**COM570: Project**

**Dissertation**

**School of Computing & Information Engineering**

**Jason McKee (B00553970)**

**Data Cleansing System**

**Supervisor: Gerry Parr**

**Second Marker: David McSherry**

**Date: 2nd May 2014**

****

Plagiarism Statement

I declare that this is my own work and that any material I have referred to has been accurately and consistently referenced. I have read the University’s policy on plagiarism and understand the definition of plagiarism as given in the Project Handbook. If it is shown that material has been plagiarised, or I have otherwise attempted to obtain an unfair advantage for myself or others, I understand that I may face sanctions in accordance with the policies and procedures of the University. A mark of zero may be awarded and the reason for that mark will be recorded on my file.

Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Acknowledgements

Thanks to anyone who contributed directly or indirectly to the project, in any way.

Contents

[Abstract 1](#_Toc383980825)

[Introduction 2](#_Toc383980826)

[Problem Statement 2](#_Toc383980827)

[Software Methodology Chosen 3](#_Toc383980828)

[Aim of Project 4](#_Toc383980829)

[Objectives 4](#_Toc383980830)

[The Project 5](#_Toc383980831)

[Report Structure 6](#_Toc383980832)

[Analysis 6](#_Toc383980833)

[Design 19](#_Toc383980834)

[Implementation, Testing and Evaluation 20](#_Toc383980835)

[Conclusions 21](#_Toc383980836)

[References 22](#_Toc383980837)

[Appendices 22](#_Toc383980838)

[A1. Analysis Models 22](#_Toc383980839)

[A2. Design Models 22](#_Toc383980840)

[A3. Code 22](#_Toc383980841)

[A4. Test Suite 22](#_Toc383980842)

[A5. Questionnaire Results 22](#_Toc383980843)

# Abstract

The problem is that the NISRA (Northen Ireland Statictics and Research Agency, 2014) receive several datasets throughout the year which contain thousands of addresses on which critical statistical data outputs rely on such as Census Data. It has been noticed that a percentage of those address cannot be validated as correct address. This is due to the addresses being captured through forms completed by people which lead to human error in the addresses. If an address cannot be confirmed it is not then reliable enough to be used in key statistical data. The current process of confirming these addresses is expensive in both time and money.

It was then proposed to create a less expensive application that would carry out the current process more quickly. The application would make use of the current software Microsoft Access and the programming language Visual Basic with Applications (VBA). This is because current applications that NISRA use are built using these utilities and therefore they are recognised by the staff and also reduces training to a minimum. Another request was the ability to carry out the process through Microsoft SQL Server (Microsoft, 2008) as this is where the data for processing is held for cleansing. This was acceptable as MS Access and MS SQL Server can communicate quite effectively. The finished solution is to carry out the majority of the processes within MS SQL Server as Access allows this via pass through queries which allows for quicker process times and also incorporates the current security procedures within NISRA. The purpose of MS Access is to provide a user friendly front end User Interface that the staff can use effectively.

To date an application was created within MS Access to act as a front end user interface for the address validation process which is carried out in the back end within MS SQL Server. Many improvements have been made over the development of the project as will be outlined in this report but the original idea has been followed with the addition of some useful features. The application works as required and this report will outline some possible future versions of this project.

This current version of this project is being used successfully by staff at NISRA.

# Introduction

## Problem Statement

Northern Ireland Statistics and Research Agency (NISRA) produce official statistics and social research on Northern Ireland every year which inform public policy and associated debate such as the Census. Many of these statistics are widely available to the public which means the results need to be as correct and accurate as possible. NISRA receive several datasets throughout the year from different government organisations which contain thousands of addresses located within the whole of Northern Ireland. The datasets which NISRA receive from other government organisations tend to be formatted in a specific way that is relevant to the organisation sending the dataset. It has been noticed that a percentage of those address cannot be validated as correct address. This is due to the addresses being captured through forms completed by people. These forms are normally read into database via Optical Character Recognition (OCR) software or again by humans. Both of these processes have disadvantages such as a person’s handwriting may not be legible by either methods or it may be just certain characters that are not legible. You will also find that some letters can be mistaken for other letters which generate an incorrect address which is quite hard to detect as that particular address may meet the visual criteria of a correct address. If an address cannot be confirmed as correct, it is not then reliable enough to be used in key statistical data and therefore has to be rejected. The current process of confirming these addresses is expensive in both time and money as the application that is currently used requires extensive training by trained technician. The current application is not specifically tailored to be used for just this purpose therefore there are quite a number of additional features within the application that are not used by the staff. As the software is so expensive only one license is affordable to purchase for NISRA and can only be installed on one computer. Also during certain times of the year this software is needed by more than one user therefore a queue for this software forms. Currently the database used to cleanse the address data for the current software can only be updated every quarter year. This means that any addresses that have been added to new datasets received by NISRA may not be included in the current version until the software is updated and therefore those addresses cannot be verified and used in the collection of statistics. The same can be said for buildings that may have been demolished between updates of the software data. The process of the updated data within the current software is that the company which produces the software collect their updated data and run several tests on this along with their processes to verify that the software continues to work before pushing the update out to the customers. This means that the update that NISRA receive for the software is already out of date when they receive the update. Depending on the growth of housing developments this can be a massive issue which will be explained further in the report.

Therefore the underlying problem for NISRA is that the software they currently use is complex, requires extensive training, expensive and out of date in terms of data.

## Software Methodology Chosen

A methodology to guide the project during development needs to be chosen it has therefore been decided that this methodology will be ‘Agile Time Development’. This methodology has been chosen as the developer wanted a methodology that they would be able to have a timeline to follow, this would make the development of this project run smoother as every step would be outlined and no time would be wasted trying to figure out what part of the project needs to be completed next. A large amount of time would be spent on developing the work plan for the development of the project at the beginning of the project, this means that only small amounts of time would be required to update this plan and this plan is called the ‘Project Backlog’. The project backlog will consist of a list of tasks, and for each of these tasks a difficulty, time frame and estimated completion date will be shown. At the start of each week the developer will develop a sprint which involves choosing specific tasks from the Project Backlog that will be tackled that week. The developer chooses tasks that they will be able to complete within the week and that would require working out the how long the tasks will take. Some of the tasks in the Project Backlog depend on the completion of other tasks so therefore can only be included in a sprint when the relied on task has been completed.

The advantages of using this methodology is that having all tasks planned out allows the project to run very smoothly as no time is wasted on planning, also because of the project backlog and some of the tasks depending on others the developer will have developed a prototype for NISRA to test and receive feedback from NISRA. When all the tasks have been assessed and given a length of time an estimated time of completion can be predicted for the developer.

The developer will need to have daily stand ups every day which they will outline what tasks will be carried out during that day. Also in this process the developer will assess how the previous days tasks went and if they need to be carried over to the next day. This process allows the developer to adjust both the tasks ahead and also the weeks sprint so that the developer knows what will succeed and what will not be completed in the week.

## Aim of Project

The solution to this problem is to create an application that can carry out the same process as the current software which is the ability to take addresses in a dataset and test their validity by running them with defined matching algorithms against an already clean address dataset. This software will save NISRA expenses in both software and training as NISRA have access to a pre-cleansed address dataset. The software must achieve at least the same accuracy percentage as the current software as well as having an easy to use interface so that all relevant members of staff can use the software. The software must be able to run on multiple computers so that if it is needed by more than one person at one time it can be done. The software must be able to be updated as and when a new updated version of the pre-cleansed dataset becomes available.

## Objectives

The main objective is to develop an application that will accept a given dataset that contains addresses that need to be validated. The application will run user chosen algorithms to validate the addresses against an up-to-date pre-cleansed dataset which will append a Unique Property Reference Number to each address that the application is able to validate. This process will need to output a percentage of valid addresses that matches or is greater than that of the previous software. Additional objectives include:

* Making use of the currently used data store Microsoft SQL Server to access the datasets of addresses.
* Having an easy-to-use interface that can be operated with little to no training. This includes the processes of the application being almost fully automated.
* Allowing the user to choose which algorithms to run to validate the addresses.
* Being able to update the pre-cleansed dataset with ease for a member of staff that may not be advanced in IT.

There are also personal objectives to be achieved and these are:

* To further my development in application development, programing languages and to understand the processes involved in creating an application.
* Gaining more experience in problem solving and being able to produce relevant documentation to back up to development of a solution as well as implementing the solution for others benefits.
* Building on current knowledge of management solutions and gaining more experience in professional communication and time management.

## The Project

Research has been carried out with in regards to what products are currently available and that can achieve the objectives listed above. The products have been tested and assessed which can be read further on in the report. A detailed analysis of what NISRA are looking for in terms of requirements has been carried out to make sure the application can meet the needs of NISRA. A large amount of the research was carried out on the development of the user interface as the back end data store was already decided on, therefor extensive research was needed on what software and programing languages operate well with the back end data store.

Throughout the project communications with NISRA were ongoing so that all areas of the project were covered and that any complications that may have arose could be dealt with and a proposed solution could be explained and then implemented.

****The project has been developed and has achieved the entire primary requirements that NISRA sought after at the beginning of the project. Currently the application has achieved validation rates matching that of the previous software. The project has also saved NISRA a substantial amount of money in both training and purchasing of the software.  
NISRA are very pleased to have this software replace their previous software and also are open to future development of this software to add further features.

Figure 6 Agile Time Management Model

## Report Structure

The remainder of this report is structured as followed:

***Analysis***

This part of the report documents the stages in finding a suitable solution to the outlined problem. It will include all research into existing solutions and an understanding to the proposed solution and why the individual components were chosen.

***Design***

In this part of the report the user interface, software architecture, data definitions, algorithms and other high-level descriptions of the system are explained in detail to give an understanding of the system in full.

***Implementation, testing and evaluation***

This part of the report describes the steps taken to implement to proposed solution as well as all the extensive testing and evaluation of the software making sure all relevant requirements are met.

***Conclusions***

In this part of the report a summary of the work carried out during the project can be found along with what problems and challenges were encountered along the way and if any of the initial requirements were amended. At the end of this chapter discussions about further development can be found.

References and appendices can be found at the end of the report and can be used alongside this report to gain a further insight to the problem and the solution. The appendices contain all relevant figures and diagrams mentioned throughout the report and should be viewed when mentioned to grasp the context in which the report explains.

# Analysis

“NISRA is an Agency of the Department of Finance and Personnel. The Chief Executive is Dr Norman Caven. NISRA is the principal source of official statistics and social research on Northern Ireland. These statistics and research inform public policy and associated debate in the wider society.” Their vision is “To be a centre of excellence in all areas of our business, dedicated to continuous improvement in the quality of our work.” And their mission “is to provide a high quality and cost effective registration, statistics and research service.” (Northern Ireland Statistics and Research Agency, 2014)

The developer of this project carried out a year’s work experience within NISRA meaning they have first-hand experience of the current system within NISRA. This includes the current software that NISRA have and the current hardware that NISRA are using. This experience is very beneficial to the developer as not only is there communication between NISRA and the developer, the developer has access to test data that can be used to verify the completed solutions effectiveness. As the developer has knowledge of the current hardware specifications of the current computers, this can be taken into account when researching current solutions to the problem. With the developer having communication with NISRA it can also be found that the technical ability of the staff can be assessed and this can be taken into account when developing the software. Also as the developer can contact the staff, this means that in terms of the user interface the developer can request that a questionnaire be developed and completed by the staff as to what they would find easiest to use in terms of interface.

