



---

Theses and Dissertations

---

2006-11-11

## On-Line Electronic Document Collaboration and Annotation

Trev R. Harmon  
*Brigham Young University - Provo*

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [OS and Networks Commons](#)

---

### BYU ScholarsArchive Citation

Harmon, Trev R., "On-Line Electronic Document Collaboration and Annotation" (2006). *Theses and Dissertations*. 816.

<https://scholarsarchive.byu.edu/etd/816>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact [scholarsarchive@byu.edu](mailto:scholarsarchive@byu.edu), [ellen\\_amatangelo@byu.edu](mailto:ellen_amatangelo@byu.edu).

ON-LINE ELECTRONIC DOCUMENT COLLABORATION  
AND ANNOTATION

by

Trev R. Harmon

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

School of Technology

Brigham Young University

December 2006



© 2006 Trev Harmon. Some Rights Reserved.

Licensed under

Creative Commons  
Attribution-NonCommercial-NoDerivs 2.5 License

and

Attribution-NonCommercial-ShareAlike 2.5

except where noted

See Licensing Section for more detail



BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Trev R. Harmon

This thesis has been read by each member of the following graduate committee and by a majority vote has been found to be satisfactory.

\_\_\_\_\_  
Date

\_\_\_\_\_  
C. Richard Helps, Chair

\_\_\_\_\_  
Date

\_\_\_\_\_  
Joseph J. Ekstrom

\_\_\_\_\_  
Date

\_\_\_\_\_  
Michael G. Bailey



BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Trev R. Harmon in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

---

Date

---

C. Richard Helps  
Chair, Graduate Committee

Accepted for the School

---

Val D. Hawks  
Director

Accepted for the College

---

Alan R. Parkinson  
Dean, Ira A. Fulton College of  
Engineering and Technology





## ABSTRACT

### ON-LINE ELECTRONIC DOCUMENT COLLABORATION AND ANNOTATION

Trev R. Harmon

School of Technology

Master of Science

The Internet provides a powerful medium for communication and collaboration. The ability one has to connect and interact with web-based tools from anywhere in the world makes the Internet ideal for such tasks. However, the lack of native tools can be a hindrance when deploying collaborative initiatives, as many current projects require specialized software in order to operate.

This thesis demonstrates, with the comparably recent advances in browser technology and Document Object Model (DOM) implementation, a web-based collaborative annotation system can be developed that can be accessed by a user through a standards-compliant web browser. Such a system, demonstrated to work on the commonly-used web browsers constituting the vast majority of web traffic, was implemented using open-source tools and industry-recognized standards. Additionally, it



accepts static copies of most standard document formats for both handwritten and typed annotations, while maintaining an archived copy of the original.

The system developed for this thesis lends itself to use in a number of different process domains, as most collaborative annotation approaches can be described by a single process model. While a number of possible usage scenarios are discussed, this thesis approaches system usage only in an academic setting, focusing on applicability of the system to electronic grading and document exchange. From here, additional system usage can be easily extrapolated.



## ACKNOWLEDGEMENTS

I would like to thank my family and friends for their support and belief in my abilities. Also, without my committee, this thesis would not have been possible. I would like to thank my committee chair, Richard Helps, for the many hours of help and consultation, Dr. Joseph Ekstrom for the nucleus idea upon which this thesis is built, and Dr. Michael Bailey for his continual support and encouragement. In addition, Ruth Ann Lowe spent a great deal of time advising and answering a near-endless stream of questions, for which I am very appreciative.



## TABLE OF CONTENTS

List of Tables.....	xxv
List of Figures.....	xxvii
Glossary.....	xxix
1. Introduction.....	1
1.1. Statement of Problem.....	2
1.1.1. Usage Scenarios.....	3
1.1.1.1. Document Collaboration.....	4
1.1.1.2. On-line Grading.....	4
1.1.1.3. Discussion Boards.....	5
1.1.1.4. In-class Discussion.....	5
1.1.1.5. Review and Accreditation Boards.....	5
1.1.1.6. In-field Presentations.....	6
1.1.2. The Big Picture.....	6
1.1.3. System Attributes.....	7
1.2. Research Questions.....	8
1.3. Constraints and Limitations.....	9
2. Literature Review.....	11



2.1. Users.....	11
2.1.1. Human.....	12
2.1.2. Computer.....	13
2.2. Process Domains.....	14
2.2.1. Academia.....	14
2.2.1.1. Electronic Grading.....	14
2.2.1.2. Annotation in the Learning Process.....	15
2.2.1.2.1. In-class Usage.....	16
2.2.1.2.2. Out-of-class Usage.....	17
2.2.1.3. Course Management Systems (CMS).....	18
2.2.2. Collaboration.....	19
2.2.3. Semantic Web.....	21
2.2.4. Social Networks.....	23
2.2.5. Ubiquitous Annotation.....	23
2.3. Implementations.....	24
2.3.1. XLibris.....	24
2.3.2. ScreenCrayons.....	25
2.3.3. MADCOW.....	26
2.3.4. Annotea.....	26
2.3.5. Digital Graffiti.....	27
2.3.6. Tablet PC.....	28
2.4. Other Observations.....	28

2.4.1. Book and Paper Metaphors.....	28
2.4.2. Free-form Annotation.....	29
2.4.3. Anchoring.....	30
2.4.4. Acetates.....	31
2.4.5. Annotation Clients.....	31
2.4.6. Bookmarks.....	32
3. Methodology.....	33
3.1. Model-View-Controller (MVC) Design.....	33
3.2. Development.....	34
3.2.1. Hardware and Operating Systems.....	34
3.2.2. Web Server.....	35
3.2.3. DBMS.....	36
3.2.4. Programming Languages.....	36
3.2.4.1. Server-side Programming.....	36
3.2.4.2. Client-side Programming.....	40
3.2.5. Software API.....	40
3.2.6. Services.....	41
3.3. Testing.....	41
3.3.1. Compliance Testing.....	41
3.3.2. Load Testing.....	43
4. Results.....	45
4.1. Security and Authentication.....	46

4.1.1. System Security.....	46
4.1.2. User Authentication.....	47
4.1.3. Document Permissions.....	49
4.1.4. Multi-tier Access (Groups).....	50
4.2. Document Acceptance.....	51
4.2.1. Interface.....	51
4.2.2. Conversion.....	53
4.2.2.1. Linux-handled Formats.....	54
4.2.2.2. OpenOffice.org Formats.....	54
4.2.2.3. Win32 Formats.....	55
4.2.2.4. The PS/PDF Strategy.....	56
4.2.3. Archival Copy.....	56
4.3. Viewing Documents.....	57
4.3.1. Display Mode.....	57
4.3.2. Read Mode.....	59
4.3.3. Annotate Mode.....	62
4.3.4. Original Document.....	63
4.4. Annotating Documents.....	63
4.4.1. Draw Annotation Mode.....	63
4.4.2. Text Annotation Mode.....	64
4.4.3. The Tracer.....	66
4.4.4. Alternative Real-time Display Methods.....	69

4.4.4.1. Client-side.....	69
4.4.4.2. Server-side and AJAX.....	70
4.4.4.3. VRML and SVG.....	70
4.5. System Administration.....	71
4.5.1. Users.....	71
4.5.2. Groups.....	72
4.5.3. Documents.....	75
4.5.4. Services.....	76
4.5.5. Document Converters.....	77
4.6. Software API.....	78
4.6.1. Annotation.pm.....	79
4.6.2. Auth.pm.....	79
4.6.3. CGI.pm.....	80
4.6.4. Converter.pm.....	80
4.6.5. DBI.pm.....	80
4.6.6. Document.pm.....	81
4.6.7. Image.pm.....	81
4.6.8. Menu.pm.....	81
4.6.9. Page.pm.....	81
4.6.10. Person.pm.....	82
4.6.11. Service.pm.....	82
4.6.12. System.pm.....	82

4.6.13. Utils.pm.....	82
4.7. Browser Support.....	83
4.8. Load Testing.....	84
5. Conclusions.....	87
5.1. Web Browsers.....	87
5.2. File Formats.....	88
5.3. Contextually Anchored Annotations.....	89
5.4. Original Document Archive.....	89
5.5. Access Control.....	89
5.6. System Performance.....	90
5.7. Other Considerations.....	91
5.8. Future Research and Improvements.....	91
5.8.1. System Performance.....	92
5.8.2. Usability.....	92
5.8.3. User Authentication.....	92
5.8.3.1. HTTPS.....	93
5.8.3.2. Cookie-based Authentication.....	93
5.8.4. OAS Extensions.....	93
5.8.4.1. Examples.....	94
5.8.4.2. Process for Extending OAS.....	94
5.9. A Final Word.....	95
Selected Bibliography.....	97

Licensing.....	109
Attribution-NonCommercial-NoDerivs 2.5.....	109
Attribution-NonCommercial-ShareAlike 2.5.....	114
Appendix A – API Documentation.....	123
A.1. OAS::Annotation.pm.....	123
A.2. OAS::Auth.pm.....	132
A.3. OAS::CGI.pm.....	140
A.4. OAS::Converter.pm.....	141
A.5. OAS::DBI.pm.....	145
A.6. OAS::Document.pm.....	147
A.7. OAS::Image.pm.....	152
A.8. OAS::Menu.pm.....	154
A.9. OAS::Page.pm.....	157
A.10. OAS::Person.pm.....	161
A.11. OAS::Service.pm.....	167
A.12. OAS::System.pm.....	170
A.13. OAS::Utils.pm.....	174
Appendix B – API Source Code.....	177
B.1. OAS::Annotation.pm.....	177
B.2. OAS::Auth.pm.....	193
B.3. OAS::CGI.pm.....	208
B.4. OAS::Converter.pm.....	209

B.5. OAS::DBI.pm.....	217
B.6. OAS::Document.pm.....	220
B.7. OAS::Image.pm.....	232
B.8. OAS::Menu.pm.....	236
B.9. OAS::Page.pm.....	250
B.10. OAS::Person.pm.....	256
B.11. OAS::Service.pm.....	267
B.12. OAS::System.pm.....	273
B.13. OAS::Utils.pm.....	280
Appendix C – Services Source Code.....	283
C.1. AddAnnotation.pm.....	283
C.2. AdminConverter.pm.....	287
C.3. AdminDeleteConverter.pm.....	291
C.4. AdminDeleteService.pm.....	293
C.5. AdminDeleteUser.pm.....	295
C.6. AdminService.pm.....	297
C.7. AdminUser.pm.....	301
C.8. AnnotateDocument.pm.....	307
C.9. ChangeDocumentOwner.....	310
C.10. CreateDocument.....	313
C.11. DeleteAnnotation.pm.....	316
C.12. DeleteDocument.....	318

C.13. DocumentPermissions.pm.....	321
C.14. EditAnnotation.pm.....	326
C.15. FileFormats.pm.....	329
C.16. GroupManager.pm.....	330
C.17. IndexMenu.pm.....	338
C.18. Logout.pm.....	339
C.19. new_user.cgi.....	340
C.20. OpenDocument.pm.....	342
C.21. ReadDocument.pm.....	344
C.22. ShowDocument.pm.....	347
C.23. UserProfile.pm.....	349
Appendix D – Converter Source Code.....	355
D.1. copy.....	355
D.2. gs_wrap.....	356
D.3. image.....	356
D.4. oas_conv_server.pl.....	357
D.5. OOo.....	361
D.6. plain_text.....	364
D.7. ps_pdf.....	365
D.8. source_code.....	367
D.9. win32.....	368
Appendix E – JavaScript Source Code.....	373



