Table of Contents

Introduction..................................................................................3
Hardware Requirements..................................................................4
  Requirements for Client side......................................................4
  Requirements for Server side......................................................4
Introductory Page.......................................................................5
  List of files for editing ..............................................................5
  New file field ............................................................................5
  Error messages..........................................................................6
Use of .INI files.........................................................................7
  .INI attributes that you can specify .............................................7
Header file/Source Code Editor ..............................................9
  Description ............................................................................9
  Required Source Code Format .................................................10
Source Editor List.....................................................................11
  Control File Module ..............................................................11
  User Control of Routine Functions ...........................................12
  Error Detection ......................................................................12
  Module Related Options .......................................................13
  Introduction Documentation .....................................................14
Routine Editor Page..................................................................15
  Default format .......................................................................15
  “Show Documentation Only” ..................................................16
  “Show Source Only” ..............................................................17
  “Show Both” ..........................................................................17
  “Save Routine” .......................................................................17
  “Display Source Line Numbers:” .............................................18
Glossary ..................................................................................19
Introduction

The Web-based Unix Source Code Editor is an editor that allows you to modify your source code via the web. All of your source files will be saved on the UNT server. This brings you an enormous advantage of being able to access your code from anywhere that has Internet access. This SRD outlines each feature, in detail, that you have requested such as: file uploads, source code modification, viewing capabilities, and INI control. There is also a date corresponding to the last time you modified your code and basic information about each routine such as: number of lines of source, number of lines of documentation and last date modified. An INI file on the server will provide customization for the interface.
Hardware Requirements

Requirements for Client side are as follows:
A. Web Browser (i.e. Internet Explorer or Netscape) 5.0 or higher
B. Internet Service Provider
C. Modem or Network Interface Card.

Requirements for Server side are as follows:
A. Unix based web server, accessible by the user via internet.
B. Unix that supports PHP scripts to be run on the server.
C. Unix that allows the user to store source code files locally.
I. Introductory Page

When the user accesses the webpage, the introductory page is displayed first. It is split into two sections: a list of files available for editing, and a field where a new file can be created.

A. List of hyperlinked files available for editing

In this section there will be a list of files that are available for editing. The files are chosen from a specified directory in the .ini file. Only files with the following extensions will be displayed:

1. .h header files
   - Clicking on this file brings up the header file editor

2. .c, .cc, .cpp source files
   - Clicking on this file brings up the source file editor

B. Field where a new file can be named and created, accompanied by a button for submission

A field box is also displayed, where the user can specify the name and extension of a new file to create. The file is created in the same directory as in the previous section, as specified by the .ini file.
C. The following errors will trigger error messages

The following errors on the introductory page will trigger messages:

1. No valid files found
   If there are no valid files in the directory specified by the .ini file, then a message stating that there are no files present will be displayed in place of a list.

2. User enters an invalid filename or extension when creating a new file
   The user must type a filename valid in a UNIX environment and the extension must be one of those listed in section 1. The inputted filename is not created.

3. User enters a filename that already exists
   If the user types a filename and extensions that already exist, then a warning will state the file already exists.
II. Use of .INI Files

This Source Editing system controls the visual appearance and defaults through the use of .ini files. The .ini files may be edited to best suit the user. Each .ini file controls the settings for one individual project. These .ini files should be placed in the same directory as the project whose source files are being edited with this Source Editor. No .ini file is required to be in this directory. However, any .ini file placed in this directory will override the default settings built into the Source Editor.

Here are some basic .ini attributes that you may specify

A. Project Name - This field contains the name that will appear at the top of the Source Editor screens. This name may consist of any characters.

B. Filename Base – This field specifies the root directory where the source files for your project will be located.

C. Source Code Window Attributes – This field allows you to control the attributes of the browser windows where you edit your source code.
D. Background Color – This field controls the background color of the browser window. You may specify the color through one of the color names that is recognized by HTML, or you may use the RGB triplet.

E. Background Image – This field controls what image, if any, is tiled onto the background of your browser window. The image may specify an URL or a local file. If left blank, no image will be used.

F. View – This field controls the default view for working with your source code. The values can be Source, which views the source code only, Docs, which will view the documentation only, or Both, which will show both the source code and the documentation.

