Software Architecture Project 2 Documentation
Microminer Web Search Engine

Group 20 - Jason Switzer, Cliff Halasz

1. Process Architecture

In designing the architecture for our solution to the Microminer Web Search Engine project, we found that there was not a lot of modification necessary from our previous build. Since we had already implemented a removal function, implementing search was as easy as copying the algorithm and collecting the indexes that matched a field instead of removing them. The team would meet after lectures to perform overall architecture design and to break down task responsibilities, and then the google talk protocol was used to communicate on a day-to-day basis. Our team of only 2 quickly came to the same architecture design, and division of tasks was simple.

A bulk of the remaining work were GUI manipulations to meet the customer's prior implicit requirement and a cleaner search form. We needed to create a new GUI window for manually entering text fields, break out the GUI for loading a text file into a separate window (previous build's main window). The only requirements that needed architectural changes were noise word reduction and multi-word searching, but these changes had minimal effect on the architecture. Since the load and create windows were independent and the existing architecture was flexible enough, we worked on them individually and simultaneously. After the GUI was completed, the remaining two tasks of noise word reduction and multi word searching were divided between the two of us. After finishing the search and the GUI modifications, we created a set of test input files and expected results to test each of the functional and non-functional requirements. These tests uncovered a few minor glitches so this effort was of great value.

2. Architecture Specification

2.1 Hybrid Object Oriented Architecture

2.1.1 Description

The Hybrid architecture aims to combine the best aspects of the ADT/OO and Implicit Invocation architectures discussed in class. This approach uses a generalized object system that builds upon the standard Java library for high re-usability and maintainability. It also uses the standard Java GUI DataModel architecture and background threads to update the interface seamlessly and quickly. Due to the use of the DataModel architecture, an implicit invocation occurs when an item is added to the list which triggers an event that the list widget (JList) registered for in order to update itself.

The system is designed to operator similar to Iterators: each module in the system is iterated over and each result is passed to the next module down the line for processing. Once the Reader reads a line, it is added to the GUI (input record list) and added to the Output list immediately. The Output list will pass the line to the Circular Shifter, and then iterate over each circular shifted result, adding it to itself (the Output list). Once the Output list is created with the fully indexed output, it can also be iterated over, with each result added to the GUI (indexed record list) directly.

2.1.2 Advantages

This design maintains high reusability and is also highly extensible. Each step can be separated and operate independently, making it possible to perform the steps in parallel. Because each module is only aware of what it expects as input and which step is next in line, the coupling is kept low, though not as high as implicit invocation. The DataModel architecture makes it easy to keep the user interface updated as frequently as we want with very custom code necessary.

2.1.3 Disadvantages

The coupling is not as low as implicit invocation because the IndexedList has to be aware of how to perform the shifting, though this is done only as an abstraction from the GUI, which acts as the main controlling unit. Since the GUI only cares about the results in the Output list, it was decided that this is a comfortable medium. The DataModel architecture is not designed for rapid updates to the underlying data model, resulting in quirky GUI behavior in certain circumstances (loading large files causes unwanted flicker). Space is not empasized, so no attempt to reduce object usage is taken; there are situations in which too many objects are created, though this is likely limited because of the iterator design (those objects are temporaries and are short lived, which is good for garbage collection).

2.2 Function Implementations

2.2.1 Deletion of noise words

The first method to remove noise words would be to eliminate them in the reader, immediately as they are input. The reader would simply store the titles without the noise words. The second method is to ignore lines that begin with noise words after performing the circular shift (by shifting until a non-noise word begins the description). The second method enables us to still store the name of the website exactly as entered, however we still ignore noise words for the purposes of search. We decided to implement the second method for the increased functionality.

2.2.2 Search

The simplest method to perform the search would be a linear search of all possible circular shifts. This is easiest, but by far the slowest. An alternative method would be to store all the indexes in a tree structure, so that searching is much more efficient. We decided that what would best fit our architecture would be to utilize the IndexList data structure which extends TreeSet and would automatically handle searching of the tree efficiently. The search is performed on each individual word in the search field through all the circular shifts, and the results are combined with repetitions omitted. Since every possible circular shift begins with a searchable word, only the first word needed to be searched, further reducing the time complexity. This was very easy to implement as we were already storing the circular shifts in this way, so we are just taking advantage of the existing object oriented design.