E.1. annotation.js.....	373
E.2. cookie.js.....	384
E.3. div_manager.js.....	385
E.4. menu.js.....	389
Appendix F – CSS Source Code.....	393
F.1. oas.css.....	393
F.2. open_document.css.....	398
Appendix G – SQL Database Schema.....	401
Appendix H – Apache Configuration File: httpd.conf.....	407
Appendix I – Supported File Formats.....	415
Appendix J – Supported Web Browsers.....	417
Appendix K – Annotation in Literature.....	419
Appendix L – Open-source Course Management Systems.....	427
Appendix M – OAS Logo.....	431

## LIST OF TABLES

Table 1: Common Document File Formats.....	10
Table 2: Document Permissions.....	49
Table 3: All Tested File Formats.....	52
Table 4: Supported Web Browsers.....	83
Table I1: Supported File Formats.....	415
Table J1: Supported Web Browsers.....	417
Table L1: Open-source Course Management Systems.....	427



## LIST OF FIGURES

Figure 1: Single Process Model.....	3
Figure 2: Basic System Overview.....	45
Figure 3: Login Prompt.....	48
Figure 4: New Document Dialog.....	53
Figure 5: Open Document Dialog.....	58
Figure 6: Document Display Mode.....	58
Figure 7: Document Read Mode.....	59
Figure 8: Z-Index Layers.....	61
Figure 9: Document Annotate Mode.....	62
Figure 10: Text Annotation Creation Dialog.....	65
Figure 11: Document Annotation View with Text Annotation.....	66
Figure 12: Tracer Image (Enlarged 10x).....	67
Figure 13: User Administration.....	72
Figure 14: User Profile.....	73
Figure 15: Group Manager.....	74
Figure 16: Document Permissions.....	76
Figure 17: Web Service Administration.....	77

Figure 18: File Format Converter Administration.....	78
Figure 19: Simplified Request Cycle Diagram for OAS.....	85
Figure M1: OAS Logo.....	431

## GLOSSARY

As several terms used in this thesis have multiple meanings or may be unfamiliar to the reader, a brief glossary is provided to clarify such terms.

**Active Reading** – “the combination of reading with critical thinking and learning” (Schilit, Golovchinsky, and Price 1998)

**Common File Format** – A file format commonly used within a given process domain

**Dynamic Hyper-Text Markup Language (DHTML)** – a dynamic markup method relying on the interaction of *JavaScript* and *CSSv2* style sheets to provide dynamic content and/or interaction

**Document Object Model (DOM)** – “a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents” (Le Hégarret, Whitmer, and Wood 2006)

**Standards-Compliant Browser** – a web browser that supports the open standards published by the W3C such as HTML, XML, XHTML, CSS, and DOM (WaSP n.d.)

**Free-form Annotation** – the ability to “mark anywhere on the document, [without constraining] the shape of the marks, [nor imposing] any structure” (Golovchinsky and Denoue 2002)

**Model-View-Controller (MVC)** – a software development paradigm and design pattern used to provide a logical partition of the system (i.e., model, view, and controller layers)

**On-line** – available over the Internet or an intranet

**Orphaning** – “when the online document changes [and] annotations lose the links to their proper positions within the document” (Brush 2002)

**Plain Old Documentation (POD)** – “a set of simple tags that can be processed to produce documentation in the style of Unix manpages” (Siever, Spainhour, and Patwardhan 1999, 25) used in *Perl* programs

**Process Domains** – different sectors within society or industry – especially in terms of processes and applications

**Reflowing** – the automatic repositioning of elements within an electronic document in response to a change in the document

**Service** – Programmatic units providing access to system functionality such as document creation, annotation creation, user maintenance, etc.

## 1. Introduction

Collaboration through the use of electronic documents is important in today's world. It provides comparatively fast access to information and the input of others, regardless of one's location in the world. However, a number of challenges face those who would more fully embrace this technology. Missing or inadequate software support, version conflicts, centralized access, version control, and other obstacles can greatly reduce the effectiveness of electronic document collaboration.

A number of different approaches have been taken to address these issues for different sectors of society (process domains), such as business and academia. Most implementations were designed to address the problems within a specific process domain, thus dictating the techniques used. However, the underlying meta-process is essentially the same in all cases. This meta-process is the basis for a single process model capable of supporting the annotation and collaboration needs across multiple process domains. In support of this, a system was designed and developed allowing one to:

1. Share electronic documents of differing file formats on-line.
2. Add text and free-form (i.e., handwritten or drawn) annotations to the documents.
3. Control access to the documents through a multi-faceted security system.
4. Archive the documents in both their original and on-line formats.
5. Perform all actions through a standards-compliant web browser.



In other words, one could upload an electronic document to the system and another party, assuming proper credentials, could access and annotate the document from another computer at work, school, home, or even at an airport kiosk.

### **1.1. Statement of Problem**

Electronic documents are becoming ever more prevalent in the modern world. In such an environment, it is necessary to not only be able to quickly and easily exchange documents, but also to annotate them. This has application in a number of different process domains.

Annotation of electronic content provides a powerful tool to aid in collaborative communications. Traditionally, such communication has required proprietary software in all but the most simplistic annotation techniques (e.g., quoting of an original message in an email reply), making such systems impractical for many possible applications because of cost or accessibility concerns.

In addition, different process domains have developed unique approaches to address these issues, often focusing only on immediate needs. However, in many cases, these can be described by the following meta-process:

1. Creation of content
2. Retrieval of content
3. Annotation of content
4. Archiving of content

This thesis suggests many of the approaches used by different process domains can, in fact, be modeled by the single process model shown in Figure 1, which is based

on the previously mentioned meta-process. Simply put, the user can interact with the system by creating, retrieving, or annotating content, which is stored in a data archive. User access is controlled via an authorization sub-system that checks for appropriate credentials. Each action causes the system to present the user with a representation of the handled data.

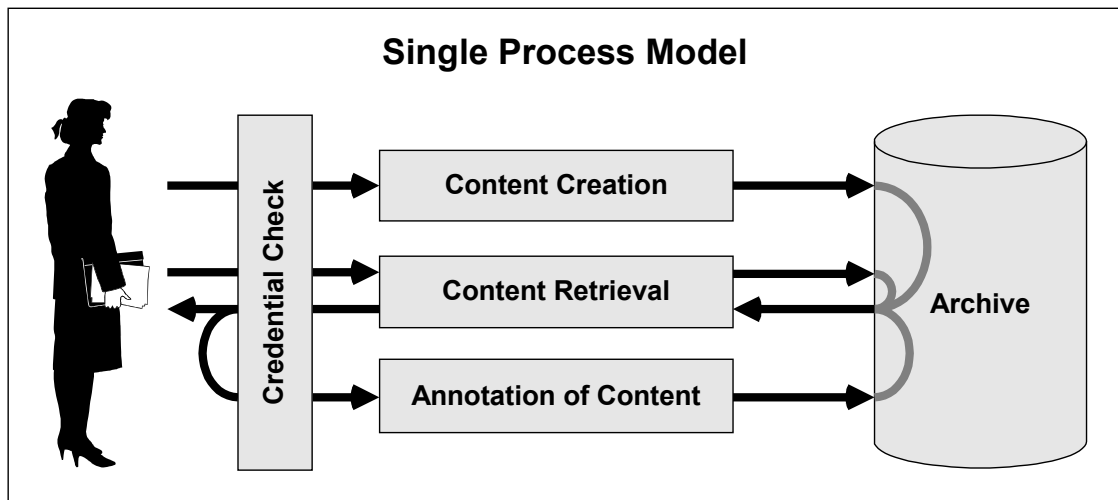


Figure 1: Single Process Model

While it was beyond the scope of this thesis to construct a multi-faceted demonstration valid for all process domains, a system based on the single process model was developed for a single process domain.

### 1.1.1. Usage Scenarios

In order to provide a greater perspective of what applications of annotation of electronic document technology provide and how the different process domains are interrelated, a number of usage scenarios are provided.

#### **1.1.1.1. Document Collaboration**

Joe Engineer works for a design firm located in California. He has just finished the design of a new cog for an important client located in England. He posts the design to the system from his workstation and makes it available to the head of the engineering department, Sally, and the client for comment. Sally, who is at a conference in New York is able to securely log into the system via one of the conference kiosks, which uses a different web browser than Sally normally uses. She notes some minor problems with the design and hand-draws some suggestions onto the document. The client, using one of their workstations, reviews Sally's suggestions and selects one to implement. The client also adds comments to the selected design. Joe, back in California, begins redesigning the problem areas of the cog design, while constantly reviewing the comments from Sally and the client.

#### **1.1.1.2. On-line Grading**

Judy, a journalism student, uses an alternative word processing software package that her English professor does not have. When she finishes typing her paper, she uploads it in its native file format to the system, which converts it to a generic format. Professor Tanaka, using his web browser of choice, is able to read and annotate the paper from home on his PC. Later, Judy accesses the system from her dorm room and is able to review the professor's comments using the web browser on her iBook.

### **1.1.1.3. Discussion Boards**

Jason is new to *Linux* and is having problems understanding all of the different widgets in *The Gimp*. He posts a screen shot to the on-line forum and asks board participants for some help setting the transparency for an image. One helpful user circles the widget Jason needs to use and then provides a basic description of how it is used. Later, several other users decide to highlight some of their favorite widgets and provide basic usage instructions.

### **1.1.1.4. In-class Discussion**

Professor Jackson wants to include more students in class discussion. He creates an in-class question and posts it to the system. During the lecture, he has the students access the question with their laptops. Each student is instructed to submit an answer to the question. Because the students only have access to submit annotations, they are not able to see the responses from the other students. However, as the responses arrive, Professor Jackson is able to review them in real-time on his own laptop, thereby checking the students understanding of the topic being presented. If a particularly interesting or thought-provoking response is found, he is able to bring it up on the digital projector for the whole class to see. This can then be used to spark additional class discussion.

### **1.1.1.5. Review and Accreditation Boards**

Professor Flores is responsible for providing the accreditation board examples of her students' work. She gets into the system and creates a group for the accreditation board. She then reviews student submissions in the system and selects those she wishes to

provide to the accreditation board. She gives their group read access to these documents and their associated annotations. Before the accreditation team arrives, they are provided with access to the system and are able to review the documents and their annotations at their leisure. When the team arrives, they are ready to discuss the students' work and the feedback given by the professor.

#### **1.1.1.6. In-field Presentations**

Joe Engineer is at a different client's site discussing the design for their new product. Unfortunately, the client's computers do not have the software necessary to display Joe's presentation and design. However, Joe submitted these designs to the on-line annotation system using his workstation before leaving his office. On-site, he logs into the system using the client's web browser and is able to continue with the presentation. Part way through, the client makes a comment on the design that Joe does not want to forget. He simply adds an annotation to the presentation while he is giving it. At the same time, the client, who is simultaneously logged into the system, is making other notes on the design. Later, when he returns to his office, he is able to review all the comments on the design in context.