The problem is that NISRA require an inexpensive, easy-to-use application to validate addresses from datasets which they receive throughout the year containing a substantial amount of addresses from location all over Northern Ireland. Addresses are a major part of generating statistics produced by NISRA as the address is used to specify a location to all persons in the datasets. Many of NISRA’s statistics can be found on their website which the link can be found in the reference section of this report.

The addresses need to be validated to prove that the address linked to a specific person on record is indeed a correct address as said above, location can be a key attribute to many statistics and if the address cannot be validated then it cannot be used as part of any statistics. Most of the data that NISRA receive has been captured from forms completed by the general public who could mistakenly input the data incorrectly resulting in human error. This could be caused by a misspelt street, town or city name or a misheard house number or postcode. When this occurs, an address which is actually located in one part of a town could be placed in the complete opposite of the town and therefore become an invalid address.

During the developer’s time spent at NISRA it was learnt that some of the tables that NISRA receive have been created by administrative staff inputting the data into tables by reading them from forms. This process involves the staff taking a form that has been completed and entering all the information on the forms into the table. The amount of forms that the staff may view and enter the data for each day could reach up to 200 forms each and each form may contain up to 20 fields (Fig. 1) (see Fig. 2 and 3 for correct and incorrect versions respectively of the same address). When this is added to the amount of datasets that NISRA receive the chance of an address being incorrect when entered into the tables is high and this chance will rise if the numbers of forms or fields rise. Therefore that is why the addresses in the database need to be validated.

It has to be noted for many statistics that NISRA produce the address does not have to be 100% correct in terms of the street address, as long as the address can be validated for that specific postal area. The process in which NISRA validate their addresses is by appending a Unique Property Reference Number (UPRN) to the address records. When querying for statistics if a UPRN is found alongside an address it is deemed as a correct address, as a UPRN is uniquely specific to every letterbox, not just every building. This means that flats and apartments are able to be validated as well.

It is then proposed that NISRA require an application that is inexpensive and can accept any given dataset that NISRA receive and validate all of the contained addresses so that the correct addresses are identified and the incorrect addresses are made known. This application must be easy to use by any non-advanced user and also can be used on any of NISRA’s computers at any given time. The benefits of this application is that it will allow more than one NISRA staff to use this application at the one time which will eliminate the waiting time for NISRA staff and the current application. Expenses will be saved through training with the use of this application as currently only a set number of staff can use the current application, but with the development of the proposed solution, the new application will be easy-to-use so that any member of staff will be able to operate it, reducing training costs.

There is an existing standalone application named Experian Quick Address (QAS) Batch (Experian, 2014) which can run on any of the NISRA staff computers but can only be installed on one of the computers due the license only allowing one computer installation. This application can carry out the validation of addresses in the manner that NISRA require as well as using the required pre-cleansed dataset. The cost of this application is £2000 for a 12 month one computer licence; this includes the pre-cleansed POINTER dataset which would be used to validate all of the addresses, as well as other extras which are outlined in Figure 4 in an email received from Experian when requesting a quote for the price of the software that NISRA receive. Although the application is quite expensive on a yearly basis for only one user, the application does have some very interesting features such as boasting the ability to cleanse any given address dataset by updating address fields to the correct street names, counties etc. as well as moving data in the datasets into the correct fields (Fig 5). The application even boasts to even inform the users if a person in the record row has moved or is deceased. Some of these features require additional data which is available for purchase although one of the datasets, named “suppression” has a cost per record to carry out the suppression function which can be seen in the mentioned email. As well as a standalone application QAS Batch is also available as an Application Programming Interface (API) which can be embedded into a current system.

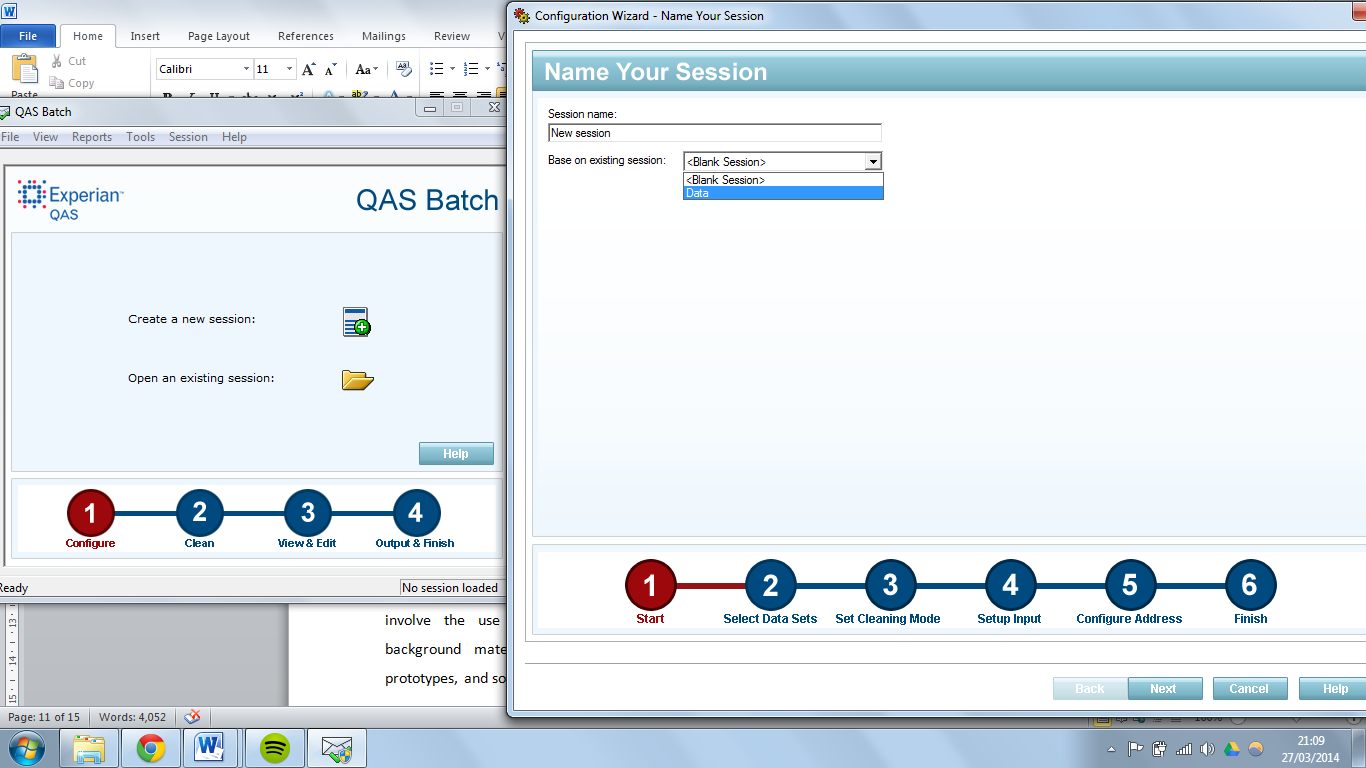
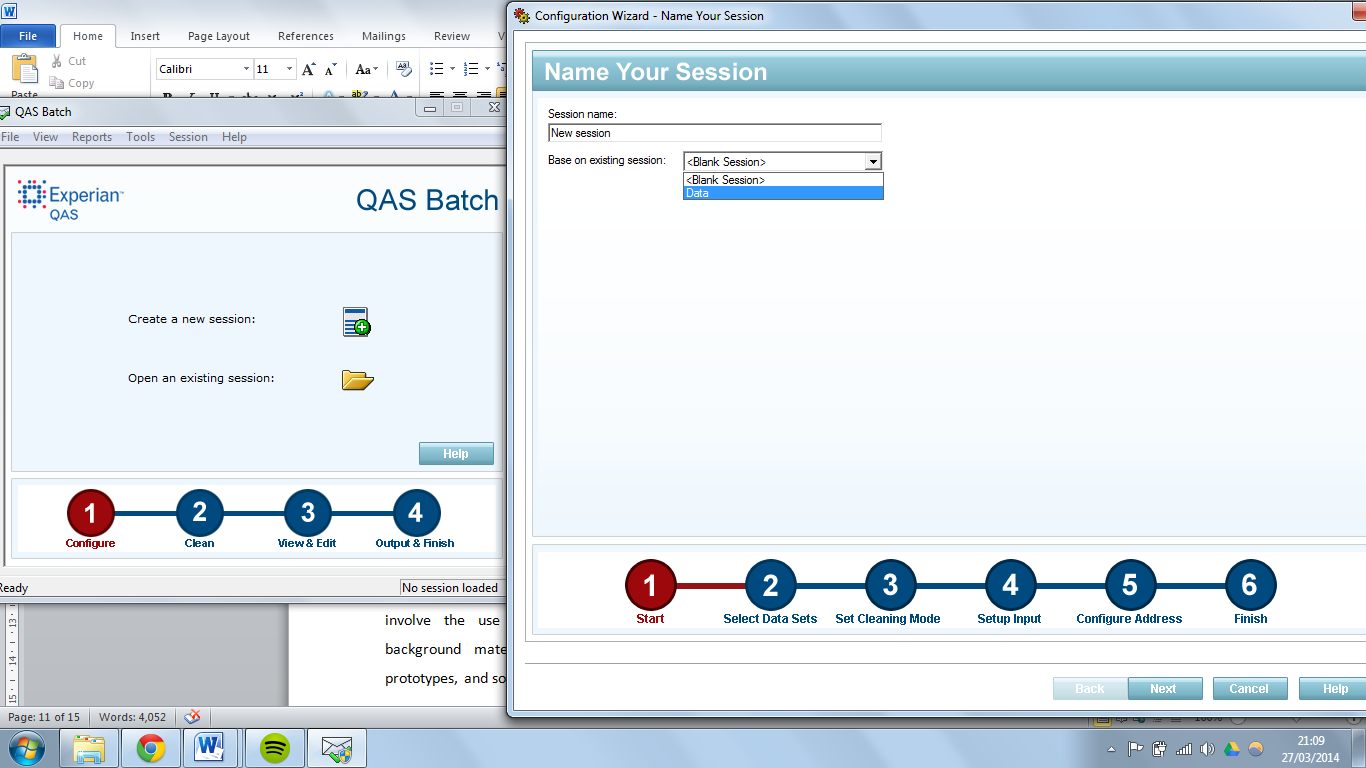
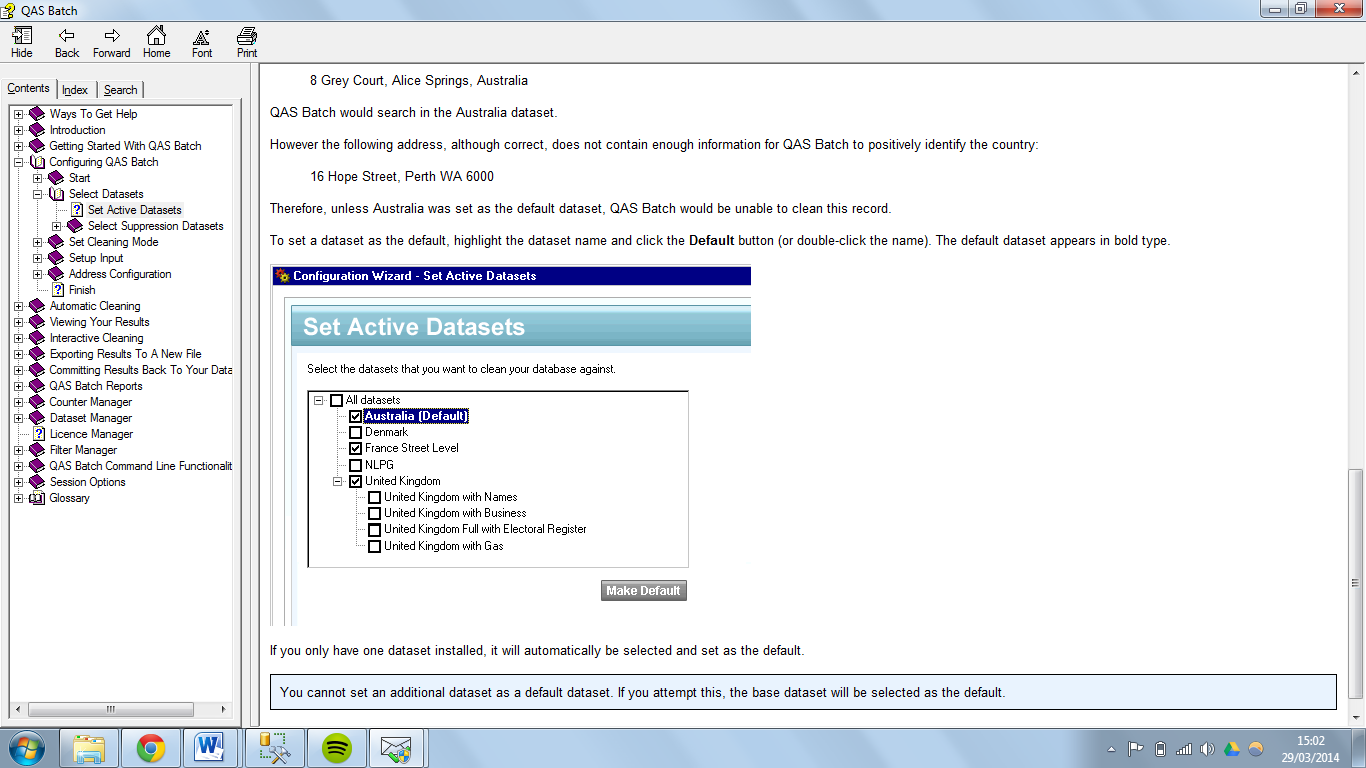
The process at which QAS carries out the validation can be seen below and the significant features that the application includes will be outlined.

