Linux Configuration Wizard

By

T. Brian Sabatini

Submitted to
the Faculty of the Information Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

University of Cincinnati
College of Applied Science
June 2006
Linux Configuration Wizard

By

T. Brian Sabatini

Submitted to
the Faculty of the Information Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Brian Sabatini 2006

The author grants to the Information Technology Program permission
to reproduce and distribute copies of this document in whole or in part.

T. Brian Sabatini

Date

Jamierehase Sangl

Date

Patrick R. Kueppl

Date
 Acknowledgements

I would like to thank my Senior Design technical Professor, Professor Tamisra Sanyal for all of his help and support through my entire Senior Design project. I would also like to thank the Company I work for, Sibco Building Products, for offering the flexibility and support to go to school and work on my Senior Design Project. I would also like to thank Professor Stockman, for helping give me the inspiration and ideas for this project. Lastly, I would like to thank Dr. Jo Ann Thompson for taking time to offer help with my reports for every one of my Senior Design quarters.
# Table of Contents

Acknowledgements .............................................................. i  
Table of Contents ............................................................ ii  
List of Figures ................................................................ iii  
Abstract ................................................................................ iv  

1. Statement of the Problem ..................................................... 1  
2. Description of the Solution ................................................... 2  
   2.1 User Profile ................................................................ 3  
      2.1.1 Administrator Profile ........................................... 3  
   2.2 Design Protocols ....................................................... 3  
      2.2.1 Technology ....................................................... 3  
      2.2.2 Organizational Scheme ....................................... 4  
      2.2.3 Use Case Diagram ............................................. 5  
      2.2.4 User Interface ................................................... 6  
         2.2.4.1 Interface Navigation .................................... 6  
         2.2.4.2 Interface Design ......................................... 7  
         2.2.4.3 Color Scheme ............................................ 8  
         2.2.4.4 Help ....................................................... 8  
3. Deliverables ......................................................................... 9  
4. Development and Design ..................................................... 9  
   4.1 Project Schedule ....................................................... 9  
      4.1.1 Senior Design I ............................................... 9  
      4.1.2 Senior Design II ............................................. 10  
      4.1.3 Senior Design III .......................................... 10  
   4.2 Project Resources .................................................... 11  
   4.3 Project Budget ....................................................... 12  
5. Proof of Concepts ............................................................. 12  
   5.1 Configuration ........................................................ 12  
      5.1.1 NIC Configuration ........................................... 13  
      5.1.2 DHCP Configuration ........................................ 16  
      5.1.3 NFS Configuration ......................................... 17  
      5.1.4 DNS Configuration ........................................ 19  
   5.2 Monitor ................................................................... 19  
   5.3 Other Tasks ........................................................... 20  
      5.3.1 Manual Edit .................................................. 20  
      5.3.2 Control Services ............................................ 21  
      5.3.3 Backup and Restore ....................................... 23  
   5.4 Help ...................................................................... 25  
6. Testing .............................................................................. 26  
7. Conclusions and Recommendations ....................................... 27  
   7.1 Conclusions ........................................................... 27  
   7.2 Recommendations ................................................... 27  
Appendix A ........................................................................ 28  
Appendix B – Code Snippets ............................................... 31  
References ........................................................................ 35
List of Figures

Figure 2.1 - Use-Case Diagram ................................................................. 5
Figure 2.2 - Application Navigation/Workflow ............................................... 6
Figure 2.3 - Main Menu ........................................................................... 7
Figure 2.4 - Service Selection ................................................................ 7
Figure 2.5 - NFS Configuration ................................................................. 7
Figure 5.1 – Network Card Configuration Screen ....................................... 14
Figure 5.2 – Network Card Configuration File ............................................. 14
Figure 5.3 - Incorrect Information Prompt ................................................... 15
Figure 5.4 - Prompt That No Changes Were Made To the NIC Configuration . 15
Figure 5.5 – DHCP Information Entered In Application ............................... 16
Figure 5.6 – DHCP Configuration File Created By Application .................... 16
Figure 5.7 - Prompt That No Changes Were Made To The DHCP Configuration . 17
Figure 5.8 - NFS Action Menu ................................................................ 18
Figure 5.9 – NFS Share Information ........................................................... 18
Figure 5.10 – NFS Share Permissions ......................................................... 18
Figure 5.11 – NFS Configuration File Before and After Configuration ............ 18
Figure 5.12 – DNS Configuration Screen ................................................... 19
Figure 5.13 – System Monitor Screen .......................................................... 20
Figure 5.14 – File Selection Screen .............................................................. 21
Figure 5.15 – Manual Edit Screen .............................................................. 21
Figure 5.16 – Selecting a Service to Control ............................................... 22
Figure 5.17 – Selecting an Action to Take .................................................... 22
Figure 5.18 – The Confirmation Screen ...................................................... 23
Figure 5.19 – Backup and Restore Menu .................................................... 24
Figure 5.20 – Backup Screen ................................................................... 24
Figure 5.21 – Restore Screen .................................................................... 25
Figure 5.22 – Help Screen ....................................................................... 26
Figure A.1 – Project Time Line .................................................................. 28
Figure A.2 – Senior Design II Timeline ...................................................... 29
Figure A.3 – Senior Design III Timeline ..................................................... 30
Abstract