#### **1.1.2. The Big Picture**

As can be seen, there exist a number of different applications for this technology. Even though these applications are in different process domains, they are really trying to solve the same problem from different perspectives. Namely, how does one present electronic content in such a way as to allow controlled access and annotation? The meta-

process and single process model described in Section 1.1 are basic guidelines for the solution to this question. Ideally, a generic solution should be able, with only minor customizations, to address each of the discussed process domains and applications.

### **1.1.3. System Attributes**

A system designed for the single process model would have many desirable attributes. Ideally, it would be able to support the needs of the different process domain applications designed using the single process model. Following is an unordered, generic list of desirable attributes for such systems. However, it should be noted not all applications or process domains would necessarily require all listed attributes.

- No specialized or proprietary software should be required to access or use any system functionality.
- The user should be able to submit documents to the system in a wide variety of file formats.
- The user should be able to annotate the documents using free-form drawing or text entry via a keyboard.
- The user should be able to take advantage of time-shifting when interacting with the system.
- The system should be available at any given time of day or night via the Internet or other intranet.
- The system should maintain copies of the original and annotated document, as well as all annotations.

- The system should support contextual annotations, regardless of the document's original file format. However, the layout of the original, underlying document should not be altered by the addition of annotations.
- Constraints should be designed into the system to control access. Administrative tools should facilitate the maintenance of such constraints.
- The design and implementation of new system extensions to support additional process domains should be simple. For example, model-view-controller design, which will be used for this thesis, supports this by separating the system into logical, independent units.
- In order to encourage continued and future the development, it is desirable that the system and *API* be built with open-source tools and technologies. This allows for improvement and innovation by outside parties without licensing and other legal entanglements.

## 1.2. Research Questions

For this thesis, a system was developed that addresses only a single process domain. However, the following questions outline the issues involved with any annotation system based on the single process model already discussed:

- Can on-line, free-form and text annotation be accomplished using a native, DOM-compliant web browser without additional add-ons, plug-ins, Active-X controls, Java applets, or other such software?
- Can the system accept multiple, common file formats for annotation, including the most widely used document formats?

- Can the system present the document in such a way that annotations can be contextually anchored regardless of the original file format?
- Can the system archive and provide access to not only the annotated document, but also the original document in its original format?
- Can the system be structured in such a way as to allow document owners to control document access through the granting of rights to other users and user-created groups?

### **1.3. Constraints and Limitations**

The system was designed using the single process model described in Section 1.1. Consequently, it should theoretically be able to handle any of the usage scenarios discussed in Section 1.1.1. However, for clarity's sake, it will be discussed in terms of its usage within the academic process domain, with special attention being paid to on-line electronic grading of student submissions. Extensive testing into usage within other process domains was not done. Therefore, such usages are extrapolations based on the technology and other results from this thesis.

Ideally, any given file format should be accepted by the system. However, only a set number of common file formats were implemented. Those file formats classified as common for this thesis are shown in Table 1. A complete list of supported file formats is found in Appendix I.



Table 1: Common Document File Formats

File Extension	File Format
doc	Microsoft Word Document
odp	OpenOffice.org Impress Presentation
ods	OpenOffice.org Calc Spreadsheet
odt	OpenOffice.org Writer Document
pdf	Portable Document Format
ppt	Microsoft PowerPoint Presentation
ps	Postscript File
rtf	Rich-Text Format Document
txt	Plain Text
xls	Microsoft Excel Spreadsheet

## 2. Literature Review

A large number of annotation systems and frameworks have been developed for commercial or research purposes, some of which have been listed in Appendix K. Proposals for systems with annotation facilities were made as early as 1945, with development of such a system occurring as early as the 1960s (Hansen 2006). Since that time, the number of approaches to computer-aided annotation has continued to grow, resulting in a plethora of system and framework implementations. Each of these implementations attempts to address the needs of the user in the given process domain.

This chapter reviews some of the users and process domains shown by literature to benefit from computer-aided annotation technologies. Following this, a number of annotation systems that are directly relevant to this thesis are discussed, along with additional observations made from the literature.

### 2.1. Users

The term *annotation* can be defined in a number of different ways. The definition tends to depend on the type of annotation and the entity that is doing the annotation. These entities can be divided into two groups: human and computer. Both have different reasons for and methodologies of annotation.

### 2.1.1. Human

Humans have been “annotating” their environment since the earliest cave paintings (Hansen 2006). Since that time, reading and viewing media and their associated annotations have progressed from stone, to paper, to digital media. In fact, today an increasing number of people peruse the majority of their reading material in a digital format. (Olsen, Taufer, and Fails 2004)

Human-created annotations play many different roles and are “an important companion activity to reading” (Cadiz, Gupta, and Grudin 2000). Research has shown annotations support a number of objectives in the learning process (Wolfe 2000). As humans read paper documents, they often annotate through the use of underlining, asterisks, and notes, thereby creating a “personal geography” of the material (Marshall and Bush 2002). This activity is a critical part of the *active reading* process, defined as “the combination of reading with critical thinking and learning....” (Schilit, Golovchinsky, and Price 1998). Also, on both cognitive and emotional levels, annotations can affect the reader's response to a document (Wolfe 2000).

Annotations are often not only for personal use. Carter et al. (2006) classified annotation systems into three categories based on the intended use of annotations by the annotation creator:

1. Personal
2. Collaborative
3. Public/social

Personal annotations are intended only for the author, while collaborative and public/social annotations are used by a wider group of individuals. The difference between the latter two is in the control the author has over the group. To some extent, the author has control over who is in a collaborative group, while there is no control in public/social situations.

The intended use changes the actual content of the annotations (Marshall and Brush 2002). Marshall and Brush (2004) found even when the technology provided for the sharing of personal annotations, “only a small fraction ... [were] made public online.”

They also noted:

Usually personal annotations reflect unselfconscious reactions to reading material, while public or shared annotations on a document reflect specific communicative intent. (Marshall and Brush 2004)

### **2.1.2. Computer**

Annotations need not only be for human users. Resources annotated with metadata can be used by computers to support a number of tasks (Wu, Zhang, and Yu 2006). Bulterman provided the following explanation:

The goal of metainformation markup [is] either to assist in content classification (for use in indexing or retrieval applications) or in providing an abstract semantic model of the media object's content for (semi-)automatic processing on the media object. (Bulterman 2003)

In fact, the gathering and use of machine-readable annotations on-line forms the basis for the Semantic Web (Cimiano, Handschuh, and Staab 2004). While the process of creating and organizing annotations may be automatic or human-supported, a framework or

ontology is needed for the results to be useful (Xu, Zhang, and Yu 2006). A great deal of research has been done in the area of defining and evaluating different frameworks and ontologies for such processes.

## **2.2. Process Domains**

Annotation means different things to different entities in different process domains. This section discusses a number of process domains where annotations have been used to support user activity and processes.

### **2.2.1. Academia**

Academia is an area where a large amount of research has been done regarding annotations, their uses, and the effects on students and instructors. This section will look at electronic grading, annotation in the learning process, and the related subject of course management systems.

#### **2.2.1.1. Electronic Grading**

Plimmer and Mason stated a “paperless environment for annotating assignments is appealing to teachers and students” (Plimmer and Mason 2006). Other researchers have found this to be the case, as well. While some students still ardently hold on to traditional methods, most are becoming more comfortable with electronic submission and grading of assignments (Harmon, Helps, and Bailey 2005).

The three major functions of electronic grading were identified by Plimmer and Mason (2006):

1. Unrestricted free-form annotation anywhere on the document
2. Ability to record scores
3. Ability to easily switch between documents

Anchoring problems, discussed in Section 2.4.3, is an issue faced by electronic annotation systems. The two most common anchoring problems deal with the reflowing of documents and the orphaning that often can then occur. When a document with existing annotations is changed, the annotations must be repositioned, in a process called reflowing, to match the new document layout. When a document is reflowed and the annotation system fails “to match an annotation to the correct location,” the annotation is described as being orphaned (Cadiz, Gupta, and Grudin 2000).

Generally, anchoring problems are not a major issue within this process domain because submitted assignments can often be considered fixed in nature (Plimmer and Mason 2006). However, if the annotation method used changes the underlying document layout, problems can arise. The grader must be aware of how such annotations affect the document's layout and, as much as possible, “decide how best to convey remarks while leaving the original submission intact” (Popyack and Herrmann 2003).

#### **2.2.1.2. Annotation in the Learning Process**

As discussed in Section 2.1.1, human reading and learning is strongly supported and influenced by annotations and the annotation process. In fact, “marks made while reading aid understanding, memorization and later retrieval” (Schilit, Golovchinsky, and Price 1998). This section looks at how annotation technology is used in the learning process both in and out of the classroom.

### 2.2.1.2.1. In-class Usage

There are a number of ways annotations can be used in the classroom setting. One such method is individual use for taking notes. Researchers have found taking notes using an on-line notepad during computer-based instruction “can lead to higher achievement than pencil and paper methods” (Wang and Chen 2004).

Another method that has undergone a great deal of research is the use of electronic whiteboards, also known as digital whiteboards, smart whiteboards or interactive whiteboards (Clyde 2004). In environments using such technologies, students have reported positive attitudes and experiences, stating:

... different packages aided effective learning by tackling problems from different perspectives, by supporting memory, and by supporting the teacher's explanation.  
(Wall, Higgins, and Smith 2005)

Annotations made on electronic whiteboards can be captured and saved for future use (Clyde 2004). Projects such as *Total Recall* even allow for annotations created on such whiteboards to be captured to hand-held computers via a wireless network (Holmquist, Sanneblad, and Gaye 2003).

While the electronic whiteboard offers many benefits, including colored text and moving images, there are a number of problems with them as well, such as the difficulty in moving and recalibrating them (Clyde 2004). Other problems include the expense for additional equipment and the possibility of equipment being damaged through misuse or abuse (Holmquist, Sanneblad, and Gaye 2003).

Academia has also benefited from the advent of Tablet PC's. Willis and Miertschin (2004) described the following educational uses for the Tablet PC annotation:

- Note-taking
- Document markup
- Interactive presentations
- Collaboration

These are all discussed elsewhere in this chapter, with the exception of interactive presentations. Interactive presentations allow a lecturer to modify content as needed during the class session. The process has been described in the following manner:

The Tablet PC, when used as a lecture tool, allows the presenter to develop well thought out presentations in advance without being tied to a completely canned lecture. Materials can be developed on the fly, especially useful in derivations and proofs for the sciences, while a complete record of the lecture can be kept for later publication and distribution to students. (Timmins 2004)

Researchers have provided anecdotal evidence that when used in this fashion, the Tablet PC is “mostly preferred ... over a traditional blackboard” (Mock 2004).

#### **2.2.1.2.2. Out-of-class Usage**

While a number of uses exist for in-class use of annotations, use of annotations outside of the classroom also offer a number of benefits for both students and instructors. One such application of the technology is in newsgroups and on-line discussion boards.

Brush noted:

Using a shared annotation system that allows a discussion to be anchored directly on a reading assignment could potentially offer many advantages over discussion boards where the discussion is divorced from the context of the document. (Brush 2002)



Many tools exist to support learning within a given discipline or environment. One such tool was developed for the viewing and annotating of medieval illuminated manuscripts, allowing the users of the system to connect and relate different images in the system based on their relationship one to another (Agosti, Ferro, and Orio 2005).