Figure 8 when you create a new session.

We can identify some of the features that QAS Batch offers from the above screen shots. In figure 7 you can see that you have two options to choose from, ‘Create a new session’ or ‘Open an existing session’. The feature of opening an existing sessions is a very useful feature if you have different settings for different processes. This feature enables you to quickly load up the correct settings for the correct process saving time configuring each session. You will also notice in figure 8 that when you choose to create a new session you have to option to base the new session on a current session. This is helpful if you have to carry out a new process that needs to be highly configured but it is similar to that of a previous session you can simply select the session you wish to base the new session off. Although these features are useful, they are not useful to NISRA as they only have one process that has to be configured and it is a basic configuration therefore this feature is not needed. In both figures 7 and 8 it can be seen at the bottom of the screenshots that there are numbers to specify what task the user is on and how many tasks are left to be completed before the process is completed. This is a useful feature that any user will find useful as a user can see clearly what tasks have been completed, what the current stage of the process is on and what tasks are left to be completed before the process is complete. This feature is not specific to QAS Batch as many applications implement a feature like this one.

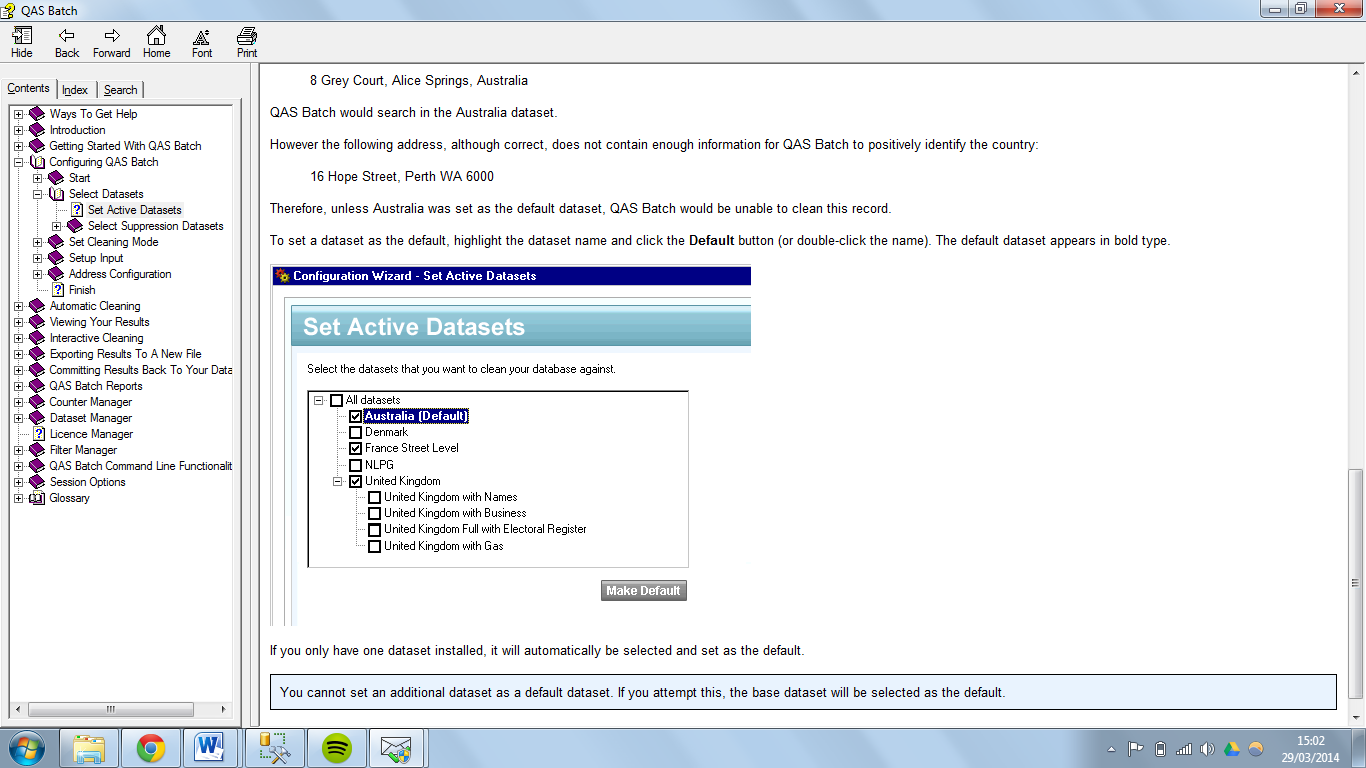
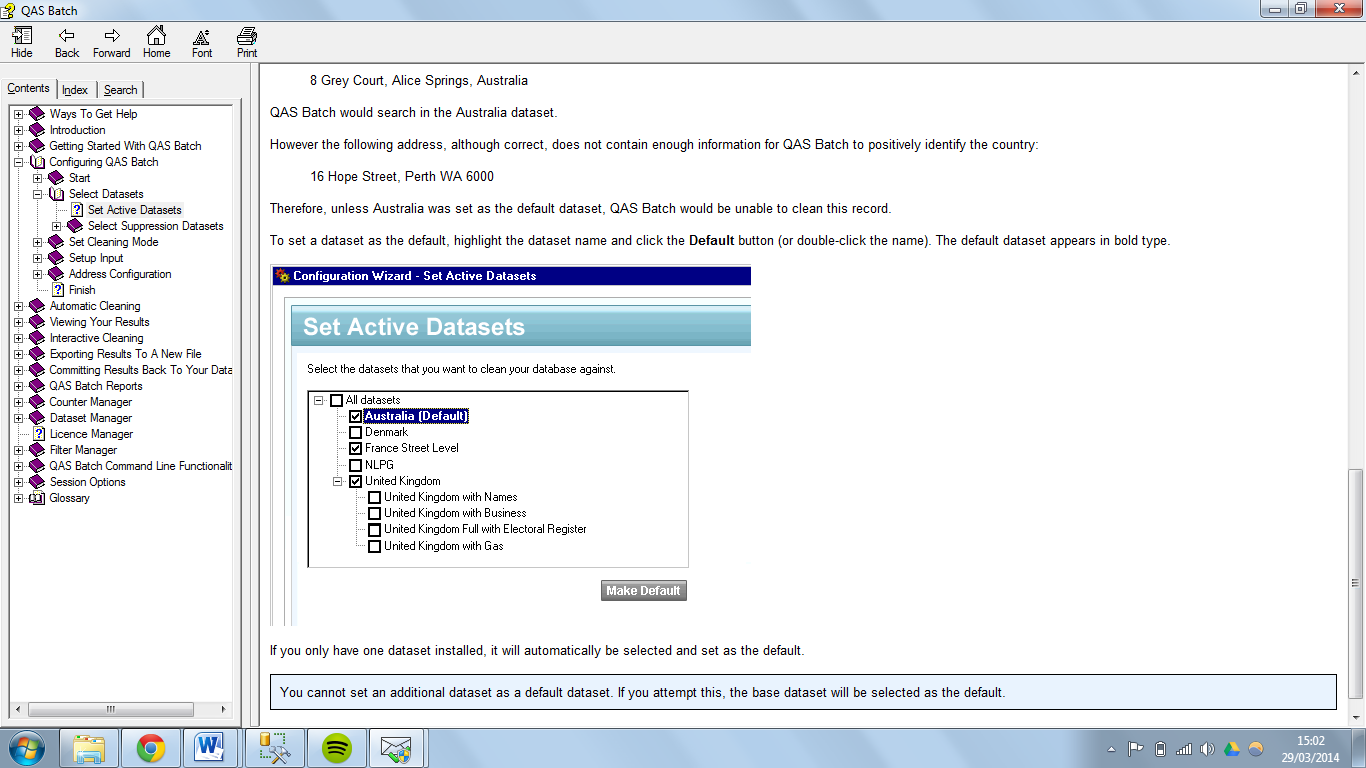
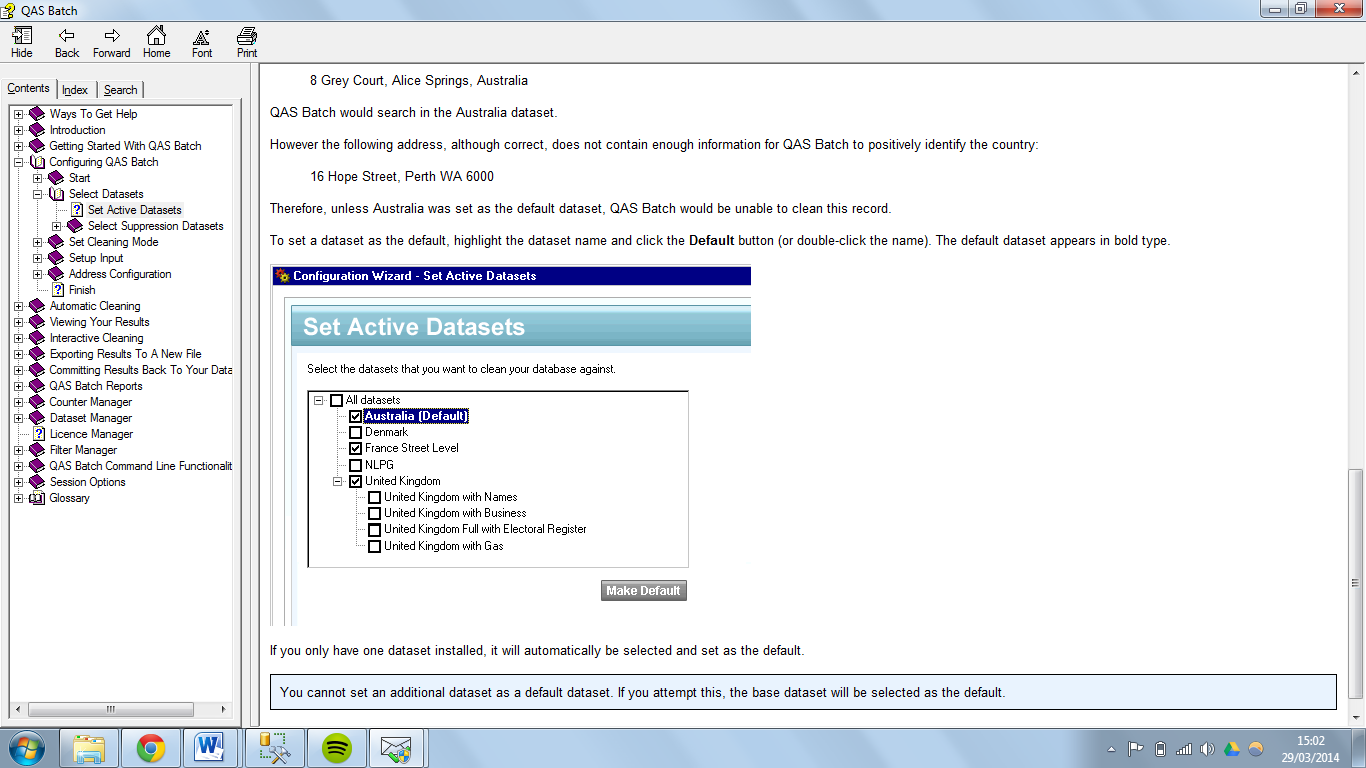
Figure 7 showing the opening screen of QAS Batch

