



## **Using Drupal: A Web Experience**

**Prepared by Web Networks  
Prepared for Association for Progressive Communications  
Author: Andre Molnar**

**Wednesday, 30 August 2006**



# Contents

- REPORT SUMMARY ..... 4**
- INTRODUCTION ..... 5**
  - AUDIENCE ..... 5
  - SCOPE ..... 5
  - STAKEHOLDERS ..... 6
  - DEFINITIONS ..... 6
  - REPUTATION FOR USABILITY: HISTORY AND BACKGROUND OF DRUPAL ..... 6
- METHOD ..... 8**
  - DEFINING COMMON ADMINISTRATIVE TASKS ..... 8
  - USABILITY GUIDELINE REVIEW ..... 8
  - STAKEHOLDER USABILITY REVIEW ..... 9
- USABILITY GUIDELINE REVIEW ..... 11**
  - GUIDELINE 1: DISTINGUISH CLEARLY AND CONSISTENTLY BETWEEN REQUIRED AND OPTIONAL DATA ENTRY FIELDS..... 12
  - GUIDELINE 2: DETECT ERRORS AUTOMATICALLY ..... 13
  - GUIDELINE 3: MINIMIZE USER DATA ENTRY ..... 14
  - GUIDELINE 4: LABEL DATA ENTRY FIELDS CLEARLY ..... 16
  - GUIDELINE 5: PUT LABELS CLOSE TO DATA ENTRY FIELDS..... 18
  - GUIDELINE 6: LABEL FORM BUTTONS CLEARLY ..... 19
  - GUIDELINE 7: ALLOW USERS TO SEE THEIR ENTERED DATA ..... 21
  - GUIDELINE 8: USE RADIO BUTTONS AND CHECKBOXES CORRECTLY ..... 23
  - GUIDELINE 9: GROUP DATA ENTRY BY METHOD TYPE ..... 24
  - GUIDELINE 10: MAINTAIN PROPER TAB INDEXING IN FORMS ..... 26
- WEB NETWORKS STAKEHOLDER REVIEW..... 27**
  - ADMINISTRATIVE TASK: CREATING CONTENT ..... 27
  - ADMINISTRATIVE TASK: EDIT EXISTING CONTENT ..... 29
  - ADMINISTRATIVE TASK: MANAGING MENUS ..... 31
  - ADMINISTRATIVE TASK: MANAGING CATEGORIES ..... 33
  - ADMINISTRATIVE TASK: MANAGING USERS AND ACCESS CONTROL ..... 35
  - ADMINISTRATIVE TASK: ATTACHING FILES AND MANAGING DIGITAL ASSETS ..... 36
- DISCUSSION, CONCLUSIONS AND FURTHER RESEARCH..... 38**
  - THE LANGUAGE OF DRUPAL ..... 38
  - IMPRESSIONS OF THE ADMINISTRATION INTERFACE ..... 38
  - SUGGESTED IMPROVEMENTS FOR THE ADMINISTRATION INTERFACE AND NEW DEVELOPMENTS ..... 39
  - CONCLUSIONS ..... 40
  - FURTHER RESEARCH ..... 41
- APPENDIX A: ADMINISTRATORS USER EXPERIENCE SURVEY NOVEMBER 2005 ..... 42**
- APPENDIX B: USABILITY EVALUATION OF CIVICSPACE INSTALLER..... 46**
- APPENDIX C: GLOSSARY..... 53**
- APPENDIX D: USABILITY GRADE SUMMARY FOR DRUPAL ..... 55**



## Document history

Author	Detail of addition/revision	Date and Time
Andre Molnar	Document Creation	08/30/06

## Further information

For further information on the information contained in this document please contact:

Name: **Dennis Robinson**  
Job Title: Operations Manager  
Address: Web Networks  
384-401 Richmond St. W.  
Toronto, Ontario  
Canada M5V 3A8

Telephone Number: 416.596.0212 x28  
Fax Number: 416.596.1374  
Email Address: **[drobinson@web.net](mailto:drobinson@web.net)**

## Acknowledgments

This document was developed with support from the UK government, Division for International Development (DFID) under the Building Communications Opportunities (BCO) program.



## Report Summary

The Drupal content management system (CMS) has gained a reputation for being a developer-centric platform that is difficult to use. While Drupal does have usability issues, the overall conclusion of this report is that the tools Drupal provides to accomplish the most common administrative tasks associated with managing a website are all usable.

This report is divided into two areas of study: a usability guideline review and the stakeholder review of Drupal usability.

The usability guideline review evaluates how well Drupal's administrative tools adhere to a subset of the Research-Based Web Design & Usability Guidelines<sup>1</sup> that are specific to web-based forms. Each of the forms Drupal provides to accomplish common administrative tasks was examined individually to determine if they:

- Distinguish clearly and consistently between required and optional data entry fields
- Detect errors automatically
- Minimize user data entry
- Label data entry fields clearly
- Place labels in close proximity to data entry fields
- Label form buttons clearly
- Allow users to see their entered data
- Use radio buttons and checkboxes correctly
- Group data entry by method type
- Maintain proper tab indexing in forms

Each Drupal form, almost without exception, adhered to all of the usability guidelines. The grades Drupal received for each guideline ranged between a 'B-' and 'A+', with an overall 'A-' average.

The stakeholder review of Drupal usability grades the usability of common administrative tools based on the experiences of Web Networks' staff and the clients they serve, taking into consideration alternate usability opinion pieces and usability surveys.

Several conclusions were made based on the stakeholder review. The first and perhaps most important conclusion was that the tools Drupal provides to complete common administrative tasks are sufficiently usable to maintain an installed website. The second conclusion was that many usability issues that have been identified in the past are regularly addressed in subsequent releases of Drupal. There is an expectation that this trend will continue.

---

<sup>1</sup> Chapter 13: Screen-based Controls of the Research-Based Web Design & Usability Guidelines  
<http://usability.gov/pdfs/guidelines.html>



## Introduction

The Association for Progressive Communications (APC) “Action Kit” project team engaged Web Networks (Web) to generate a report on the usability of the Drupal content management system (CMS).

### **Audience**

The primary audience for this report is the APC, APC member organizations and the “Action Kit” project team. Secondary audiences include organizations considering Drupal as their content management system and members of the Drupal development community who may find the report useful in identifying usability concerns.

### **Scope**

Web Networks was asked to identify and synthesize usability research on Drupal and distributions of Drupal. Web was also asked to report on its experiences in regards to developing Drupal-based sites for their clients, including supporting such clients once their sites have been deployed.

After a discussion with the “Action Kit” team, it was determined that the report should primarily focus on the usability of deployed Drupal websites from an administrator’s perspective and that Web should draw heavily on their own experiences supporting clients when making conclusions. Rather than attempt to review every administrative feature Drupal makes available to an administrator, Web narrowed the focus further to review only the administrative features most commonly used by its clients once they have assumed control of their sites.

In an effort to make this report useful to the broadest possible audience, including those in the Drupal development community, a decision was made to focus only on core features and functionality available in a non-modified Drupal 4.7 installation. There are dozens of contributed modules that are available for download that extend Drupal’s core functionality. Many of those modules are considered essential by developers because they meet common client needs. Despite this, we felt the evaluation of the additional modules was beyond the scope of this report. However, they will be considered for future documentation.

During the preparation of this report, we discovered that several usability ‘problems’ that have been identified by clients are, in fact, feature requests. Care has been taken to distinguish clearly between the usability of the tools Drupal provides and a perception of poor usability due to a feature or expected functionality that does not exist.

Many confuse user-experience and information architecture design with usability. Design for the purposes of good user-experience can enhance usability. Likewise, optimal information architecture can result in improved usability. However, all three are technically different areas of study. This report attempts to focus primarily on usability, but does consciously include some discussion about Drupal’s information architecture and user-experience related issues.



## **Stakeholders**

In conducting the usability evaluation, key stakeholders were identified to include:

- APC
- APC member organizations
- APC “Action Kit” Project Team
- Web Developers
- Web Administration Users
- Web Networks
- Web Networks’ Clients

The APC “Action Kit” project team approached Web Networks because of its ability to represent key stakeholder groups – specifically, users, support staff, web developers and organizations who use Drupal in the delivery of their online strategies.

Web Networks has been providing content management systems (integrated tools, services, and internet-based solutions) to assist and support labour groups, Unions, NGO’s, Government, and ethical businesses in delivering their missions. For over a year now, Web has focused on developing web applications based on the Drupal CMS to support its clients.

Web’s Drupal projects range in size and complexity from simple CMS deployment with a custom theme to complex projects with large organizations in alignment with custom business requirements. The largest projects have been with organizations with 250 to 300 employees, whose websites serve audiences of several thousand members. Web has specialized in the customization of Drupal creating new functionality in areas such as document management, in-line help, translation, e-learning, e-commerce, process workflows, membership management, and political advocacy, for example.

## **Definitions**

For the purpose of this report, it is important to define usability. Usability is an objective or subjective measure of a software product’s ease of use.

For a list of additional terms used in this document please refer to the glossary in Appendix C. The glossary also contains a list of common Drupal terminology.

## **Reputation for Usability: History and Background of Drupal**

Drupal was originally developed by Dries Buytaert in the year 2000. From its humble beginnings as a university web-based discussion board, Drupal has grown to become a full-grown, feature-rich content management system which has been adopted by several high-profile organizations. Drupal is highly regarded by developers because of its extensible architecture and robust source code. Many developers also admire the scalability of Drupal and its ability to handle high traffic loads with features such as page caching and component/feature throttling.