Another such tool, *Diary Composer*, was developed for childhood education. This system allows for the annotation of video media by children ages four to eight, allowing them to “reflect upon and annotate episodes from their everyday life” (Sevasti and Christos 2000).

Proper use of annotation tools can greatly influence the learning process (Weng and Chen 2004). *Active reading*, discussed in Section 2.1.1, relies heavily on annotation. Speaking of this, researchers have stated the “combination of reading with critical thinking is a fundamental part of education and knowledge work” (Schilit, Golovchinsky, and Price 1998). Indeed, “annotation can serve as a bridge between reading and writing and is often a tangible reflection of a reader's engagement with the text” (Wolfe 2000).

### **2.2.1.3. Course Management Systems (CMS)**

Since the advent of the computer as an educational tool, a large number of different approaches and educational software tools have been used to integrate the computer into traditional teaching methods. Course management systems (CMS) are one such tool. While the definition is not yet fixed, web-based CMS has been defined as:

... a comprehensive software package that supports some or all aspects of course preparation, delivery, communication, participation and interaction and allows these aspects to be accessible via a network. (Collis and De Boer 2004)

Even though CMS do not necessarily provide native annotation facilities, they often provide a framework for collecting and distributing such content within an academic setting (Harmon, Helps, and Bailey 2005). Examples of commercial CMS include Blackboard and WebCT. An extensive list of open-source CMS is contained in Appendix L.

Even though CMS are becoming more prevalent in academia, they have met some resistance from faculty for a number of reasons (Gehring 2003). Some of these include the lack of native annotation capabilities, difficulties in easily presenting information to the external public from a closed system, migration of existing web pages to the CMS, and interoperability issues with third party web development tools.

Researchers have observed that CMS are “new tools for teachers” and “teachers must learn to use them in a technical sense as well as a meaningful sense.” They have identified the “two major learning curves for teachers” (Coolis and De Boer 2004):

1. “learning to set up and manage a web environment that best fits their own course and their students”
2. “learning to design and support new types of learning activities where both students and the teacher take on new roles”

Integration of annotation capabilities into CMS would help support the requirements for effective on-line grading described by Plimmer and Mason (2006).

### **2.2.2. Collaboration**

Collaborative writing and development is an important task in today's world. It is used in business, research, and academic settings to produce results from a multiplicity of

backgrounds and experiences. Weng and Gennari describe the current situation in collaborative work as follows:

An essential part of iterative reviewing and revisions in collaborative writing is annotation – where text is marked up with comments and meta-information by participants in the collaboration. Annotation has been well recognized as an important approach to document-centric discussions. In collaborative writing, however, it is often poorly supported. Much time delay, communication overhead, and cognitive burden is added to collaborative annotation and annotation incorporation. (Weng and Gennari 2004)

Summarizing the work of Cadiz, Gupta, and Grudin (2000), Weng and Gennari listed five major problems impeding the use of collaborative annotation, noting one of the major problems was the required static nature of documents and their annotations (Weng and Gennari 2004):

1. Orphaning of annotations – the failure “to match an annotation to the correct location when the text is edited” (Cadiz, Gupta, and Grudin 2000)
2. Lack of change notification – the need to appropriately notify users when document changes or annotations have been made
3. Unresponsiveness of users – user perspective that “annotations [were] not fast enough when a quick response was required” (Cadiz, Gupta, and Grudin 2000)
4. Non-public nature of annotations – the nature of annotations affected user annotation usage
5. Insufficient richness of annotations – the difficulty of presenting some ideas using only plain-text

Even with these problems, the use of annotations can be very useful in the collaborative writing and development processes. Researchers have suggested a system could be constructed on the Internet, making content “accessible from a web browser anytime and anywhere” (Cadiz, Gupta, and Grudin 2000). They continued:

“In-context” annotations can be tightly linked ... with threads visible in the document, access control to regulate viewing and editing, and a notification subsystem to inform relevant people when new annotations are added. (Cadiz, Gupta, and Grudin 2000)

This notification system is important in many, though not all, process domains. Process domains requiring immediate communication between users often need a notification system. This is because in an electronic annotation setting, the interaction mainly occurs between the user and document, as opposed to between users, often resulting in slow or cumbersome user-to-user communication (Brush et al. 2002). However, in process domains where this immediate communication is not needed, notification is less important. In fact, in some circumstances, users have stated they did not need a notification system or have been frustrated with an incorrectly implemented notification system (Cadiz, Gupta, and Grudin 2000). If needed, a notification system should be user-customizable to support different usage patterns. Such a system can quickly become very complicated (Cadiz, Gupta, and Grudin 2000).

### **2.2.3. Semantic Web**

Handschuh, Staab, and Maedche (2001) quoted Dan Brickley as saying, “The Web is about links; the Semantic Web is about the relationships implicit in those links.”

They then stated the following:

Research about the WWW currently strives to augment syntactic information already present in the Web by semantic metadata in order to achieve a Semantic Web that human and software agents alike can understand. (Handschuh, Staab, and Maedche 2001)

This semantic metadata is effectively a type of annotation, which links often divergent data into a web of information. This can be leveraged by search services (Dmitriev et al. 2006) and social networking applications, as described in Section 2.2.4. Bottoni et al. stated the Semantic Web, once implemented, could improve:

... browsing the Internet for significant content by providing some precise description of such content, so that it can be usefully put in the context of its possible usages, while allowing automated agents [sic] retrieve interesting pieces of knowledge based on such descriptions. (Bottoni et al. 2004)

The developers of *CREAM*, a Semantic Web framework, outlined the areas that must be addressed when developing an effective Semantic Web annotation system (Handschuh, Staab, and Maedche 2001):

- Consistency
- Proper Reference
- Avoid Redundancy
- Relational Metadata
- Maintenance
- Ease of Use
- Efficiency

Effectively addressing these areas helps assure that annotated items are properly referenced and can be effectively used by both human and computer users.

#### 2.2.4. Social Networks

User-supplied tags, comments, and bookmarks in social networks and blogs are examples of on-line annotation. They serve as part of the solution to the “knowledge capturing problem” inherent in the Semantic Web (Handschuh, Staab, and Maedche 2001). The tags and bookmarks in social networks such as *del.icio.us*<sup>1</sup> and *Flickr*<sup>2</sup> provide a treasure trove of semantic information (Wu, Zhang, and Yu 2006).

Research has been done on the use of annotations in the *Friendster*<sup>3</sup> network to allow for improved communication and coordination between users (Appan et al. 2005). This concept can also be seen in the comment capabilities used by many blogs, on-line magazines, and video network *YouTube*<sup>4</sup>. Often, the very nature of these services encourages large number of users to effectively annotate the web semantically by providing a simple user interface that does not enforce predefined formal ontologies or taxonomies on user-defined tags or comments (Wu, Zhang, and Yu 2006).

#### 2.2.5. Ubiquitous Annotation

In earlier times, in order for an annotation to be effective, it had to be physically close or attached to the annotated object. Hansen discussed some of the problems that result when annotation is moved to the digital realm, especially when annotating physical objects or locations in the digital world:

---

<sup>1</sup> <http://del.icio.us/>

<sup>2</sup> <http://www.flickr.com/>

<sup>3</sup> <http://www.friendster.com/>

<sup>4</sup> <http://www.youtube.com/>

However, when it comes to digital annotation systems, it cannot be assumed that the annotations can be added directly to the annotated resource .... (Hansen 2006)

Projects such as *Google Earth*<sup>5</sup> allow users to create annotations for real-world objects represented in a digital format. The *Plasma Poster Network* provide public annotation kiosks for posting of multimedia content in the public square (Carter et al. 2004). The annotation of real-world objects in the digital realm is known as *ubiquitous annotation* and has the possibility of bringing semantic information to users through a variety of user interfaces and devices (Hansen 2006).

### 2.3. Implementations

This section reviews a small number of the annotation systems and frameworks found in Appendix K. Many of these provide ingenious ways of approaching the different issues in the annotation of digital content.

#### 2.3.1. XLibris

Presented in 1998, *XLibris* was a hardware device described by the authors as an “active reading machine” (Schilit, Golovchinsky, and Price 1998). About the size of a large book, this device allows users to not only read electronic documents, but also to annotate them “to organize their reading for later review and retrieval” and provided “different colors of highlighters and pens to increase users' flexibility of expression” (Schilit, Golovchinsky, and Price 1998).

---

<sup>5</sup> <http://earth.google.com/>

*XLibris* has been viewed by some as a benchmark system. One set of researchers praised it as providing “a rich set of annotations and nice mechanisms for summarizing documents and searching for other documents based on the annotations” (Olsen, Taufer, and Fails 2004).

A number of studies have referenced Schilit, Golovchinsky, and Price's research, partly because they summarized a number of reasons why people often prefer reading hard copies of documents (Schilt, Golovchinsky, and Price 1998). These reasons include:

- Tangibility
- Free-form ink annotations
- Page orientation
- Multiple displays
- Sharing
- Legibility

See Appendix K for additional references.

### **2.3.2. ScreenCrayons**

Through the use of screen captures, the *ScreenCrayons* application was developed for “collecting annotations on *any* type of document or visual information from *any* application” (Olsen, Taufer, and Fails 2004 – emphasis added). Because screen captures were used to create a static copy of the digital content, the original file format software was not required to view the content and its annotations. However, this resulted in the following problem:



[A] disadvantage is that some of the non-visible context is lost. This would include portions of a document currently scrolled out of sight. (Olsen, Taufer, and Fails 2004)

Naturally, this can be problematic for certain applications and process domains.

### 2.3.3. MADCOW

Originally introduced in 2004, *MADCOW* hoped to ease the overhead needed for manual (i.e., human interaction required) creation and editing of on-line content annotations problem found in many Semantic Web projects by introducing a simple web browser toolbar plug-in or extension supporting both *Internet Explorer* and *Firefox*, respectively (Bottoni et al. 2004). Extensive annotations were represented by a small placeholder on the web page (Bottoni et al. 2004). An revised version of the plug-in was presented in 2006 by a reorganized research team (Bottoni et al. 2006).

### 2.3.4. Annotea

Originally released in November 2002, *Annotea* has been described as:

... a Web-based shared annotation system based on a general-purpose open RDF infrastructure, where annotations are modeled as a class of *metadata*. (Kahan et al. 2001)

Capable of producing annotations via a keyboard, it is based on a number of W3C<sup>6</sup> standards and is part of the W3C's Semantic Web Advanced Development (SWAD)<sup>7</sup> project (Koivunen 2005). The project also defines a protocol allowing users to: “attach

---

<sup>6</sup> <http://www.w3.org/>

<sup>7</sup> <http://www.w3.org/2000/01/sw/>

data to Web pages such that other users may ... see the attached data when they later browse the same pages ... without modifying the original document” (Swick et al. 2002).

In addition to the framework, the W3C also developed an integrated web browser and editor to leverage the power of *Annotea* and other W3C recommendations (Vatton 2006). This project, called *Amaya*<sup>8</sup>, has been described by researchers as follows:

Amaya is a full-featured web browser and editor developed by W3C for experimenting and validating web specifications at an early stage of their development. Amaya supports CSS, MathML, XHTML, HTML, and also provides a basic implementation of XLink and XPointer. Libwww is linked to Amaya and provides HTTP/1.1 support and an RDF parser. (Kahan and Koivunen 2001)

The *Annotea* and *Amaya* projects show the power of annotation functionality when it is built into the browser as a native feature. Additional related resources for both are listed in Appendix K.