Figure 9 Selecting the active datasets to use to cleanse the user dataset.



Australia

United Kingdom with Pointer (Default)



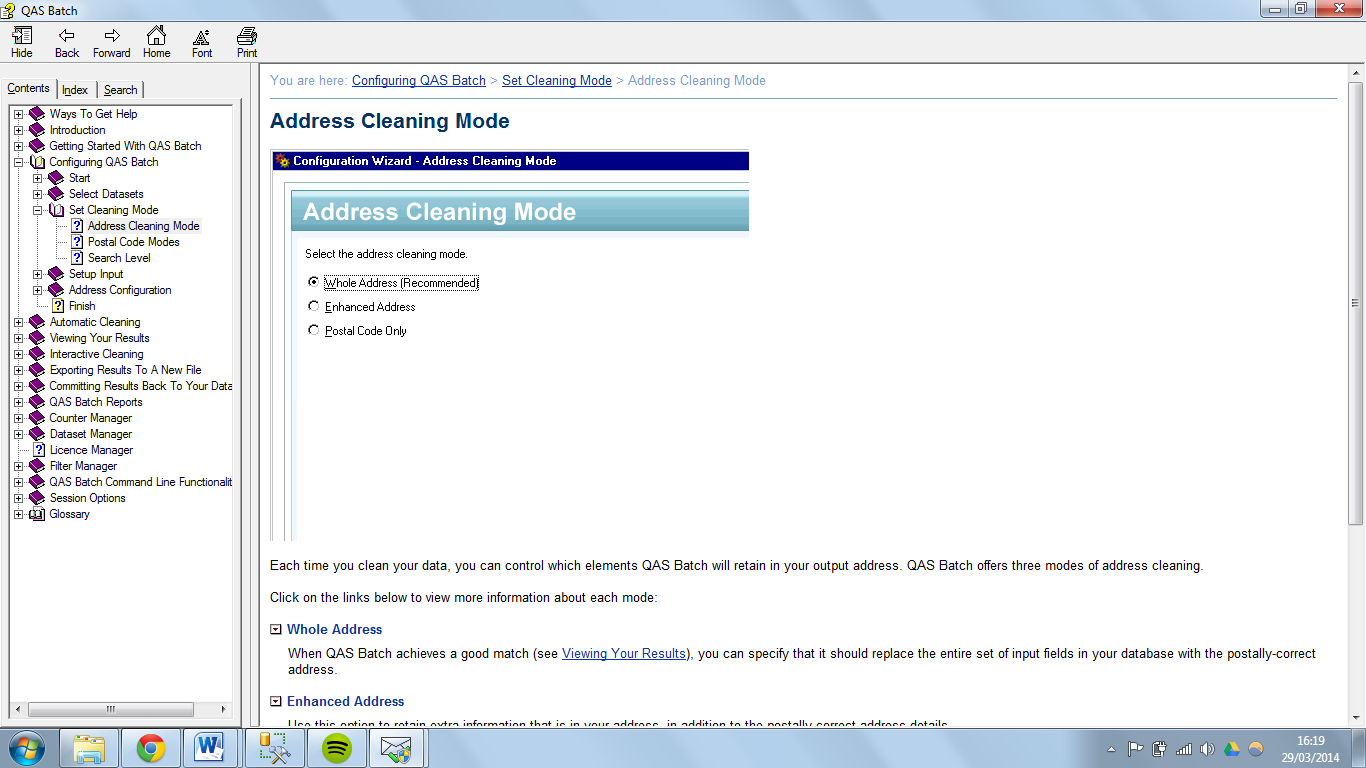
From figure 9 you can see you are given the option to choose what datasets you would like to use to clean the user dataset. Only the datasets that you have loaded onto your computer and that have the valid licenses will be shown. You can choose more than one dataset depending on the data you are cleansing but as NISRA will only be cleansing address data only ‘United Kingdom with Pointer’ needs to be checked. The ability to choose a default dataset is very efficient feature that if a user has several datasets that they use to cleanse their own datasets they can choose the most used dataset and that dataset will become the default dataset to use when carrying out the cleansing process. This feature is not a requirement for NISRA as they only use the one dataset to cleanse their data which is the Pointer dataset.

Figure 10 Choosing the address cleaning mode.

Every time you clean a user dataset, you can control which of the attributes QAS Batch will keep in the output address and which attributes QAS Batch will drop. As you can see from figure 10 there are 3 options;

‘Whole Address’ – When QAS Batch identifies a good match (explained later) you can specify that it should replace all the input fields in the user dataset with the postally-correct address in the chosen QAS Batch dataset.

‘Enhanced Address’ – This option retains extra information that is in the address from the user dataset, as well as the postally-correct address details. E.g. if a house was named “Home Farm” by the owner (and this information may be present in the user address data), the name of the house may not be part of the postally-correct address and would be lost if you selected the ‘Whole Address’ option. It has to be noted that enhanced address matching will not always retain all additional information. For example, when different partial matches are found for a certain address then any attributes that are not shared across the matches will be lost, even under ‘Enhanced Address’ matching.

‘Postal Code Only’ – with using this option you can specify QAS Batch to only update any incorrect postcodes in the user datasets.

For NISRA the only option that is ever used is the ‘Whole Address’ option. Although the ‘Enhanced Address’ option saves any extra information about the address, NISRA only require the postal address as this is what will be used to match against the Pointer dataset.