Because of its extensibility, developers often view Drupal as a development platform that happens to have CMS functionality. This popularity among developers has given Drupal a



reputation of being a CMS by developers for developers. Of the creators' own admission, Drupal has a reputation of being a developer-centric system that is difficult to use<sup>2</sup>; in the past many developers in the Drupal community have been less than apologetic for Drupal's apparent disregard for usability.

Indeed, many peoples' first impressions of Drupal have been that it is difficult to install and initially difficult to use. My own first impression of Drupal 4.4 from a user and usability perspective (which now spans over two years and three versions), was that there was much room for improvement. However, despite its reputation, every new release of Drupal addresses usability issues previously identified in earlier releases, and there is evidence within the Drupal development community of a growing focus on usability.

---

<sup>2</sup> <http://buytaert.net/ockhams-razor-principle-of-content-management-systems>

## Method

### ***Defining Common Administrative Tasks***

Web Networks identified six common administrative tasks that their clients perform on a regular basis to maintain their Drupal-based websites.

These tasks were identified as:

- Adding new content to the website
- Editing existing content on the website
- Managing website navigation menus
- Categorizing content
- Managing users, roles and access control (for the purposes of delegating administrative tasks)
- Attaching digital assets to content pages

This data was obtained by reviewing the functional requirements of all the Drupal-based projects in which Web Networks has been involved. An interview of Web Networks' Drupal trainer also confirmed that clients were most interested in learning these tasks during training sessions.

While one survey we reviewed had identified "Administering the style, layout, or presentation of the site" as a common administrative task, it was not included in this report because style and presentation are primarily 'theme' development tasks. Theme development is not a common task performed by Web Networks' clients and, consequently, was deemed to fall outside the scope of the current evaluation.

### ***Usability Guideline Review***

We conducted a review of the tools that come with a standard Drupal installation in order to determine whether they adhere to common usability guidelines. This was done in the hopes of establishing a baseline to compare this review with the more subjective stakeholder usability review. For example, if no common issues were identified in the stakeholder review, it may be due to users overcoming obstacles but failing to report them.

Once we decided to conduct a guideline review, we had to choose a set of usability guidelines to gauge our results by. We chose the Research-Based Web Design & Usability Guidelines, developed by the Communication Technologies Branch (CTB) of the [U.S.] National Cancer Institute (NCI) in the U.S. Department of Health and Human Services.<sup>3</sup> This standards document contains several hundred web design and usability guidelines that were gathered from research done in the field of usability. Each guideline is ranked by 'relative importance' and 'strength of evidence'. The 'strength of evidence' ratings and clear citations were ultimately what led us to choose this guideline over others for our

---

<sup>3</sup> <http://usability.gov/pdfs/guidelines.html>

examination of Drupal. The guide covered a broad range of usability and design issues, but we focused primarily on those found in Chapter 13: 'Screen-based Controls (Widgets)'.

Most web developers would identify these as forms since they are built using HTML form elements. Most content management systems' administration interfaces are a collection of web-based HTML forms that allow users to accomplish various tasks. Drupal is no different in this respect. Users must interact with at least one form to accomplish each of the common administrative tasks identified in this report.

In total, the Research-Based Web Design & Usability Guidelines includes twenty-five (25) guidelines for screen-based controls. Some of these guidelines were not reviewed because the 'strength of evidence' supporting them was low. Others were excluded simply because they did not apply (e.g. Label Units of Measurement). In some cases, guidelines that were sufficiently similar were combined into a single guideline for the review.

The following is the final subset of guidelines reviewed:

- Distinguish clearly and consistently between required and optional data entry fields
- Detect errors automatically
- Minimize user data entry
- Label data entry fields clearly
- Put labels close to data entry fields
- Label form buttons clearly
- Allow users to see their entered data
- Correct use of radio buttons and checkboxes
- Group data entry by method type
- Maintain proper tab indexing in forms

Each guideline review took into consideration all of the forms provided to accomplish the common administrative tasks identified in this report. In most cases the general tendency of forms were reported, but in some cases each of the forms was reported on separately.

Drupal's adherence to the guidelines were awarded letter grades determined by asking the following set of questions:

- Was the guideline followed? (If not, a failing grade would be assigned)
- Was the guideline followed completely or partially?
- Was the guideline followed consistently in the forms related to the common administrative tasks?
- Was the guideline followed consistently throughout all of Drupal's administrative pages?

An 'A+' represented the top grade. An 'F' represented a failing grade.

### ***Stakeholder Usability Review***

In order to prepare the stakeholder usability report, Web had to look both externally and internally. The external portion involved examining existing studies, surveys and opinion



pieces on the usability of Drupal. The internal portion involved interviewing support, training and development staff at Web Networks and client communication.

External information was gathered over a period of two weeks and included several discussions about usability of the Drupal developer and documentation mailing lists<sup>4</sup>, Drupal forums<sup>5</sup>, and individual usability opinion pieces published on various websites<sup>6</sup>. Special attention was paid to discussions of the ongoing work of Civicspace Labs in the area of usability.

The interviews of Web's staff were conducted over a period of two days at Web Networks' offices. The interviews were structured discussions with questions focused on each interviewee's experiences with clients, with special attention being paid to the different phases of a typical project. Interviewees were also asked about: specific projects and their functional requirements, specific training sessions, client support calls, and post-launch communications for each project. When appropriate during the discussions, staff members were asked to recount specific user experience feedback. If the interviewees struggled, some leading questions like 'did client X ever ask about feature Y' were posed to help stimulate the discussion.

Once the external and internal information-gathering was complete, the resources were grouped into two areas: those that related to the common administrative tasks identified in this report, and those related to other tasks. Information on the usability of contributed modules, no matter how commonly used, was ignored. In addition, information on the usability of administrative tools outside the scope of this report was not used. An analysis of the resources was then conducted to help grade Drupal's administrative tools based on the following questions:

- How easily could our clients find the tools used to accomplish each task?
- Did clients understand the tools and forms?
- How complex were the forms that clients had to use to complete each task?
- Did clients have difficulty with any options available on the forms?
- What problems did clients encounter completing common tasks?
- Were the problems common? Between forms? Between clients?

Web also looked at problems identified by others researching the topic of Drupal usability, and whether Web's experiences with clients were consistent with those findings. Usability issues identified by others in the past were examined to determine whether they have been addressed in the current release of Drupal, or were being addressed in the upcoming release of Drupal. Finally, where appropriate, we looked at how easily usability obstacles were overcome by users and whether the obstacles represented real system-wide problems or were just quirks.

---

<sup>4</sup> Drupal Documentation mailing list archives <http://lists.drupal.org/archives/documentation/>  
Drupal Development mailing list archives <http://lists.drupal.org/archives/development/>

<sup>5</sup> Drupal Usability Feedback Forum <http://drupal.org/forum/6>

<sup>6</sup> See Appendix D: Further Reading



## Usability Guideline Review

Each guideline review took into consideration all of the forms used to accomplish the common administrative tasks identified in this report. In most cases the general tendency of forms were reported, but in some cases each of the forms was reported on separately.

The rendering or 'look' of the administrative pages in Drupal is dependent on the 'theme' installed on the site. For the purposes of this report, however, we have reviewed the default theme 'blue marine'. Where appropriate, we have tried to identify features that could be changed by a theme developer.



***Guideline 1: Distinguish clearly and consistently between required and optional data entry fields***

**Grade: B-**

**Reasoning for Guideline**

Users should be able to easily identify fields that are required from those that exist to reduce the likelihood of a form being submitted and returning an error.

**Drupal Review**

Required fields in Drupal are identified by a red asterisk. However, there is no text to indicate that fields designated with an asterisk are required. This is consistent across all administrative forms (not only those forms used to complete common administrative tasks).

When a form is submitted without having all required fields populated by user input, text appears at the top of the page indicating the missing fields.

The default site theme will also highlight the required form fields with a red border if left empty.

Despite the lack of text explaining that fields marked with asterisks are required, users are quick to discover their meaning the first time they miss a required field. Site developers or administrators also have the ability to provide text that will appear on every content creation and editing form by changing a setting for each content type via administer >> settings >> content types.

**Recommendations**

Include text that indicates that fields marked with an asterisk are required.

## **Guideline 2: Detect errors automatically**

**Grade: B+**

### **Reasoning for Guideline**

Users should not be expected to make error-free entries. It is the responsibility of the developer to anticipate potential user errors and display appropriate error messages.

### **Drupal Review**

As discussed in the previous section (see: *Guideline: Distinguish clearly and consistently between required and optional data entry fields*), the Drupal forms automatically detect if required fields are missing and display appropriate error messages indicating which fields must be completed.

Other potential errors are also caught when forms are passed through validation functions in the Drupal code base. Some examples include: not allowing two pages of content to share the same path alias (path module), and warning users that two menu items point to the same page of content (menu module).

Validation functions appear in all of the modules related to the common administrative tasks.

Error detection is module-dependent. Error messages will be displayed on the screen but the way the errors are displayed will depend on the theme of the site.

Generally speaking, the error messages provided are sufficient to indicate the nature of the problem.

### **Recommendations**

Consider providing a link to additional help besides error messages, or provide an option for more verbose error reporting.

### **Guideline 3: Minimize user data entry**

**Grade: A**

#### **Reasoning for Guideline**

Requiring re-entry of data imposes additional tasks on a user. Data entry errors should not result in users having to re-enter fields that did not contain errors. Ideally, users will be required to make as few entries as possible.