### 2.3.5. Digital Graffiti

*Digital Graffiti* is a project for annotating content on *Plasma Posters*, “large-screen, interactive, digital community bulletin boards that are located in public spaces” (Carter et al. 2004). This project is special in the fact that annotations are made via portable devices, such as PDAs and cell phones. Allowing access through such devices is a very user-centric approach – allow the users to choose their own tools. This is one answer to the following question put forth by the developers:

How can we support people in more easily acquiring multimedia content that is published in public spaces, marking it up, and publishing the marked-up content back to public places.[sic] (Carter et al. 2004)

<sup>8</sup> <http://www.w3.org/Amaya/>

### 2.3.6. Tablet PC

While not technically an annotation system itself, when coupled with *Windows XP Tablet PC Edition*, the Tablet PC provided the necessary hardware and software support for annotating digital content. Tools such as *Microsoft Journal* and *OneNote* provided free-form annotation capabilities on the Tablet PC (Mock 2004). *Journal* even allowed for the “importing” of any printable file format through the use of a virtual printer driver (Willis and Miertschin 2004). However, additional software is needed to view the native file formats of these annotation packages (Timmins 2004). Free-form ink annotations are supported as an add-on to *Word 2002* and natively in *Word 2003 Beta* (West 2003).

## 2.4. Other Observations

This section reviews other observations made in the literature. Most of these topics describe different issues with the crucial integration of “content capture, aggregation and annotation.” (Carter et al. 2004)

### 2.4.1. Book and Paper Metaphors

People have a propensity to mimic the real world inside the computer's digital realm. This can be seen in the desktop metaphor used by almost all GUI-based operating systems. It is therefore not surprising many annotation systems rely either a page or book metaphor. *XLibris'* use of the metaphor is reinforced by the device's form factor and is described as follows:

The metaphor includes a display that emulates the appearance of a sheet of paper, an interface for viewing pages in the linear order of the document, the ability to mark in any place on any page, and cues about the size of a document and about the reader's location within it. (Schilit, Golovchinsky, and Price 1998)

Another system that employs a slightly modified version of this metaphor is *ALT*, which is an A4-size device “enabling users to annotate and sketch on paper in collaboration with a remote peer” (Gabrielli and Law 2003). Here the metaphor is so strong it is used in the actual description of the system by the developers.

*3Book*, another annotation system, displays the digital content as a three-dimensional representation of a book, complete with animated page turns (Hong, Chi, and Card 2005). A unique problem in this case is how to make annotations track appropriately with the page turns.

#### **2.4.2. Free-form Annotation**

Free-form annotation “allows the reader to mark anywhere on the document, does not constrain the shape of the marks, and does not impose any structure on them” (Golovchinsky and Denoue 2002). This need to be able to annotate anywhere on the digital content was reiterated by Plimmer and Mason (2006). The reasoning behind this can be seen in the following:

Although [free-form ink annotation] marks may not have explicit meaning to the computer, they have rich meaning to the user and thus support episodic memory. (Schilit, Golovchinsky, and Price 1998)

Since an annotation system should support the user goals, free-form annotation became very important for human users. See Sections 2.1.1 and 2.2.1.2 for discussions on the effects annotation has on human reading and learning.

### 2.4.3. Anchoring

Even though Golovchinsky and Denoue were discussing free-form annotations when they said the following, it has bearing on all forms of annotation:

From the user's point of view, freeform annotations are attached to a particular representation of some content, like a line of text, a paragraph, etc. We call this content the anchor of the annotation. For some annotations, the anchor is very specific ... [and for] others, it is not ....  
(Golovchinsky and Denoue 2002)

The anchoring of annotations to digital content is a difficult problem as long as the content is allowed to change (Plimmer and Mason 2006). Brush stated:

Robustly positioning annotations so that they are not orphaned by changes to the document is a difficult problem, particularly in the most interesting cases where users are allowed to make annotations anywhere on a digital document and can annotate documents that they do not have permission to modify. (Brush 2002)

He defined *orphaning* as “when the online document changes [and] annotations lose the links to their proper positions within the document” (Brush 2002).

Orphaning is a problem unique to digital content, because “each time the content of a digital document reflows to a new layout, any digital ink annotations must also reflow to keep up with it” and while “real ink annotations often end up in the recycle bin, digital annotations can persist throughout the lifetime of a document” (Barger and Moscovich 2003).

This problem became even more complicated with the need to *life-cycle* annotations (i.e., removal of old annotations that are no longer relevant). When describing their system, Weng and Gennari stated:

A physical annotation can be thrown away easily once we do not need the message on it. In our system, we make annotations detachable, versioned, and life-cycled. Therefore, when an annotation reaches the end of its life cycle, it will be deactivated but archived in the shared space. (Weng and Gennari 2004).

Whatever system is developed, annotations must be controllable and anchored appropriately. Without this, annotations easily lose their contextual or full relevance.

#### **2.4.4. Acetates**

The use of the term “acetates” for describing the approach used by this thesis was originally proposed by Dr. J. Ekstrom<sup>9</sup>. Simply put, annotations are like sheets of acetate that are laid on top of the original document and each other. This was similar to *glosses* described by Zellweger et al. when presenting their *Fluid Documents* system (Zellweger et al. 2001). While they said this approach would work, Hong, Chi, and Card (2005) warned it can give rise to “Z-fighting, a classical 3D graphics problem” if the layers are not carefully constructed.

#### **2.4.5. Annotation Clients**

Appendix K lists over eighty annotation implementations, most of which are proprietary. While many were created only for research purposes, there was the question

---

<sup>9</sup> Dr. J. Ekstrom of Brigham Young University, Ira A. Fulton College of Engineering and Technology, School of Technology, Information Technology, personal communication, March 2002, Brigham Young University.

of how an annotation system can become generally accepted. Researchers have put the problem facing many framework developers in context:

“Even when the interface to the server is public, the small installed base of a single system does not encourage external development of clients.” (Kim, Slater, and Whitehead 2004)

Another problem facing the developers of annotation clients was being locked into a given technology or paradigm. Many annotation clients require a specific operating system or web browser to work correctly. A subset of the problem has been succinctly described in the following manner:

“An annotation system that depends upon specific application implementations is awkward to use and is frequently deimplemented[sic] by new releases of the software.” (Olsen, Taufer, and Fails 2004)

If annotation is to become pervasive, an approach needs to be used that provides a client that can be easily accessible to all.

#### **2.4.6. Bookmarks**

Weng and Gennari (2004) identified bookmarking as one of the attributes of physical annotations. While discussing the *Digital Graffiti* system, which lacked this functionality, users “expressed interest in saving bookmarks to review and edit later...” (Carter et al. 2004). An annotation system which relies on the paper or book metaphor described in Section 2.4.1 should provide bookmarking functionality either by annotations or other means.

### 3. Methodology

With any non-trivial system, design decisions have a significant effect on the resulting product. As can be seen in Chapter 2 and Appendix K, a large number of different annotation implementations exist, each with its own paradigm and approach to the problem of digital annotation. A single design cannot include or support all possible approaches. This chapter explains what design decisions were made, the reasons for these choices and what effects they had on the resulting system. To simplify things, the system will be called the *On-line Annotation System (OAS)* throughout.

#### 3.1. Model-View-Controller (MVC) Design

The model-view-controller software design pattern was used because it provides a logical partitioning of the system. The basic component layers and their related implementations are as follows:

- Model Layer – The underlying database and file system for the system, as well as the associated web service daemons (e.g., *Apache*) and system security.
- *API* – *Perl* software linking the Model Layer to the View Layer.
- View Layer – *Perl* scripts and *Apache* modules to provide access through the *API* to the Model Layer to the Controller Layer.
- Controller Layer – A standards-compliant web browser.



It should be noted this is only one possible way to implement this or a similar system. However, the modular design allows for system components to be easily updated or replaced completely.

### 3.2. Development

This section describes the hardware and software packages used in the development of *OAS*.

#### 3.2.1. Hardware and Operating Systems

The *OAS* software could have been developed on any modern *Linux* desktop computer or server. The actual development system consisted of a *Fedora Core 5 Linux* installation built on an HP Compaq dc7100 SFF desktop computer system with the following configuration:

- Intel Pentium 4 CPU 3.20GHz
- 1GB of RAM
- 36 GB Harddrive (Samsung SP0411C ) formatted *Ext3*

*Linux* was used because of the availability and stability of open-source tools such as *Apache*, *MySQL*, and *Perl*, which can be integrated tightly together. In addition, a number of open-source file format conversion tools are also readily available, which reduced development time.

Also, a second server with *Windows XP Home Service Pack 2* was used to handle the conversion of the *Win32*-specific file formats (e.g., *Microsoft Office* formats). It was a Dell Inspiron 8200 laptop with the following configuration:

- Intel Pentium 4 Mobile CPU 1.60GHz
- 1GB of RAM
- 36 GB Harddrive (Toshiba MK6021GAS) formatted *NTFS*

While *OpenOffice.org* on *Linux* can be used to convert many *Win32*-specific file formats, native tools provide cleaner conversions. In addition, there are certain file formats, such as *Microsoft Journal*, for which *Linux* tools do not yet exist. Even though this particular file format was not addressed in this thesis, it was important to provide the system architecture to handle this and other *Win32*-specific file formats in the future.

### 3.2.2. Web Server

For *OAS*, an *Apache v2.2.x* series web server was used as the system web server. This server was chosen because it is stable and supports dynamically linked modules. The two modules that provided specific functionality to *OAS* are as follows:

- *mod\_perl2* – Discussed in further detail in Section 3.2.4.1, this module provided faster response times than that which is available in a standard *CGI* environment.
- *mod\_auth\_mysql* – This module allowed authentication credentials to be stored in a *MySQL* database, which simplified the design and administration of the system by placing credentials in a single location.

Because of its flexibility, *Apache* provided a good base for *OAS*. By leveraging the functionality of the two modules noted above, system response time and security were improved over a normal *CGI* environment.

### 3.2.3. DBMS

While any number of *DBMS* could have been used for *OAS*, *MySQL v5 Community Edition* server was chosen. This *DBMS* not only had support for *Perl 5*, but also the *mod\_auth\_mysql* for *Apache*. While discussing *MySQL*, which is open-source, Di Giacomo<sup>10</sup> stated “the open source products have achieved enterprise-level quality” and “MySQL's speed and reliability have made it a popular alternative to proprietary database systems” (Di Giacomo 2005).

### 3.2.4. Programming Languages

As with many systems built to work over the web, a number of different programming languages and technologies had to be utilized in order to get the desired result. Server-side and client-side programming will be discussed separately.

#### 3.2.4.1. Server-side Programming

For the server-side programming, *Perl 5* was chosen as the programming language for the majority of *OAS*. *Perl 5*, with its long tradition of web system development, had many advantages, including the following:

- *Language Integration* – *Perl 5* has the ability to integrate other languages into it. This allows the programmer to leverage the advantages of other languages. For example, wrapper classes for *C* libraries can be written, allowing *Perl 5* to take advantage of the speed of the *C* library, which is often considerably faster than native *Perl 5*. This was particularly important for the graphic routines in *OAS*.