The Linux Server Configuration Wizard is an application designed to help reduce the difficulties associated in implementing a server with a Linux operating system. The main focus of the application is for small businesses. Small businesses usually have limited decisions regarding the selection of what I.T. resources are brought into their operation. These limitations are the result of deficiently funded budgets and the incapability to support resources in-house. The application was developed to give a small business complete control over a Linux server. With the application, several essential network services can easily be set up and configured, which can provide all the needed components for an operational network. The services that the application can be used to configure include: DHCP, DNS, Samba, and NFS. Not only does the application offer the ability to configure network services, but it also provides the ability to monitor them as well. Along with monitoring network services, other vital system statistics can be viewed. Other capabilities that the application provides are: the ability to backup and restore system configurations, the ability to control the state of network services, and the ability to manually edit system configuration files.
Linux Server Configuration Wizard

1. Statement of the Problem

Many small businesses have a need for computer resources but are often unable to obtain them. Small businesses are restricted to resources that they are capable of supporting and able to afford. To be able to support these resources, a business must either have a capable staff or hire outside consulting aid in the support. Hiring outside consultants can become expensive, leaving a business to go with the decision to support the resources in-house. Choosing to support resources in-house often eliminates the decision to go with open-source products since they are difficult to maintain. Therefore, when deciding to go with a computer resource such as a server operating system, many businesses choose to go with a product like Microsoft Windows. They choose this for its ease of installation and configuration. The negative aspect of a product like Microsoft Windows is that the operating system itself, along with additional licenses that may be needed to be purchased, can become very costly. Some small businesses cannot afford this expense.

There is a need for a user-friendly configuration wizard for a Linux server that could help a small business easily implement an open-source server. This wizard would eliminate the difficulty in configuring an open-source operating system, giving a small business the ability to use it in their operation. This application would need to provide an environment similar to the Microsoft Windows 2003 Server Roles feature currently available. It should provide a place for central management and configuration for a server. It needs to be easy to use by providing a graphical user interface and wizard type workflow. It must configure the essential services for a server such as DNS, DHCP,
network file storage, and Samba for file and printer sharing. An application such as this would let a small business implement open-source computer resources for little cost and minimal difficulty.

2. Description of the Solution

My solution to reduce the difficulty in implementing a system with a Linux operating system is to develop the Linux Server Configuration Wizard. This application will allow the setup and configuration of a Linux server to be done with minimal knowledge needed. This application will allow more small businesses to implement a system because they will not need to hire outside consultants or additional staff to support the system.

Key features of this application are:

- Setting up common networking services, which include
  - DHCP
  - DNS
  - NFS
  - Samba
- Configuring a NIC (network card)
- Checking the status of the server in real-time
- Automatically configuring security settings
- Letting services be controlled (stopped, started, and restarted)
- Providing a manual edit window for custom or advance configurations, allowing everything to still be done from one place
- Providing a backup and restore functionality for configured services
2.1 User Profile

There is one main user for this application: the administrator, the user who is responsible for the server configuration.

2.1.1 Administrator Profile

The administrator is the person who will be using the application to configure the server. The administrator will be able to use the application to configure a desired service, and be able to start, stop and restart those services. The administrator will also be able to go into the application to check the current status of a service to make sure it is operating correctly.

2.2 Design Protocols

2.2.1 Technology

The application will be written with Perl. The graphical user-interface will be provided by the Perl module Cmenu. This is a module that was built to bind with Perl to interface with the Curses library. Cmenu will allow the interface to be displayed within a terminal window, allowing the application to be used in Linux at “runlevel 5” as well as “runlevel 3.” For parts of the application that need to store or recall information, a simple database will be used, since what is needed is very limited. After looking in Perl DBI, I found that a DBM database would be the best decision.
2.2.2 Organizational Scheme

The Linux Application Configuration Wizard is an application that is divided into three sections: configuring, monitoring, and completing special tasks. It will provide a wizard-like workflow environment, taking the user through a process one step at a time. When a user opens the application, it will bring up a welcome window where a user can select a task to perform. Figure 2.1 shows the use-case diagram for the different functions for the application. Figure 2.2 shows the different steps a user can take throughout the application. Most tasks can be done for multiple services, and while the steps will be different for each service, the process is the same. For example, if a user wants to configure the DHCP service, the user would select *configure* and then choose DHCP from the list of services, which would take the user through the steps related to that service while the other services would have different steps.
2.2.3 Use Case Diagram