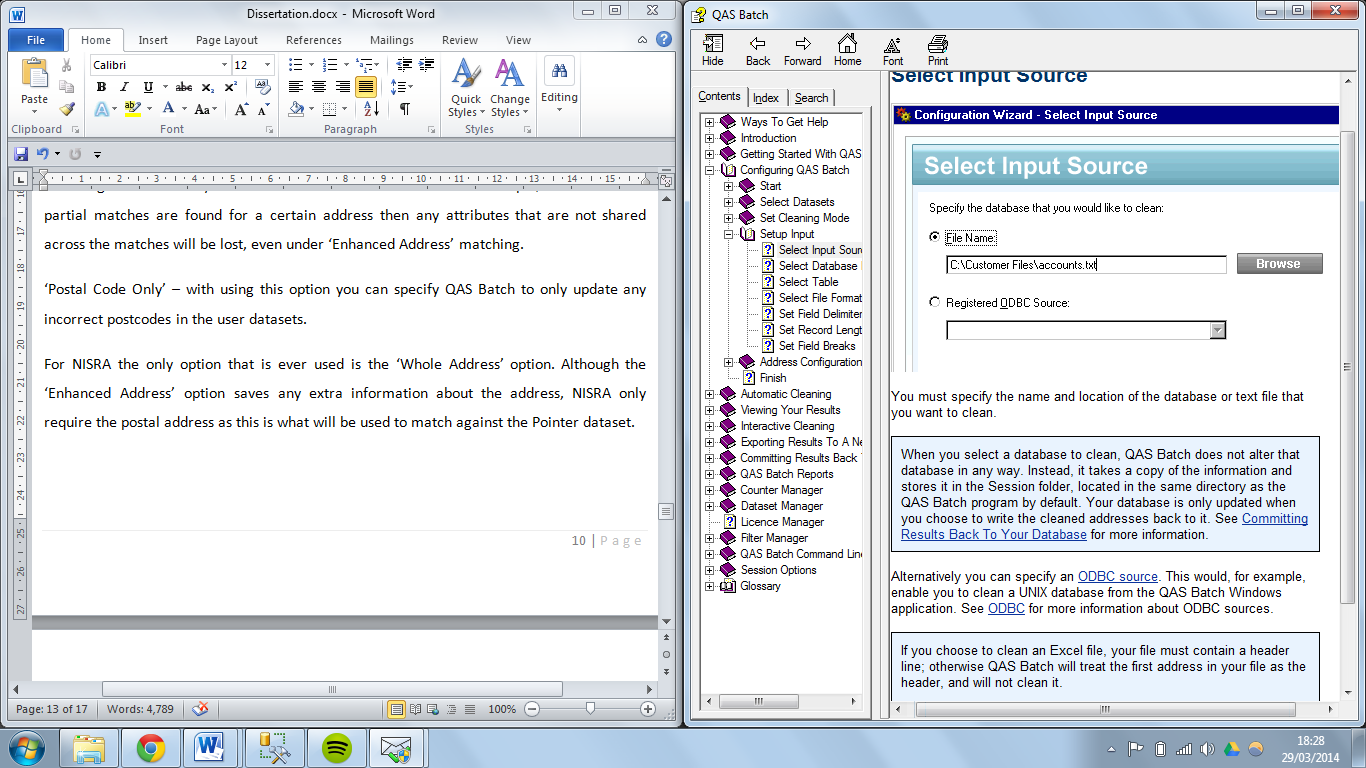
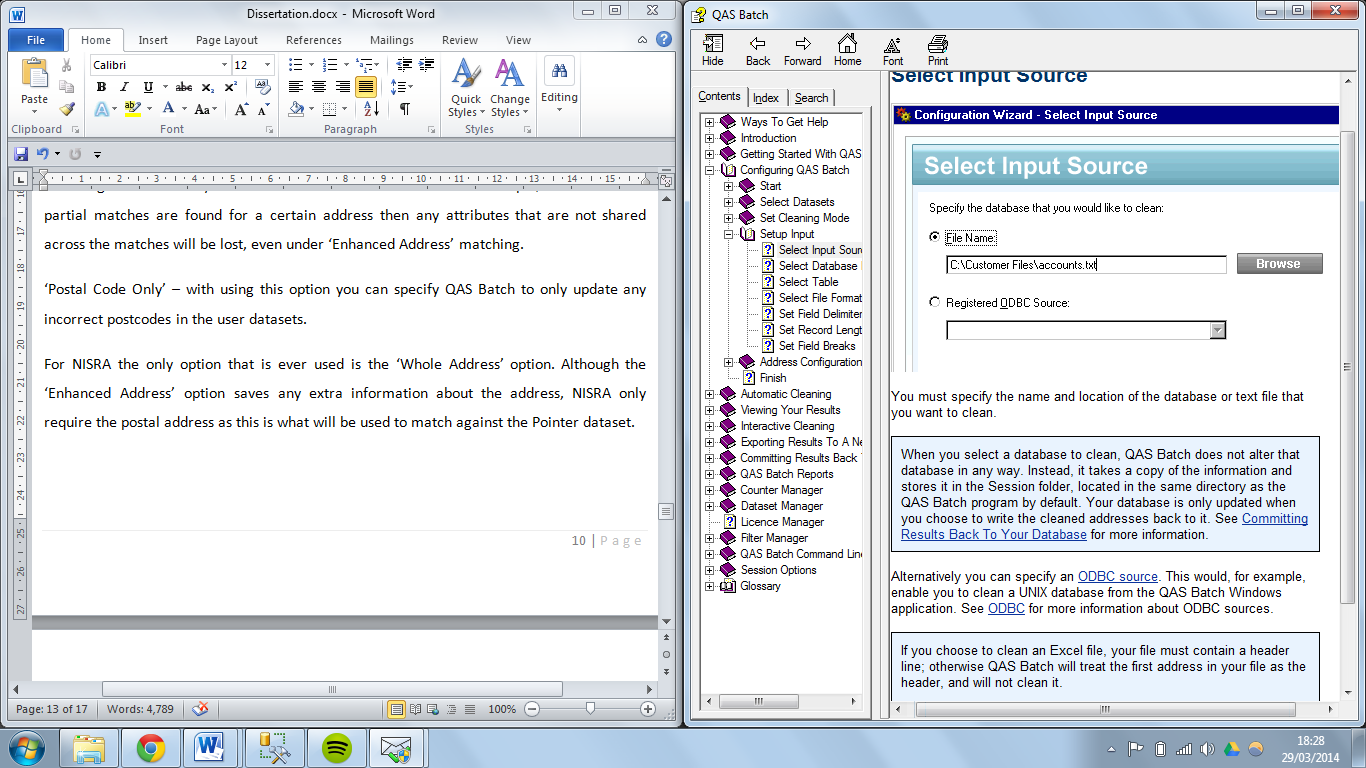


Figure 11 Specifying the location of the database to be cleaned.

When choosing an input source you have two choices as shown in figure 11, you can locate the file if it is stored locally on your computer by specifying the name and location of the file, later in the wizard you have the option of choosing what format the file is in. Alternatively you can choose the option to use an ODBC (Open DataBase Connectivity) source which enables you to link a table within the connected database to be cleaned. It has to be noted that the database you select to be cleaned, whether it is a file or within an ODBC connection, QAS Batch does not alter that database in any way. Instead it takes a copy of the database and stores it in a session folder. Your database is only updated when you choose to write the cleaned addresses back to it.

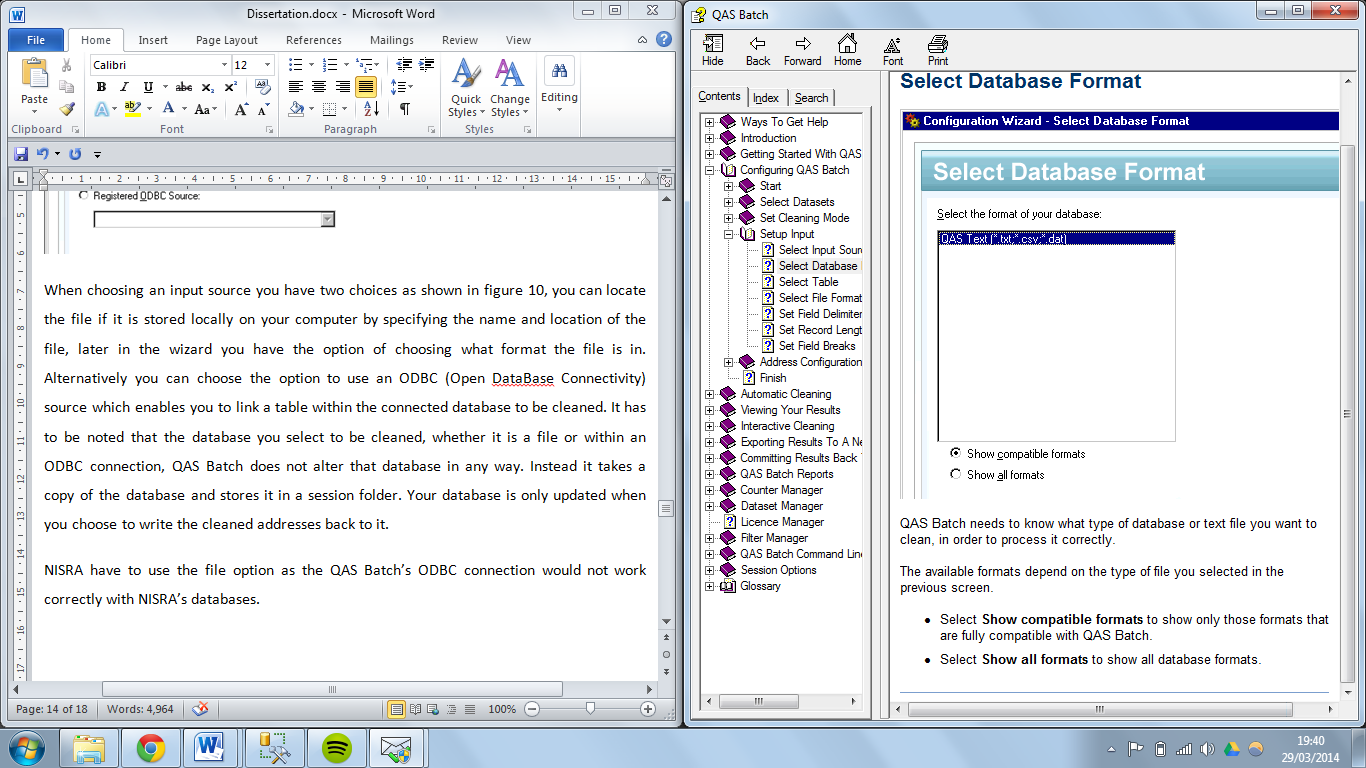
NISRA have to use the file option as the QAS Batch’s ODBC connection would not work correctly with NISRA’s databases.

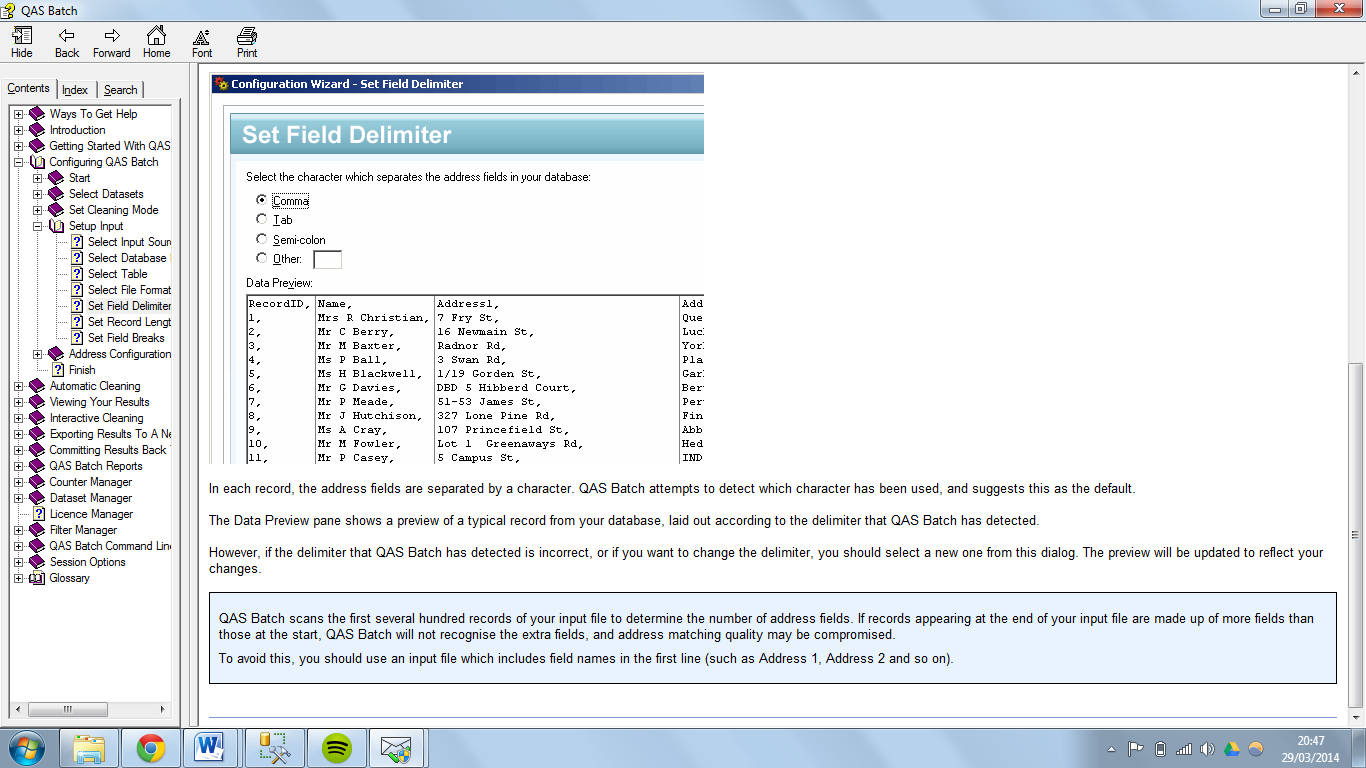
Figure 12 Selecting the format of the database file.