---

<sup>10</sup> Los Alamos National Laboratory Research Library's *Library Without Walls* team member

- *Available Third-party Modules* – A large number of third-party modules are available for *Perl 5*. The largest such repository is CPAN (the Comprehensive *Perl* Archive Network), which currently has over 10,000 open-source modules available for download. This archive provided the needed modules for interfacing with *Apache*, *MySQL* and *Image::Magick*.
- *Text Manipulation* – *Perl 5* has a number of built-in tools for text manipulation, which is one of its biggest strengths (Dominus 1998). This includes a powerful regular expression engine (Siever, Spainhour, and Natwardhan 1999, 63-70). In addition, multiple string creation techniques exist. This allows for text, such as *HTML* code, to be in-lined directly into the *Perl 5* source code with limited delimitation issues.
- *Taint Checking* – *Perl 5* supports taint checking. This means data coming from a source external to a *Perl 5* program is tainted and is not allowed to be used in a way that can affect an external system (e.g., file system, system call, etc.) without first being verified via a regular expression. This provides an excellent tool for forcing data verification. This feature was especially important in the design of web-accessible portions of *OAS*, as many traditional hacker attacks have relied on sending malformed or unexpected data.
- *mod\_perl2* – One of the largest advantages of *Perl 5* was the ability to integrate it directly into the *Apache* web server, using *mod\_perl2*. This allowed *OAS* to be built using *Apache* response handlers, instead of only *CGI* scripts. Handlers, which were compiled when the server started, could share database connections,

memory, and compiled module instances (Bekman and Cholet 2003, 508-510, 549). This removes the overhead found in the standard *CGI* model. In addition, *mod\_perl2* gave direct access to all stages of the *Apache* Request model, providing complete control over all phases of request handling (Bekman and Cholet 2003, 720-721).

- *Cross-platform* – Ports for *Perl 5* exist for all major system platforms. For this system, *Perl 5* was used both on the *Linux* and *Win32* servers.
- *Flexibility* – One of the main design concepts of *Perl 5* was *TIMTOWTDI* (There Is More Than One Way To Do It). The language has a rich syntax, though archaic at times, which allowed the programmer multiple paths to complete a given task. The method used can be optimized for the design goals of the current task. While this has many advantages, it can cause problems as well, which are discussed later in this chapter.

Many of the advantages of *Perl 5* have been discussed. However, there were a number of drawbacks, as well. As many of these have a direct effect on the system and the development process, they are discussed below. Also included are possible approaches for mitigating the effects.

- *Speed of Execution* – Because *Perl 5* has to be interpreted and then compiled for each execution, its speed suffers (Dominus 1998). However, through the use of *mod\_perl2*, this overhead can be greatly reduced. With a built-in *Perl 5* interpreter, *Apache* loaded, compiled and stored the handler once, as opposed to the multiple times used for a traditional *CGI* request. Because the majority of

services could be run as *mod\_perl2* handlers, only a small number of services required this overhead.

- *TIMTOWTDI* – *Perl 5*'s rich syntax can be quite archaic at times. Consequently, programmers often use only a subset of the language, which leads to the creation of language dialects (Dominus 1998; Maher n.d.; Wall n.d.). This can lead to problems for future development and support, as individual programmers can not be guaranteed to “speak” the same dialect. Therefore, the following style guidelines were established to help mitigate this problem:

1. Procedural sections were to resemble *C/C++* code as much as possible in terms of style and construct.
2. Descriptive variable and function names were to be used throughout. Names of internal functions were to begin with one or more underscore(s).
3. All complex regular expressions were to use the extended format allowing for white-space and comments.
4. Code was to be modularized as much as possible.
5. Plain Old Documentation (*POD*) was to be used in all *API* modules.  
Documentation was to include the arguments and return types for all functions, including those internal to the module. Context-sensitive or variable return types and determination factors were to be clearly stated.
6. Functions with context-sensitive or variable return times were to be used as little as possible. Generally, they were to be restricted to internal functions.
7. Large data structures were to be passed by reference.

8. All constants were to be defined as such and contained in a single location in each logical programming unit.

9. Dynamic scope usage was to be minimized.

While *Perl 5* had some disadvantages, these could be addressed through careful design and development. Therefore, *Perl 5* was chosen as the language of choice for the majority of the server-side programming. As an additional note, a limited number of *sh* shell scripts were also used for document conversion and development support.

#### **3.2.4.2. Client-side Programming**

Because one of the main goals of *OAS* was that it will run on any given standards-compliant web browser, *JavaScript* was the natural choice for the client-side programming language. All major and most minor web browsers natively supported either *JavaScript* or *JScript*, which is generally interchangeable with *JavaScript*.

While not technically programming, markup played an integral part in not only presentation, but also interacting with the client-side programming to provide a vibrant interface. Markup was done using *HTML* and *XHTML* in conjunction with *CSSv2* style sheets. User- and system-controlled dynamic content was controlled through *DHTML*, the interaction of *JavaScript* and *CSSv2* style sheets.

#### **3.2.5. Software API**

As this is a non-trivial system, an *API* was almost an absolute necessity in the development of *OAS*. An *API*, written in *Perl*, supported the implementation of a number of important system design goals:

- Code modularity
- Standardized documentation
- Maintainability improvements

The *API* also provided a platform for extending *OAS* in the future by providing the basic system tools and system framework.

### 3.2.6. Services

Services for *OAS* were created as either *CGI* scripts or *Apache* handlers. These provided the user access to system functionality such as document creation, annotation creation, user maintenance, etc.

Stored in the database, service information was used to create dynamic menus and system redirects. It also facilitated the creation of administrative tools for updating or changing services and service information.

### 3.3. Testing

A series of tests were needed to ascertain web browser compliance and system performance. Each is discussed in its own section.

#### 3.3.1. Compliance Testing

One of the main requirements was that *OAS* work with any standards-compliant web browser. As it was not feasible to test every browser build on every operating system, a set of standards-compliant web browsers on multiple operating systems was chosen based on information in . In order for a web browser to be considered compatible



with *OAS*, the user had to be able to perform all of the following tasks using the web browser:

- Log into *OAS*
- Open and view documents
- Create new documents
- Create free-form annotations
- Create text annotations
- View annotations
- Edit text annotations
- Delete annotations
- Delete documents

Each of these tasks was considered to be atomic, with only two considered states: successful or unsuccessful. As each of these tasks deals either with submitting data and/or rendering the resulting *HTML*, either the information was submitted correctly and the *HTML* was rendered correctly or else it was not. A case could be made for a more specific metrics on the creation of free-form annotations. However, due to the complexity of the different scenarios posed by the multiplicity of web browsers, operating systems, and hardware combinations, an issue discussed more thoroughly in Section 4.4.3, this would constitute a full study in and of itself. Therefore, if a reasonable free-form annotation rendering could be created, this task was considered complete.

### 3.3.2. Load Testing

While not an integral part of this thesis, it was desirable to know how *OAS* would react under load. In order to ascertain this, automated load testing was used. The load testing was to mimic a scenario where a class of students were all simultaneously accessing a multi-page document in *OAS* and annotating it.

A test script was developed using *Perl 5's LWP* modules, which created virtual web users. Each virtual web user would do the following three steps, which mimic the actual chain of actions that occur with a real user using a web browser:

- Download a random page in the test document.
- Add a random sample annotation to the page.
- Re-download the page.

This process was done multiple times from multiple hosts and the results analyzed.



## 4. Results

*OAS* was designed and implemented pursuant to the description in Chapter 3. In general, *OAS* performs as expected, though the results of this research have been mixed. This chapter details how well *OAS* met the design goals presented by the research questions in Section 1.2.

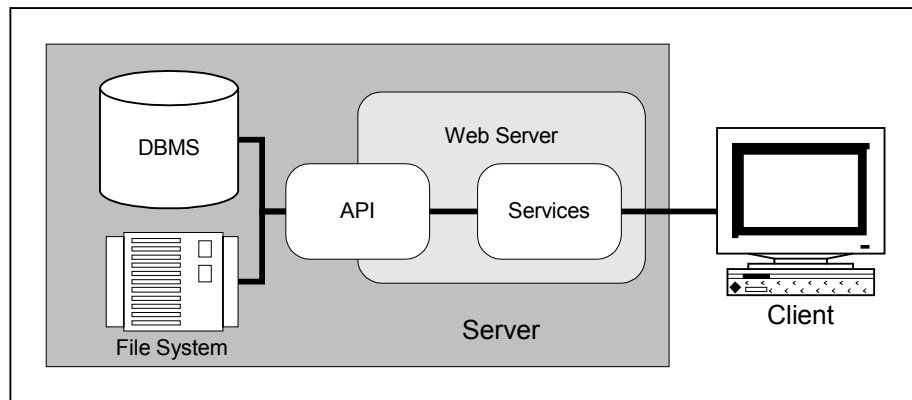


Figure 2: Basic System Overview

Figure 2 shows the basic layout of the *OAS* system. The client connects to the services provided by the web server. Services interact with the *DBMS* and the file system via the *API*. The *API* is partially included in the web server because the web server worker processes compile the *Apache* handler services and their associated *API* methods

as part of their initialization. However, the *API* is still available to programs external to the web server.

As the development of *OAS* progressed, security and user authentication became one of the most important parts of the system. The underlying security implementation and paradigm had a large effect on the development of the overall system. Consequently, security and authentication will be discussed first, thereby giving the reader a basis for the discussions which follow.

#### **4.1. Security and Authentication**

Security is an important consideration when designing and implementing any on-line system. This section outlines the security measures that were taken with *OAS*, from basic system security through controlled access to system resources.

##### **4.1.1. System Security**

Because *OAS* was used only for academic research, the *Linux* server was not hardened to the level of a bastion host, which would have required each publicly-accessible service to be relegated to its own *chroot* file system (Bauer 2003, 40-97). However, several important security measures suggested by Bauer were implemented. Following is a list of some of the steps taken to secure the *Linux* server.

- All unnecessary operating system daemons were removed or disabled.
- *OAS*, *Apache* and *MySQL* were all given their own user and group. Only the *OAS* user had shell access, which was only given because it was necessary to spawn *vncserver* for the conversion of *OpenOffice.org* document formats.

- File ownership and permissions were set restrictively in the *API*, *OAS* directories and *Apache* directories.
- Inbound connections to the database were limited to *localhost*.
- Connections to the *vncserver* and *X* were limited to *localhost*.
- *Perl 5*, which has automatic buffer-overflow protection and taint checking, was used for all *OAS* services.
- *ipchains* was used to limit access to operating system services.

While these measures do not cover all possible avenues of attack, they do provide a reasonable level of protection for the system in the given environment.

#### 4.1.2. User Authentication

Several methods of doing user authentication were explored during development. All were variations on the standard basic or digest authentication models. The final solution relied on *mod\_auth\_mysql* (described in Sections 3.2.2 and 3.2.3), which utilizes the *MySQL* database. Users' information and login credentials (username and encrypted password hash) were stored in the *person* table. Each request for protected content triggered a database lookup, which compares the presented credentials against those stored in the database.