Figure 2.1 - Use-Case Diagram
2.2.4 User Interface

2.2.4.1 Interface Navigation

Figure 2.2 - Application Navigation/Workflow
2.2.4.2 Interface Design

Figures 2.3, 2.4, and 2.5 show example screenshots from the actual application.
2.2.4.3 Color Scheme

Since the application is terminal-based, the color scheme is rather simple and restricted to the capabilities of the Cmenu module. The color scheme is as follows:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>Blue</td>
</tr>
<tr>
<td>Background Text</td>
<td>Light Blue</td>
</tr>
<tr>
<td>Current Window</td>
<td>Gray</td>
</tr>
<tr>
<td>Current Window Guide Text</td>
<td>Yellow and Gray</td>
</tr>
<tr>
<td>Current Window Option Labels</td>
<td>Yellow</td>
</tr>
<tr>
<td>Current Window Option Text</td>
<td>Black</td>
</tr>
<tr>
<td>Editable Field Text</td>
<td>Red</td>
</tr>
<tr>
<td>Confirmation/Info Screen</td>
<td>Orange</td>
</tr>
<tr>
<td>Confirmation/Info Screen Text</td>
<td>White</td>
</tr>
<tr>
<td>Warning Screen</td>
<td>Red</td>
</tr>
<tr>
<td>Warning Screen Text</td>
<td>White</td>
</tr>
</tbody>
</table>

Table 2.1 - Application Color Chart

2.2.4.4 Help

There will be a specialized help screen for each menu in the application. The user will be able to access these screens by using the help button on each menu. Each help screen will give in-depth information about the screen the user is currently on. This information will include:

- Information about the part of the application the user is in
- An overview of the options the user has for the current menu or screen
- Detailed information for parts related to services
  - What the service is and what it does
  - Information about the fields the user needs to populate
3. Deliverables

To construct a complete and easy-to-use system, a selected list of deliverables has been developed. The deliverables listed below came from the design phase for the Linux Server Configuration Wizard:

1. Graphical Interface provided through Perl modules
2. Detailed help files with each process
3. Configuration of a network card
4. Configuration of network services
   - Automatically install service when configured
   - Automatically set service to run at startup
   - Make any needed security configuration changes
   - Replace current configuration with application template when configuring
   - Allow the option to cancel configuration
5. Ability to monitor system
   - Show what services are installed/configured
   - Show the status of configured services
   - Give a summary of the settings for configured services
6. Ability to perform special tasks:
   - Manually edit a configuration file
   - Backup the current configuration settings
   - Restore saved configuration settings
   - Start services that are stopped
   - Stop services that are running
   - Restart services that are running

4. Design and Development

This section covers the project’s timeline, budget, and resources through the entire Senior Design process.

4.1 Project Schedule

Tasks and goals were delegated out over three different quarters. Below are the objectives met for each of the Senior Design sections.

4.1.1 Senior Design I

During Senior Design I the following was performed:

- Generated ideas for development
During the creation of the project, a lot of time was spent trying to find what the most needed network services were. After the services were selected, they were used as the foundation for the project. The application was designed and structured to configure those selected services. Once the decision on the services was made, there was a lot of research done to find out what configuration was needed for each service.

4.1.2 Senior Design II

During Senior Design II the following was performed:

- Developed the overall layout and scheme
- Created the user interface
- Added functionality to the application
- Started minor testing
- Developed Design Freeze documentation
- Presented project to faculty and class

The design and creation of the user interface required a great deal of effort. The application being for a Linux system did not allow for ease or a wide use of development tools. The program being written in Perl also did not allow for the most flexibility in development either. Once the proper modules were found, things eventually started to come together.

4.1.3 Senior Design III

During Senior Design III the following was performed:
• Completed the user interface
• Completed the application functionality
• Completed testing
• Completed project documentation
• Presented project at Tech Expo
• Presented project to faculty and class

Developing the functionality for the manual edit feature was a difficult task. While the goal was to be able to perform the task within the application, due to module limitations it was not able to be done. As a result an external call to a text editor in a new console was used.

Detailed timelines for the project and selected quarters can be found in Appendix A.

4.2 Project Resources

Table 4.1 is a presentation of the resources that the project requires. This table represents all of the resources necessary for completing the project, such as the software, labor, and reference material.

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Type</th>
<th>Material Label</th>
<th>Std. Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perl</td>
<td>Material</td>
<td>Software</td>
<td>$0.00</td>
</tr>
<tr>
<td>Labor</td>
<td>Work</td>
<td></td>
<td>$50.00/day</td>
</tr>
<tr>
<td>Books</td>
<td>Material</td>
<td>Books</td>
<td>$136.00</td>
</tr>
<tr>
<td>Manuals</td>
<td>Material</td>
<td>Books</td>
<td>$14.00</td>
</tr>
</tbody>
</table>

Table 4.1 – Project Resources

Resource Notes:
• Perl is an open source scripting language and can be used and used for development freely.
• Since being that this project is not for a specific company, the labor was given an estimated rate of $50.00 dollars a day with and every day of the project is calculated as a working day.
• The book and manual cost are a combined total of purchases and the cost is distributed accordingly for each task.
• There were two books used for technical reference: Perl Core Language and Linux Administration.
4.3 Project Budget

Table 4.2 represents the cost of completing the project. Each task of the project requires a different amount of resources, and this reference shows the cost (software, labor, and reference material) distributed across each task accordingly.

