JavaScript files from the previous page.

```

function queryParam(name) {
    name = name.replace(/[\/],"/,"\\[\/],\\/");
    var regexS = "[\\?&]" + name + "=(^&#)*";
    var regex = new RegExp( regexS );
    var results = regex.exec( window.location.href );
    if( results == null )
        return "";
    else
        return results[1];
}

function cleanString(str) {
    var res = "";

    for (var i = 0; i < str.length; i++) {
        if (str.substr(i, 3).match(/%[0-9a-fA-F]{2}/)) { // Replace the codes
            var hex = str.substr(i+1, 2);
            var num = parseInt(hex, 16);
            var ch = String.fromCharCode(num);
            res = res + ch;
            i += 2;
        } else if (str.substr(i, 1) == "+") { // Replace with spaces
            res = res + " ";
        } else {
            res = res + str.substr(i, 1);
        }
    }
}

return res;

```

```
}
```

```
function addDash(str) {  
    var res = str;  
  
    res = res.replace(/ /g, "-");  
    res = res.replace(/[V]/g, "");
```

```
    return res;  
}
```

table_setup.js

Establishes the theme of the results tables, and includes the links from certain columns to the page relating to those columns.

```
$(document).ready(function() {  
    $('#info').dataTable({  
        "sPaginationType": "full_numbers",  
        "bJQueryUI":true  
    });  
  
    $('#eqresults').dataTable({  
        "sPaginationType": "full_numbers",  
        "bJQueryUI":true,  
        "aoColumnDefs": [  
            {  
                "fnRender": function(oObj) {  
                    return "<a href=\"EquipmentInfo.html?id=\" +  
                        oObj.aData[4] + \">\" + oObj.aData[2] +  
                        "</a>";  
                },  
                "aTargets": [2]  
            }  
        ]  
    });  
  
    $('#docinfo').dataTable({  
        "sPaginationType": "full_numbers",  
        "bJQueryUI":true,  
        "aoColumns": [  
            null,  
            null,  
            { "fnRender": function(oObj) {  
                return "<a href=\"DocumentInfo.html?id=\" +  
                    oObj.aData[0] + \">\" + oObj.aData[2] +  
                    "</a>";  
            } },  
            null,  
            null,  
            null,  
            { "fnRender": function(oObj) {  
                return "<a href=\"StoredFiles/\" + addDash(oObj.aData[3]) + \"/\" +  
                    addDash(oObj.aData[4]) + \"/\" + oObj.aData[6] + "\">  
+  
                    oObj.aData[6] + "</a>";
```

```

    }}

    ]
  });

  $('#conresults').dataTable({
    "sPaginationType": "full_numbers",
    "bJQueryUI": true,
    "aoColumns": [
      null,
      { "fnRender": function(oObj) {
        return "<a href=\"ContactInfo.html?id=\" +
          oObj.aData[7] + \"\">\" + oObj.aData[1] +
          "</a>";
      } },
      null,
      null,
      null,
      null,
      { "fnRender": function(oObj) {
        return "<a href=\"http://\" +
          oObj.aData[6] + \"\">\" + oObj.aData[6] +
          "</a>";
      } },
      null
    ]
  });
});