When a user tries to access a restricted resource, the user is prompted for credentials through a browser-generated login screen, such as the one shown in Figure 12. Per the method outlined in RFC 2617<sup>11</sup>, credentials are automatically transmitted by

<sup>11</sup> Frank, J. et al. "HTTP authentication: basic and digest access authentication." *Network Working Group*. Standards Track. June 1999. Available at <http://www.ietf.org/rfc/rfc2617>; accessed 9 October 2006.

the web browser and checked by the web server for every transaction. Therefore, the user is only presented with the login screen once.

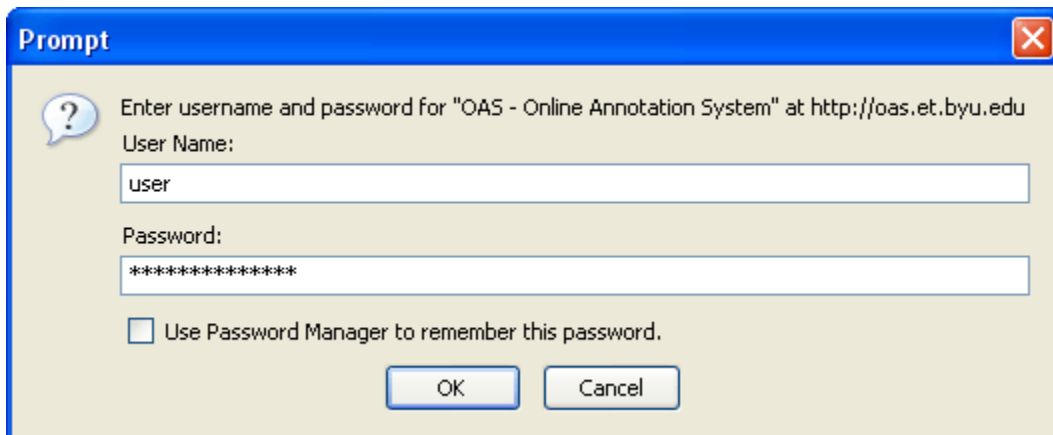


Figure 3: Login Prompt

When a document is created, a *.htaccess* file was created in the document's directory in the file system. Therefore, authenticated users who do not have access rights to a certain document can not by-pass the document display handlers and directly access the document's files on the web server.

Because the client in this method of user authentication stores the credentials in the current browser instance, logging a user out of the system can be complicated. If a second browser window was open, simply closing the window containing the system would not remove the credentials. In order to address this, a service was created which forces the web browser to present invalid credentials to the server. When these credentials are rejected, the web browser clears all credentials for the authentication domain, effectively logging the user out of *OAS* (Documet 2006).

While this user authentication method worked, it does have a few drawbacks. These and alternative approaches for addressing them are discussed further in Sections 5.5 and 5.8.3.

#### 4.1.3. Document Permissions

All users are allowed to create documents. However, once a document is created, access to it is controlled by its own set of document permissions, which are administered by the document owner. Table 2 shows these permissions, along with their standardized abbreviation and a brief description. These permissions are only for documents, and do not extend to other system facilities. The processes by which permissions are granted and revoked is described in Section 4.5.3.

Table 2: Document Permissions

Abbrev.	Permission	Description
R	Read Document	The user has the right to view the document.
D	Delete Document	The user has the right to delete the document. The deletion is permanent, and cannot be undone.
A	Annotate Document	The user has the right to add new annotations to the document.
Ra	Read Annotation	The user has the right to read annotations owned by other users. The user always has rights to read annotations they own.
Da	Delete Annotations	The user has the right to delete annotations owned by other users. The user always has rights to delete annotations they own.
M	Moderate	The user has moderator rights and can see who owns what annotations.



Enforcement of permissions is generally left to the scripts and handlers, though the *API* does enforce it to some degree. As would be expected, the document owner and administrative users are granted all rights automatically by the system.

#### **4.1.4. Multi-tier Access (Groups)**

Multi-tier access simply means that groups can be created that also have document permission rights. The model used for *OAS* follows the classic user/group model. Users can be granted permission rights. A user could be a member of any number of groups, each of which can be granted specific rights. The user inherits all document permissions granted to those groups. The effective document permissions rights of a user is therefore the union of all document permission rights granted individually to the user and to any groups in which the user is a member.

Groups can not be members of other groups.

All users have rights to create groups. The user who creates the group becomes the group's owner and has the ability to control group membership. This allows for a great deal of flexibility in *OAS*. Following are some examples of how this could be used in an academic setting:

- Classes are constructed as a group in the system. Class document may only be read by members of the classes group.
- The instructor allows students to form their own study groups. A member of the group posts a blank document to use as a graphical wiki and sets the document permissions such that only the group can read or update the file.

- The instructor uses *OAS* to create a discussion group for all of his classes. She sets the permissions such that all of her classes have read, annotate and read annotation permissions.

As can be seen, the multi-tier access paradigm combined with the fact that all users can create and manage groups opens a number of possibilities.

A special *ADMIN* group was created for system-wide administrators. This group has no owner, but each administrator can control group membership. Administrative users have wide-ranging power on the system. Because of this, it is preferable for Administrators to use an unprivileged user account for normal system usage.

## **4.2. Document Acceptance**

One of the main goals of *OAS* was to allow users to submit documents in a wide range of file formats. There are literally thousands of data formats in existence. Table 3 shows the file formats supported by *OAS*. Multiple plain-text formats (e.g., source code) are listed because of context-highlighting applied by *OAS* during the conversion process.

### **4.2.1. Interface**

When a user selects *New* from the system's *File* menu, the user is presented with the dialog shown in Figure 4. A standard *HTML* upload form is used, which allows the user to select the document for submission from their local file system.

All documents in *OAS* are rendered and displayed as images. Therefore, the user can select the render quality for many file formats. The higher the quality, the clearer the

fonts, the larger the file and the more screen real estate is needed. Unfortunately, this option only affects documents utilizing the *PS/PDF* Strategy described in Section 4.2.2.4.

Table 3: All Tested File Formats

File Extension	File Format
c	C Source Code
c++	C++ Source Code
css	CSS Source Code
csv	Comma-delimited Spreadsheet
doc	Microsoft Word Document
gif	GIF Image
h	C/C++ Header File
htm/html	HTML Source Code
jpg/jpeg	JPEG Image
js	JavaScript Source Code
odg	OpenOffice.org Draw Drawing
odp	OpenOffice.org Impress Presentation
ods	OpenOffice.org Calc Spreadsheet
odt	OpenOffice.org Writer Document
pdf	Portable Document Format
pl	Perl Script Source Code
pm	Perl Module Source Code
png	PNG Image
ppt	Microsoft PowerPoint Presentation
ps	Postscript File
rtf	Rich-Text Format Document
sh	Shell Programming Source Code
txt	Plain Text
wpd	Word Perfect Document
wrl	VRML Source Code
xls	Microsoft Excel Spreadsheet

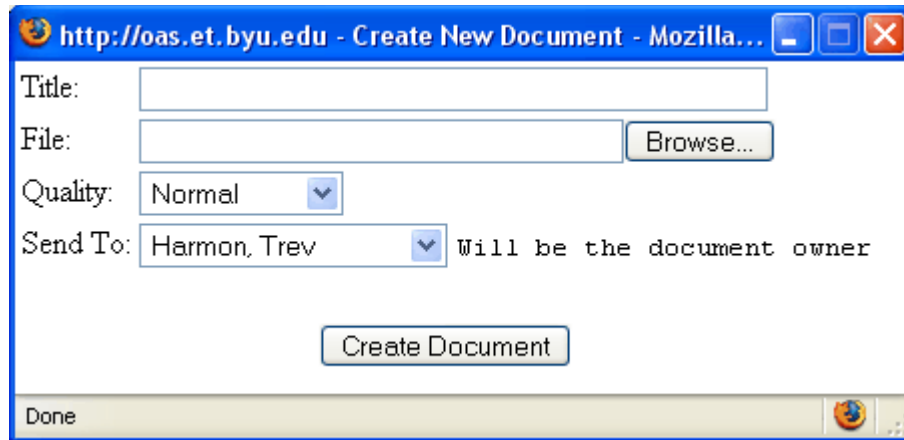


Figure 4: New Document Dialog

Users can also choose to “send” the document to a different user. This option effectively allows the user (the author) to specify a new owner for the document. This mechanism can be used in instances such as assignment submission in an academic setting. In this case, the instructor is selected as the document's new owner, while the student would remain its author.

#### 4.2.2. Conversion

When the user submits a new document, it has to be rendered to one or more image files. This rendering is accomplished by a group of conversion programs (converters). In addition to rendering documents to image formats, the converters also handle the necessary database and file system updates.

When a document is received by the server, the file format is determined using the file extension. The appropriate converter is then determined by a database lookup using that extension. The document is then sent to this converter, where it is rendered to one or more images. Several different techniques were used to accomplish this process.

#### 4.2.2.1. Linux-handled Formats

The majority of file format conversions are handled by the *Linux* server. This includes all formats except for some *Microsoft Office* file formats. Most converters change the document into either a *Postscript* or *PDF* file, which is then sent to the *PS/PDF* converter for final conversion, a process described in Section 4.2.2.4. For some image formats, the files are simply copied to their new location and the necessary database updates made.

#### 4.2.2.2. OpenOffice.org Formats

*OpenOffice.org* documents proved to be a special challenge, as a bug in version 2.x did not allow for headless *OpenOffice.org* servers to directly create *PDF* files. Consequently, *vncserver* was used to create a virtual *X* server where a full version of the *OpenOffice.org* server could be instantiated. This posed some security issues, as the virtual *X* server created with *vncserver* was owned by the *OAS* user id but needed to be accessible by the *Apache* user. The risk was minimized by locking inbound connections to the virtual *X* server to *localhost*.

With the virtual *X* server in place, the converter can load *OpenOffice.org* into a full GUI environment. As part of the command line, an *OpenOffice.org BASIC* macro is called, which handles the conversion of the document to *PDF*. This is then converted to the final format using the *PS/PDF* Strategy described in Section 4.2.2.4.

As an additional note, *OpenOffice.org* can convert all of the *Microsoft Office* formats. However, it was found these conversions were generally not as good as those produced using native *Microsoft Office* programs. The only exception to this was

*PowerPoint*, which was rendered using *OpenOffice.org*. Therefore, the *Win32* server with *Microsoft Office* installed is used for these conversions.

However, if need be, *OAS* can run without the *Win32* server, utilizing *OpenOffice.org* for the *Microsoft Office* formats with a degradation in performance and output quality based on the version of *OpenOffice.org* used for the conversion.

#### **4.2.2.3. Win32 Formats**

In order to get the highest quality output from *Microsoft Office* document file formats, a secondary conversion server was created on a *Windows XP* system. This conversion server was written using *ActivePerl* and utilized the *Win32::COM* module to access the application objects for the different *Microsoft Office* applications. The application objects allowed *API* calls to be made to handle the conversion of the documents. The process follows the following steps:

1. The original document is received by the *Windows* conversion server.
2. Based on the file extension, the application object is created and bound. *API* calls are used for the following:
  - i. The document is opened.
  - ii. The document is printed to a file using a *Postscript* printer driver.
  - iii. To avoid system hangs caused by a dialog initiated by automatic file conversions, the document is saved.
  - iv. The document is closed.
3. The resulting *Postscript* file is returned to the main *OAS* server.
4. The *Windows* conversion server waits for a new connection.

