

Drillhole Import Tool

Newmont Mining Corporation



Colorado School of Mines

Craig Carlson

Jennifer Francis

Lars Walen

Table of Contents

Introduction

Client Description 2

Product Vision 2

Requirements

Functional 3

Non-Functional 3

Risks 4

System Architecture

Graphical Representation 5

System Architecture 5

Technical Design

Design Diagrams 10

Design Aspects..... 11

Design Decisions and Implementation

Decisions..... 12

Implementation..... 16

Results

Performance Testing Results..... 17

Results of Usability Tests..... 17

Future Work 17

Appendix 18

Introduction

Client Description

Newmont Mining Corporation, based in Greenwood Village, Colorado, is the second largest mining company in the world specializing in gold mining, with operations in the United States, Australia, Peru, Indonesia, Ghana, New Zealand, and Mexico. Newmont's revenue topped \$9.8 billion in 2012 alone, attributed 50 million ounces of gold and copper, and had an operating cash flow of \$2.4 billion. Newmont's mission is to, "build a sustainable mining resource that delivers top quartile shareholder returns while leading the environmental stewardship and social responsibility." ("Newmont Mining Corporation." *About Newmont*. N.p., n.d. Web. 30 May 2013.) To find potential mines, Newmont performs exploratory drilling to investigate the mineral and rock types of a given area before mining. The exploration consists of drillholes and analyzing the rock cores produced.

Product Vision

The Drillhole Import Tool project consists of importing the rock core data into a user interface that the group will create. The program then proceeds to validate that the information is both complete and that the data makes sense. The validation is able to identify multiple problems in the imported CSV files and then allows a user to correct any mistakes. The program then converts the imported and validated data into a DHL type file (which is a file specific to Newmont's TSS Suite) which can then be exported to Newmont's software, TSS. The program is able to run in both the developers' user interface and behind the scenes using a command prompt. The main focus of the project was on validating and exporting the drillhole data.

Previously, users of TSS entered the downhole data manually; this program benefits the client by allowing the users to bypass the manual entry of the drillhole data and import the data through a program by converting a CSV file. This ultimately saves the users' time and, ideally, helps increase productivity.

Requirements

Functional

ASCII/CSV Input Specification UI

- Specify one *Collar File*, one *Survey File* and one or more *Assay Files*.

ASCII/CSV File Layout UI

- A file layout UI is needed for each of the import file types to facilitate column destination and assignment.

Validation UI

- This UI uses generic behind the scenes data structures since there may be additional importer modules that read data in different formats.
- The following validations are performed on the data and a user is allowed to correct, omit, or skip the invalid data for each error.
 - No survey records for a collar
 - Collar depth is not equal to survey depth
 - No assays for a collar
 - Overlapping intervals
 - Gaps or missing intervals
 - Duplicated collar/survey/intervals in different holes (i.e., the same collar locations which creates duplicate coordinate points)
 - Downhole survey readings with high dog-leg severity (i.e., drillhole fishhooks)
 - Negative or zero value intervals
 - Missing collar/assay/survey
 - Any of the three could be missing
 - Data intervals greater than the sample length (result of overlapping values)
 - Also, for each Assay value the following information is displayed
 - Alpha fields: a list of all values and their frequency
 - Numeric fields
 - Minimum, maximum, mean, and standard deviation

Export UI

- This user interface allows a user to save the imported drillhole data as a TSS DHL file. The user will need to specify the DHL file name.

Non-Functional

- The program needs to be written in C# so the developers at Newmont may alter the code in the future.
- The use of a Model View Viewmodel (MVVM) is used to separate the user interface from the functionality of the code.
- The graphical view uses Windows Presentation Foundation (WPF) to create a user interface that conforms to the current software used at Newmont.

- The client recommended using Infragistics to perform view binding while creating the interface.
- The software is not only able to run in the group's GUI but it is also able to run in the background using a command prompt.
- An XAM datagrid is used instead of a grid from Visual Studio.
- The program has a properties file so alterations can be made to the format of the input file.
 - Different orientations of the collar and drillholes
 - Different order of parameters in CSV file
- The group functioned off the assumption that the input CSVs are standardized.
 - If time permitted, the program is to allow for non standardized CSV files to be imported

Risks

Technology Risks

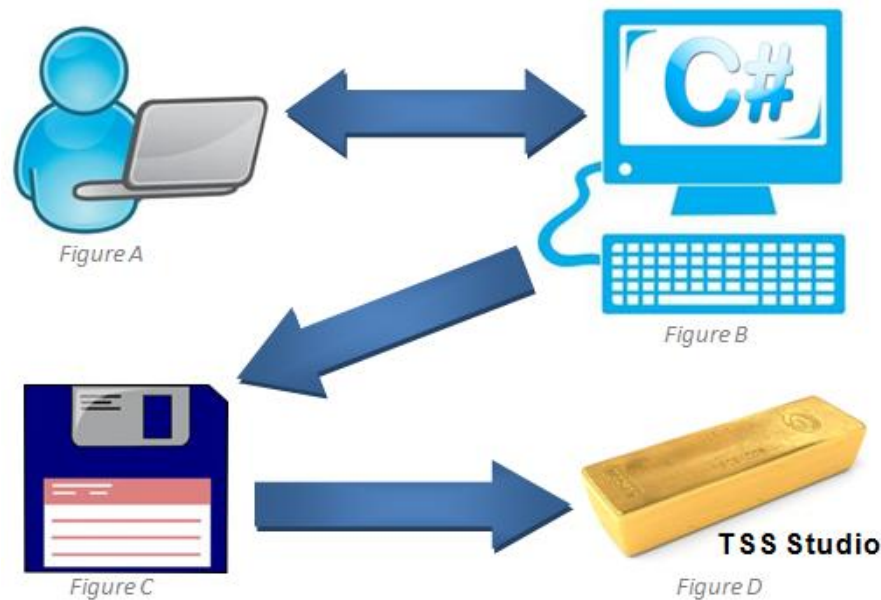
- The entire program functioning incorrectly due to improper code and lack of skill within the team was a possible risk.
- A user opting to skip or omit the errors flagged that should be altered while importing the file and essentially importing faulty data is currently a risk.

Skills Risks

- Initially, none of the team members were familiar with C#; several members of the team had to learn it.
- None of the team members were previously exposed to TSS, the team needed to develop a basic understanding of the product.
- The team also needed to become knowledgeable with WPF, MVVM, and XAM data grids.

System Architecture

Graphical Representation



System Architecture Representation

Figure A represents the user input of the three or more CSV files – one collar file, one survey file, and one or more assay files. *Figure B* represents the program written by the group for the client, Newmont Mining Corp in the language of C#. *Figure C* represents the DHL file wrapper provided by Newmont to convert the cleaned CSV files to TSS Studio's DHL. *Figure D* represents Newmont's TSS Studio.

The diagram begins by a user importing at least three CSV files containing collar, survey, and assay data that will eventually be exported to TSS Studio. The program, written in C#, will validate the file type and numerous data aspects. Any errors in the files will be displayed to the user, where then the user can edit, delete, or skip the error(s). Since the user is given the errors, and has the ability to fix them, the association between *Figure A* and *Figure B* is represented by a double edged arrow.

Once all errors have either been removed, skipped, or fixed the data is converted to a DHL file needed to export to TSS Studio via the wrapper provided by Newmont (*Figure C*).