#### **Drupal Review**

##### **General Observations**

In the event of errors during form submission, no forms require users to re-enter data into fields that were completed correctly. It is evident that attention has been paid to providing default values in form fields to reduce repetitive data entry tasks.

##### **Content Creation**

###### **Form Location(s): Primary 'Navigation' menu under 'create content'**

Some form fields (such as those for comment settings and publishing options) are pre-populated based on settings users can configure for each of the available content types (via 'administer >> settings >> content types').

The 'author' field also defaults to the current user and the 'date authored' field defaults to the current date/time. This is the case for all content types.

##### **Editing Existing Content**

###### **Form Location(s): Primary 'Navigation' menu under 'administer >> content >> edit'**

When editing content, all fields are pre-populated based on the last saved settings for the content page. There are no fields that would require a user to re-enter information prior to re-saving the form.

##### **Menu Administration**

###### **Form Location(s): Primary 'Navigation' menu under 'administer >> menus'**

**Also integrated in content creation and editing found in the primary 'Navigation' menu under 'create content' or 'administer >> content'**

When creating menu items, a default value for weight and parent menu are provided.

##### **Categorization of Content**

###### **Form Location(s): Integrated in content creation and editing found in the primary 'Navigation' menu under 'create content' or 'administer >> content'**

If the vocabulary associated with a content type is designated as a 'free tagging' vocabulary, users are provided with an auto-complete form that will search for existing categories that match the users keystrokes as they enter a category name. If a match is found at any time, a user may click the auto-completed value and stop typing.

For all other types of vocabularies the user may make a selection from a list of terms.



### **Users, Roles and Access Control**

**Form location(s):** Primary 'Navigation' menu under 'administer >> access control' and 'administer >> users'

There are very few fields to be entered and none that could be pre-populated.

### **Attaching Files**

**Form locations(s):** Integrated in content creation and editing found in the primary 'Navigation' menu under 'create content' or 'administer >> content'

When a file has been uploaded via the 'attach file' interface in the content creation forms, by default the file will be 'listed'. In other words, the user is not required to click the 'list' checkbox after they have uploaded the file.

### **Recommendations**

None

#### **Guideline 4: Label data entry fields clearly**

**Grade: B**

##### **Reasoning for Guideline**

Users should never have to guess what information should be entered into any field. Labels should be easy to understand, unambiguous and jargon-free.

##### **Drupal Review**

###### **General Observations**

While the language and terminology used by Drupal can sometimes be difficult for users to grasp, the administration interface and forms that administrators must use to complete common tasks are free of most Drupal jargon. All fields are labelled on all forms. Most fields are easy to understand and are for the most part unambiguous. Where there might be some confusion, additional descriptive text is provided beneath the field. As discussed in an earlier section, required fields are identified by asterisks but the asterisks themselves are not explained.

###### **Content Creation**

###### **Form Location(s): Primary 'Navigation' menu under 'create content'**

All fields are clearly labelled and easy to understand.

###### **Editing Existing Content**

###### **Form Location(s): Primary 'Navigation' menu under 'administer >> content >> edit'**

All fields are clearly labelled and easy to understand.

###### **Menu Administration**

###### **Form Location(s): Primary 'Navigation' menu under 'administer >> menus'**

**Also integrated in content creation and editing found in the primary 'Navigation' menu under 'create content' or 'administer >> content'**

All fields are clearly labelled and easy to understand.

###### **Categorization of Content (Creating Vocabularies and Terms)**

###### **Form Location(s): Primary 'Navigation' under 'administer >> categories >> add vocabulary' and 'administer >> categories >> add term'**

**Also integrated in content creation and editing found in the primary 'Navigation' menu under 'create content' or 'administer >> content'**

All fields are clearly labelled.

The purpose of some fields (e.g. hierarchy) found within the vocabulary creation form are somewhat ambiguous, and the descriptive text provided does not provide clarity. A group of checkboxes and their purpose is also somewhat ambiguous. Some work could be done to combine the text found on the main categories administration page with the text found on this form.



### **Users, Roles and Access Control**

**Form location(s):** Primary 'Navigation' menu under 'administer >> access control' and 'administer >> users'

All fields are clearly labelled and easy to understand.

### **Attaching Files**

**Form location(s):** Integrated in content creation and editing found in the primary 'Navigation' menu under 'create content' or 'administer >> content'

All fields are labelled.

The form that is displayed to the user once a file has been uploaded is unclear. The purpose of the 'delete' and 'list' checkboxes is not adequately explained.

### **Recommendations**

All form elements are clearly labelled, but some forms (e.g. vocabulary creation) could use additional text to explain the purpose of some of the fields that they contain. Text explaining that fields with asterisks are required would also help.



### ***Guideline 5: Put labels close to data entry fields***

**Grade: A+**

#### **Reasoning for Guideline**

Users should be able to quickly identify which field a label refers to.

#### **Drupal Review**

The rendering or 'look' of the administrative pages in Drupal are dependent on the theme installed on the site. Theme developers have the ability to change how forms and form elements look, but most do not drastically change the placement of field labels relative to the fields they describe.

Without exception, in the default Drupal theme, field labels for all form elements in the default theme are directly above the fields to which they belong. This provides users with a consistent, close-proximity labelling standard.

#### **Recommendations**

No recommendations at this time.

## **Guideline 6: Label Form Buttons Clearly**

**Grade: A**

### **Reasoning for Guideline**

Users should have a clear understanding of what action can be expected when clicking a button.

### **Drupal Review**

#### **General Observations**

Most forms conform to Drupal's own button-naming convention. Most actions that will result in a form being submitted and saved provide a button labelled 'submit'. Any action resulting in a content item or system setting being deleted is labelled 'delete'.

In general, form buttons are clearly labelled and describe the action that will be taken upon clicking them.

#### **Content Creation**

##### **Form Location(s): Primary 'Navigation' menu under 'create content'**

Two buttons available by default on content creation and editing forms are 'preview' and 'submit'. While it may be clear that the 'submit' button will submit a form, it is unclear whether that submission will complete the content creation process.

Pressing the 'preview' button results in the content that is being created to be 'previewed' on the screen.

#### **Editing Existing Content**

##### **Form Location(s): Primary 'Navigation' menu under 'administer >> content >> edit'**

The content editing forms provide both the 'submit' and 'preview' buttons as well as a 'delete' button.

#### **Menu Administration**

##### **Form Location(s): Primary 'Navigation' menu under 'administer >> menus'**

**Also integrated in content creation and editing found in the primary 'Navigation' menu under 'create content' or 'administer >> content'**

Buttons on menu or menu item creation and editing forms maintain the standard submit, delete naming convention.

#### **Categorization of Content**

##### **Form Location(s): Integrated in content creation and editing found in the primary 'Navigation' menu under 'create content' or 'administer >> content'**

Buttons on the vocabulary or term creation or editing forms maintain the standard submit, delete naming convention.



## **Users, Roles and Access Control**

**Form location(s):** Primary 'Navigation' menu under 'administer >> access control' and 'administer >> users'

The add user form has a button called 'create new user'. This is very clear.

The add role form has a button called 'add role'. This is likewise very clear.

The access control form provides a button labelled 'save permissions'.

## **Attaching Files**

**Form location(s):** Integrated in content creation and editing found in the primary 'Navigation' menu under 'create content' or 'administer >> content'

The interface for attaching files provides two buttons. One is named 'browse' and the other is named 'attach'. There is no button to 'delete' an attached file.

The only potential confusion of the 'attach' button is whether the post to which the file is being attached is saved during the attachment process.

## **Recommendations**

Consider changing the labels of buttons currently labelled as 'submit'. In the case of content creation forms, consider providing a variable button label that would change based on publishing options. For example, if the content being created was flagged to be published to the front page, providing a button that is labelled 'save and publish to front page' would be helpful.

## **Guideline 7: Allow users to see their entered data**

**Grade: B-**

### **Reasoning for Guideline**

When completing a form, users should be able to see as much of their entered data as possible without needing to scroll. Data should be easy to locate, and it is difficult to do so if it is no longer visible on the screen. Whenever possible, the entire form should fit within one screen. Single-line text fields should be wide enough to accommodate entries of a reasonable length. Multi-line text areas should provide as much space as possible for data entry.

### **Drupal Review**

The size and appearance of data entry fields are ultimately determined by the theme enabled on the site. However, in the default theme, nearly all data entry fields provide a sensible amount of space for users to enter the requested information, while allowing visibility of this information at the same time (e.g. a title field is wide enough to enter a title of reasonable length and not have part of it hidden). The text areas provided for entering the body of content in a post include dragable corners so that the space can grow to provide more space as needed. This is particularly helpful when editing large bodies of text so that the editor is able to establish the context of the section of text they are editing by having more of the text visible within the text area field.

Drupal 4.7 introduced forms with 'hidden' or 'collapsed' field sets. Depending on your perspective, this enhances or decreases usability of forms. Both perspectives are related to this guideline. On the one hand, collapsing certain sets of fields reduces the amount of space required to render a form and therefore makes it possible for very long forms to be displayed without forcing the user to scroll. On the other hand, fields are hidden from view. If a user opens a collapsed field set and enters data, and then re-collapses the field set, the data they entered will be hidden from view.

While a grade of B- was given, it should be noted that Drupal 4.7 is much improved over Drupal 4.6 as it relates to this guideline. The improvement is in large part due to the introduction of expandable text area fields and collapsible field sets. However, as noted above, collapsible field sets pose potential problems.