3. Program Specification

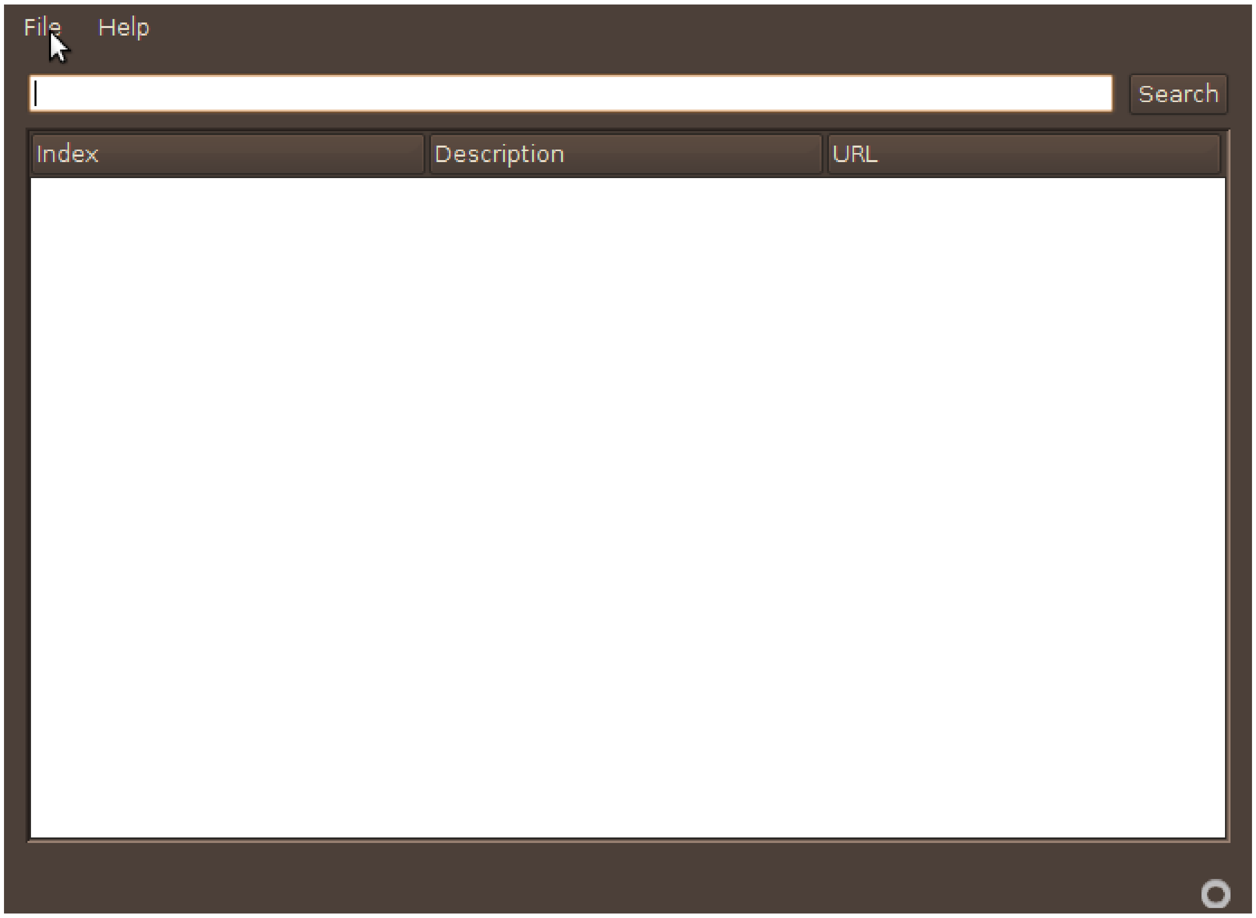
Main extends SingleFrameApplication and sets up the MainWindow, which controls the GUI interface and performs the tasks of the Printer. This can essentially be considered the Main Control module of the program. When loading a file, MainWindow passes the file path to InputReader which performs the work of the Reader module. The InputReader is iterated over to obtain the input lines, which is stored in an IndexedString. When entering text the same process is done with the text box.

The IndexedString objects are passed directly to the IndexedList, which acts as an abstraction to the Alphabetizer. The IndexList also provides an abstraction to the CircularShifter (see the IndexList.add function), which performs the role of the Shifter by creating new IndexedString objects for all of the circular shifts of the IndexedStrings it was passed. The IndexList will iterate over the CircularShifter and add each circularly shifted result to itself. The underlying data type of the IndexList was chosen for sorted inserts (TreeSet), so no further work is needed.

Once the InputReader terminates the iterator, the MainWindow (Main Control module) will iterate over the IndexList as well, adding each IndexedString directly to the user interface, which represents the Output module. The GUI uses DataModels, which will implicitly fire events to trigger the update of the appropriate widgets. This is all abstracted and provided by the Java JRE. Search functionality is achieved with a simple function on MainWindow that calls on the TreeSet, and then the results are output to the main GUI DataModel.