Finally the complete DHL file is exported into TSS Studio (*Figure D*) where the user can then view the various drillholes through a 3-D representation generated by TSS.

Software Architecture

With regards to the flow and processing of the data, the group started by inputting three or more CSV files – one collar file, one survey file, and at least one assay file. Based on the file being imported, the program checks the headers and proceeds to assign the known column names accordingly. After assigning the column the program moves to validation, to ensure all files are

accounted for and the files contain the required pertinent information. The data then is converted into file class objects. If there are any errors or irregularities, the program sends a notification to the user and allows the user to correct the errors, which is easy to access given all non-validated data is stored in a separate data collection. Finally, a DHL file is created, the formatting is verified, and the created file is exported.

GUI

The user begins with a simple and relatively desolate **Data Validation** window (see *Figure 2.1*). The user must begin by import files by clicking *File > Import Files...* or clicking the link in the center of the screen before the tabs may be enabled.

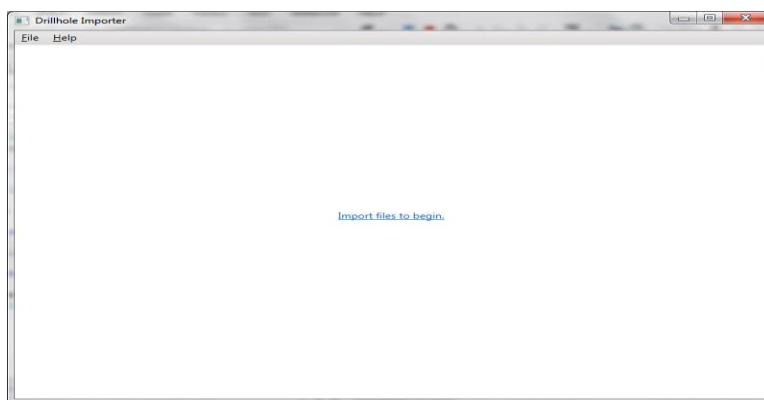


Figure 2.1

When the user clicks either of these, the **Import Files** window opens, which contains two tabs – the *Browse* tab and the *Assign* tab. From there the user may import a collar file, survey file, and one or more assay files.

The *Browse* tab of the **Import Files** (see *Figure 2.2*) window contains a **Browse** button for a collar file and a survey file, and an **Add...** button one or more assay files. A user is able to browse for multiple assay files at once. If there is not at least one path for each type of file, the **Import** button will not be activated.

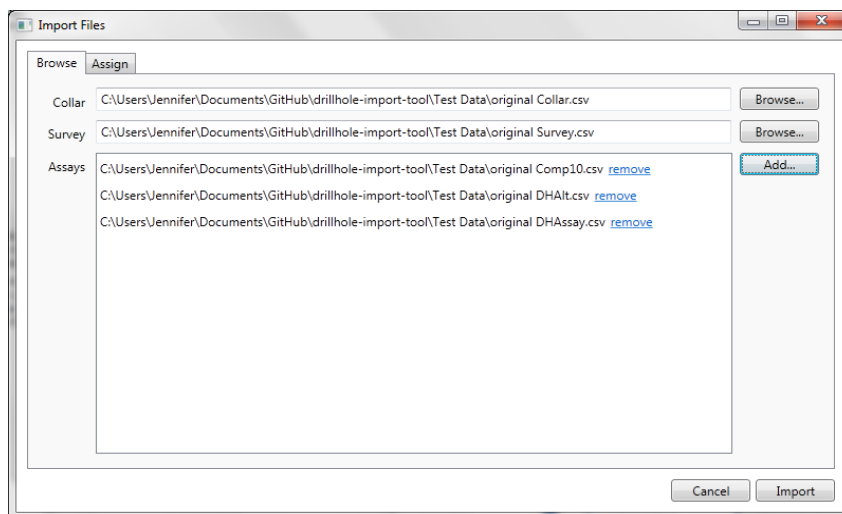


Figure 2.2

The *Assignment* tab of the **Import Files** window allows the user to specify the header row and which, if any, rows to skip for each of the CSV files. It also asks the user to enter the column names of each of the CSV files manually (see *Figure 2.3*).

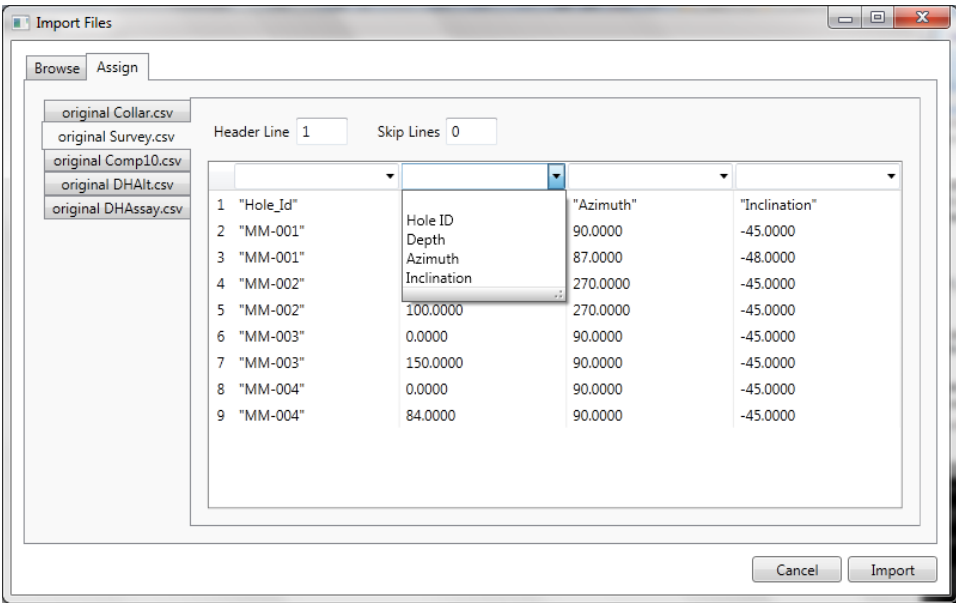


Figure 2.3

Once all required information is entered in the *Assign* tab, the **Import** button is activated and the user can import and move on to the now populated **Data Validation** window.

The **Data Validation** window consists of two tabs – an *Invalid* tab and a *Valid* tab. The *Invalid* tab includes a pane with all errors for the imported files organized in a tree structure (see *Figure 2.4*). For each error, the user must either choose to 'Fix', 'Omit', or 'Skip' the error. Relevant panes are displayed for each error type.