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. <strong>Overall Design of Application</strong></td>
<td>$114.00</td>
</tr>
<tr>
<td>2. Develop the overall layout/scheme</td>
<td>$114.00</td>
</tr>
<tr>
<td>3. <strong>Menu Creation</strong></td>
<td>$313.60</td>
</tr>
<tr>
<td>4. Design main menusplash screen</td>
<td>$50.00</td>
</tr>
<tr>
<td>5. Create sub-menus for main menu selections</td>
<td>$113.60</td>
</tr>
<tr>
<td>6. Specialized configuration menus for each service</td>
<td>$363.60</td>
</tr>
<tr>
<td>7. Monitor screen design</td>
<td>$63.60</td>
</tr>
<tr>
<td>8. Manual edit screen design</td>
<td>$63.60</td>
</tr>
<tr>
<td>9. Other task selection menus</td>
<td>$63.60</td>
</tr>
<tr>
<td>10. Other task selection sub-menus</td>
<td>$113.60</td>
</tr>
<tr>
<td>11. <strong>Adding Functionality</strong></td>
<td>$354.40</td>
</tr>
<tr>
<td>12. Service configuration functionality for each service</td>
<td>$363.60</td>
</tr>
<tr>
<td>13. Control services functionality</td>
<td>$113.60</td>
</tr>
<tr>
<td>14. Monitor screen functionality</td>
<td>$163.60</td>
</tr>
<tr>
<td>15. Manual edit screen design</td>
<td>$213.60</td>
</tr>
<tr>
<td>16. <strong>Testing</strong></td>
<td>$200.00</td>
</tr>
<tr>
<td>17. Navigation workflow</td>
<td>$100.00</td>
</tr>
<tr>
<td>18. Functionality testing</td>
<td>$100.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$4,000.00</strong></td>
</tr>
</tbody>
</table>

Table 4.2 – Project Budget

5. Proof of Concepts

This section will show how the deliverables of the project were fulfilled and information about completing the task. This section is broken down into three parts, one for each main part of the application. The three sections are: configuration, monitoring, and other tasks.

5.1 Configurations

This section will show how the deliverables were met for the configuration part of the application. The main purpose for this part of the application is for the configuration
of network services and for configuration of a network card. The four services being configured are DHCP, DNS, NFS, and Samba. When the user configures a service, the user is prompted for the needed information to configure the service. Once the information is entered, a configuration file that the system uses is populated.

### 5.1.1 NIC Configuration

The user can use the application to set up a network card. The configuration file for the NIC is populated with static information, along with information for which the user is prompted. Figure 5.1 shows the screen that prompts the user for information in the application. Figure 5.2 shows the configuration file that is populated for the NIC. If the user leaves a field blank or enters an invalid parameter, the system will prompt the user and not update the configuration wizard. Figure 5.3 shows the prompt the user would receive for incorrect information being entered. The difficult part about this was checking the syntax of the I.P. address the user entered, since there could be different combinations of numbers used. To check the address a one-line regular expression was used. The address was compared to the regular expression to check for valid digits, length, and structure. If the user leaves this part of the application without entering any information, there is a prompt to let the user know that no changes were made. Figure 5.4 shows this prompt.
Figure 5.1 – Network Card Configuration Screen

Figure 5.2 – Network Card Configuration File
Figure 5.3 - Incorrect Information Prompt

Figure 5.4 - Prompt That No Changes Were Made To the NIC Configuration
5.1.2 DHCP Configuration

Figure 5.5 shows a prompting screen for the DHCP service, and Figure 5.6 shows the configuration file that was created using the information supplied. If this section of the application is entered and then exited without any information being entered, the user will be prompted that no changes were made. This prompt can be seen in Figure 5.7.

![Figure 5.5 – DHCP Information Entered In Application](image)

![Figure 5.6 – DHCP Configuration File Created By Application](image)
5.1.3 NFS Configuration

For the NFS configuration a user has two options: deleting a share or adding a share. Figure 5.8 shows the menu where the user can select what action is to be performed. From this point the user can go one of two ways depending on the selection made: either delete a share or add a share. Figures 5.9 and 5.10 show share information and permissions being entered for a share that will be added. Figure 5.11 shows the configuration file before and after the share was added.
Figure 5.8 - NFS Action Menu

Figure 5.9 – NFS Share Information

Figure 5.10 – NFS Share Permissions

Figure 5.11 – NFS Configuration File Before and After Configuration
5.1.4 DNS Configuration

For a DNS server a caching name server is configured. This was chosen because it would be the easiest for a user to configure while still offering many benefits and better performance. The input information is limited, only an ISP DNS server address and a domain name are needed. Figure 5.12 shows the DNS configuration screen.