### **Recommendations**

Earl Miles, who has been working on usability enhancements for the overall administrative interface of Drupal and Civicspace, recommends that in certain cases collapsed field sets be replaced by tabs<sup>7</sup>. Moshe Weitzman recommends keeping the collapsed field sets, but having them show more information than just their title, such as the value of the fields hidden from view<sup>8</sup>. We tend to agree that additional information about the field set should be visible in its collapsed state. In the cases where the collapsed state with the field values visible provides

---

<sup>7</sup> Earl Miles Angry Donuts [http://www.angrydonuts.com/another\\_attempt\\_at\\_administrativ](http://www.angrydonuts.com/another_attempt_at_administrativ)

<sup>8</sup> Moshe Weitzman [http://www.angrydonuts.com/another\\_attempt\\_at\\_administrativ#comment-99](http://www.angrydonuts.com/another_attempt_at_administrativ#comment-99)



no space savings, the collapsed state should be removed. In other cases, such as 'publishing options' or 'authoring information', space could be saved while keeping the values of fields visible.



## ***Guideline 8: Use radio buttons and checkboxes correctly***

**Grade: A**

### **Reasoning for Guideline**

Radio buttons should only ever be used for mutually exclusive selections. Checkboxes should only ever be used for options where more than one option may be selected. Binary options should use two radio buttons, one for each of the states. If used properly, these elements limit the user's ability to make data entry errors or provide input that is unexpected (e.g. more than one option where only one option is expected).

### **Drupal Review**

All forms in the Drupal administration interface use both checkboxes and radio buttons correctly. The only deviation from the guideline among the common administrative tasks is in the menu item creation or form editing which provides an 'expanded' option (whether the menu is expanded by default or not), but provides this option via a single checkbox instead of two radio buttons.

### **Recommendations**

The guideline suggesting the use of two radio buttons for binary options is primarily based on expert opinion rather than research evidence. However, a minor change to the menu item creation form to use two radio buttons rather than a single checkbox is a reasonable request.

## **Guideline 9: Group data entry by method type**

**Grade: A**

### **Reasoning for Guideline**

The ordering of fields in a form makes a great deal of difference to users, but it is something that is often overlooked. Requiring users to continually change between data entry actions that require keystrokes and those that require mouse clicks can substantially slow their data entry speed.

### **Drupal Review**

#### **General Observations**

The forms Drupal provides to complete common administrative tasks do a very good job of grouping data entry methods together so that user are not required to switch back and forth between typing on the keyboard and clicking with a mouse (e.g. entering text and then clicking radio buttons and then returning to entering text). It is obvious that there has been attention paid to this detail.

### **Content Creation**

#### **Form Location(s): Primary 'Navigation' menu under 'create content'**

The title and body fields of the content creation forms are the first fields visible to a user. A user can enter the title and body of a post and only have to reach for the mouse once to submit their post. Other optional elements hidden behind the collapsible headings also group text entry fields together followed by actions that require mouse clicks (within their own context). Exposing the 'hidden' fields requires a mouse click, but this action can take place after the major data entry portion of content creation is complete.

### **Editing Existing Content**

#### **Form Location(s): Primary 'Navigation' menu under 'administer >> content >> edit'**

See content creation.

### **Menu Administration**

#### **Form Location(s): Primary 'Navigation' menu under 'administer >> menus'**

**Also integrated in content creation and editing found in the primary 'Navigation' menu under 'create content' or 'administer >> content'**

There are six fields in the menu item creation or form editing areas. Three require keystrokes and three require mouse clicks. The keystroke items come first in a logical sequence followed by items requiring mouse clicks, with the final click reserved for the 'submit' button.

### **Categorization of Content**

#### **Form Location(s): Integrated in content creation and editing found in the primary 'Navigation' menu under 'create content' or 'administer >> content'**

Fields requiring keystrokes are grouped together ahead of fields requiring mouse actions.



### **Users, Roles and Access Control**

**Form location(s):** Primary 'Navigation' menu under 'administer >> access control' and 'administer >> users'

Fields requiring keystrokes are grouped together ahead of fields requiring mouse actions.

### **Attaching Files**

**Form locations(s):** Integrated in content creation and editing found in the primary 'Navigation' menu under 'create content' or 'administer >> content'

Attaching files is part of the content creation form. The form elements provided for attaching files primarily require mouse clicks. These fields are displayed after other fields that primarily require keystrokes.



## ***Guideline 10: Maintain Proper Tab Indexing in Forms***

**Grade: A**

### **Reasoning for Guideline**

When a user presses the tab key, focus should shift to the next available form element. Pressing the tab key should never bounce a user around a form.

### **Drupal Review**

This guideline should go without saying, but it is included to state explicitly that all forms in the administration area of Drupal do maintain a proper tab index.



## Web Networks Stakeholder Review

### *Administrative Task: Creating content*

**Grade: A**

### **Drupal Review**

#### **Pre-configuration**

In most sites deployed by Web Networks, the types of content that may be published are well defined before the site is delivered to the client. For example, a client may require the ability to add basic pages of content to their site, some time-based events, and perhaps some specialized content such as a course description. Web Networks often pre-configures content-type choices so that they are available to the client when they take control of their site.

#### **Ease of Use**

In any event, when the client opens the administration interface and clicks the ‘add content’ menu item, they are presented with a choice of content types. Every client, being familiar with which type of content they want to add, easily finds the option they are interested in and has little difficulty entering content via the web-based forms. In most cases, the clients express how easy it is to add content.

The results of the 2005 user experience survey also strongly indicates that content publishing is regarded as easy.

<b>Percentage of Respondents<sup>9</sup></b>	<b>Rating</b>
78%	<i>Easy / Very Easy</i>
17%	<i>Somewhat Easy</i>
3%	<i>Difficult</i>

Web’s clients have not had any difficulty understanding the ‘publishing’ options available to them when they are creating content. Some of those options include, ‘publish’, ‘promote content to the front page’, and ‘sticky at top of lists’. There is sometimes hesitation with the options ‘create revision’, and ‘in moderation queue’, but when these options are explained clients often respond that the features are nice, but not likely to be used very often. Finally, the ‘input formats’ options are easily understood. For example, the explanations of ‘filtered HTML’, ‘full HTML’ and ‘PHP code’ within the interface are sufficient. Web clients are increasingly requesting a WYSIWYG editor in order to avoid adding HTML elements.

<sup>9</sup> See Appendix A Administrators User Experience Survey November 2005  
Page 27 of 55

### Creating Drupal Content

It has been suggested that in previous versions of Drupal, some of the content creation forms were not user-friendly because of the large number of options displayed to the user in lengthy detail (requiring vertical scrolling). This issue has been addressed in the latest version of Drupal by the introduction of collapsible form element containers that 'hide' options from the user unless they click to reveal them. These collapsible containers do indeed reduce clutter on the screen which, in our opinion, is a user experience enhancement rather than a usability improvement. From a usability perspective, having to reveal hidden options adds to the number of steps required to complete a task. We do not, however, have any evidence at this time to suggest that the new forms are any more or less usable, but do admit that the aesthetics of the forms are much improved.

### Enabling Additional Content Types

While Web Networks pre-configures content types for clients, by default, Drupal only has two content types enabled: 'page' and 'story'. Additional content types can be made available to a user by enabling content type modules or 'node modules'. Some content modules available in a standard installation of Drupal include: community blogs, forum posting, hierarchical collaborative books, and polls. Currently, there are also thirty-seven (37) additional contributed content modules available for download from the drupal.org website. The user experience survey of 2005 indicated that 60% of administrators found installing modules to be easy or very easy.

<b>Percentage of Respondents</b> <sup>10</sup>	<b>Rating / Comment</b>
60%	<i>Installing modules easy or very easy</i>
54%	<i>Very easy to enable module</i>
35%	<i>Easy to enable module</i>
9%	<i>Difficult to install module</i>
1%	<i>Difficult to enable module</i>

### Recommendations

Implement new module installation system to reduce the number of manual steps required by a user to install a content type modules, particularly those that require the creation of additional tables in the database.

\*Note: It appears as though an improved module installer will be available in Drupal 4.8/5.0.

<sup>10</sup> See Appendix A Administrators User Experience Survey November 2005  
Page 28 of 55



## **Administrative Task: Edit Existing Content**

**Grade: A**

### **Drupal Review**

Web Networks has worked on several projects where we have been required to pre-populate a client's website either by pre-entering some content or importing large quantities of data from legacy CMS applications. For this reason, Web has had the opportunity to witness clients interacting with the content editing controls before the launch of their website.

Since the actual user interface of the content editing form is the same as the content creation form, users are immediately accustomed to the interface and generally find it easy to use.

In earlier versions of Drupal, when content was retrieved into the form, it was sometimes discovered that the form's text area was too small to edit very large bodies of text. The default size of the text area is determined by the theme or 'look' of the theme applied to the site. In Drupal 4.6 based sites, the solution has been to adjust the theme to allow for larger text areas, but Drupal 4.7 has addressed this concern by including a JavaScript tool that allows the user to dynamically grow or shrink the size of the text area by dragging its bottom right corner. (See review of usability: *Guideline: Allow users to see their entered data*).

### **Finding where to edit content**

There are two ways in which a user can find the content they wish to edit. One is via the 'administer >> content' menu item where a user is presented with a list of existing content. The other is by browsing the website while logged in as the administrator and viewing pages. When logged in with 'administer nodes' privileges, an 'edit' tab is visible at the top of the content page. Clicking on the 'edit' tab opens the content editing form.