G. Line Width – This field controls the line width of the editing window in characters.

H. Doc Lines – This field controls how many lines will be displayed in the documentation editing window.

I. Source Lines – This field controls how many lines will be displayed in the source code editing window.

J. Intro lines – This field controls how many lines will be displayed in the introduction editing window.

K. Header Intro Lines – This field controls how many lines will be shown in the header file editing window.

III. Header File Source Code Editor
This section will detail the basic controls and a description of the header file format needed to run with our software.

**A. Description**

1. This is the name of the current project. This name may be specified in the project’s `.ini` file.
2. This is the path of the file you are editing.
3. This line displays the last time the file you are editing was updated.
4. This is the editing window. It works exactly like a simple text editor, such as Notepad or JOE.
5. This is the save button. This saves the file you are working on and saves a backup of the old file with a `.bak` extension. For example, if you saved the file as `myfile.h`, the backup would be saved as `myfile.bak`.

**B. Required Source Code Format**
In order to make it possible for our Source Editor script to recognize each individual routine and its corresponding comments inside a source module, specifically coded comments should be included at the beginning of each designated routine. These comments will have the following general appearance:

```plaintext
/*! ///////////////////////////////////////////////////
** Routine Name
** Date Modified: date or "Unknown"
** View: B or S or D
*
* Programmer-specified routine-related comment lines
*
!*/
```

Your actual source code goes here, immediately after each routine related comment for each routine.

The "Date Modified:" and "View:" lines are optional and can be left out. The "Date Modified:" line will be added whenever the current module is modified by the Source Code Editor. If the programmer modifies the view of this routine using one of the "Show Source Only", "Show Documentation Only", or "Show Both Source and Documentation" buttons, the "View:" line is added to indicate the specific visual display for this routine when it is viewed in the future. (Don Retzlaf)
IV. Source Editor List for “ini.cpp”

When the user clicks on **ini.cpp** file, a new page will open. This page contains the Source Editor Routine list for the **ini.cpp** file. This window contains three parts:

1. **Control File Module**
2. **Module Related Options**
3. **Introduction documentation for the ini.cpp file.**

A. Control File Module

In this table, the user will be able to view the following information:

1. The List of the Routine Names.
2. The number of lines of the documentation part for each specific routine.
3. The number of lines of the source code part for each specific routine.
4. The last date that each routine was updated; where the date will be unknown for the first time use.
5. Along with routine names, the user will see check boxes.
6. **Delete selected routines** button will be placed under the table.

See the figure below:

![Web-based Source Editing System](image-url)
B. User Control of Routine Functions

1. The user will be able to delete the checked routines through checking the box beside the specified routine and then click on the **Delete selected routines** button. See the following figure:

   ![Delete selected routines](image)

   **Note:** A pop out confirmation window will open for the user as soon as the user click on Delete selected routines button. This window will ask the user if he still needs to delete the selected routine or the user can click on Cancel to return back to the original window.

2. The user will be able to open a new routine in a separate window by clicking on the specified one.

C. Error Detection associated with Control File Module

The application must handle errors that might occur while the user is working with INI Control File Module. There are potentially two errors that might occur:

1. If the user doesn’t check any box and click on Delete selected routines button then a window will pop out and ask the user to check the required box and the hit on the delete button

2. If the user adds a new routine that doesn’t exist, an error message window will pop-up telling that the routine doesn’t exist
D. Module Related Options

1. Add New Routine

The user should type the name of the specified routine name in the specified text area. Then the user should click on the “Add New Routine before first selected” button.

Note: if the user decides to add the new routine in a particular place then he must select the routine that he wants to place it above and then click on “Add New Routine before first selected” button. Check the image below:

![Module-Related Options](image)

2. Rearrange Routines

The user should select the required routine that he wants to place it at the top and then click on the “Rearrange Routines to this order” button.

3. Error Detection associated with Module-Related Options Window

The application must handle errors that might occur while the user is working with the Module-Related Options Window. There is one potentially error that might occur:
a. If the user decides to add a new routine and he types a name of an existing routine, then a window will pop out which tells the user he typed a wrong name.
b. If no routines are listed in the Control File Module, then the user must add a routine

**E. Introduction Documentation for the ini.cpp file**

The window below is the third in a series of windows on the webpage interface. The characteristics of the documentation window are stated below:

![ini.cpp Documentation](image)

1. The window contains the documentation for the “ini.cpp” file.
2. The user has the ability to modify the documentation file.
3. Once the user adds the documentation for the “ini.cpp” file in the above window, the user should then click on the “Save Introduction” button to save the introduction.
4. After the file is saved, the user can always go back and modify the documentation for the “ini.cpp” file.
5. Once the file is modified, the “Save Introduction” button is clicked.
6. A backup for the file is saved.