As mentioned above you have to choose the format of the file that you have selected (figure 12). The options that are displayed are relevant to the file that you upload. QAS will try to determine the file type before this screen is shown.

If the database file that you chose contains more than one table you will have to specify which table that you want to cleanse. QAS Batch will display a preview of the tables attributes and a preview of the content in each attribute in the database to allow for easy identification of the table you wish to select.

Figure 13 configuring the format of the text file.

(Note that the data shown is dummy data and not of any connection with NISRA)

When presented with the screen shown in figure 13 you are able to choose the format of the table from your file you selected is in. The options are ‘Delimited’ which means the file has fields that are separate by a specific character which, if chosen, will be declared later in the process. The other option is ‘Fixed width’, this option means the file has fields that are the same throughout the file, and in other words the information is aligned in columns. Notice that the check box allows you to specify if the first row of the file contains field names. Also the data preview box allows you to see how the above choices will affect your data.

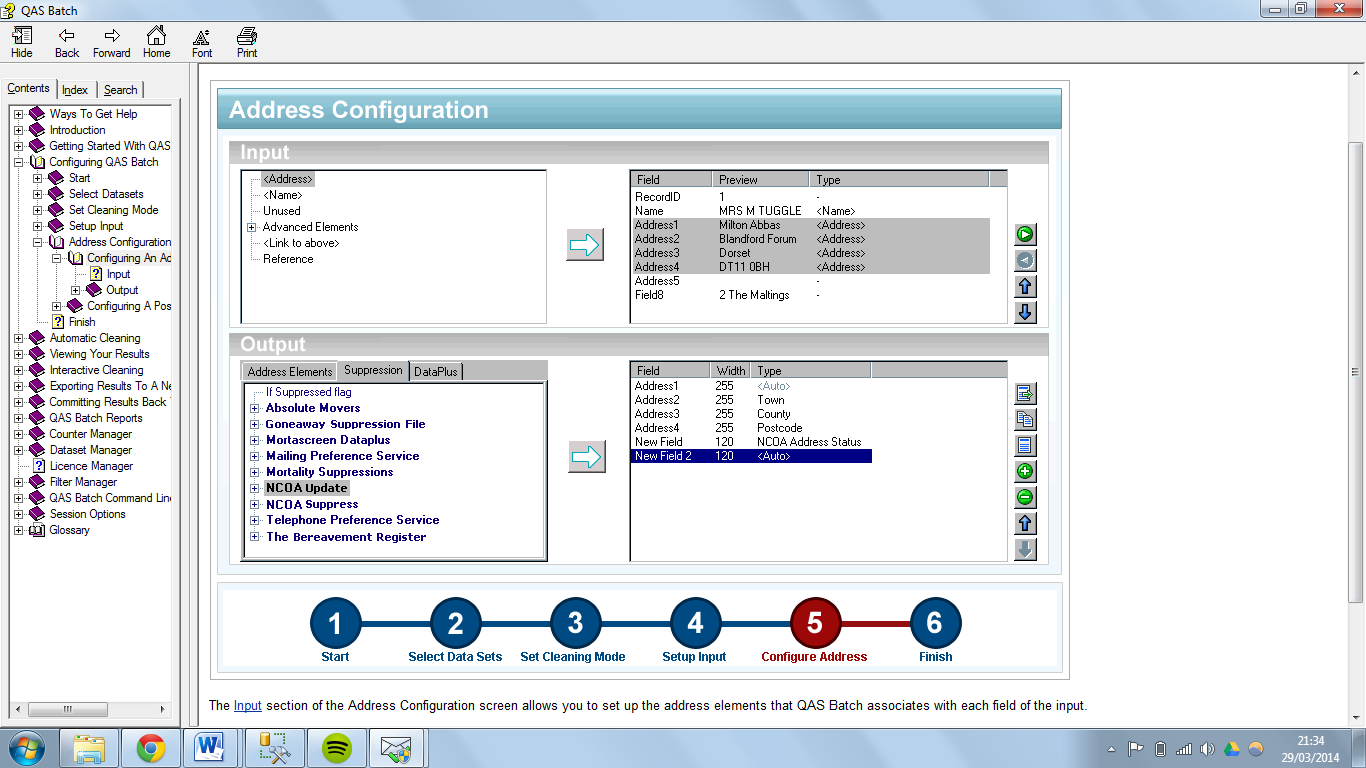
This step of the configuration, shown in figure 14, allows you to specify the character that will separate the fields in the table of the database. As you can see there are a few default options that would be most used and the ability to define your own character.

Figure 15 configuring the input and output address elements.

(Note not actual data. Only for explanation purposes.)

Figure 14 selecting the character that separates the fields in the database.

This part of the process, as shown in figure 15, is a very important part of the process as the more accurate you are with configuring both the input and output elements the more accurate the cleansing process will be. On the left hand side of the input configuration you have a list of all the elements available to select such as the basics of ‘<Name>’ and ‘<Address>’. There are also some more specific selections such as ‘Unused’ which means QAS Batch will ignore this element as it contains irrelevant data, ‘Reference’ is used when elements contain data that you do not want QAS Batch to search on but wish this data to be visible both during and after the cleaning process. Within the ‘Advanced Elements’ node in the tree you will find specific address types such as a Street Line or PO Box, these elements will make cleansing more accurate when matching.

In the output section of figure 15 you will see the same layout, on the left hand side you will find a list of all the elements that can be selected for output. Normally you will select the elements that were configured in the input section but you can add more elements that were not in the input section. E.g. if in the input section you configured the elements ‘<Address>, <Street> and <Postcode>’ but for statistics you need the county of address you could add the ‘<County>’ element. This would add this information into the output of the address even though you did not have this in the input data. It does this by using the information in the chosen cleaning datasets.

This feature is a very smart feature if used correctly and when used correctly can save the user a substantial amount of cleaning time. Also the way in which the input elements are configure impacts how well the data will be cleaned, so the better the configuration the better the cleaning. The use of this feature requires extensive knowledge of the data that you are cleaning and the attributes that the data includes. The way in which the data to be input is presented has a massive factor in the configuration as if an address in one row has county in address line 3 whereas other rows may have county in address field 4. This can cause difficulties when configuring the inputs as you may not be able to select county for both inputs. As NISRA receive several datasets which are all formatted differently this process does take some time to configure each time it has to be carried out.

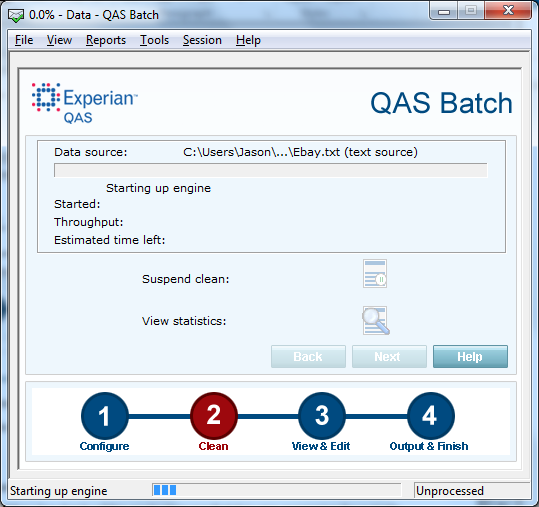
As you can see from figure 16 the progress screen is very basic but has all the essential information required such as the time started and the time that is remaining. The progress bar enables the user to see how much of the cleaning process is remaining and you will also notice the percentage of the work completed at the top of the window. All of these features are essential for a process such as this as it does take some time to complete therefore when the user has an estimate of how long is remaining they can rest at ease and plan around this.

Figure 16 progress screen after the configuration is complete the cleaning has started.

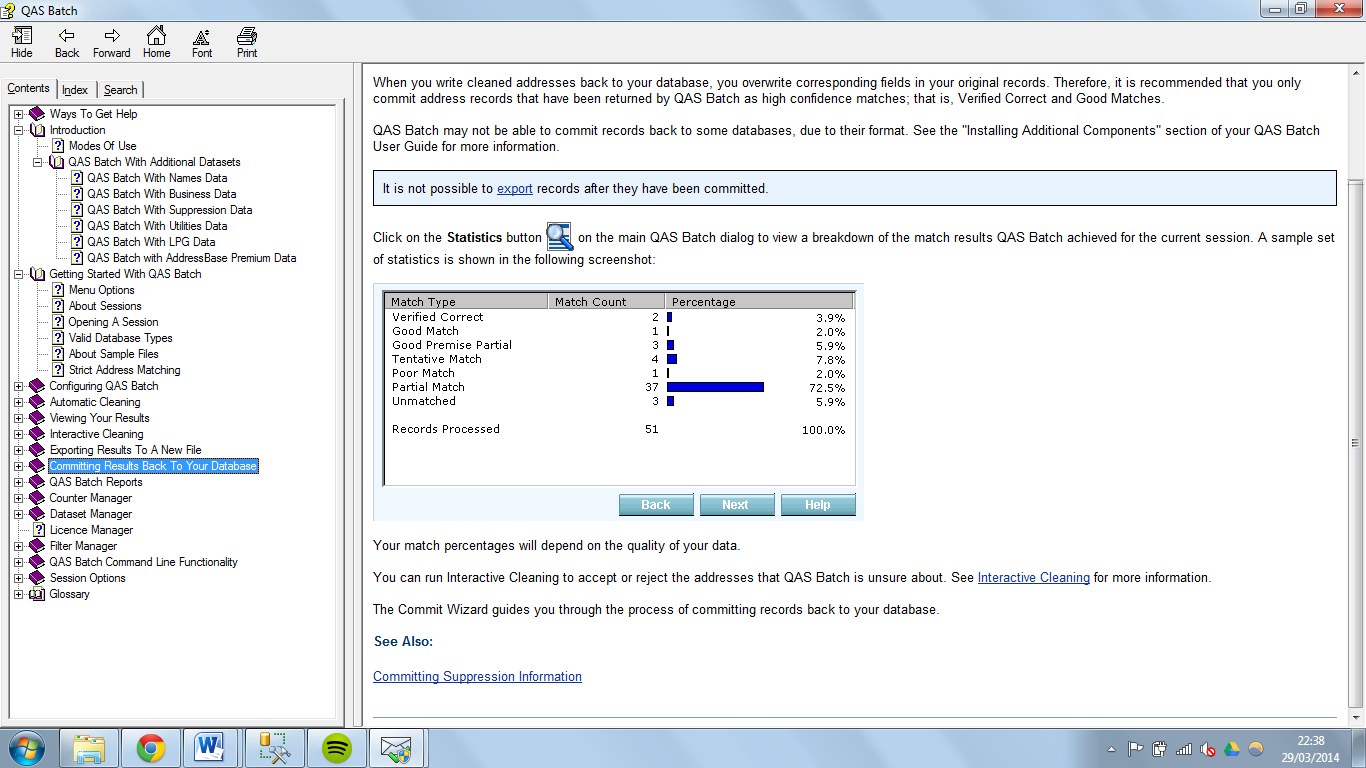


Figure 17 viewing the statistics of the cleaning process

This screen shown in figure 17 is the statistics screen for the cleaning process, and this can be seen both during the process and after the process. The screen shown is after the matching process and you can see the detail in which the cleaning process achieved. These results reassure the user of the data that they may be re-importing into their database that it has in fact been cleaned and how much of the data has been cleaned. You will notice that only 3.9% of the records were verified as correct, this means that all of those records were correct before the cleaning process started and that they did not have to be changed during the process. All of the other records were either changes or no match was found. This feature is essential for the user to view as they can be assured of what process has been carried out on their data. Remember that the data that was cleaned is a copy of the original data meaning that if the user is not satisfied they can go back and change some of the elements. The output statistics may change depending on what elements are selected during the input configuration as this will have an impact on cleaning.