Users show a clear preference to editing content simply by browsing the site. Since they are familiar with their own site architecture, they easily find the content they wish to edit. In situations where the administrator wishes to edit content that is entered but not yet 'published', they are required to use the 'administer >> content' menu item.

Drupal provides a way of filtering the content listed on the 'administer >> content' page. Content may be filtered by status (published, unpublished, promoted to front page, etc.), content type (e.g. book page), and category if categories are used on the site. These filtering options prove to be very useful for clients with a large number of content pages on a site.

While this filtering option is clearly presented at the top of the 'administer >> content' page under the heading 'show only items where...', we have discovered that users often overlook the feature. We are not certain how this feature could be more clear, but clients have asked, "how do I find and edit an unpublished page?" When given the answer, they are usually surprised to find the tool in clear sight.



If the Drupal search module is installed on a site, a search tab is presented in the 'administer >> content' page. The search module indexes a Drupal site and allows users to perform keyword queries. The search module then displays the result to the users in ranked order. The functionality of the search tools are familiar, easy-to-use, and return decent results. This functionality is very useful for finding 'lost' content.

## ***Administrative task: Managing menus***

**Grade: B-**

### **Drupal Review**

Compared to creating and editing content, menu management has historically presented more usability challenges. Users do not typically find it difficult to find the menu management tools. They are either found directly inside of content creation forms, or via the 'administer >> menus' menu item in the administration menu. Generally speaking, users do not find the interface for adding, editing or deleting menus to be difficult to understand, but there is some room for improvement.

### **Adding new menu items**

In the latest version of Drupal, a content page can be added to a menu as the content is being created, by specifying the desired menu item name and parent menu directly inside of the content creation form. This feature is a great improvement over the menu item creation options available in version 4.6. Users of Drupal 4.6 could still add menu items, but were required to engage in a more cumbersome manual process that required they know the destination or internal site 'path' of the menu item they were creating before creating it. There was no easy way to find or browse for the destination path once the user had started to 'add' the menu item. The ability to manually add menu items from the menu administration page still exists in Drupal 4.7, but can be avoided.

A remaining usability concern with the administration of menu items becomes apparent to users when they attempt to add an item or move a menu item in an already large menu hierarchy. When adding or editing a menu item from the menu administration interface ('administer >> menus') or the content creation form, one of the required tasks is to choose the item's place in the menu hierarchy by choosing its parent menu item. The only interface element provided to do this is a dropdown list containing all existing menu items. When this list is long, it is extremely difficult to locate the desired parent item.

### **Creating new menus**

Another issue related to menu management is that it is not clear why newly created 'custom' menus (ones created and defined by the user) do not immediately appear in one of the navigation areas (e.g. one of the side columns). A user is required to go to the 'block' administration page and enable a block that contains the newly created menu. This is documented if you click on the help link on the menu administration page, but is not entirely intuitive.

Of the common administrative tasks, the 2005 user experience survey showed content structuring to be the area that needed the most improvement. Administering of weights (the order in which lists of content pages appear), and managing parent-child relationships in menus was rated as follows by respondents:

<b>Percentage of Respondents</b> <sup>11</sup>	<b>Rating / Comment</b>
9%	<i>Very Easy</i>
26%	<i>Easy</i>
31%	<i>Somewhat Easy</i>
26%	<i>Difficult / Very Difficult</i>

It must be noted that these results were based on Drupal version 4.6 or earlier and that we have found the ability to add content to a menu from the content creation pages in the most recent version.

There is continuing work being done to address menu usability issues for the next release of Drupal. There is currently a developer participating in the Google summer of code project that is creating an AJAX tool to help make structuring of menu trees easier.

### **Recommendations**

Implementing JavaScript 'drag and drop' functionality to assist in the ordering of large menu hierarchies should be a top priority.

Consider including a link to the 'blocks' administration interface (currently found in the administration menu under 'administer >> blocks') from within the menu administration page to make it easier for users to find where they can enable their newly created custom menus.

---

<sup>11</sup> See Appendix A Administrators User Experience Survey November 2005  
Page 32 of 55



## ***Administrative Task: Managing categories***

**Grade: C**

### **Drupal Review**

There are two tasks related to categorizing content. The first is to create the categories and their terms, and the second is to associate content with those categories.

The user interface for creating a vocabulary of category terms is easily found by users in the administration menu under 'administer categories'.

The usability of the form used to create new vocabularies from a usability guideline perspective is technically no less usable than the form used to create new content. However, understanding the language of the category system (the taxonomy module) and understanding the options available when creating vocabularies requires some learning on the part of the user. The latest version of Drupal has much improved inline documentation to assist users with the task of adding vocabularies and terms compared to previous releases. There are always calls for improved documentation, but the latest version makes understanding the basics of the taxonomy system and vocabulary types much easier without 'dumbing it down' or downplaying the power of the system.

### **Using Vocabularies and Taxonomies**

Web Networks tends to create vocabularies for our clients prior to delivering sites to them. By doing so, we also pre-configure vocabulary to content-type relationships. For this reason, our clients have been quick to learn how to add terms to an existing vocabulary without having to fully understand the taxonomy system beforehand.

Once a vocabulary is created and associated with a content type, a dropdown list of terms in that vocabulary is displayed to a user when they create or edit that content type. Making a piece of content belong to a category becomes just another option when creating content.

The introduction of 'free tagging' in Drupal 4.7 has made categorization even easier and enhanced usability. If a vocabulary is created and configured to allow 'free tagging', the process of creating terms and associating content with a term is combined in the content creation interface. When a user creates content a free tagging field is made available. If the user begins to type a term into the field, the AJAX auto-complete functionality will attempt to suggest an existing term from the vocabulary (but will not keep the user from adding a completely new term). Once the content form is submitted, the term and the association with the new content is stored. What was a two-step process can now be completed in one.

### **Taxonomy paths**

The taxonomy module provides a way to display lists of content grouped by category by browsing to a specific Drupal path. For example, to display a page on your site that had all published content that belonged to the category 'dog', you would need to enter the URL <http://example.com/taxonomy/term/1> (where example.com is your site's domain and '1' is the



term id number for the term 'dog'). If you wanted to include content from multiple categories you might use the path 'taxonomy/term/1,3,12'.

As one might see from the examples above, an administrator is required to know the term ids for categories entered via the categories administration interface. However, the term ids are not displayed anywhere on the 'administer >> categories' page.

This feature, though widely used by advanced site administrators, is entirely un-documented in the administration interface of Drupal 4.7. The help page found in the categories interface suggests installing the contributed 'taxonomy browser' module to "build a custom view of your categories". One must ask why a contributed module is required for such a common task.

If an administrator wants to add a link to a page containing a list of content related to a category or a custom menu, they would need to use the cumbersome taxonomy path as the menu item path. This path would ultimately be displayed to a site visitor if they clicked on the menu item. More often than not, administrators opt to create 'path aliases' (more human readable paths) to hide these types of awkward Drupal system paths. This introduces yet another task to accomplish a goal common to many administrators.

Typically, Web Networks pre-configures pages listing categorized content (along with their path aliases) for clients simply because it is difficult enough for experienced users, let alone first-time administrators.

### **Recommendations**

If the help files recommend that the 'taxonomy browser' tool be used to assist with 'building custom views of categories', the module (or similar module) should be included in Drupal by default.



## ***Administrative Task: Managing users and access control***

**Grade: B**

### **Drupal Review**

#### **Creating new users**

The creation of user accounts in the Drupal CMS has not historically had many identified usability problems. It has been Web Networks' experience that clients have no difficulty finding where to add users and have no trouble adding the users via the available interface. Of the forms provided for common administrative tasks, the user creation form is one of the shortest and most intuitive.

#### **Administering access control**

The administration of user rights or 'access control', on the other hand, is an area that could use some improvement. Users can administer access control via the 'administer >> access control' menu item. The interface of this administration page can be overwhelming. The interface is a matrix of user roles along the top and available permissions down the left-hand side, grouped by module name. To grant permissions to a user role, the administrator must check off checkboxes beside the available permission.

Drupal has two roles defined by default. One is 'anonymous user' (a user not logged in) and 'authenticated user' (a user who is logged in). By default, there are only ten (10) modules with twenty-four (24) available permissions. This creates a 2x24 matrix. As new roles get added the matrix grows horizontally. As new modules are enabled, the matrix grows vertically. A site with a modest number of user roles and modules soon creates a very large matrix that causes the browser window to scroll in both directions. If an administrator scrolls down, the names of the roles are no longer visible and it becomes difficult to know which column of checkboxes should be checked when granting a new permission. Likewise, horizontal scrolling can cause the names of the permissions to become invisible.

While this interface is very easy to understand, it can become very difficult to use.

#### **Creating user roles**

The interface for creating user roles is available via a tab on the 'administer >> access control' page. This is perhaps one of the simplest administrative tools available. The user is presented with a list of existing roles and a single text field in which they can enter the name of a new role. Clicking on 'add new role' next to the field immediately adds the new role.

### **Recommendations**

Web's clients are often confused between access to modules and access to edit content on specific areas of a website. Some care should be taken to create simple concepts to clarify these functions in the access control interfaces.



## ***Administrative Task: Attaching files and managing digital assets***

**Grade: D-**

### **Drupal Review**

By default, Drupal ships with the 'upload' module which, when enabled, allows administrators to attach files to a page of content. With the module enabled, the content creation and page editing include an additional option (under the heading 'file attachments') that allows the user to browse for files on their local system and 'attach' them to the page by uploading the file to the server.