4. User Manual

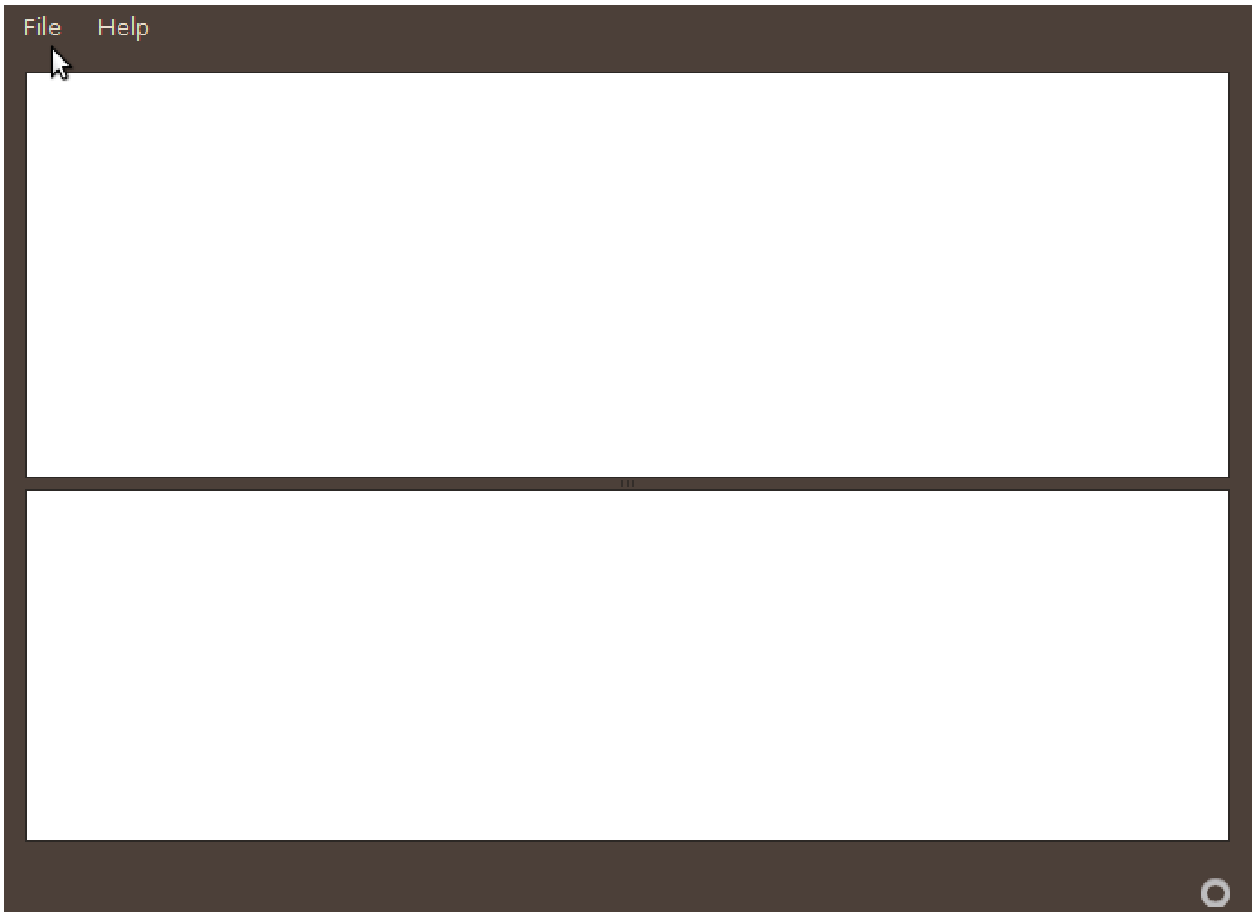
This is the basic Microminer Web Search window:



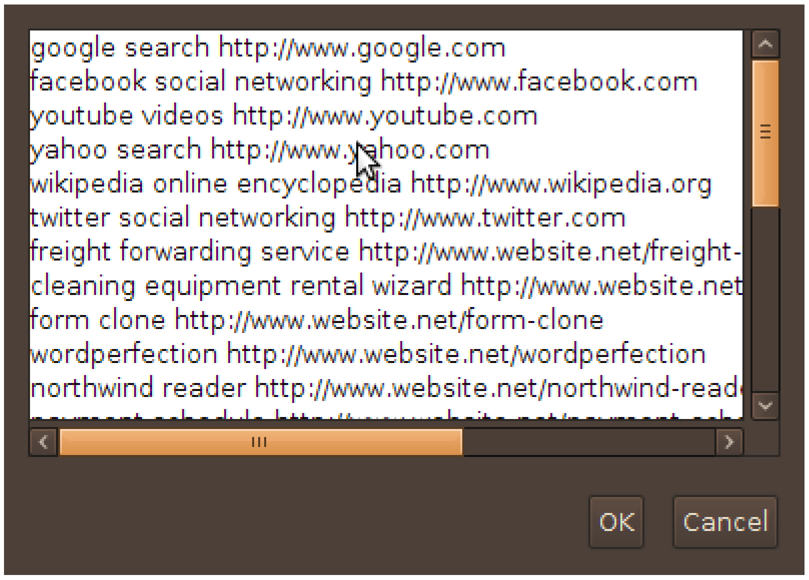
4.1 Manually entering data

If you wish, you can enter data into the search engine manually in a text box format. This works well for copy and pasting lists.

Start by expanding the *File* menu and selecting the *Create Index...* menu item:



An empty text box will open, allowing the user to type or paste any text:



google search http://www.google.com
facebook social networking http://www.facebook.com
youtube videos http://www.youtube.com
yahoo search http://www.yahoo.com
wikipedia online encyclopedia http://www.wikipedia.org
twitter social networking http://www.twitter.com
freight forwarding service http://www.website.net/freight-
cleaning equipment rental wizard http://www.website.net
form clone http://www.website.net/form-clone
wordperfection http://www.website.net/wordperfection
northwind reader http://www.website.net/northwind-read

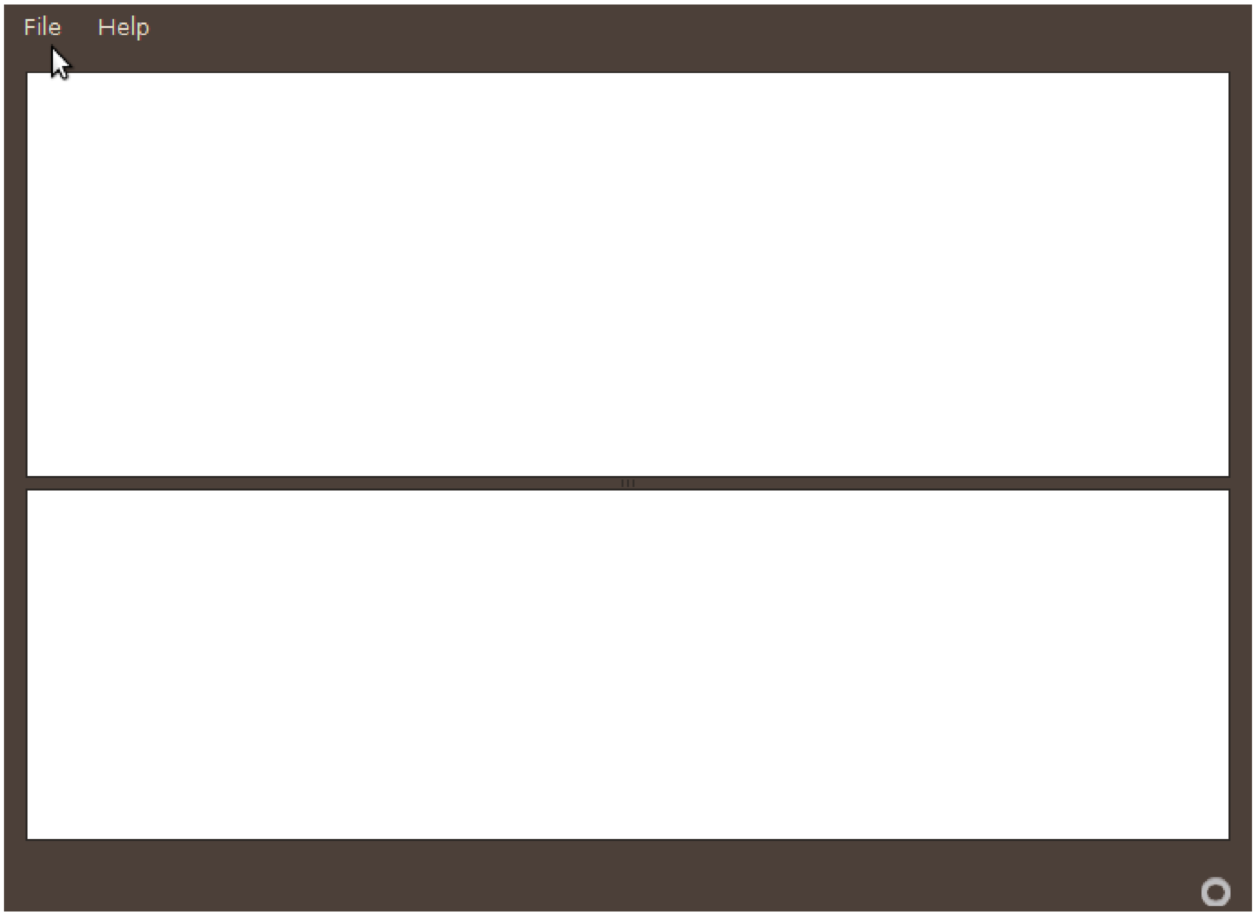
The text box will automatically add scroll bars if the text becomes larger than the size of the window.