As NISRA’s datasets which they receive vary in format and layout this process can take some time to configure for each dataset which is very time consuming for staff. Configuring the input elements can cause problems with the output statistic as the data can keep continually changes every set they receive.

When the user is satisfied with the output statistics they can continue to either store the cleaned data as a text file to be imported to the users database at a later point in time or the user can directly output the cleaned data via ODBC directly to the table that it originated from. This process is the reversed process of selecting the input file.

All in all this application does carry out the process that NISRA needs in regards to appending the UPRN to the datasets and can also carry out data cleaning if NISRA were to require this at a later time. Unfortunately as mentioned above, the pricing in regards to the purchasing of the application and the licencing of multiple users as well as the training of the staff to use the application is extensive. As you can see from the walkthrough of the process there are a lot of features in the application and training would definitely be required to some if not all of the staff so that they could efficiently continue to use this software. It has to be noted that currently there is only one person in NISRA that can use the current version of the software but they have not been trained in doing so therefore the efficiency of this is affected. Also if the version of the software changes more training may be required which again will cost money.

**Feedback from Questionnaires**

NISRA staff was presented with questionnaires (found in the appendices section of this report) in regards to the user interface of the new application and the idea of this questionnaire was to outline what the staff would find simplest to navigate. It was found that the majority of the staff wanted a simplistic design that allowed them to press a few buttons and the process would be complete. When asked about current user interfaces that they might use within work or even at home it was clear that the staff wanted a well instructed interface so that if new staff were to be employed they would be able to easily operate the application due to the on screen instructions. A number of people made reference to the easiness of an application installation and how this was a well instructed process. When asked to list applications with user interface that they preferred the most common answer among the employees was Microsoft Access. This would be because many of NISRA’s applications involve the use of this application as a front end to NISRA’s databases. Microsoft Access will be considered as a user interface in the design section of this report.

**The Solution**

The developer plans to design a “Data Cleansing” application that will allow the user to select a ‘dirty’ database with addresses to be cleaned. This application will use the existing address fields in the dataset to match those addresses to the addresses in a pre-cleaned dataset, when a match is found; the UPRN of the matched address from the pre-cleaned dataset will be appended to the ‘dirty’ matching address.

The matching process for the existing application is not available to view for licensing reasons so therefore it is proposed to create a new matching system which will require the use of what the developer is calling ‘MatchKeys’. These matchkeys will be created using an ‘Unique Postcode Identifier’ (UPCID) which will be made up of different parts of the address (this will be discussed and explained in the design section of the report). These UPCID’s will be created for all records in both the user dataset and the pre-cleansed dataset and it will be on these fields that the matching process will be carried out.

The user interface of the application will be simplistic and easy to use for the staff at NISRA. The user interface must be developed in a way that if NISRA employ new staff they are able to operate the application without training.

The application will be based off the current application that NISRA are using and where possible features will be ported.

The data handling application that will be used will be Microsoft SQL Server as this is where the current processing of the addresses is carried out and many of the front end software developments are able to communicate with this. As NISRA contain millions of records it would not be feasible, both for time and money to change the data store.

**Requirements**

Through researching the current problem and through feedback from questionnaire as well as time spent at NISRA the developer and NISRA came up with a list of functional and non-functional requirements. Diagrams were used to help understand the requirements.

* A user should be prompted to log into the SQL server database with their SQL user name and password when the application opens. If the user does not enter both of these or if they are incorrect the application should prompt the user again.

User is prompted to enter SQL username and password

SQL server: Validate Username and password

Incorrect

Correct

Application opens

`

* A user should be able to select a table name in the SQL database from an automatically populated list.
* A user should be able to click a minimum of one button which will carry out the cleansing process with minimum user intervention
* Throughout the entire cleansing process the user should be informed if a process has already been completed on a chosen table.
* A user should be able to view a report for the matching process, outlining how many UPRNs were added to the table and what percentage of the records that covers.
* A user should be shown a form after the match process is complete displaying all the records from the table used in the match process that were assigned a UPRN along with the records in the pre-cleaned table that have the same UPRN.

User selects table from dropdown list

User clicks button to carry out match process

SQL Server: matches UPCIDs and adds UPRNs

Error message displayed

Failed

Success message displayed and comparison form opens

Completed

* The application should have an easy to use graphical user interface with clear navigation options that will allow a user of any level to use.
* A user should be able to use this application without any prior knowledge to SQL server.
* If more than one button is required for a certain process then these buttons should be found together as to not confuse the user.

# Design

**Software Chosen**

In this section of the report the software that was chosen will be explained.

*Backend Database*

Microsoft SQL Server will be used to handle both the storing of data and also most of the processing of the data. The reason this software was chosen was because NISRA already are using this software for many of their other data processing tasks, this means that no extra expense will occur for this part of the project. The developer found that MS SQL Server can communicate with an extensive list of front end software applications which made the choice of this application easier as it does not limit the software that can be chosen for the front end system. The developer has previous knowledge of this application and therefore will know how to optimise functionality within the project, optimising this project will be essential as it will be very process heavy due to the amount of records being processed.

Other database applications were considered for this project such as MySQL (Oracle, 2014) but were not decided on in the end. This application is another brilliant database system that is very popular and one of the most used database applications available but the reason the developer did not use this application was it would not be efficient for the users of the application. NISRA receive datasets in text file format which could be simply read into this application, the data would have to be exported to a text file when the cleaning was complete and then it would have to be imported into MS SQL Server as that is where all NISRA’s data and statistic processing is carried out. Therefore it was proposed that due to efficiency MySQL would not be used due to this and not due to overall rating. When NISRA receive the datasets they can simply import the data into MS SQL Server and all the processing, both for data cleansing and for statistics can be carried out here. Another advantage of MS SQL Server is that NISRA perform backups of all data stored in MS SQL Server so any cleansed data will get backup, therefore if MySQL was chosen, a separate backup system would have to be created to be compliant with NISRA’s data act.

*Frontend Application*

When choosing the application for the frontend many options were considered such as; Java, C++, Microsoft Access, C# and also the possibility of a frontend web application using PHP and XML. All of the above mentioned were researched fully with consideration to the development options available in regards to a user interface as well as the integration of MS SQL Server. All of these methods had the ability to connect and communicate with MS SQL Server either via API or via ODBC, although when researched many of the methods had several disadvantages. Some of these disadvantages included the lack of skill and knowledge that the developer had in regards to the mentioned languages. The amount of time and effort that the developer would have to spend learning the language would not be feasible if there is already an application that the developer can use that they already know how to operate. For developing in the languages mentioned above the developer would have been able to create a user interface using the languages but a substantial amount of time would have been wasted in researching and developing both the MS SQL Server connection as well as the processes to be carried out.

It was then proposed to develop the front end application within MS Access as not only does the developer have great experience in using this software but certain NISRA staff also have knowledge of this software. This is a great advantage these members of staff will be able to troubleshoot problems should problems arise or also be able to identify what the problem is and pass that information on to the developer to be fixed. This application was also chosen as not only can you connect to a back end database to carry out processes, you can also carry out processes within this application itself, this will be useful when outputting information back to the user in terms of estimated time of completion etc. By using this application you can further develop the front end of the project without having to compile code meaning that small changes to the UI can be made without large down times. The ability to edit these features can be set so that it is only available to certain users with certain rights on the system. This also leads to advanced security within this application, when using MS Access the developer can create the application so that users have to log in with their correct information before any user can start any of the processes, this means that any user that should not have access to the system cannot log in and view data they are not cleared to view.

*Software Version Control*

From the developers previous application development experience it was decided that version control should be implemented. As NISRA take backups of their data system frequently the backend database does not need to be backed up and therefore the only version control that will be needed will be for the frontend application. Extensive research has been carried out to find the best software version control for the frontend application. As it has been decided that MS Access will be used to develop the frontend user interface it has been discovered that there are no official applications that suit the need of version control with MS Access. Many of the suitable applications found cost money to implement and do not provide trials as they are third party applications therefore they could not be tested to see if they work with MS Access. The developer was able to acquire open source code that is available to freely use which requires the user to run a single line of code (ExportAllSource) from inside the Visual Basic with Applications (VBA) window, and that code will export all the code from inside the MS Access application, which includes Form Code, Macros and queries as well as tables, to a specified location on the current computer. The developer then implemented the use of the software Git to upload and store any increments of the software on a web server. Git allows users, with the installed framework, to run command line commands to upload files from a specified folder to a web server with 3 commands, ‘git Add -A’, ‘git commit’ and ‘git push’.

* ‘git Add -A’ – there are several instances of ‘git Add’ that can be used depending on what files you wish to upload. I.e. if you have a project that is large in size and only want to upload several files that you have changed then you could specify this. As any time this project will be modified all the code will have changed this is the code needed to upload the new software.
* ‘git commit’ – this command simply prepares the files that have been added for the upload to the server. From this command you have to write a heading for the upload, this is to identify one upload from another. You also have the choice to add more text to describe the files you are uploading if they need more explanation.
* ‘git push’ – this command will start uploading the files to the git depository on the web server and notify the uploader when this is complete.

It has to be noted that each upload does not overwrite previous uploads, rather new uploads are stored alongside previous uploads. This is very useful in case a modification is made and uploaded without discovering that there has been a problem, the user can then simply roll back a commit so that a previous upload is now the main version.

This process will be very useful for the future development of this project as the developer has access to all the code of the project from anywhere they wish and can be downloaded onto any computer, it will also allow the developer to upload new versions of the software so as NISRA can download them and simply import the new code.

It has been proposed to upload code of any changes made after that change has been made but one a two week basis an upload of the whole project will be made and this will be a major version release.