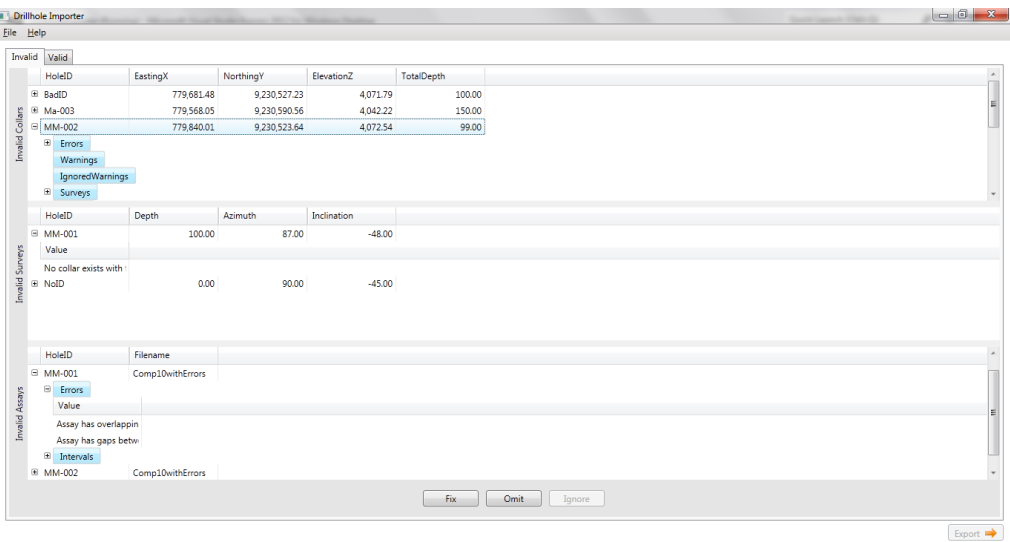


Figure 2.4

The *Valid* tab of the **Data Validation** window displays the valid data and it’s associations with other data, such as how the assay data is connected to the collar data (see *Figure 2.5*).

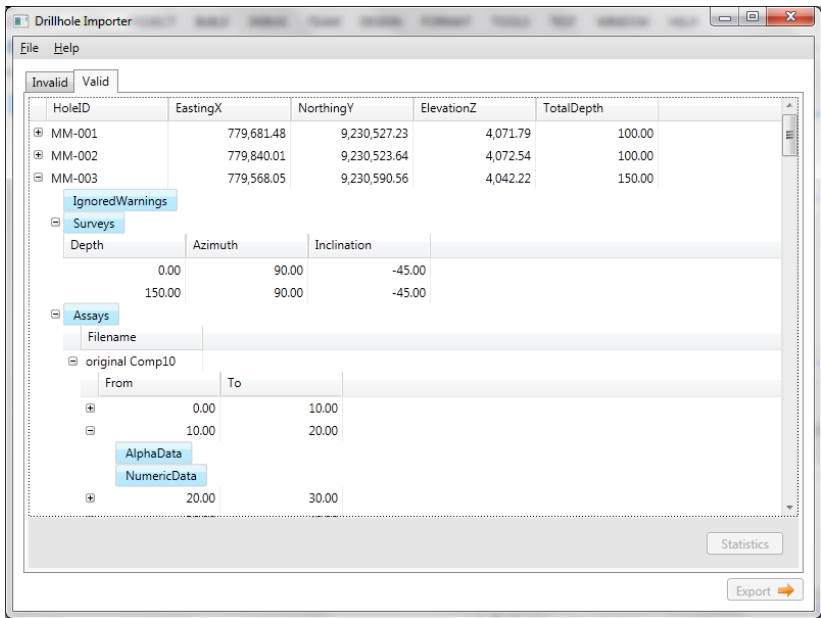


Figure 2.5

Assay files are also displayed with their associated statistics, as well as a list of each rock type and the number of times it appears in the file (see *Figure 2.6*). The statistics calculate the minimum, maximum, mean, and standard deviation. The calculations involved are outlined in the *Calculations* section of the *Appendix*.

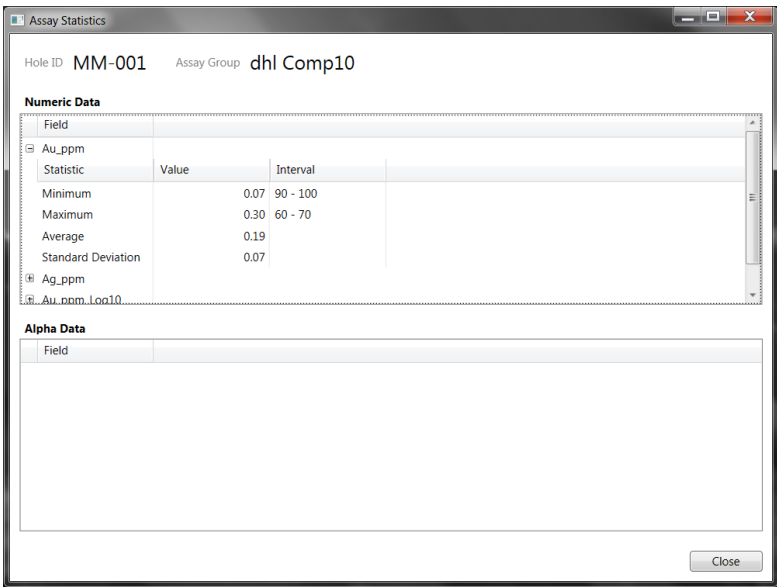


Figure 2.6

When all errors have been accounted for, the user is allowed to specify an output file path and then click the **Export** button. This should correctly create a DHL file with the imported data.

Console

The console version of the program focuses on running non-interactively. The `-b` flag is used to run the program without a GUI. The minimum information the program may be run with is one collar file, one survey file, one assay file, and an output DHL file path.

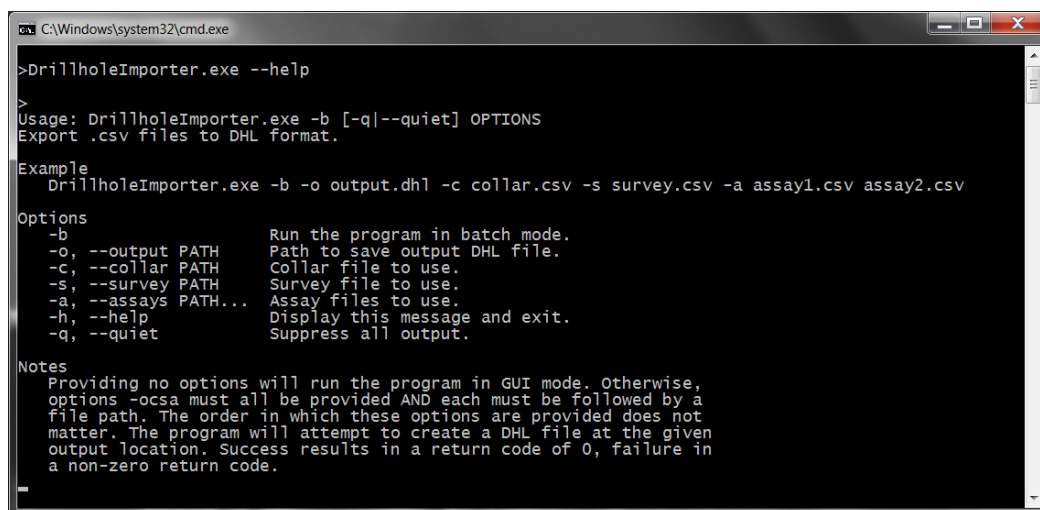
Example:

```
> DrillholeImporter.exe -b -o Output.dhl -c Collar.csv -s Survey.csv -a Assay.csv
```

In default mode, the program prints all errors preventing the creation of a DHL file, as well as any warnings that may be a result of malformed data. The output can then be redirected to a log file of the user's choice.

The optional argument, `-q`, does not print any output; it instead creates the DHL, if possible, or returns with a failure return code.'