Text data is expected in the format <name> <http(s)://><url> and lines that do not conform will be ignored. If satisfied, press *OK* to load this text data into the engine. If you wish, you can press *Cancel* to return to the main window without loading any data.

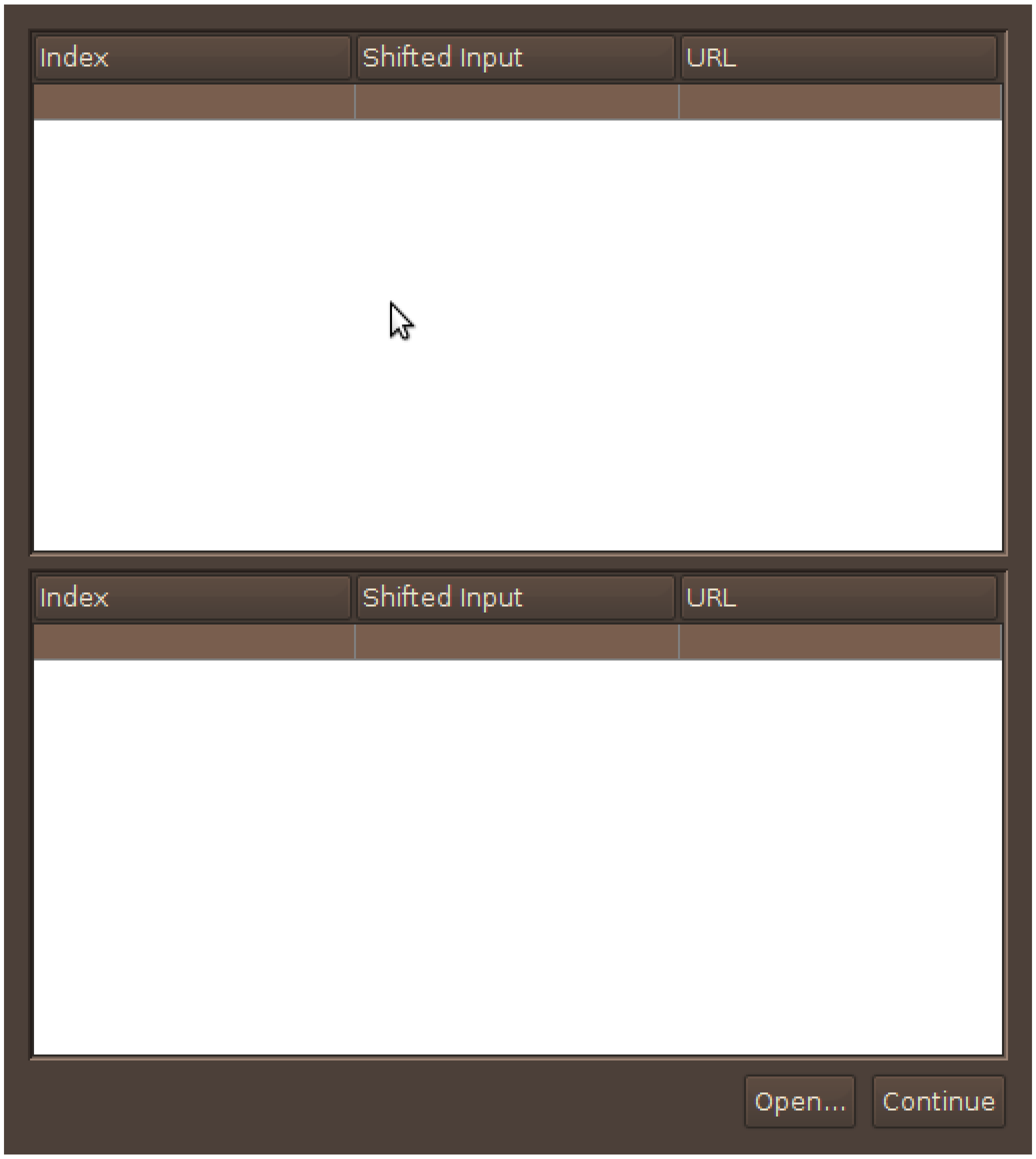
4.2 Loading data from a file

If you wish, you can enter data into the search engine manually in a text box format. This works well for copy and pasting lists.