**Pre-cleaned dataset**

It has already been decided that the pre-cleansed dataset to be used will be the Pointer dataset from LPS (Land & Property Services, n.d.).

‘Pointer is the address database for Northern Ireland and is maintained by Land & Property Services (LPS), with input from Local Councils and Royal Mail (RM). This is now the common standard address for every property in Northern Ireland. It is important to note that Pointer is a dataset for addressable buildings in Northern Ireland. Each building has a UNIQUE\_BUILDING\_ID, which uniquely identifies a Primary Addressable Object (PAO). A PAO is defined as the ‘physical footprint’, i.e. the building shell. Each property has a Unique Property Reference Number (UPRN). The UPRN represents the Secondary Addressable Object (SAO) e.g. a residence or business within a building.

Pointer has been allocated a set of UPRNs from the national hub, which are allocated to all addresses within the dataset. This will ensure consistency of UPRNs across Northern Ireland and Great Britain.’ (Ordnance Survey of Northern Ireland, n.d.)

The reasons for choosing this dataset are;

* It is the leading address dataset for Northern Ireland,
* It has full Northern Ireland coverage,
* It is maintained by Local Authority staff,
* It is the only NI spatial address database,
* It contains information on multiple occupancy, building use and organisation type, and
* It has full incorporation of Townland names

This dataset, as mentioned above is sold with QAS Batch as an additional dataset but you can also buy this separately, although NISRA are contracted to receive this dataset already therefore it is efficient for the developer to use this dataset. It is updated frequently and NISRA can acquire a copy of this dataset on a regular basis without having to wait for this to be released.

**Matching methodology**

This part of the project will take the longest to develop as it is the main body of all processes in the project. As stated above in the report the existing application carries out this process and this is the process that needs to be replicated but the method QAS Batch use cannot be seen due to licensing. The idea behind the developer’s methodology relates back to the problem of the dirty addresses being recorded as shown below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| User Address Data | | | | | | |
| Address1 | Address2 | Address3 | Address4 | County | Postcode |  |
| 65 | Castle Street |  |  | Armahg | BT12 3AB |  |
| Pointer Data | | | | | | |
| Address1 | Address2 | Address3 | Address4 | County | Postcode | UPRN |
| 65 | Castle Road |  | Tandragee | Armagh | BT12 3AB | 12345 |

The above example data shows how simple mistakes can give an incorrect address. It can be seen that the ‘Address2’ field in both datasets are different, this could be down to the fact that a person misheard the information when they were keying in the data. It can also be noticed that in the user data the ‘County’ field has a spelling mistake in the word, this again could be down to careless typing, none the less the address has been made incorrect. The numbering of the address suggest that there are a substantial amount of houses in that street therefore the postcodes could not be identical so again the address is incorrect.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User Address Data | | | | | | | | | | | | | |
| Address1 | Address2 | Address3 | Address4 | County | Postcode | UPCID\_N\_WP | | | UPCID\_N\_C\_NP | | … |  | |
| 65 | Castle Street |  |  | Armahg | BT12 3AB | 65BT123AB | | | 65ArmahgB123 | |  |  | |
| Pointer Data | | | | | | | | | | | | | |
| Address1 | Address2 | Address3 | Address4 | County | Postcode | | UPCID\_N\_WP | UPCID\_N\_C\_NP | | … | | | UPRN |
| 65 | Castle Road |  | Tandragee | Armagh | BT12 3AB | | 65BT123AB | 65Armagh123 | |  | | | 12345 |

A method of creating matchkeys has been created by the developer to be able to tackle these misspelt and misinterpreted addresses. The basis of these matchkeys are down to different combinations of the addresses split up and concatenated together to form a unique postcode identification (upcid). These upcids would be created for all records in both the user dataset and the Pointer dataset via an automated process in the backend of the application. An example of a upcid can be seen below using the same sample data from above.

UPCID\_N\_WP = UPCID made up from Building Number (N) and the Whole Postcode (WP)

UPCID\_N\_C\_NP = UPCID made up from Building Number (N), the County (C) and the Number from the Postcode (NP)

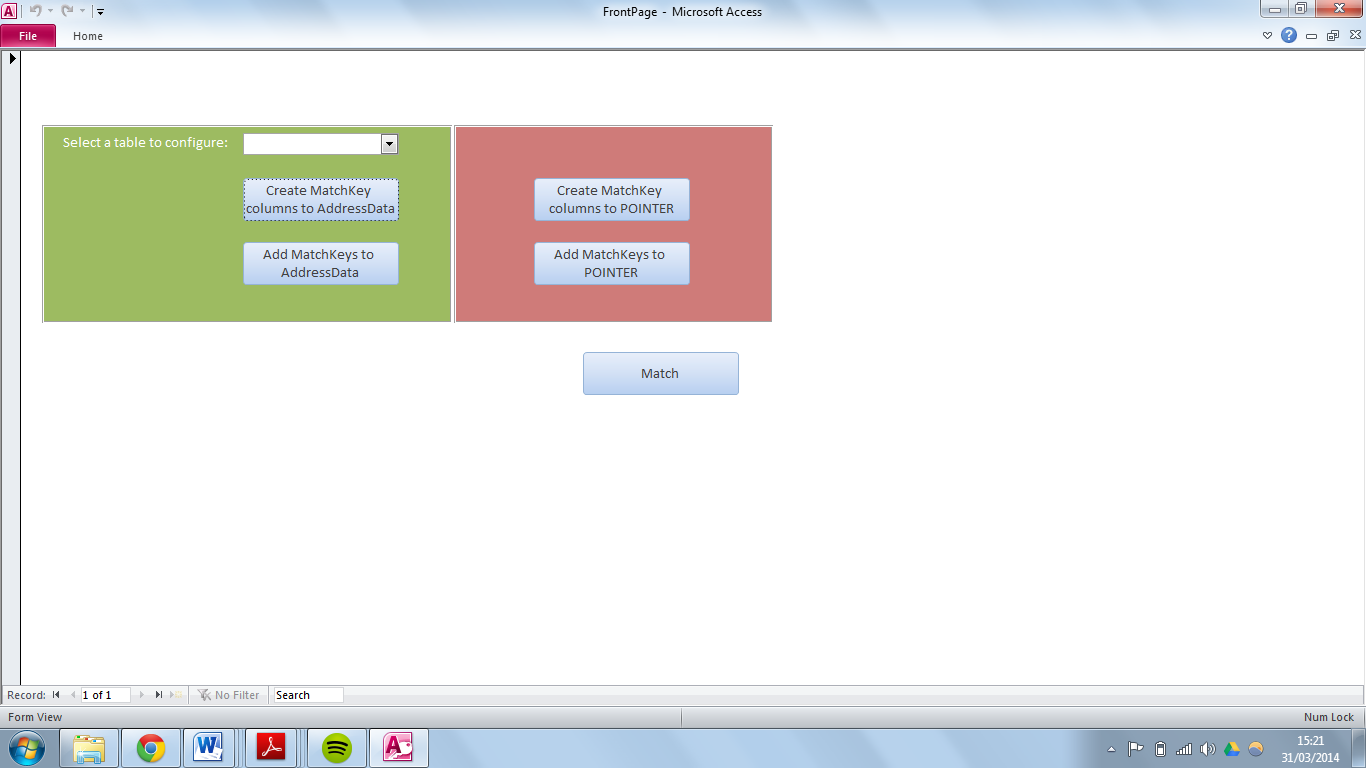
**Initial Prototype**

Figure 18 showing the initial prototype of the project

Design covers the user interface, software architecture, data definitions, algorithms and other high-level descriptions of the system you have created. Ideally, a good system design document is one that can be passed to someone else to implement.

It is expected that during the design phase various options will have been considered before any final decision was taken. These options should be identified and the rationale for each decision presented.

You are encouraged to use descriptions and models suitable for your own circumstances. For example,

* In describing how the user will interact with your system you may want to present a block diagram identifying key parts of the user interface in addition to showing screen shots. If you considered HCI guidelines, you should explain how these influenced your design
* If you are using web pages, you may want to include a map of your site, indicating how users navigate through it
* If you are using a database, you will need to describe the design schema, including details of any normalisation involved. You may also want to include an entity-relationship diagram
* For a Java program, you will need to start with the class design, identifying your main classes and indicating their purpose
* For some processing it may be necessary to make use of complex algorithms, which should be described and illustrated appropriately

Length: ~14 pages

# Implementation, Testing and Evaluation

The content of the implementation, testing and evaluation chapter is largely self-explanatory.

First, under implementation, you should describe the approach you took to development, indicating what was learned from the prototype and successive versions produced, identifying any other incremental steps followed to achieve the final system.

Secondly, you should describe any specific tools used to support the development process, including details of any new languages that you needed to learn.

Also under implementation, you should include details of any significant aspects of the code you have produced, including the use of specific algorithms. If you have used code from other sources, this should also be identified, indicating how it was integrated into your system.

Your approach to testing should then be described, identifying the test cases that you have used to verify the correctness of the software. These should demonstrate that your testing has been appropriate and thorough.

Under evaluation, you should assess the perceived value of your system to its intended users against the specified requirements. You are not expected to create a ‘perfect’ system so marks are awarded more for the thoroughness of your validation than any praise obtained. Indeed criticism gives you an opportunity to explain the issue raised and suggest improvement.

Note that when planning your project, sufficient time should be included after implementation for testing and evaluation as these activities have a significant impact on the final quality of your system and the write up in your dissertation.

Length: ~16 pages

# Conclusions

In the final chapter you should summarise your project work overall and assess it critically. This should indicate what lessons you have learned and so clarify what you might do differently if faced with the same situation again. In particular, you should identify and discuss how the project plan evolved as the project progressed.

The limited time available for implementation means that you are likely to have ideas for further work. These should also be included in the conclusions.

The Conclusions chapter, like the Introduction, should be freestanding, allowing the reader to understand what the project has achieved without studying other chapters.

Length: ~4 pages

# References

Experian, 2014. *Data Cleansing Solutions.* [Online]   
Available at: http://www.qas.co.uk/solutions/data-quality-software/data-cleansing/batch.htm  
[Accessed 26 03 2014].

Microsoft, 2008. *SQL Server.* [Online]   
Available at: http://www.microsoft.com/en-us/server-cloud/products/sql-server/  
[Accessed 31 03 2014].

Northen Ireland Statictics and Research Agency, 2014. [Online]   
Available at: http://www.nisra.gov.uk/  
[Accessed 25 03 2014].

Northern Ireland Statistics and Research Agency, 2014. *About NISRA.* [Online]   
Available at: http://www.nisra.gov.uk/aboutus/index.html  
[Accessed 26 03 2014].

# Appendices

The appendices are an opportunity to provide secondary material in support of the description in the body of the dissertation. In principle, the reader need not look at the appendices and no specific marks are awarded for this section.

Sample content:

## A1. Analysis Models

e.g. SSM models

## A2. Design Models

e.g. database schema

## A3. Code

Code developed through the project

## A4. Test Suite

Full set of tests applied to the software

## A5. Questionnaire Results

Results of questionnaires used to evaluate the software or identify requirements