The user may use the `-h` flag to see the help dialog (see *Figure 2.7*).



```
C:\Windows\system32\cmd.exe
>DrillholeImporter.exe --help
>
Usage: DrillholeImporter.exe -b [-q|--quiet] OPTIONS
Export .csv files to DHL format.

Example
  DrillholeImporter.exe -b -o output.dhl -c collar.csv -s survey.csv -a assay1.csv assay2.csv

Options
  -b                Run the program in batch mode.
  -o, --output PATH Path to save output DHL file.
  -c, --collar PATH  Collar file to use.
  -s, --survey PATH  Survey file to use.
  -a, --assays PATH... Assay files to use.
  -h, --help         Display this message and exit.
  -q, --quiet        Suppress all output.

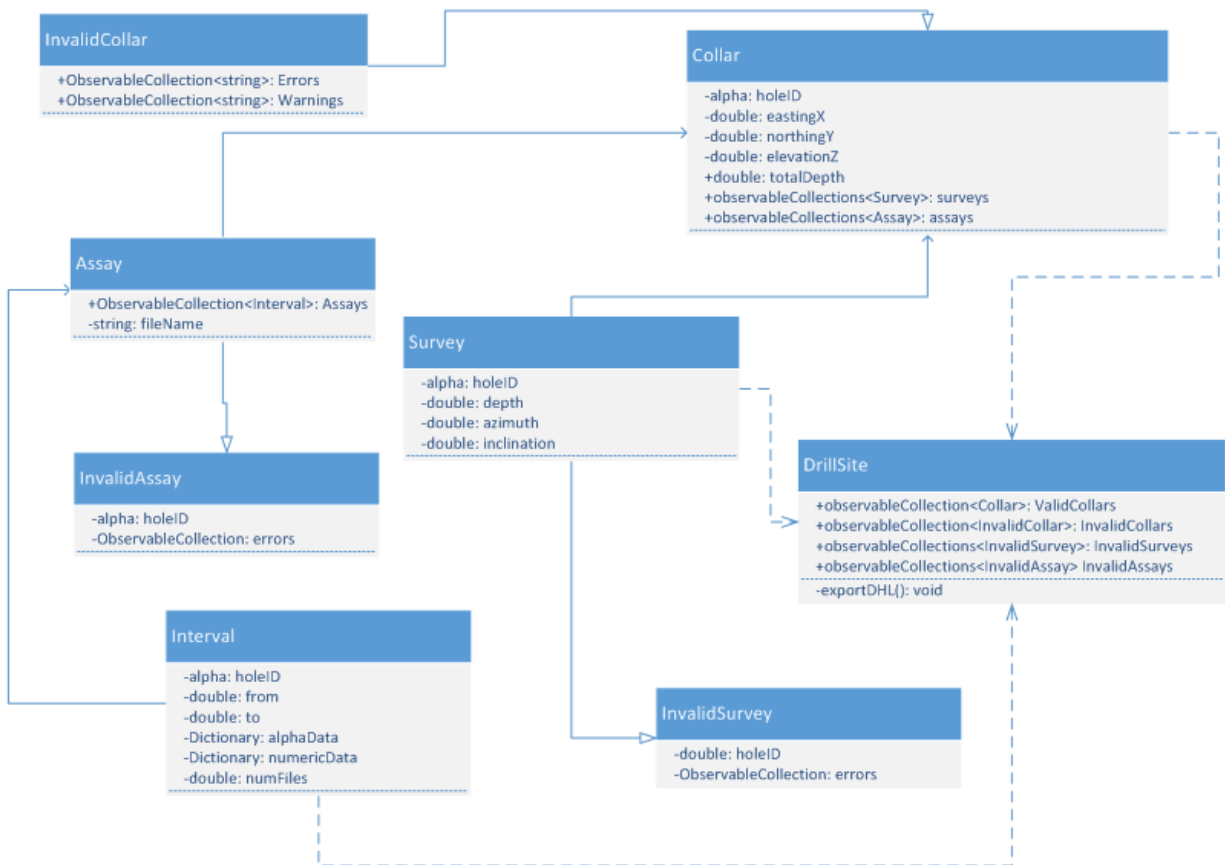
Notes
  Providing no options will run the program in GUI mode. Otherwise,
  options -ocsa must all be provided AND each must be followed by a
  file path. The order in which these options are provided does not
  matter. The program will attempt to create a DHL file at the given
  output location. Success results in a return code of 0, failure in
  a non-zero return code.
```