Of all the Drupal features commonly used by Web Networks' clients, this is the one with the greatest room for improvement.

### **Attaching files**

Users generally do not have problems browsing for files or clicking the 'attach' button. However, we have observed clients that were unsure of what, if anything, happened when they clicked the button. After clicking 'attach', the interface shows a table with two checkboxes (one under the heading 'delete' and the other under 'list'), a text area with the filename, and the size of the file. There is no indication if the file has in fact been uploaded. Nor is there any indication whether the content page to which they are attaching the file has been saved.

Once users come to understand and are familiar with the process of attaching new files, they often ask how to attach a file that they have already uploaded and attached to a different page. In other words, there is an expectation that the 'file attachments' tool would allow them to attach files that are already on the system. Sadly, it does not offer this functionality.

Another common request by clients is to view all files that have been uploaded or attached to any page. Once again, there is an expectation that if a file is uploaded, there must be a way to see what has been uploaded. There is no such ability in the current interface.

### **Removing attached files**

While the interface to attach a file requires the user to click an 'attach' button, there is no corresponding 'delete' or 'remove' button for removing a file. Instead, a user must check a box under the heading 'delete' and click the 'save' button of the content editing form. Still, there are no explicit instructions explaining this. It must somehow be inferred from the text on the screen that says, 'Changes made to the attachments are not permanent until you save this post'.

Overall, the tools available for associating files with pages of content are difficult to use and very limited.

### **Recommendations**

A digital asset management tool must be developed for Drupal. Web Networks has begun work on a contributed module for such asset management.



The upload module should be stripped down to provide nothing more than simple upload functionality. All file management should be offloaded to a specialized asset management module.



## Discussion, Conclusions and Further Research

### ***The Language of Drupal***

One of the first challenges new Drupal users face is learning the language of Drupal. Terms like node, taxonomy, module and path can be sometimes confusing for first-time administrators. Compounding these terms to create ‘node module’ or ‘module path’ or ‘path module’ each with distinctly different meanings only adds to the confusion some may experience. In the Administration User Experience Survey of 2005<sup>12</sup>, 47% of respondents found understanding Drupal terminology to be difficult, while only 29% found it easy.

Nearly every Web Networks client has at one point or another asked, “what exactly is a node?” The answer is: a single unit of posted content, excluding ‘comments’. The term ‘node’ does not appear on either the ‘create content’ page in Drupal or in the ‘administer >> content’ interface. Yet, ‘node’ appears regularly in the help files and extensively through the Drupal handbooks. It is conceivable that a user may encounter the term several times without ever making the connection to ‘content’.

One general usability guideline for all software is to avoid jargon. Drupal unfortunately has a great deal of it.

Web Networks’ experience has been that users do learn the language of Drupal relatively quickly. It has also been our experience that the initial lack of understanding of Drupal terminology does not prevent users from easily accomplishing basic tasks such as publishing content. Still, unfamiliar terms can be unsettling and do not help provide a positive user experience.

### ***Impressions of the Administration Interface***

Unlike several other content management systems, the ‘look’ of Drupal’s administration interface is not standardized. The look of the administration interface is entirely dependent on the theme installed on the site. To be clear, the administration menu items are standardized, but the presentation of the menu is not.

Nearly all of Drupal’s administration tools are found via the ‘administer’ menu item which becomes available to a user once they have signed in / logged onto the CMS. Drupal introduced localized tasks (tab-based navigation) to administrative pages in version 4.5 so that similar tasks could be grouped together on fewer pages, thereby reducing the number of items in the ‘administer’ menu tree. The introduction of this user interface element was positive, but in some cases its overuse resulted in several administrative tasks being displayed via sub-tabs (tabs under tabs). This sort of deep linking had been identified as a usability concern in version 4.5 and 4.6. The user experience survey from November 2005 showed that 41% of respondents found it difficult to find deeply linked tasks and only 16% found deeply linked tasks easy to find.

---

<sup>12</sup> See Appendix A: Administrator User Experience Survey

Over the past two years, there have been several improvements to the administrative menu structure based on sound usability guidelines and suggestions from the community of developers. Some of these changes included standardizing the language of the menu structure to improve readability, assigning more meaningful names to menu items (eliminating generic names like 'configure'), and moving deeply linked items higher in the menu hierarchy. The topic of menu structure continues to be discussed, but there is very little evidence to suggest that any further changes to the current default menu structure (version 4.7) are actually required to improve usability.

Web Networks will occasionally receive technical support calls to help users find non-common administrative tools (e.g. content type configuration) within the interface of sites built using version 4.6. However, all common administrative tasks are very easy to find in the first level of the 'administer' menu tree.

It should also be noted that the administration menu tree is can be customized by the administrator. In other words, the default menu structure (or names of menu items) can be changed to suit the tastes of the end user via the 'administer >> menu' menu item.

### ***Suggested Improvements for the Administration Interface and New Developments***

It has been suggested several times within the Drupal community that the administration interface be the same across all installations of Drupal. The primary argument is that an attractive, standardized interface would improve user experience. The truth is that the default Drupal theme is rather bland, but this in itself does not affect functionality.

Besides standardizing the look of the interface there has been a call to replace the menu-driven administration interface with a 'control panel' approach that groups tasks by goal or objective. There is sound research that shows grouping tasks by task types or goals improves usability.<sup>13</sup> This 'control panel' would be displayed to the user upon logging in. Currently, the first page a user is presented with when logging in to the administrative interface is the 'watchdog' or event log.

The Civicspace distribution of Drupal included such a control panel via a pre-installed add-on module called the 'administration' module which was developed based on usability research Civicspace conducted in 2005. The developer behind the administration module has been working on improving the module and has recently had it added to the core of Drupal. This means that it will be installed by default for the next release of Drupal (4.8/5.0).

---

<sup>13</sup> Task Knowledge Structures <http://www.cs.bath.ac.uk/~hci/TKS/home.html>

## **Conclusions**

The usability guideline review established that overall, the forms that Drupal provides to accomplish common administrative tasks do not themselves make tasks difficult to accomplish. The majority of the guidelines reviewed are followed by Drupal.

The stakeholder usability review likewise found that usability issues were, at most, partially related to the design of forms. The acts of creating or editing content, adding or removing menu items, creating vocabularies or terms, and creating or removing users and roles were, generally speaking, easily accomplished by Web Networks' clients as they had little difficulty using the actual forms provided by Drupal.

These conclusions do not mean that Drupal is without usability issues. Several recommendations have been made throughout this document indicating areas where usability could be improved. In most cases usability was sufficient to accomplish a task, but could be improved by simply providing more robust tools (e.g. drag and drop menu ordering). In one case, usability was poor because a feature present in the system (creating views of categories), had no administrative interface.

The issues related to attaching files to published content represented a special case where the current interface provided to accomplish the task was mediocre, but in general was found to be unusable because it lacked desired functionality. Technically speaking, this is not a usability issue with the upload module. The module was never designed to be a digital asset manager. Still, it is felt that a lack of digital asset management is a usability concern for our stakeholders.

Drupal has a history of following recommendations that improve usability. Historically, Drupal has made several usability improvements with each new release.<sup>14</sup> Based on personal experience using Drupal, version 4.7 is the most usable release to date. Furthermore, there is growing evidence of a shift within the Drupal development community to focus more closely on usability. This may be a result of Dries Buytaert (the original creator of Drupal) urging developers to change their focus, or the efforts of groups like Civicspace that lead by example and persevere to have their usability enhancements included in the Drupal code base. Regardless of the catalyst, there is a sense that usability will continue to be a priority among Drupal developers.

The overall conclusion of this report is that Drupal provides sufficiently usable tools to allow administrators to effectively maintain their installed websites.

---

<sup>14</sup> Drupal 4.7 release announcement including a listing of usability improvements for the release.  
<http://drupal.org/drupal-4.7.0>



## ***Further Research***

### **Expanded Scope**

This report focused exclusively on functionality provided in a default installation of Drupal. There are several contributed modules that are commonly used by our stakeholders that were not included in this review. Further research into the usability of such modules would be of benefit.

### **Personas**

The identification of the stakeholders of this document is similar to an exercise in creating a persona for the purposes of evaluating usability. In the case of this review our stakeholders represent a group of organizations that require websites that are content-rich and therefore primarily require tools to manage and organize their content. The administrators of these sites typically have the same goals and technical requirements.

Even so, this group only represents a tiny sliver of potential Drupal users, and their usability experiences are going to differ from others because they will tend to use different tools and features provided by Drupal. A similar study could be completed for any one of a number of 'personas'.

### **Usability Guideline Reviews**

The usability guideline review focused on a handful of core modules and their administrative interfaces. However, there were several core modules that were not included in the review (e.g. search, path, forum, etc.). There are also hundreds of other contributed modules (many of which are commonly used), which could be reviewed in a similar fashion.



## Appendix A: Administrators User Experience Survey November 2005






This online survey was conducted in November of 2005. The survey was presented at Drupal Con 2005 by Kieran Lal of Civicspace Labs as part of his presentations on usability testing. There were over 200 respondents to the survey in the 16 hours following a post to the Drupal developer mailing list. It can be assumed that a large number of respondents are subscribers to that mailing list. Subscribers to the list span a broad range of experience levels working with and developing for Drupal. The survey was also posted to the main page of Drupal.org (the main portal to the Drupal project).


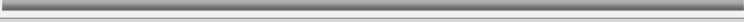
A new version of this survey is in the process of being developed by Kieran Lal. Future research should compare the results of the two surveys as they span a major release in Drupal. At the time of this survey Drupal 4.7 had not yet been released.