![Figure 5.12 – DNS Configuration Screen](image)

5.2 Monitor

This section will show how the deliverables were met for the monitoring part of the application. The application will also provide a central location for the user to monitor the status of the server itself, along with the status of the services running on it. Figure 5.13 shows a system monitor screen from the application which is showing the status of the system.
5.3 Other Tasks

This section will show how the deliverables were met for the other tasks section in the application.

5.3.1 Manual Edit

For advanced and custom configurations, the application comes with a manual edit feature. Due to limitations of the module being used, it was not possible to be able to perform manual editing within the application itself. The reason was because there is no scroll functionality available for the windows, so if a configuration file was longer than the window only the top half would be shown. An alternative method was used instead to provide manual editing. This method was done by calling an external shell to open a configuration file with a text editor, then when the editor is closed focus returns back to the main application. Figure 5.14 shows the selection window where a user can select a...
configuration file to edit. Figure 5.15 shows a text editor that was opened externally by the application.

![Figure 5.14 – File Selection Screen](image1)

![Figure 5.15 – Manual Edit Screen](image2)

5.3.2 Control Services

A user can also use the application to control services. The user can select the service that will be controlled, and then select to stop, start, or restart the service. Figure
5.16 shows the service that the user wants to control being selected. Figure 5.17 shows the action being selected that will be taken against the service. Figure 5.18 shows the confirmation of the action being taken against the service after it is complete.
5.3.3 Backup and Restore

To backup and restore the application uses tarball files, which is a file type used in Linux for archiving. The user can select a location to back a file up to or select a file to restore. When backing up, the application archives all of the configuration files together. Figure 5.19 shows the backup and restore menu. Figures 5.20 and 5.21 show the backup and restore screens respectively.
Figure 5.19 – Backup and Restore Menu

Figure 5.20 – Backup Screen
5.4 Help

The application also provides help screen for the user for each process that can be taken. Figure 5.22 shows an example help screen from the application.
6. Testing

Each part of the application (configuring, monitoring, and completing special task) is broken down into its own package within the code. Each package is split into two parts: one that provides the menus and screens, and one that performs the functionality. Being broken down into parts like that, each part could be individually tested as it was developed. Both parts of a package were tested during development. First the workflow of the menus was tested to make sure that all the right objects were displayed. Then the functionality was tested to make sure the configuration was done correctly. Testing for the service configurations included making sure the configuration file was accurately populated, the service was started, and the service functioned properly.

The testing went on alongside with the development phase to make sure that everything was developed correctly. Once the application development was done, it was put onto a new system without any services configured. Then the application was used to configure all of the services for the system. Once all of the services were then set up, further testing was performed to verify that the services were running correctly on the new system.
7. Conclusions and Recommendations

7.1 Conclusions

This project was developed to help a small business be able to easily implement an open-source system into its operation. By doing so, the total cost of implementation is reduced making it a realistic option. Perl was the tool used to develop an application to configure a server and provide a graphical user interface. Deliverables were developed based upon what a small business would need for a fully operational and manageable network. Over the course of three Senior Design quarters the project was completed and met all the deliverables.

7.2 Recommendations

The development of this application was a true learning process. Trying to develop a console application for the console of a Linux operating system was difficult. The biggest problem was trying to find the right technology to use to provide menus and screens for Linux. To make this even trickier, this was done using a scripting language instead of a programming language. I would recommend using a native Linux programming language such as C or C# instead. The other difficulty was that everything was developed with a text editor only. There are not a lot of options in Linux for the use of an IDE as there are for Microsoft, such as Visual Studio. When using a language like C in Linux then there are options available such as QT or kdevelop which could be used.
# Appendix A