Figure 2.7

Technical Design

Design Diagrams

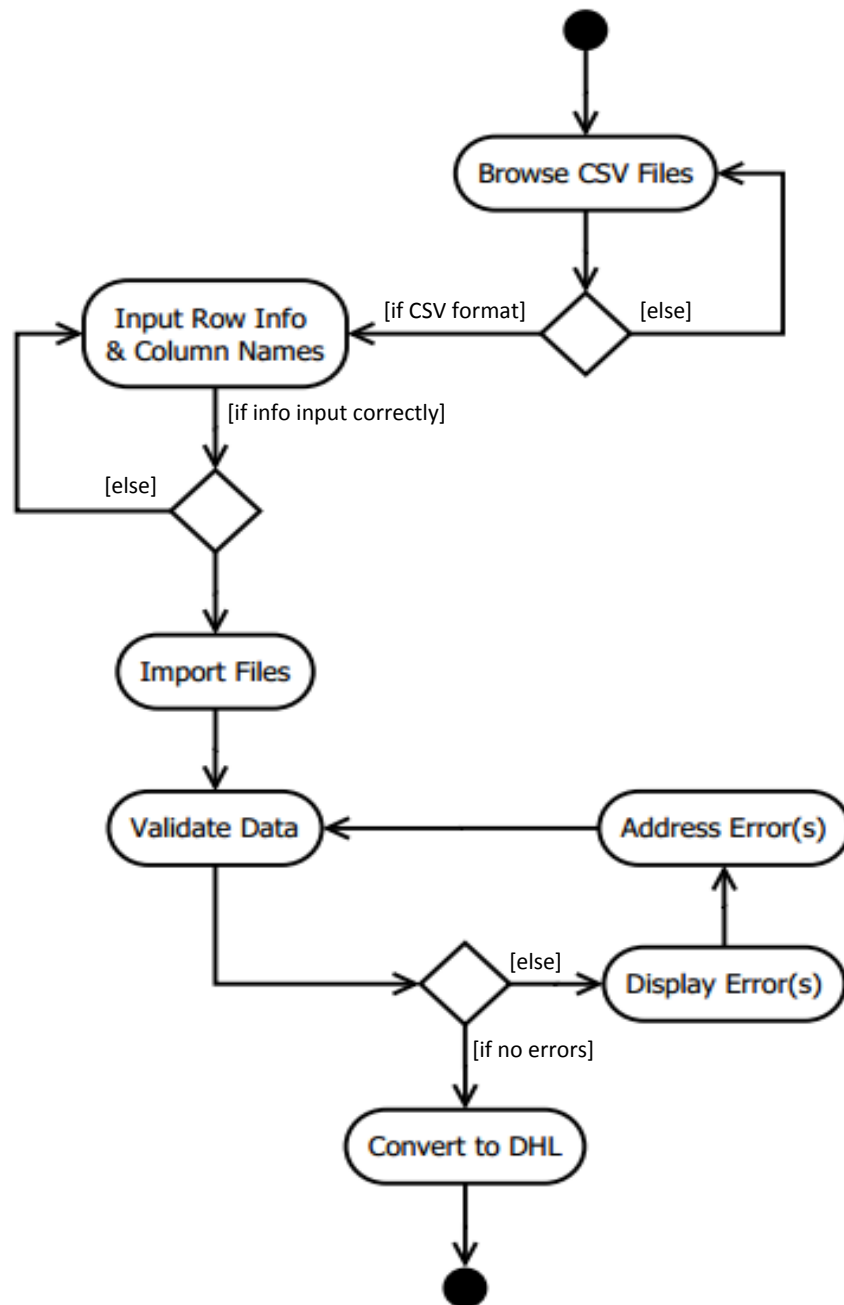
UML



Drillhole Importer UML Representation

In the main class, **DrillSite**, there are four observable collections. Of these collections, three of them are collections of invalid data, one for invalid collars, and another for invalid surveys, and finally invalid assays. The final collection holds the valid collars. Each of these collars have two of their own observable collections: one for the corresponding surveys and one for each assay file imported. The collar class also holds the coordinates x, y, z, and total depth of the collar. The survey class holds the depth, azimuth, and inclination for each interval of the drillhole. There is a level of abstraction between the intervals and its corresponding collar, called **Assay**, which partitions the intervals of a particular collar into different groups based on the files from which they originated. Each assay contains the intervals of the drillhole measured and the geological data associated with the intervals. Once all of the imported data has been allocated it is validated and sent to the DHL wrapper to be converted into a readable file for their software.

Activity Diagram



Activity Diagram Representation

The activity diagram illustrates the workflow involved in converting the file from a CSV to a DHL. The program hinges on correct user input, shown in the three diamonds which require correct file formats imported, correct row and column name input and alterations to any errors.

Design Aspects

One of most important design aspects for the drillhole import tool is the use of Windows Presentation Foundation (WPF). Windows Presentations Foundation, referred to as WPF, is a graphical subsystem used to create user interfaces in windows based applications. Previously known as Avalon, this subsystem is the primary connection between a user and the program and aims at unifying the user experience by having the program feel as integrated as possible with the Windows operating system. It was originally implemented using a .NET framework and rather than using the older GDI subsystem it runs using DirectX and is aimed at trying to standardize user interface design throughout windows applications. WPF allowed the group to create windows and pages within the application that the user inherently has knowledge of. The majority of the user interface was implemented using WPF; including buttons, browse features, and multiple tabs.

WPF is constructed using a version of XML called XAML (Extensible Application Markup Language) which allows for XAML elements to be mapped directly to object instances with its attributes map to that object's properties and methods. This process, known as data binding, allows for a user to easily interact with the underlying data structures as seamlessly as possible. This tool is valuable for the validation and data correlation pages because it allows a user to view the data stored in the collection as an easy to understand table and make changes to the data structures without knowledge of the code's presence. This necessity for an easy, practical user interface meant WPF needed to be the focal point of the program's design.

Design Decisions and Implementation

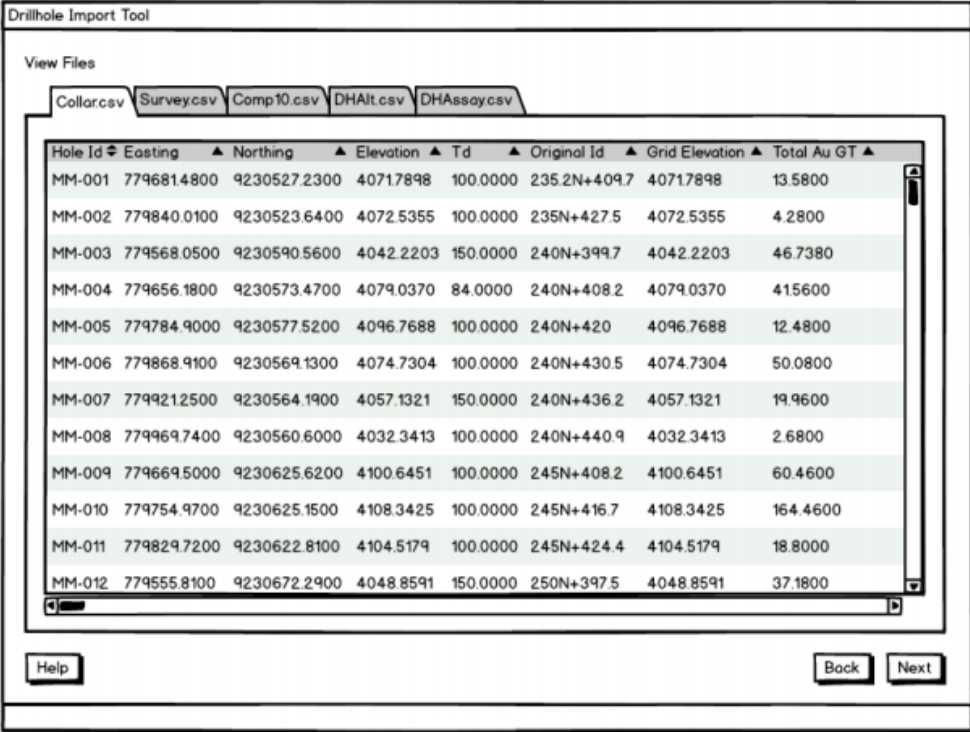
Decisions

Initial Decisions

Most of the big decisions regarding the program were made by the client. Such decisions were using Infragistics for binding while creating the interface, Windows Presentation Foundation (WPF) to create the user interface, writing the program in C# so Newmont may alter the code in the future, and using an XAM datagrid in lieu of a grid from WPF. These requirements were given to the group on day one in the meeting with Newmont.

Initial Interface Design

The interface went through several drafts before it landed on the current design. The initial interface design (see *Figure 5.1*) was based off of Leap-Frog.



The screenshot shows a software window titled "Drillhole Import Tool". Inside, there's a "View Files" section with five tabs: "Collar.csv", "Survey.csv", "Comp10.csv", "DHAIt.csv", and "DHAssay.csv". The "Survey.csv" tab is selected, displaying a table with 12 rows of drillhole data. The table has columns for Hole Id, Easting, Northing, Elevation, Td, Original Id, Grid Elevation, and Total Au GT. The data is as follows:

Hole Id	Easting	Northing	Elevation	Td	Original Id	Grid Elevation	Total Au GT
MM-001	779681.4800	9230527.2300	4071.7898	100.0000	235.2N+409.7	4071.7898	13.5800
MM-002	779840.0100	9230523.6400	4072.5355	100.0000	235N+427.5	4072.5355	4.2800
MM-003	779568.0500	9230590.5600	4042.2203	150.0000	240N+399.7	4042.2203	46.7380
MM-004	779656.1800	9230573.4700	4079.0370	84.0000	240N+408.2	4079.0370	41.5600
MM-005	779784.9000	9230577.5200	4096.7688	100.0000	240N+420	4096.7688	12.4800
MM-006	779868.9100	9230569.1300	4074.7304	100.0000	240N+430.5	4074.7304	50.0800
MM-007	779921.2500	9230564.1900	4057.1321	150.0000	240N+436.2	4057.1321	19.9600
MM-008	779969.7400	9230560.6000	4032.3413	100.0000	240N+440.9	4032.3413	2.6800
MM-009	779669.5000	9230625.6200	4100.6451	100.0000	245N+408.2	4100.6451	60.4600
MM-010	779754.9700	9230625.1500	4108.3425	100.0000	245N+416.7	4108.3425	164.4600
MM-011	779829.7200	9230622.8100	4104.5179	100.0000	245N+424.4	4104.5179	18.8000
MM-012	779555.8100	9230672.2900	4048.8591	150.0000	250N+397.5	4048.8591	37.1800

At the bottom of the window, there are three buttons: "Help", "Back", and "Next".