Of the questions in the survey, #7 provides the most insight into which features or tasks are considered easy/difficult by Drupal administrators.


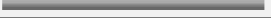
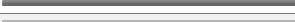





1. How would you describe yourself as a Drupal administrator?			
		Response Percent	Response Total
New user, confused but persistent		20.3%	178
Web developer: Site installer and configurer		52.8%	463
Site maintainer: Can do maintenance, but do not understand what's underneath		21.7%	190
Power user: very technical user capable of significant changes including programming		40.8%	358
Other role		3.4%	30
<b>Total Respondents</b>			877
(skipped this question)			0

2. How frequently do you administer your Drupal site?			
		Response Percent	Response Total
Once every three weeks		12.3%	98
Once a week		18.5%	147
Several times a week		32.9%	262
Once a day		15.3%	122
Several times a day		22.7%	181
Other (please specify how often)		4.3%	34
<b>Total Respondents</b>			796
(skipped this question)			81

3. How long do you administer your Drupal site in a single sitting approximately?			
		Response Percent	Response Total
A few minutes at a time.		24.9%	198
A little less than an hour at a time.		45%	358
Multi-hour sessions		26%	207
Days at a time		2.6%	21
Other (please specify length of time in a single period)		1.5%	12
<b>Total Respondents</b>			<b>796</b>
(skipped this question)			81

4. Can you get to your Drupal administration pages?			
		Response Percent	Response Total
No		3%	21
Yes		97%	669
<b>Total Respondents</b>			<b>690</b>
(skipped this question)			187

5. What are some common Drupal administration tasks that you do? Select all tasks that you commonly do with your sites.			
		Response Percent	Response Total
Administer content on the site such as comments, or pages.		82.7%	570
Administer users such as reset passwords, change user permissions.		41.1%	283
Administer a new site by installing new modules, set permissions, configure users		68.4%	471
Administer the style, layout, or presentation of the site.		66.9%	461
Administer new modules or features to learn what is possible		82.3%	567
other (please specify common Drupal Administration Tasks)		11.5%	79
<b>Total Respondents</b>			<b>689</b>
(skipped this question)			188

6. What are some infrequent Drupal administration tasks that you do?			
		Response Percent	Response Total
Configure modules		38.5%	265
Administer comments		34.7%	239
Moderate content		38.8%	267
Administer the theme		46.9%	323
Review site statistics		44.1%	304
Administer users		39.9%	275
Administer content types (book, page, story, blog, etc)		41.8%	288
Other (please specify)		3.6%	25
<b>Total Respondents</b>			<b>689</b>
(skipped this question)			188

7. When I administrate my Drupal site, I find it is easy to?							
	Very Easy	Easy	Somewhat Easy	Difficult	Very Difficult	N/A	Response Average
Review the site activity in the watchdog	24% (121)	<b>31% (159)</b>	23% (120)	8% (42)	4% (18)	10% (53)	2.30
View a list of users	33% (169)	<b>38% (198)</b>	16% (81)	6% (32)	1% (7)	6% (31)	1.99
Publish content such as news	<b>40% (207)</b>	38% (195)	17% (86)	3% (14)	0% (1)	3% (16)	1.82
Configure roles and their permissions	14% (75)	<b>33% (173)</b>	32% (169)	14% (74)	3% (16)	2% (13)	2.57
Enable and disable modules	<b>54% (280)</b>	35% (181)	10% (51)	1% (6)	0% (2)	0% (2)	1.59
Install Drupal modules	21% (110)	<b>39% (206)</b>	30% (154)	7% (37)	2% (10)	1% (5)	2.29
Administer module permissions	14% (74)	32% (168)	<b>35% (180)</b>	14% (72)	2% (11)	3% (13)	2.56
Administer module settings	12% (60)	<b>39% (202)</b>	39% (201)	9% (47)	1% (7)	1% (3)	2.50
Administer menus	10% (51)	24% (125)	<b>37% (192)</b>	23% (118)	5% (25)	2% (13)	2.88
Get to administration pages due to slow loading times	10% (50)	23% (119)	<b>29% (151)</b>	11% (59)	3% (13)	24% (124)	2.66
Update site to a new version of Drupal	2% (9)	10% (50)	24% (125)	<b>27% (141)</b>	17% (86)	21% (110)	3.60
Manage inconsistency in themes	2% (9)	6% (30)	20% (103)	<b>37% (194)</b>	18% (92)	18% (92)	3.77
Compare Drupal files	4% (23)	9% (45)	22% (112)	28% (144)	9% (45)	<b>28% (146)</b>	3.39
Administer categories with the taxonomy module	8% (43)	25% (132)	<b>32% (166)</b>	19% (100)	10% (53)	5% (28)	2.98
Administer multiple comments	5% (26)	15% (75)	24% (123)	13% (64)	3% (17)	<b>40% (206)</b>	2.90
Order and organize content	6% (33)	24% (126)	<b>39% (203)</b>	25% (129)	4% (23)	2% (9)	2.97
Experiment and learn new features and modules	7% (36)	24% (126)	<b>43% (226)</b>	22% (115)	3% (15)	1% (3)	2.90
Find pages that are deeply linked	3% (17)	13% (67)	27% (138)	<b>31% (160)</b>	8% (40)	18% (91)	3.33
Administer the structure of content	5% (25)	18% (95)	<b>37% (191)</b>	30% (155)	6% (33)	3% (16)	3.15
Use Unix commands such as setting permissions and moving files	<b>24% (123)</b>	23% (120)	17% (87)	12% (60)	9% (48)	16% (81)	2.52
Test your site	19% (97)	<b>32% (165)</b>	29% (150)	14% (71)	3% (17)	4% (23)	2.49
Set access permissions for organizational business roles	4% (23)	19% (96)	<b>30% (156)</b>	22% (113)	8% (41)	17% (88)	3.12
Administer the structure of a site	4% (22)	18% (96)	<b>36% (187)</b>	32% (169)	7% (35)	2% (12)	3.19
Administer the theme	10% (50)	22% (115)	<b>35% (182)</b>	21% (111)	9% (46)	3% (17)	2.98
Administer weights and organize parent relationships(Menu/Book)	9% (48)	26% (133)	<b>31% (160)</b>	21% (110)	5% (28)	8% (41)	2.87
Understand Drupal terminology and synonyms	8% (40)	19% (100)	29% (149)	<b>30% (156)</b>	13% (69)	1% (6)	3.22
Administer themes with a programming mindset	6% (30)	17% (87)	<b>27% (140)</b>	25% (129)	12% (63)	13% (69)	3.24
Add and position blocks	20% (106)	<b>36% (186)</b>	28% (144)	10% (52)	4% (23)	2% (9)	2.41
Configure blocks for various roles	6% (30)	16% (83)	<b>30% (156)</b>	30% (153)	8% (39)	11% (57)	3.19
<b>Total Respondents</b>							538
(skipped this question)							339



8. What are some other important Drupal administration tasks that we did not cover above?			
		Response Percent	Response Total
Configure menu open and closed defaults		27.7%	134
<b>Mailing functionality</b>		<b>51.3%</b>	<b>248</b>
Access to all modules		19.9%	96
Management of style		41.4%	200
Make the screens look better		41.4%	200
Administer time sensitive nature of communication		22.6%	109
Group tasks logically		50.7%	245
Other (please specify tasks we may have overlooked)		26.7%	129
<b>Total Respondents</b>			<b>483</b>
(skipped this question)			394

## Appendix B: Usability Evaluation of Civicspace Installer

This appendix contains an incomplete study of the Civicspace Installer. Initially it was assumed that a review of installation procedures of both Drupal and Civicspace was required for the APC usability report. In the end, installation fell outside of the scope of the report and the study was abandoned.

However, installation is widely recognized as a major usability issue with Drupal. At the time of review, Drupal itself did not come with an installer. The upcoming release of Drupal 4.8/5.0 will include an updated version of the installer system reviewed in this document. After reviewing initial releases of this new installer, it is clear that the install process for the core distribution of Drupal is much simpler than the steps described in this document. Likewise, there have been changes made to the installer for the Civicspace distribution of Drupal. Therefore, much of what is contained in this document has been addressed or changed.

A usability review was conducted after performing a usability test with a single test subject. These are the incomplete results of our review.

### Automated Installation

For the purposes of evaluating the automated installation process, we have assumed that the user has some experience using the core technologies used by Drupal / Civicspace (i.e. Linux, Apache, MySQL, PHP (LAMP), or Window Apache MySQL PHP (WAMP)) and is capable of transferring files and using the file system of their chosen platform. It is also assumed the user has read the install.txt file.

### The install.php page

The introduction to the install.php page is precise with links to additional resources to help those who may not fully understand what is required. Text assuring users that they may proceed even if they are unsure of the required configuration is helpful.

### Recommendation

A link explaining how to set file permissions may be useful (currently, such understanding is assumed).

### Automated Installation

#### Install.php (Server Check-up)

The pre-install checklist clearly identifies the correctly configured components (use of green checkmark icons) and provides useful warnings with links to resources to help resolve the warning or provide additional information if the warning is unclear.



1. **Server check-up**
2. Install
3. Database setup
4. Finishing up
5. Configure your site

We're going to run a few checks to make sure that your server is setup correctly for CivicSpace...