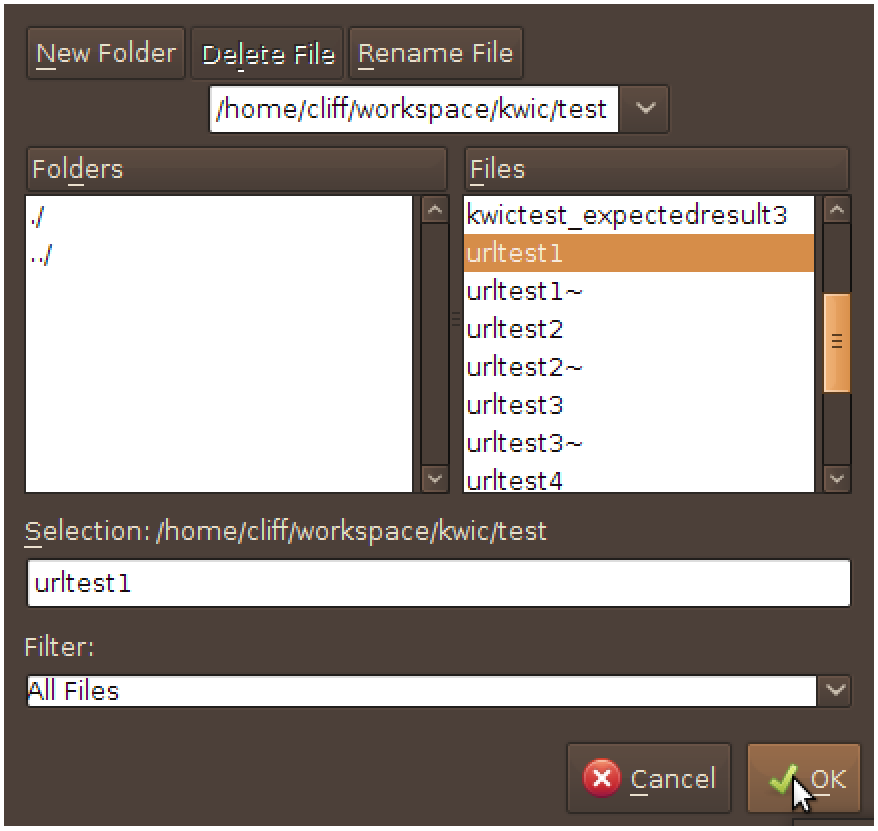
Start by expanding the *File* menu and selecting the *Load Index...* menu item:



The file display window will come up, initially empty:



click on the *Open...* button to open the file select dialogue:



/home/cliff/workspace/kwic/test

./
../

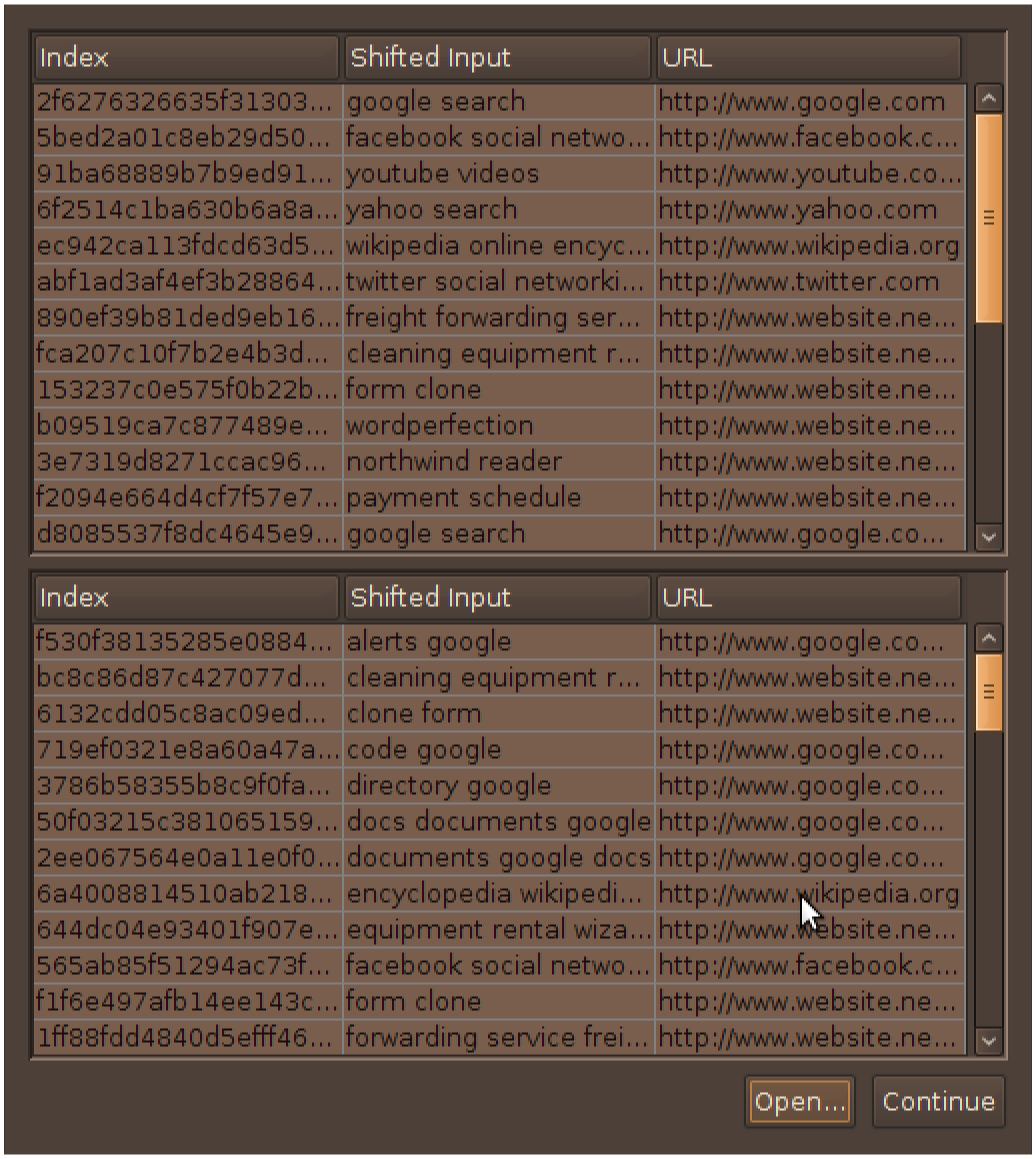
kwictest_expectedresult3
urltest1
urltest1~
urltest2
urltest2~
urltest3
urltest3~
urltest4

Selection: /home/cliff/workspace/kwic/test

urltest1

Filter: All Files

Where you can navigate to the file you would like to load. After selecting the file, press *OK* to continue, or *Cancel* if you wish to go back:



Index	Shifted Input	URL
2f6276326635f31303...	google search	http://www.google.com
5bed2a01c8eb29d50...	facebook social netwo...	http://www.facebook.c...
91ba68889b7b9ed91...	youtube videos	http://www.youtube.co...
6f2514c1ba630b6a8a...	yahoo search	http://www.yahoo.com
ec942ca113fdcd63d5...	wikipedia online encyc...	http://www.wikipedia.org
abf1ad3af4ef3b28864...	twitter social networki...	http://www.twitter.com
890ef39b81ded9eb16...	freight forwarding ser...	http://www.website.ne...
fca207c10f7b2e4b3d...	cleaning equipment r...	http://www.website.ne...
153237c0e575f0b22b...	form clone	http://www.website.ne...
b09519ca7c877489e...	wordperfection	http://www.website.ne...
3e7319d8271ccac96...	northwind reader	http://www.website.ne...
f2094e664d4cf7f57e7...	payment schedule	http://www.website.ne...
d8085537f8dc4645e9...	google search	http://www.google.co...