```

uploadDocument.js

Handles a request to upload a document; that is, it copies the file to the correct folder, then adds an entry to the Access database.

```
var myDB = finddb();

var fileName = cleanString(queryParam("FileName"));
var category = cleanString(queryParam("Category"));
var subCat = cleanString(queryParam("SubCat"));
var division = cleanString(queryParam("Division"));
var association = cleanString(queryParam("Association"));
var description = cleanString(queryParam("Description"));

if (fileName == ""
    || category == ""
    || subCat == ""
    || division == ""
    || association == ""
    || description == "") {
    alert("All fields are mandatory.");
    history.back();
} else {
    //Check if entry already exists
    var sql = "select ID from Documents where " +
        "Description = '"+description+"' and " +
        "Category = '"+category+"' and " +
        "Subcategory = '"+subCat+'";";

    if (myDB.query(sql)) {
        document.write("An entry already exists for " + description + " in " +
            category + ", " + subCat + ".");
    } else {
        //Obtain the file
        var mybeginning = location.pathname.indexOf(":") - 1;
        var myend = location.pathname.lastIndexOf("/");
        var mypath = location.pathname.substring(mybeginning, myend+1);
        mypath = mypath.replace(/%20/g, " ");
        mypath = mypath.replace(/\\/g, "\\");
        var myObject = new ActiveXObject("Scripting.FileSystemObject");
        var newFileName = mypath + "StoredFiles\\" + addDash(category) + "\\" + addDash(subCat) +
            fileName.substring(fileName.lastIndexOf("\\"));
        //document.write(fileName+"<br />"+newFileName);
        myObject.CopyFile(fileName, newFileName);

        //Add the file info to the database
        sql = "INSERT INTO Documents (Division, Description, Category, Subcategory, Association, Link)
" +
            "VALUES ('"+division+", '"+description+", '"+category+", '"+subCat+", '"+association+",
            "'"+fileName.substring(fileName.lastIndexOf("\")+1)+"');";
        //document.write(sql);

        myDB.query(sql);
        document.write("Document added.");
    }
}
```

```

    }
}
results.js
Generates results for a search for items.

var room = queryParams("room");
var division = queryParams("division");
var system = queryParams("system");
var description = queryParams("description");
system = cleanString(system);
description = cleanString(description);

var myDB = finddb();

var things = table_list();
var systems = system_list();

var SQLs = [];
var roomClause = "";
if (room) {
    var roomClause = " and Rooms.Number = '\" + room + '\"";
}

for (var i in things) {
    var m = things[i][0];
    var t = things[i][1];
    var d = things[i][2];
    var s = systems[d-1];

    if (division && division != d) continue;
    if (system && s.indexOf(system) < 0) continue;

    var SQL = "select "+d+" as Division, " +
        "\""+s+"\" as System, " +
        t+".FamilyName as Description, " +
        "Rooms.Number + \" \" + Rooms.Name as Room, " +
        m+".Id as Id " +
        "from "+m+", "+t+", RoomAssociations, Rooms " +
        "where "+m+".TypeId = "+t+".Id " +
        "and "+m+".Id = RoomAssociations.Id " +
        "and RoomAssociations.RoomId = Rooms.Id";
    SQL = SQL + roomClause;
    if (description) {
        SQL = SQL + " and "+t+".FamilyName like \"%"+description+"%\"";
    }
    SQLs.push(SQL);
}

if ((!room || room == "ROOF") &&
    (!system || system == "Thermal & Moisture Protection" ) &&
    (!division || division == 7)) {
    var SQL = "select 7 as \"Division (level?)\", " +
        "\"Thermal & Moisture Protection\" as System, " +
        "RoofTypes.FamilyName as Description, " +

```

```

        "\"ROOF\" as Room, " +
        "Roofs.Id as Id " +
        "from Roofs, RoofTypes " +
        "where Roofs.TypeId = RoofTypes.Id";
    if (description) {
        SQL = SQL + " and RoofTypes.FamilyName like '%" + description + "%'";
    }
    SQLs.push(SQL);
}

if (SQLs.length > 0) {
    var BigSQL = SQLs.join(" union all ") + ";";
    var resultSet = myDB.query(BigSQL, {table:{stringOut:true, id:"eqresults"}});
    if (resultSet) {
        //for (var i in things){
        //var str=resultSet[i];
        //document.write("<a href=\"EquipmentInfo.html\">");
        //document.write(str);
        document.write(resultSet);
        //document.write("</a>");}
    } else {
        alert("Search returned no results.");
        history.back();
    }
} else {
    alert("Search returned no results.");
    history.back();
}
}

```

search_functions.js

Functions used in search.js, division.js, space.js and system.js, so they won't be duplicated.

```
var myDB = finddb();

var max_division = 16;
var systems = ["General Information",
    "Civil Work",
    "Concrete",
    "Masonry",
    "Metals",
    "Wood & Plastics",
    "Thermal & Moisture Protection",
    "Doors and Windows",
    "Finishes",
    "Specialties",
    "Equipment",
    "Furnishings",
    "Special Construction",
    "Conveying Systems",
    "Mechanical",
    //"% & , Foo %26",
    "Electrical"];

function search_init() {
    document.write("<form action=\"EqResults.html\" method=\"get\">");
    document.write("<table border=\"0\">");
}

// Room
function search_room() {
    document.write("<tr><td align=\"right\">Space:</td>");
    document.write("<td align=\"right\">");
    room_dropdown("&lt;All&gt;", "roomnumber");
    document.write("</td></tr>");
}

// System
function search_system() {
    document.write("<tr><td align=\"right\">System:</td>");
    document.write("<td align=\"right\"><select name=\"system\">");
    document.write("<option value=\"\">&lt;All&gt;</option>");
    for (var i=0; i < systems.length; i++) {
        document.write("<option value=\"\" + systems[i] + \"\">");
        document.write(systems[i]);
        document.write("</option>");
    }
    document.write("</select></td></tr>");
}

// Division
function search_division() {
    document.write("<tr><td align=\"right\">Division:</td>");
```

```

document.write("<td align=\"right\"><select name=\"division\">");
document.write("<option value=\"\">&lt;All&gt;</option>");
for (var i=0; i <= max_division; i++) {
    document.write("<option value=\"\" + i + \"\">");
    document.write(i);
    document.write("</option>");
}
document.write("</select></td></tr>");
}

// Description
function search_description() {
    document.write("<tr><td align=\"right\">Description:</td>");
    document.write("<td align=\"right\"><input style=\"width:98%\" name=\"description\">");
    document.write("</td></tr>");
}

function search_end() {
    document.write("<tr><td /><td align=\"right\"><input type=\"submit\" value=\"Submit\"
/></td></tr>");
    document.write("</table></form>");
}

```