**Warning:** Our memory check indicates that you currently have a memory limit of 8M which is less than the recommended 24M. You may proceed with the installation, but it is *strongly* recommended that you configure your PHP to provide you with at least 24M of memory. [More help](#)

**Notice:** You appear to be installing CivicSpace on a private IP address or on 'localhost'. If you would like your site to be publicly accessible, you need to restart this installer from a public domain or IP. [More help](#)

**Warning:** The installer has detected that your server's PHP is configured to report all levels of errors. This can be a problem if you are running a production site. It is recommended that you lower your level of error reporting to exclude PHP notices at the very least. [More help](#)

**Success!** PHP's "safe\_mode" is set to "Off".

**Success!** `sites/default/settings.php` is writable.

Please **make sure you turn off write permissions to this file when you are done with the installation.** (We'll remind you at the end of the installation.)

## Install.php (Install)

Currently the page is called 'Install' and provides users a choice of install type (single or multiple site) and also an option to install additional CiviCRM components.

There is some confusion here. The text at the top reads: "Optional multi-site install. If you want to set up a single site, you can ignore this page and click on Setup database". However, there is an option to install CiviCRM components at the bottom of the page which may well be missed if the user does indeed "ignore this page".

While the page does provide links to external resources regarding 'multi-site' installs, the links take the user to an entirely new set of installation instructions on the Civicspace site. These instructions are different from those found on the download page on the Civicspace site.

### Recommendation:

Rename the page to 'choosing installation type'.

Indicate that the default installation for most users is 'single'.

Provide a link to a page describing the differences between single and multi-site installation if the user is unsure of which type to select.



Remove the optional CiviCRM installation component from this page. Optional components should be dealt with separately.

Either replace the installation instruction link on the download page with a link to the instructions on this page, or link to the instructions found on the download page. In either case the user should refer to a consistent set of external installation instructions.



1. Server check-up
2. **Install**
3. Database setup
4. Finishing up
5. Configure your site

## Install

### Multi-Site Install

Optional multi-site install. If you want to set up a single site, you can ignore this page and click on **Setup database**

Single site set up

Just keep this option checked if you want to set up a single site. The configuration file will be placed in `sites/default/settings.php`

Multiple site set up

Multi site set up. You will be able to set up more sites with this single CivicSpace install. It is **strongly** recommended that you consult the [CS 0.8.2 installations guide](#)'s page on [setting up multiple sites](#).

Parameters for multi-site set up

Base URL:

`http://localhost/civicspace`

This will be the base URL for your CivicSpace installation

Install path:

`sites/localhost.civicspace/settings.php`

The configuration file for this site will be placed in

`sites/localhost.civicspace/settings.php`

### CiviCRM Install Settings

#### CiviCRM Installation Settings

Enable CiviContribute

Use the CiviContribute component for collecting and managing online and



## **Install.php (Database Setup)**

### **Recommendations**

Page should indicate that the settings being entered are those that were previously configured.

### **CiviCRM Database Configuration:**

Prior to this point in the installation or the installation instructions, there has been no indication that a separate database or database settings may be required for CiviCRM.

While it is indicated that the settings in the CiviCRM form section are not required, the text comes after the fields are presented to the user. Present this information before presenting the form.

Table prefix:

**Tip:** If you are using a shared database account or only have one database account, you may wish to prefix your database tables to avoid a "naming collision." For example: "my\_new\_site\_".

The settings in this box are for the CivicSpace database.

CiviCRM

User name:

**Tip:** Allowed characters include letters, digits, '\_', and '-' with a maximum of 16 characters.

Password:

**Tip:** Allowed characters include **letters**, **digits**, and any of the following:  
-\_!#\$%^&\*()+={}|<>;?;~,~

Host:

**Tip:** If your database is on the same machine as your CivicSpace installation, just enter "localhost". Leaving this item blank will default the value to "localhost".

Database name:

The settings in this box are for the CiviCRM database. If you want to use the same database for CivicSpace and CiviCRM, leave the form items in this box empty. Note that table prefixes will not be applied to the CiviCRM database. The installer will install the CiviCRM database for you if your database's version is recent enough. You will be notified of success or failure.

Extras

### Install.php (Finishing Up)

In the early stages of installation users were asked to change file permissions and at this stage they are asked to change them back.

The information is presented as if there were errors and that the user must now correct these errors.

## Recommendations

The information describes why the settings need to be changed back, but it should be made clear that the user has not done anything wrong.

CIVICSPACE<sup>BETA</sup>



CivicSpace Installer

1. Server check-up
2. Install
3. Database setup
4. **Finishing up**
5. Configure your site

We're going to run a few checks to make sure that your CivicSpace installation is secure...

- ! **Whoops!** `sites/default/settings.php` is writeable and can be a security hazard. [More help](#)
- ! **Whoops!** `sites/default/civicsm.settings.php` is writeable and can be a security hazard. [More help](#)
- ! **Whoops!** The `sites/` directory is writeable and can be a security hazard.. [More help](#)

Try checks again

[Skip these checks and go to the configuration wizard, I understand the risks involved \(Click here if some checks "stall" and you cannot continue\).](#)

### **Configuration wizard:**

#### **Observations and Recommendations**

Introduction text of the configuration wizard accurately describes what this page is.

#### **Configuration wizard – administrator:**

A 'show help' link is offered to the user in the top right corner. Clicking on the link displays the information on the page.

Up until this stage such guidance had been offered by default. As a matter of consistency, consider showing this guidance by default throughout the configuration process.

#### **Configuration wizard – basic settings:**

Default options are presented as examples. It is clear that these settings may be changed at a later time and that they are not required.

#### **Configuration wizard – getting started:**

User was unsure of where they would be taken if they chose to end the configuration wizard.



### **Configuration wizard – continued:**

The wizard in general has solved many complaints about Drupal / Civicspace configuration.

Before:

"I want my site to do x - how do I get it to do that?" Answer: install module x.

"Module x is installed - now what?" Answer: enable the module in administer/modules.

"The Module is enabled but how do I change behaviour y?" Answer: go to administer/settings/x and change behaviour y.

Now the site asks what functionality the user would like and enables the appropriate module. Then it asks how the user would like the module to behave and sets the appropriate behaviour.



## Appendix C: Glossary

### API

Application Programming Interface - a standardized way for programs to communicate or interact.

### Category

When used in the context of Drupal, category refers to a term that is a part of a Vocabulary. Content within the Drupal CMS may be categorized by associating pages with a category term. Categories are managed via the Taxonomy module (see: Vocabulary, and Taxonomy).

### CivicSpace

A distribution of the Drupal CMS maintained by CivicSpace Labs.

### Content Management System (CMS)

A content management system (CMS) is a system used to manage the content of a website. A CMS element allows the content manager or author, who may not know Hypertext Markup Language (HTML), to manage the creation, modification, and removal of content from a website without needing the expertise of a webmaster. The features of a CMS system vary, but most include web-based publishing, format management, revision control, indexing, search, and retrieval.

### Content Type

A type of content that can be posted or published to a Drupal website. Content Types are defined by Modules.

### Drupal

An open source content management system originally developed by Dries Buytaert.

### Module

A Drupal term that refers to a program that extends the functionality of the Drupal CMS. Modules may provide completely new functionality or extend the functionality of other Modules.

### Node

Any content post to a Drupal website that is not a comment. Nodes can be any one of many different Content Types. (See: Content Type)



## **Path**

In the context of Drupal, a Path is a variable that the Drupal CMS interprets to decide which page of content to display to the user (e.g. node/13432). The path variable is visible in the address bar of the browser as either `http://example.com/q=node/13432` or more commonly `http://example.com/node/13432`

## **Path Alias**

An alias of a Drupal Path that makes it more human readable. Instead of 'node/13432' a Path Alias may read '/events/companypicnic'

## **Permissions**

Permissions are tied to user roles, and control access to or usage of a module's functionality.

## **Taxonomy**

The theory and practice of categorization.  
The Drupal categorization module.

## **Template**

A file that defines the HTML structure of pages within the site.

## **Theme**

The look and feel for a site. A theme is usually composed of a set of template files.

## **Vocabulary**

A container for category terms in the Drupal taxonomy system. Vocabularies can be for classification (controlled vocabularies), thesauri (controlled vocabularies that indicate the relationship of terms), taxonomies (controlled vocabularies where relationships are indicated hierarchically), and free vocabularies where terms or tags are defined during content creation.

## Appendix D: Usability Grade Summary for Drupal

Criteria	Grade
<b>Against Common Administrative Tasks<sup>15</sup></b>	
Adding new content to the website	<b>A</b>
Editing existing content on the website	<b>A</b>
Managing website navigation menus	<b>B-</b>
Categorizing content	<b>C</b>
Managing users, roles and access control	<b>B</b>
Attaching digital assets to content pages	<b>D-</b>
<b>Against Usability Guidelines<sup>16</sup></b>	
Distinguish clearly and consistently between required and optional data entry fields	<b>B-</b>
Detect errors automatically	<b>B+</b>
Minimize user data entry	<b>A</b>
Label data entry fields clearly	<b>B</b>
Place labels in close proximity to data entry fields	<b>A+</b>
Label form buttons clearly	<b>A</b>
Allow users to see their entered data	<b>B-</b>
Use radio buttons and checkboxes correctly	<b>A</b>
Group data entry by method type	<b>A</b>
Maintain proper tab indexing in forms	<b>A</b>

<sup>15</sup> See ‘Method’ section for definition of common administrative tasks

<sup>16</sup> Chapter 13: Screen-based Controls of the [Research-Based Web Design & Usability Guidelines](http://usability.gov/pdfs/guidelines.html)  
<http://usability.gov/pdfs/guidelines.html>