Figure 5.1

The interface was based on Leap-Frog's design because Newmont's preliminary design request given to the group included screen shots from Leap-Frog.

Second Interface Design

After sending the preliminary design document to Newmont for review the client requested adding more tabs in the interface on the view files form. A *Data Association* tab and a *Data Validation* tab were added (see *Figure 5.2*).

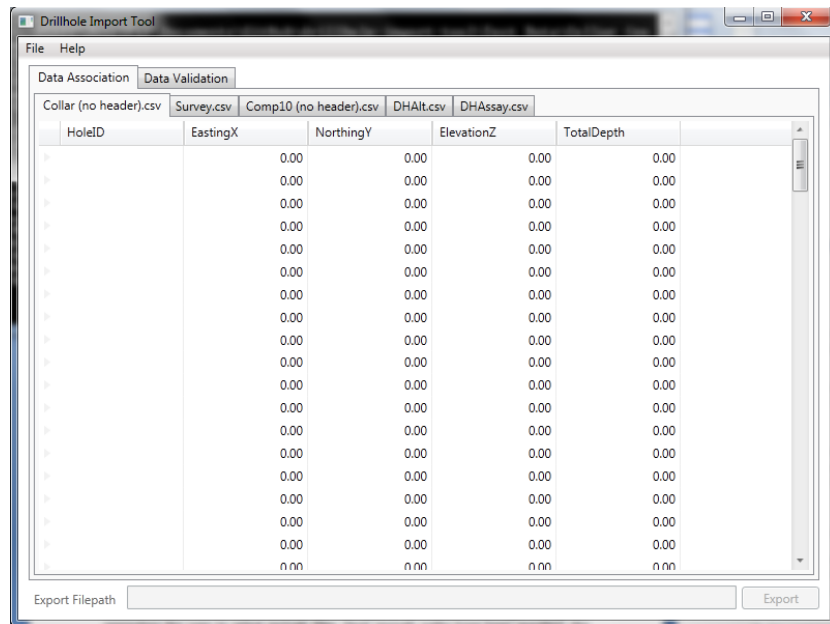


Figure 5.2

The reason behind the addition of the tabs is to differentiate between data association and data validation. Initially, only a validate data form was displayed (as seen in *Figure 5.1*), not a data association form. Eventually, the client requested the *Data Association* tab be removed, this is explained further in the *Third Interface Design* section.

Third Interface Design

The initial interface design included a page that displayed the data from the CSV files and allowed the user to select the first line of data to read, preventing the program from reading any unnecessary header in the file (see *Figure 5.3*); then another data display window which would allow the user to select the correct column name for the data in each imported file from various dropdown menus (this was not fully implemented before the design was changed, therefore no screen shot is included).

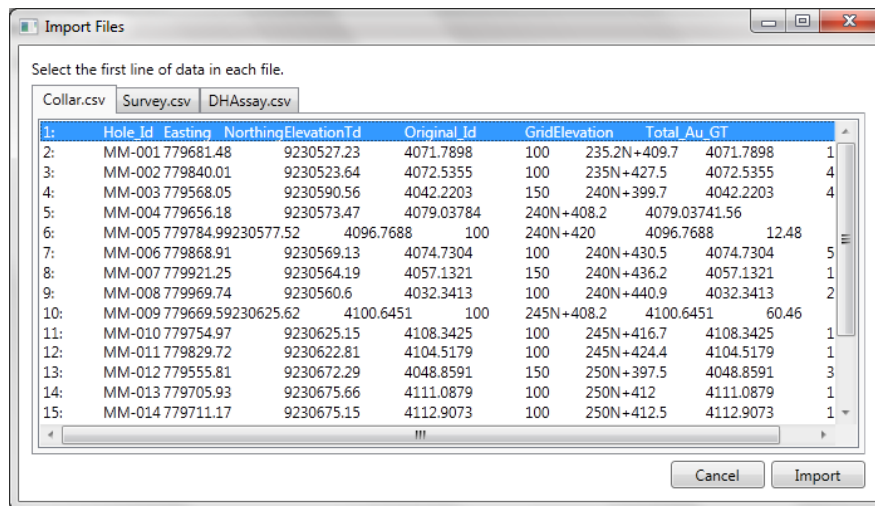


Figure 5.3

However, this design was changed once the client requested the two pages be combined into one validate and import page. This page is called the *Assign* tab on the **Import Files** window, which allows the user to state which rows of each file the data begins, or which rows need to be skipped along with allowing the user to manually input the column names for each file (see *Figure 5.4*).

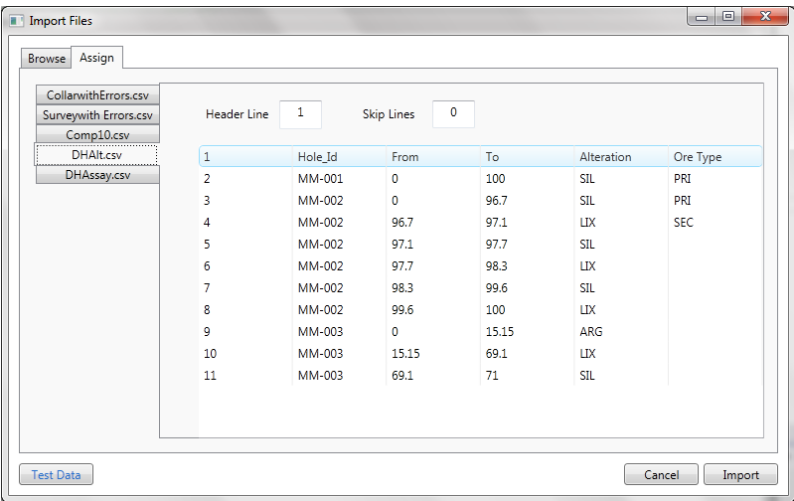


Figure 5.4

This change caused the *Data Association* tab to be superfluous, therefore was no longer in the program; though a similar tab was re-added upon the client’s request which is explained in the *Final Interface Design* section.

Final Interface Design

The final interface design contains the same **Import Files** window. The **Drillhole Importer** window was changed. The window now includes two tabs – a *Valid* tab and an *Invalid* tab (see *Figure 5.5*). (*Figure 5.5* illustrates the two tab design, but only displays the *Invalid* tab.) The two tabs were added at the request of the client. The client felt the single pane **Drillhole Importer** window was too cluttered and asked the group to separate the data from the errors.