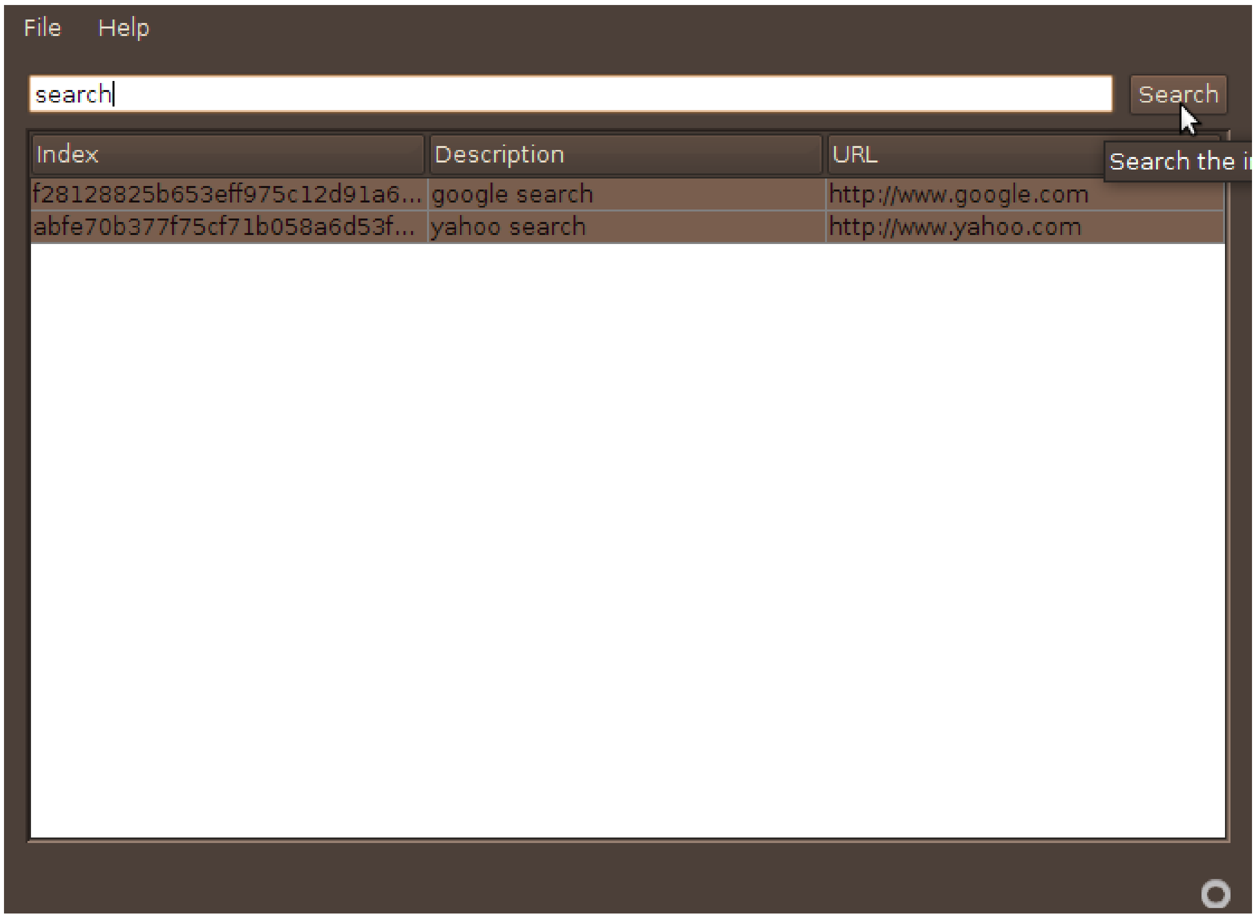
Index	Shifted Input	URL
f530f38135285e0884...	alerts google	http://www.google.co...
bc8c86d87c427077d...	cleaning equipment r...	http://www.website.ne...
6132cdd05c8ac09ed...	clone form	http://www.website.ne...
719ef0321e8a60a47a...	code google	http://www.google.co...
3786b58355b8c9f0fa...	directory google	http://www.google.co...
50f03215c381065159...	docs documents google	http://www.google.co...
2ee067564e0a11e0f0...	documents google docs	http://www.google.co...
6a4008814510ab218...	encyclopedia wikipedi...	http://www.wikipedia.org
644dc04e93401f907e...	equipment rental wiza...	http://www.website.ne...
565ab85f51294ac73f...	facebook social netwo...	http://www.facebook.c...
f1f6e497afb14ee143c...	form clone	http://www.website.ne...
1ff88fdd4840d5efff46...	forwarding service frei...	http://www.website.ne...

The file select dialogue will go away, and the file display window will now be populated with the data from the file. If you wish you can review the data using the scroll bars, and you may choose to select any input line or circular shift line and remove it manually. When satisfied with the data, you can press *Continue* to return to the Main Window.

4.3 Searching

In order to perform a search, some data must first be loaded into the system by using the Create Index or Load Index. On opening Create Index or Load Index you will override any Index currently in the system.

Once you have completed creating or loading an index, you can perform a search of the index by entering text in the Search Bar on the Main Window and then simply press the *Search* button:



Index	Description	URL
f28128825b653eff975c12d91a6...	google search	http://www.google.com
abfe70b377f75cf71b058a6d53f...	yahoo search	http://www.yahoo.com

The results will be displayed in the window below the Search Bar, and a scroll bar will be added if necessary.