**Note:** The user has the ability to view the date when the documentation for the “ini.cpp” file was last modified.
V. Routine Editor Page

The routine editor page generates an initial default format for each routine that the user has for editing. The default settings display both documentation and source code, both in their own text-editing area on the web page. The user will then have the option to select between two other displays, source only and documentation only. These settings will then be saved so that the next time the routine is opened it retains the last specified display settings.

A. The default format divides into three sections:

1. The first section displays:
   a. The routine name,
   b. The source filename,
   c. The project name, and
   d. The last or unknown date that modified the routine.

2. The second section displays:
   a. The documentation text box that lists any comments for the routine source code.
   b. The source text box that lists any program codes for the routine.

3. The last section displays:
   a. The “Show Documentation Only” button,
   b. The “Show Source Only” button,
   c. The “Save Routine” button,
   d. The “Display Source Line Numbers” button, and
   e. The copyright text box that lists:
      i. The project name with the latest version,
      ii. The copyright logo, and
      iii. The author name with the email address.
B. When the user clicks on the “Show Documentation Only” button in the last section of the page:

1. The page refreshes as the documentation format:
2. The first section displays same as the default format.
3. The second section displays only the documentation text box that listed any updated comments for the routine source code.
4. The last section displays buttons for:
   i. The “Show Both Documentation and Source”
   ii. The “Show Source Only”
   iii. The “Save Routine”
   iv. The “Display Source Line Numbers”
   v. The same copyright text box that is in the default format
C. When the user clicks on the “Show Source Only” button in the last section of the page

1. The page refreshes as the source format
2. The first section displays same as the default format
3. The second section displays only the source text box that listed any updated program codes for the routine.
4. The last section displays buttons for:
   i. The “Show Both Documentation and Source” button,
   ii. The “Show Documentation Only” button,
   iii. The “Save Routine” button,
   iv. The “Display Source Line Numbers” button, and
   v. The same copyright text box in the default format.

D. When the user clicks on the “Show Both Documentation and Source” button in the last section of the page

1. The page refreshes as the default format with any updated information in the documentation and source text boxes, and
2. The program saves the routine in the source file.

E. When the user clicks on the “Save Routine” button in the last section of the page

1. The page refreshes as the current format with any updated information in either the documentation, source, or both text boxes, and
2. The program saves the routine in the source file
F. When the user clicks on the “Display Source Line Numbers” button in the last section of the page

1. The page refreshes as the current format with any updated information in either the documentation, source, or both text boxes
2. The program saves the routine in the source file
3. A new window will be displayed that shows the user the source code that is being worked on with corresponding line numbers. (This is a read-only file)
4. At this time you will have two browser windows open
   i. Line numbers (read-only)
   ii. The page you are currently working with
## Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration File</td>
<td>A file that contains configuration information for a particular program.</td>
</tr>
<tr>
<td>INI-file</td>
<td>Text based data file that allows the user to control information for the Windows environment and many of its application.</td>
</tr>
<tr>
<td>Internet browser</td>
<td>An application that allows the user to browse the Internet.</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>A specific application that enables the user to view Web pages.</td>
</tr>
<tr>
<td>Internet Service Provider</td>
<td>A company that provides access to the Internet.</td>
</tr>
<tr>
<td>Log File</td>
<td>A file on the server that is a log of all the traffic on the server.</td>
</tr>
<tr>
<td>Modem</td>
<td>A modem is a device or program that enables a computer to transmit data over, for example, telephone or cable lines.</td>
</tr>
<tr>
<td>Network Interface Card</td>
<td>An expansion board you insert into a computer so the computer can be connected to a network.</td>
</tr>
<tr>
<td>Netscape</td>
<td>A specific application that enables the user to view Web pages.</td>
</tr>
<tr>
<td>Pop-out Window</td>
<td>A separate browser window that open in front of the main browser window.</td>
</tr>
<tr>
<td>Server</td>
<td>A computer or device on a network that manages network resources.</td>
</tr>
<tr>
<td>Web-based</td>
<td>Refers to the use of the Internet and Web browsers.</td>
</tr>
<tr>
<td>Web-based Application</td>
<td>An application that utilizes the features of common Web browsers to present information to a geographically disperse audience.</td>
</tr>
<tr>
<td>Web browser</td>
<td>A software application used to locate and display Web pages.</td>
</tr>
</tbody>
</table>