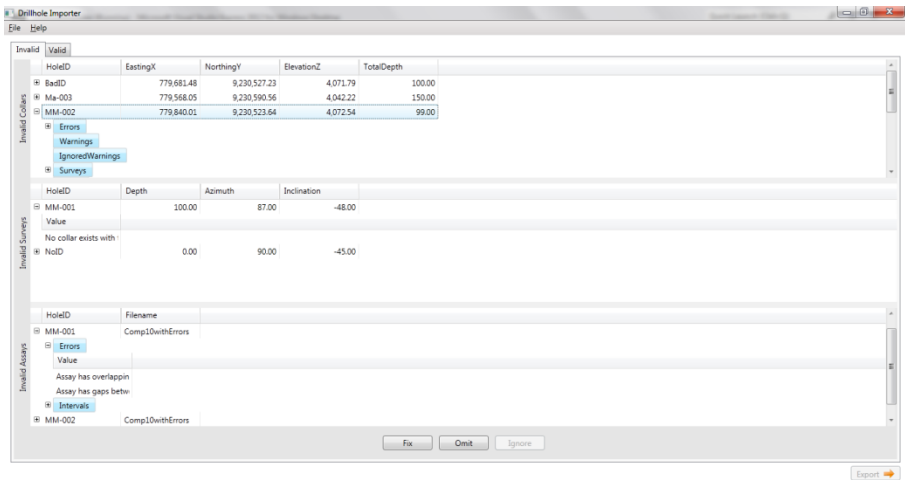


Figure 5.5

The *Browse* tab of the **Import Files** window is the same as the initial **Browse** window from the first user interface mockup (see *Figure 5.6*).

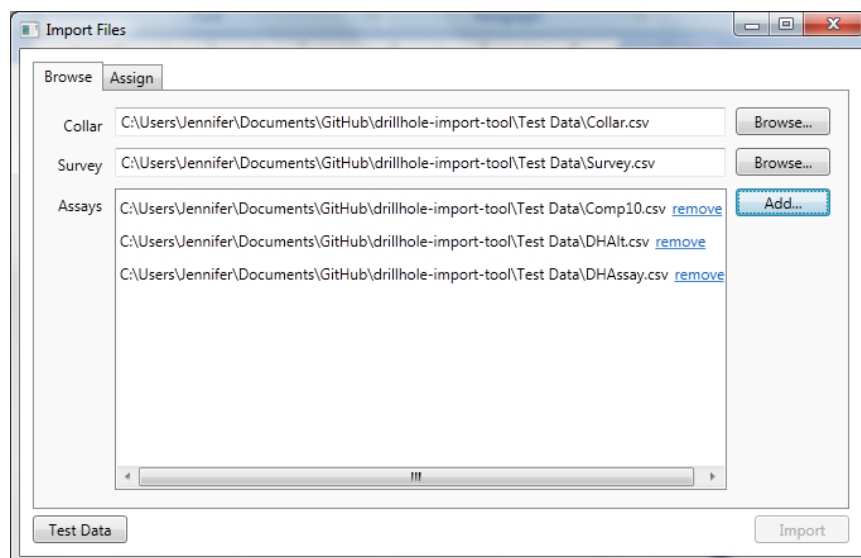


Figure 5.6

Implementation

The greatest issue encountered during the implementation process was the ever changing user interface. Unfortunately, the client's ideal interface was often misinterpreted by the group, causing several interface restructures. Because the group was unfamiliar with drillhole data, there was a lack of understanding in the optimum way to display data errors or entering column titles. As the project progressed, so did the group's knowledge of drillhole data, greatly benefiting the user interface design.

The open dialog with Newmont was another benefit that helped the interface evolve into the final product. The client was easy to contact and was more than willing to help when a road block was encountered. Weekly client/group updates were valuable since the meetings were a way to present any progress made with the program, and any changes that will need to be made. The open dialog allowed discrepancies to be caught early, thus enabling the program to be altered along the way as opposed to dealing with large scale changes during the last week of the project.

Data binding was an additional road block encountered during the program's implementation. Ultimately, the group felt the theory of data binding is fantastic, since it allows for a level of abstraction of the code from the user that is otherwise impossible. It allows for one to restrict the user to specified data and allow the user to interactively change the hidden data structures with ease. However, it comes at a price; it requires new, observable data structures that can be accessed by the textboxes and graphs. The complexity of data binding intensifies with both one-way and two-way binding; which determines what is displayed to the user and whether or not the data can be altered. This requires the need for specific variables to become binding sources, which is a complex and less intuitive version of getter/setter. If a user does alter the data then the changes must be applied immediately to both the hidden data structures and potentially displayed data in other sections of the user interface.

Results

Performance Testing Results

The project was to import a set of CSV files related to a drillhole Newmont Mining Corporation samples to explore new areas to mine. Read in the files, and allow a user to interact with the data to see how each drillhole was laid out before converting the files into a DHL file. The group accomplished this with the **Import Files** window and *Assign* tab, in which a user imports multiple files simultaneously and may specify what the data represents.

The project requirements then requested to validate the imported data to ensure the values are correctly formatted, and each drillhole has all the required parameters needed to model in Newmont's software, TSS Suite. The program contains a validation window, the **Drillhole Importer** page, which displays the hierarchy of the valid data and tables of the invalid data with corresponding errors. The program allows a user to 'Fix', 'Omit' or 'Ignore' each error that has been detected. Once all the errors have been addressed, the user may then export a DHL file. This process takes the validated data and sends it to a DHL wrapper application supplied by the client.

The client wanted a version of the application that could be run from the command line, which was successfully achieved. However, there are some features that were not accomplished due to time constraints. The client wanted the option of importing non-CSV files, along with the ability to drag and drop files into the import window.

Results of Usability Tests

The client is pleased with the outcome of the drillhole import tool. While there were concerns about the group completing the program on time during the final week of implementing the data validations, the product is fully functioning and has met all client requested requirements. The program successfully converts files and allows a user to view the downhole data in Newmont's TSS Studio. This program helps cut time by permitting a user to bypass the manual entry of the downhole data into TSS by using the program instead.

Future Work

The client, Newmont Mining Corporation, specifically requested the program be written in C# for future changes/upgrades. A possible future alteration of the program would be implementing a drag/drop feature instead of browsing for the CSV files. This would save Newmont's users file browsing time, particularly if the folder with the collar, survey, and assay files is already open.

Further potential changes made to the drillhole import tool include allowing the import of various file types, not only CSV files. This functionality was explained by the client during the first meeting. However, given the time frame of the project, it was understood that such functionality would not be feasible, though it will most likely be an enhancement implemented in the future by Newmont. Allowing various file types to be imported will benefit the user base by not forcing the users to create CSV files, which will ultimately save time.

Appendix

Appendix Contents

Product Installation Instructions	19
Development Environment Description.....	19
Calculation Details.....	19
Coding Conventions	20
Program Use Directions	20

Product Installation Instructions

The client, Newmont Mining Corporation, requested the hard code from the program. They have taken it upon themselves to install the program as they see fit. There are currently no installation instructions for this program.

Development Environment Description

The development environment utilized by the group was a breakout room in Brown Building West of the School of Mines campus. The room provided a neutral space to meet and work together. Work was also preformed onsite at Newmont. Working at the client site enabled the group to ask questions and receive client feedback instantly. This increased productivity and prevented any road blocks from forming.

Pair programming was also employed during certain aspects of development, particularly during the more difficult features of the program. Pair programming allowed the group to bounce ideas off one another and problem solve as a team.

Calculation Details

Dogleg Severity Calculation

The equation for calculating dogleg severity from the imported survey files is as follows,

Let $DLS = \text{DogLeg Severity} : DLS \in [0^\circ, 180^\circ]$,

let $I_1 = \text{Initial Inclination}, I_2 = \text{Final Inclination} : I_1, I_2 \in [-90^\circ, 90^\circ]$,

and let $A_1 = \text{Initial Azimuth}, A_2 = \text{Final Azimuth} : A_1, A_2 \in [0^\circ, 360^\circ]$,

$$DLS = \frac{180^\circ}{\pi} |\cos^{-1}[(\cos I_1 \times \cos I_2) + (\sin I_1 \times \sin I_2) \times \cos(A_2 - A_1)]|.$$

A dogleg severity greater than 10° is flagged as an error.