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Start</th>
<th>Finish</th>
<th>Sep 25, T5</th>
<th>Nov 10, T5</th>
<th>Jan 1, B5</th>
<th>Feb 19, B5</th>
<th>Apr 8, B5</th>
<th>May 20, B5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Past &amp; Current Task</td>
<td>Thu 10/29</td>
<td>Mon 12/5</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
</tr>
<tr>
<td>2</td>
<td>Idea/Plan Development</td>
<td>Thu 10/5</td>
<td>Mon 12/5</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
</tr>
<tr>
<td>3</td>
<td>Progress Report 1</td>
<td>Thu 10/5</td>
<td>Thu 10/11</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
</tr>
<tr>
<td>4</td>
<td>Problem/Need/Area of Inquiry</td>
<td>Fri 10/4</td>
<td>Fri 10/11</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
</tr>
<tr>
<td>5</td>
<td>Draft Proposal</td>
<td>Thu 10/7</td>
<td>Thu 10/11</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
</tr>
<tr>
<td>6</td>
<td>Progress Report 2</td>
<td>Thu 10/27</td>
<td>Thu 10/31</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
</tr>
<tr>
<td>7</td>
<td>Final Proposal</td>
<td>Thu 11/3</td>
<td>Thu 11/7</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
</tr>
<tr>
<td>8</td>
<td>Oral Presentation</td>
<td>Thu 12/1</td>
<td>Thu 12/5</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>M</td>
</tr>
<tr>
<td>9</td>
<td>Senior Design 2 Task</td>
<td>Tue 12/8</td>
<td>Tue 12/12</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>10</td>
<td>Senior Design Assignments</td>
<td>Tue 12/8</td>
<td>Sun 12/10</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>11</td>
<td>Interface Design/Handoff</td>
<td>Tue 12/8</td>
<td>Fri 12/11</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>12</td>
<td>Interface Development</td>
<td>Mon 12/10</td>
<td>Wed 12/13</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>13</td>
<td>Functionality Development</td>
<td>Thu 12/13</td>
<td>Thu 12/17</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>14</td>
<td>Combine Interface and Functionality</td>
<td>Fri 12/15</td>
<td>Wed 12/17</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>15</td>
<td>Changes and Improvements</td>
<td>Thu 12/18</td>
<td>Tue 12/26</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>16</td>
<td>Senior Design 3 Task</td>
<td>Mon 1/4</td>
<td>Fri 1/8</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>17</td>
<td>Senior Design Assignments</td>
<td>Mon 1/4</td>
<td>Fri 1/8</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>18</td>
<td>Testing and Debugging</td>
<td>Mon 1/11</td>
<td>Sun 1/16</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>19</td>
<td>Obtain User Feedback</td>
<td>Mon 1/4</td>
<td>Fri 1/8</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>20</td>
<td>Make Changes</td>
<td>Mon 1/11</td>
<td>Tue 1/13</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>21</td>
<td>Test and Debug Changes</td>
<td>Wed 1/12</td>
<td>Sun 1/16</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>22</td>
<td>Create Distribution Medium</td>
<td>Mon 1/9</td>
<td>Fri 1/13</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>23</td>
<td>Test Full Production Medium</td>
<td>Mon 1/16</td>
<td>Fri 1/20</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
</tbody>
</table>

Figure A.1 – Project Time Line
Figure A.2 – Senior Design II Timeline
Figure A.3 – Senior Design III Timeline
Appendix B – Code Snippets

This section shows and explains various code snippets to demonstrate the technology and structure used in the development of this application.

Main Menu Code

This section shows the code for the main menu, which is the simplest menu in the application, but demonstrates the main functions.

```plaintext
# initialise the application (main window)
menu_initialise("Linux Server Configuration Wizard");
$top_item=0; # sets top window to zero for navigation

# ---< main menu >-----------------------------
do {
    # initialise the main menu screen
    &menu_init("Main Menu", "Select a task to perform");
    
    # create buttons for menu
    &menu_button_set(1, "Select");
    &menu_button_set(2, "Help");
    &menu_button_set(3, "Exit");
    
    # create menu items
    &menu_item("Configure Services", "configures");
    &menu_item("Monitor Services", "monitor");
    &menu_item("Other Task", "other");
    
    # use menu_display to display the menu
    chop($sel=&menu_display("Choose your option", $top_item));
    
    # based on the return value, chain to a sub-menu
    SUB_MENU: for($sel) {
        /configures/ && do { # sub for the configure menu
            &menu01;
            $top_item=0;
            last SUB_MENU;
        };
        /monitor/ && do { # sub for the monitor menu/screen
            &menu02;
            $top_item=1;
            last SUB_MENU;
        };
        /other/ && do { # sub for the other task menu
            &menu03;
            $top_item=2;
            last SUB_MENU;
        };
    };
} until($sel eq "%EMPTY%");
```
Configuration

This section uses the code to configure a NIC to demonstrate how the configuration information was used throughout the application. This section contains the code that provides the screen and the code that performs the functionality.

The first part is the code that provides the screen and checks the information entered. This code also shows the use of the regular expression to check for valid I.P. addresses.

# package information starts here
BEGIN {}
# sub for the nic configuration menu
sub nic {
    # setting $data_link variable to empty string
    # to error handle farther down
    for($i=1;$i<=3;$i++) {
        $data_nic[$i] = "";
    }

    # setting the buttons used for this menu
    &menu_button_set(1, "Edit/Change");
    &menu_button_set(2, "Help");
    &menu_button_set(3, "Accept");

    # initializing the window
    &menu_init("Edit NIC information", "Enter the needed information", "/help_files/nic.txt");

    # setting the items in the menu
    &menu_item("", "IP Address (edit)", 6, "", "25 0 0 ");
    &menu_item("", 'Netmask (edit)', 6, "", "25 0 0 ");
    &menu_item("", "Gateway Address (edit)", 6, "", "25 0 0 ");