Mean Calculation

The equation for calculating the mean from the imported assay is as follows,

Let $\bar{x} = \text{Mean}$

and let $N = \text{Number of data entries} : N \in \mathbb{R}^+$,

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

Standard Deviation Calculation

The equation for calculating the standard deviation from the imported assay is as follows,

Let $S = \text{Standard Deviation}$

and let $N = \text{Number of data entries} : N \in \mathbb{R}^+$,

$$S = \sqrt{\frac{1}{(1-N)} \sum_{i=1}^N (x_i - \bar{x})^2}$$

where

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

Coding Conventions

A handful of variables names were chosen by the client; but for the rest of the variables, camel case is used. Proper indentation is displayed through the program and each bracket is written on a separate line. Comments appear above the variables and methods, not below or beside. The group also ensured each method is separated by a blank line for better readability.

Program Use Directions

This section contains all program use documentation located in the **Tools -> Help** page of the program. The Help page has two tabs – the *Drillhole Import Screen* tab for the Drillhole importer page of the program, and the *Import Files Screen* tab, for the Import Files page of the program.

Drillhole Importer Screen

To begin importing files, click on the **Import files to begin icon** in the center of the screen or select **File -> Import Files...** at the top left corner of the page. To exit the page either click **File -> Exit** from the top left corner of the page or click the red **X** in the top right corner of the page.

Note: See the **Import Files** screen section to properly select the files to be imported.

Once the files have been properly imported from the **Import Files** screen, you will be taken back to the **Drillhole Importer** screen where you may view your data and your potential error(s). The **Drillhole Importer** screen contains two tabs – an *Invalid* tab and a *Valid* tab.

On the *Valid* tab of the **Drillhole Importer** screen you are able to view your data associations via a nested view. Each 'Hole ID' is displayed and may be expanded to view the survey and its associated 'Hole ID' information and the assays and their 'Hole ID' information. To expand and view a 'Hole ID', subsequent survey or assay, or an assay's subsequent information, click on the **+** box next to the data you wish to expand.

The *Invalid* tab of the **Drillhole Importer** screen states the error(s) and their locations in the data. Errors are categorized as 'Invalid Collars', 'Invalid Surveys', and 'Invalid Assays'. Each of these error categories has its own field. You have the option to 'Fix', 'Omit', or 'Skip' each error. To view the errors expand

each error set by clicking on the + box next to the data you wish to expand. There are error sets for each field – ‘Invalid Collars’, ‘Invalid Surveys’, and ‘Invalid Assays’. Once the errors have been expanded, you will be able to view the invalid data and the reason why it was flagged as an error. To address an error, click the error you wish to address and click either the **Fix** button, **Omit** button, or the **Ignore** button depending on how you wish to handle the error. The buttons are located beneath the ‘Invalid Assays’ field at the bottom of the *Invalid* tab on the **Drillhole Importer** screen.

The **Fix** button allows you to fix the error by typing in the necessary changes. To change the error, click on the invalid data and type in the correct data, and then click the **Fix** button. Once the **Fix** button has been clicked, the data will be rechecked for errors. If the corrected data is still flagged as an error, you will see it reappear in the ‘Errors’ section. If the corrected data is now acceptable to import it will be moved to the *Valid* tab of the **Drillhole Importer** screen.

The **Omit** button allows you to decline to import the invalid data completely. Once the **Omit** button has been clicked, the error will move from the ‘Errors’ section to the ‘Omitted Errors’ section.

The **Skip** button allows you to ignore the error and move on to the export process. Once the **Skip** button has been clicked, the error will be removed from the ‘Errors’ section and will be included in the final export without changes.

You may view Assay Statistics if you choose. These statistics and the Minimum, Maximum, Mean, and Standard Deviation. To view these statistics click the **Statistics** button on the *Valid* tab on the **Drillhole Importer** page.

Once all errors have been addressed, the ‘Export Filepath’ field will activate and you may enter the file path in the ‘Export Filepath’ field, and then click the **Export** button. This will convert your error-free data into a *.DHL file readied for TSS.

You may import more files by selecting **File -> Import Files** and following the steps to import files outlined in the [Import Files Screen](#) tab of the [Help](#) page.

Import Files Screen

The **Import Files** screen allows you to import files, view the first several lines of the files, and assign the column names of any imported files without a header. It also allows you to skip any unnecessary file header in the imported file. The screen has two tabs visible in the top left corner – *Browse* and *Assign*. To switch between the tabs, click on the tab you wish to view. To exit the page either click the **Cancel** button at the bottom left corner of the page or the red **X** in the top right corner of the page; this will bring you back to the **File Importer** screen.

Note: All imported files must be *.csv files.

Click the **Browse...** buttons on the *Browse* tab to add Collar and Survey files. To replace a file already populated in the Collar or Survey field, click the **Browse...** button again and select the new file you wish to import. To remove a Collar or Survey file, click inside the Collar or Survey field with your mouse and delete the file.

Click the **Add...** button on the *Browse* tab to add one or more Assay files. You can add more than one Assay file at a time. To remove an Assay file from being imported, click the **Remove** icon next to the file you wish to remove in the Assay field. You may click the **Add...** button and add as many Assay files as you need.

Note: A Collar file, Survey file, and one or more Assay files must be selected before any files may be imported. Once all criteria are met, the **Import** button at the bottom right corner of the page will become enabled.

Note: If you attempt to import the same file into one or more fields, an error message will appear. Duplicate files are not allowed.

To view and verify the files before importing them, navigate to the *Assign* tab. Select the file you wish to view by clicking the file name in the list on the left of the **Import Files** page. You may use the scroll bar at the bottom of the file name section on the left to view the entire file name.

Once the file you wish to view has been selected, you can designate the 'Header Line' by clicking in the 'Header Line' field and entering the line number of the file that the header of columns are located. If there is not a header line in the file, enter the number '0' into the 'Header Line' field.

For files without headers, you will need to click the top of the column and type in the header name for each column in the file.

If there are any other lines in the file you do not wish to be imported, click in the 'Skip Lines' field and enter the number of lines you wish to skip. As you enter the skipped lines, your data will automatically update in the data display directly beneath the 'Skip Lines' field. If a number is not entered in the 'Skip Lines' field the program will default to '0'.

Note: If something other than a number is entered in either the 'Header Line' field or the 'Skip Lines' field, the text will turn red as a warning. No error message will be displayed, the 'Header Line' field will simply default to '1' and the 'Skip Lines' field will default to '0' once you either click out of the field or click the **Import** button.

Note: The lines skipped will be at the beginning of the file. For example, if you enter '1' into the 'Header Line' field and '3' into the 'Skip Lines' field, then your data will display the header and the data will begin on line 5. If you enter a number in the 'Skip Lines' field greater than the number of lines contained in the file, the number entered will turn red and the **Import** button will deactivate until the problem has been corrected.

Once all the correct 'Header Line' and 'Skip Lines' information has been entered for each file and is ready to be imported, click the **Import** button at the bottom right of the **Import Files** screen. You will then return to the **Drillhole Importer** screen.