    # displaying the menu
    chop($sel=&menu_display("Editing NIC Information"));

    # setting buttons for next screen
    &menu_button_set(1, "Continue");
    &menu_button_set(2, "");
    &menu_button_set(3, "");

    # setting test variable to 0 for farther down
    # for checking that everything is filled
    $filled=0;
# separating all entered values
@x=split(/$Cmenu::menu_sep/,$sel);
# checking to see if anything was even entered
# if not goes to error screen telling user no changes were made
if($#x==0) {
  &menu_show("NO CHANGE", "No changes were made to the NIC configuration", ERROR);
} else {
  for($i=1;$i<=$#x;$i++) {
    # breakout field label and new data
    (undef,$newdata)=split(/$Cmenu::menu_sepn/,@x[$i]);
    # sets seperated data into array to pass to functionality
    $data_nic[$i]=$newdata;
    # checking to make sure all fields were entered
    if ($data_nic[$i] eq "") {
      $filled=1;
      $data_nic[$i] = " ";
    }
  }
}

# loop to make sure that all fields entered are valid ip addresses
# checks that structure is valid not actual address
# (i.e. xxx.xxx.xxx.xxx)
for($i=1;$i<=3;$i++) {
  unless ($data_nic[$i] =~ /d{1,3}\./d{1,3}\./d{1,3}\./d{1,3}/) {
    $filled=1;
  }
}

# creating a string to display a verification message
# of what information was entered
$text= " IP Address set to <" . $data_nic[1] . " >\n";

# using that variable created above to check if information is valid
# if it is not a error message is displayed
# if it is a verification is displayed and the data sent to the
# sub-function in the package that contains the functionality
unless ($#x==0) {
  if ($filled==1) {
    # warning screen for invalid paramaters
    &menu_show("Data edited","A field was left blank or is invalid, NIC info not updated!",WARN);
  } else {
    # confirmation screen the configuration was edited and passes to
    # configuration function
    &menu_show("Data edited"," $text",WARN);
    configure_func::nic(@data_nic);
  }
}
The second part is the code that provided the functionality. This code is what populated the configuration file and made the proper system calls.

```perl
# subfunction for the functionality of nic configuration
sub nic {
    shift(@_);

    $nic_file = '/etc/sysconfig/network-scripts/ifcfg-eth0';
    open(NIC, ">$nic_file") || die "could not open file";

    print NIC "DEVICE=eth0\n";
    print NIC "BOOTPROTO=none\n";
    $nic_ip = shift(@_);
    print NIC "IPADDR= " . $nic_ip . "\n";
    $nic_mask = shift(@_);
    print NIC "NETMASK= " . $nic_mask . "\n";
    print NIC "ONBOOT= yes\n";
    print NIC "TYPE=Ethernet\n";
    print NIC "USERCTL= no\n";
    $nic_gate = shift(@_);
    print NIC "GATEWAY= " . $nic_gate . "\n";
    print NIC "IPV6INIT= no\n";

    close(NIC);

    chop($nic_net = substr(`ipcalc -n $nic_ip $nic_mask`, 8, 15));
    $broad = "ipcalc -b $nic_ip $nic_mask";
    chop($nic_broad = substr(`$broad`, 10, 15));

    dbmopen(%lscw_conf, "lscwDBM", 0666);
    $lscw_conf{"nic_set"} = 1;
    $lscw_conf{"nic_ip"} = $nic_ip;
    $lscw_conf{"nic_mask"} = $nic_mask;
    $lscw_conf{"nic_gate"} = $nic_gate;
    $lscw_conf{"nic_net"} = $nic_net;
    $lscw_conf{"nic_broad"} = $nic_broad;
    dbmclose(%lscw_conf);

    `service network restart`;
}
```
References


Linux Configuration Wizard

By

T. Brian Sabatini

Submitted to
the Faculty of the Information Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

University of Cincinnati

College of Applied Science

June 2006
Linux Configuration Wizard

By

T. Brian Sabatini

Submitted to
the Faculty of the Information Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Brian Sabatini 2006

The author grants to the Information Technology Program permission
to reproduce and distribute copies of this document in whole or in part.

_______________________________   ______________
T. Brian Sabatini                   Date

_______________________________   ______________
Professor                        Date

_______________________________   ______________
Head                              Date
Acknowledgements

I would like to thank my Senior Design technical Professor, Professor Tamisra Sanyal for all of his help and support through my entire Senior Design project. I would also like to thank the Company I work for, Sibco Building Products, for offering the flexibility and support to go to school and work on my Senior Design Project. I would also like to thank Professor Stockman, for helping give me the inspiration and ideas for this project. Lastly, I would like to thank Dr. Jo Ann Thompson for taking time to offer help with my reports for every one of my Senior Design quarters.
# Table of Contents

Acknowledgements .......................................................... i  
Table of Contents .......................................................... ii  
List of Figures .............................................................. iii  
Abstract ........................................................................ iv  

1. Statement of the Problem .............................................. 1  
2. Description of the Solution ............................................. 2  
   2.1 User Profile .............................................................. 3  
      2.1.1 Administrator Profile ........................................... 3  
   2.2 Design Protocols ..................................................... 3  
      2.2.1 Technology ....................................................... 3  
      2.2.2 Organizational Scheme ...................................... 4  
      2.2.3 Use Case Diagram ............................................ 5  
      2.2.4 User Interface ................................................. 6  
      2.2.4.1 Interface Navigation ....................................... 6  
      2.2.4.2 Interface Design .......................................... 7  
      2.2.4.3 Color Scheme ............................................. 8  
      2.2.4.4 Help .......................................................... 8  

3. Deliverables ............................................................... 9  
4. Development and Design ............................................... 9  
   4.1 Project Schedule .................................................... 9  
      4.1.1 Senior Design I ............................................... 9  
      4.1.2 Senior Design II ............................................. 10  
      4.1.3 Senior Design III .......................................... 10  
   4.2 Project Resources .................................................. 11  
   4.3 Project Budget ...................................................... 12  

5. Proof of Concepts ........................................................ 12  
   5.1 Configuration ......................................................... 12  
      5.1.1 NIC Configuration .......................................... 13  
      5.1.2 DHCP Configuration ........................................ 16  
      5.1.3 NFS Configuration ......................................... 17  
      5.1.4 DNS Configuration ......................................... 19  
   5.2 Monitor .............................................................. 19  
   5.3 Other Tasks .......................................................... 20  
      5.3.1 Manual Edit ................................................... 20  
      5.3.2 Control Services ........................................... 21  
      5.3.3 Backup and Restore ....................................... 23  
   5.4 Help ................................................................. 25  

6. Testing ........................................................................ 26  
7. Conclusions and Recommendations ................................. 27  
   7.1 Conclusions ........................................................... 27  
   7.2 Recommendations .................................................. 27  

Appendix A ................................................................. 28  
Appendix B – Code Snippets ............................................. 31  
References ..................................................................... 35
List of Figures

Figure 2.1 - Use-Case Diagram..................................................................................5
Figure 2.2 - Application Navigation/Workflow.........................................................6
Figure 2.3 - Main Menu................................................................................................7
Figure 2.4 - Service Selection....................................................................................7
Figure 2.5 - NFS Configuration..................................................................................7
Figure 5.1 – Network Card Configuration Screen....................................................14
Figure 5.2 – Network Card Configuration File..........................................................14
Figure 5.3 - Incorrect Information Prompt.................................................................15
Figure 5.4 - Prompt That No Changes Were Made To the NIC Configuration........15
Figure 5.5 – DHCP Information Entered In Application..........................................16
Figure 5.6 – DHCP Configuration File Created By Application..............................16
Figure 5.7 - Prompt That No Changes Were Made To The DHCP Configuration....17
Figure 5.8 - NFS Action Menu..................................................................................18
Figure 5.9 – NFS Share Information..........................................................................18
Figure 5.10 – NFS Share Permissions.......................................................................18
Figure 5.11 – NFS Configuration File Before and After Configuration....................18
Figure 5.12 – DNS Configuration Screen.................................................................19
Figure 5.13 – System Monitor Screen......................................................................20
Figure 5.14 – File Selection Screen..........................................................................21
Figure 5.15 – Manual Edit Screen............................................................................21
Figure 5.16 – Selecting a Service to Control..............................................................22
Figure 5.17 – Selecting an Action to Take................................................................22
Figure 5.18 – The Confirmation Screen....................................................................23
Figure 5.19 – Backup and Restore Menu................................................................24
Figure 5.20 – Backup Screen....................................................................................24
Figure 5.21 – Restore Screen....................................................................................25
Figure 5.22 – Help Screen........................................................................................26
Figure A.1 – Project Time Line..................................................................................28
Figure A.2 – Senior Design II Timeline.................................................................29
Figure A.3 – Senior Design III Timeline..................................................................30
Abstract

The Linux Server Configuration Wizard is an application designed to help reduce the difficulties associated in implementing a server with a Linux operating system. The main focus of the application is for small businesses. Small businesses usually have limited decisions regarding the selection of what I.T. resources are brought into their operation. These limitations are the result of deficiently funded budgets and the incapability to support resources in-house. The application was developed to give a small business complete control over a Linux server. With the application, several essential network services can easily be set up and configured, which can provide all the needed components for an operational network. The services that the application can be used to configure include: DHCP, DNS, Samba, and NFS. Not only does the application offer the ability to configure network services, but it also provides the ability to monitor them as well. Along with monitoring network services, other vital system statistics can be viewed. Other capabilities that the application provides are: the ability to backup and restore system configurations, the ability to control the state of network services, and the ability to manually edit system configuration files.