As stated in Step 3, the document is returned to the server as a *Postscript* file, which is passed to the *Postscript* converter for final conversion. This approach can be expanded to support any *Win32* file format that supports printing to a file through its application object.

There is a risk in relying solely on the file extension for file format identification. However, this is a widely used convention, and so the risk is minimal. An alternative approach is to process each file looking for key characteristics or markers in an attempt to determine file type. This process can be complicated and is not completely accurate either. Therefore, the simpler solution relying on convention was chosen for *OAS*.

#### **4.2.2.4. The PS/PDF Strategy**

Because many document programs cannot render directly to an image format, it was found converting to *Postscript* or *PDF* is the best approach with current technology. For document programs without *PDF* creation support, the document is printed to a file using a *Postscript* printer driver.

*Ghostscript* is used to convert the *Postscript* or *PDF* file to the *JPEG* image format. While rendering, *Ghostscript* performs the necessary pagination, splitting a multi-page document into multiple images. Effective and reliable pagination was one of the main reasons most converters in *OAS* utilized this strategy, even if another was available.

#### **4.2.3. Archival Copy**

When any document is submitted to *OAS*, an archival copy in the original file format is created in the file system. This allows users with read access to get a copy of

the original document, without any annotations added in *OAS*. This feature satisfies the requirement to have access to the original document.

When a document is deleted from *OAS*, the original copy and all related annotations are also removed.

### **4.3. Viewing Documents**

When a document is accepted and converted by *OAS*, it is available for viewing by all who have appropriate document permissions. This section discusses the several view modes provided by *OAS*. Once the user opens the document using the dialog shown in Figure 5, all view modes accessible to the user are listed in the menu system under *View*. Also, each view of each document has a unique URL, allowing for browser-level bookmarking for later access or reference.

#### **4.3.1. Display Mode**

Display Mode is the default view when a document is opened or successfully converted by *OAS*. This mode, shown in Figure 6, simply displays the converted document image. While the user can move between pages, no annotations are displayed. This is also the only mode where the document image can be accessed by a *right-click* in the web browser.



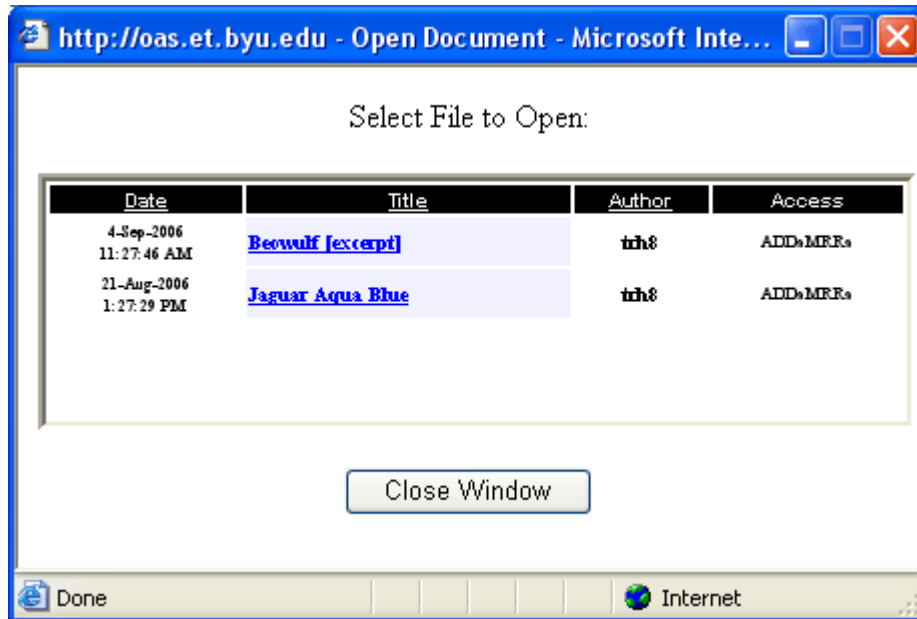


Figure 5: Open Document Dialog

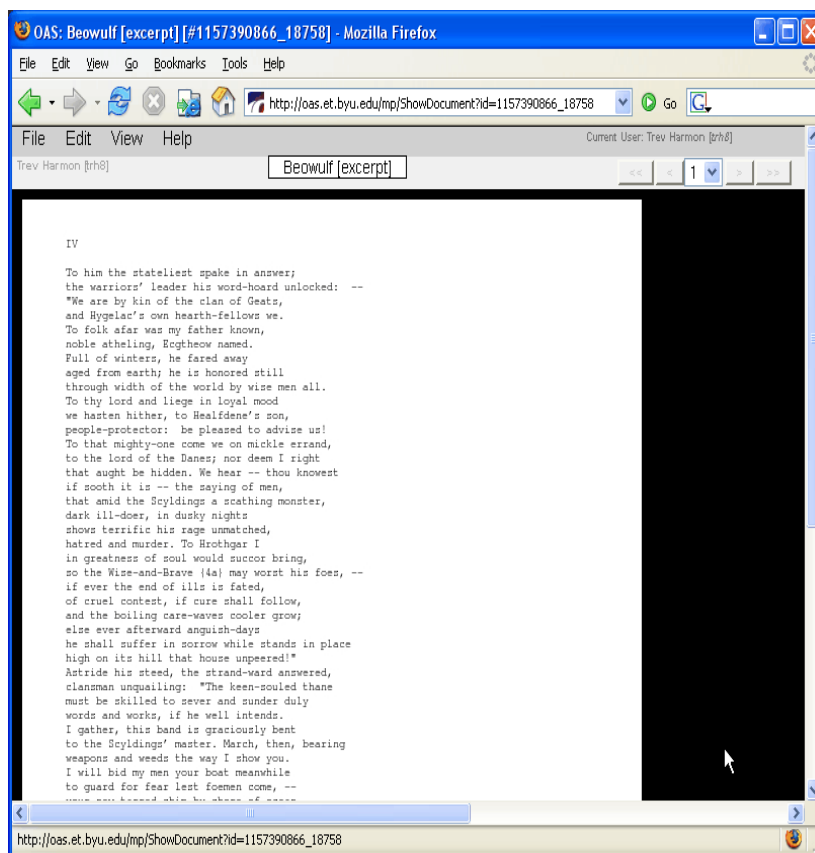


Figure 6: Document Display Mode

### 4.3.2. Read Mode

Read Mode is used to view the document and the annotations for which the user has permissions. A sidebar is added, which allows the user to manage the annotations' visibility and to edit text annotations (per permissions). Figure 7 shows a document being viewed in Read Mode.

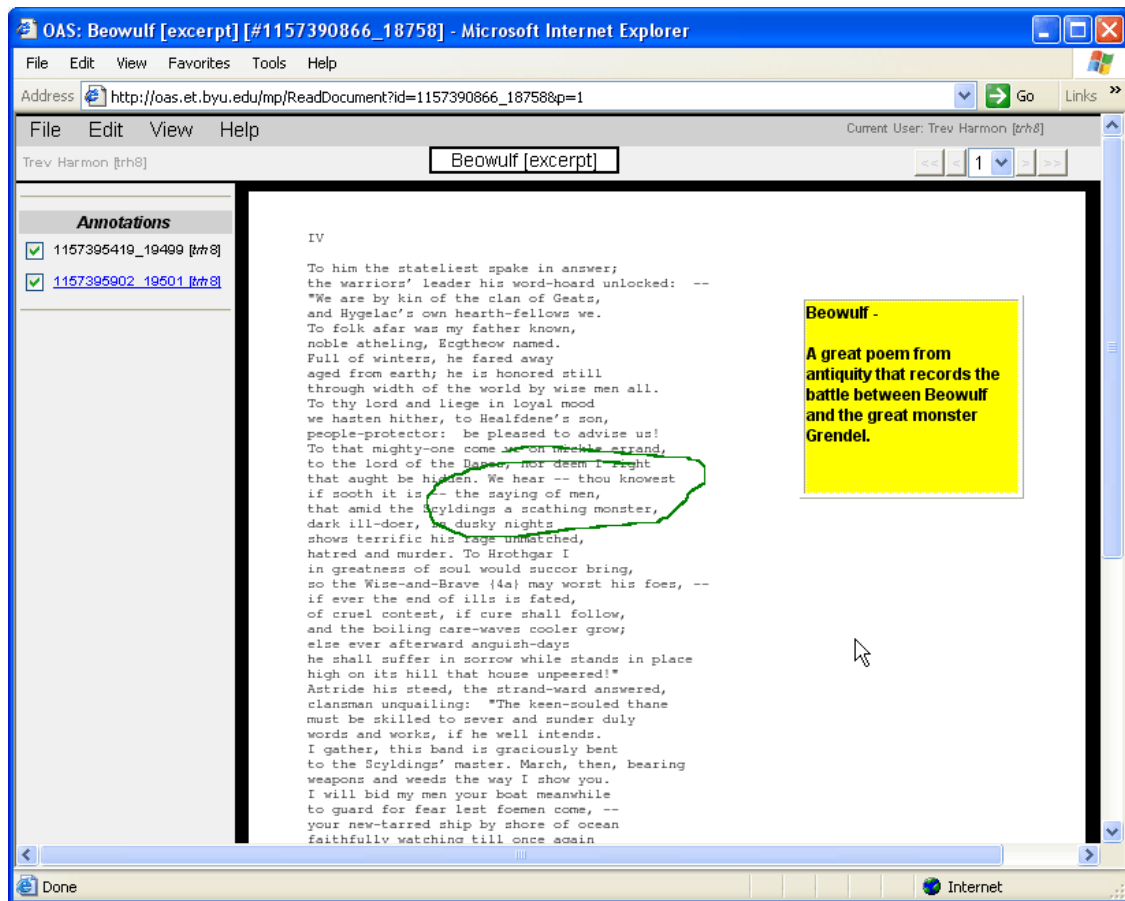


Figure 7: Document Read Mode

This document has two annotations. The green circle is a free-form annotation, while the yellow box is a text annotation. Both annotations are listed on the left, along

with their ID number and a *checkbox* that controls their visibility. The second annotation ID is a hyper-link because the user viewing the document is also its owner. Clicking on this link will allow the user to edit the annotation's content.

Annotations are positioned in the document through the use of *DHTML*. Specifically, display elements are grouped within `<div>` tags, which can be positioned using *DHTML*. The *z-index* refers to the element's layer. An element with higher *z-index* value was stacked above one with a lower *z-index* value. This process can be described as layering sheets of acetate, each of which contains an annotation, on top of the document and each other. Figure 8 shows the *z-indexes* that were by *OAS*.

As can be seen, the document's image was actually the web page's background image. Menus, sidebars, etc. were located in *z-indexes* with a value over 1000. Annotations were auto-stacked by the web browser in the region below this. This allowed for a large number of annotations on a single page. Annotation visibility was controlled by toggling the element's *visibility* property. When multiple elements located on the same position on the page are visible, the element with the larger *z-index* appear to float over elements with smaller *z-indexes*. Consequently, even though the Text Annotation Control element resides in the same area of the screen as annotations, it will always appear on top when visible because it has the highest possible *z-index*. All elements are arranged in the stack in such a way as to allow proper access and layout.

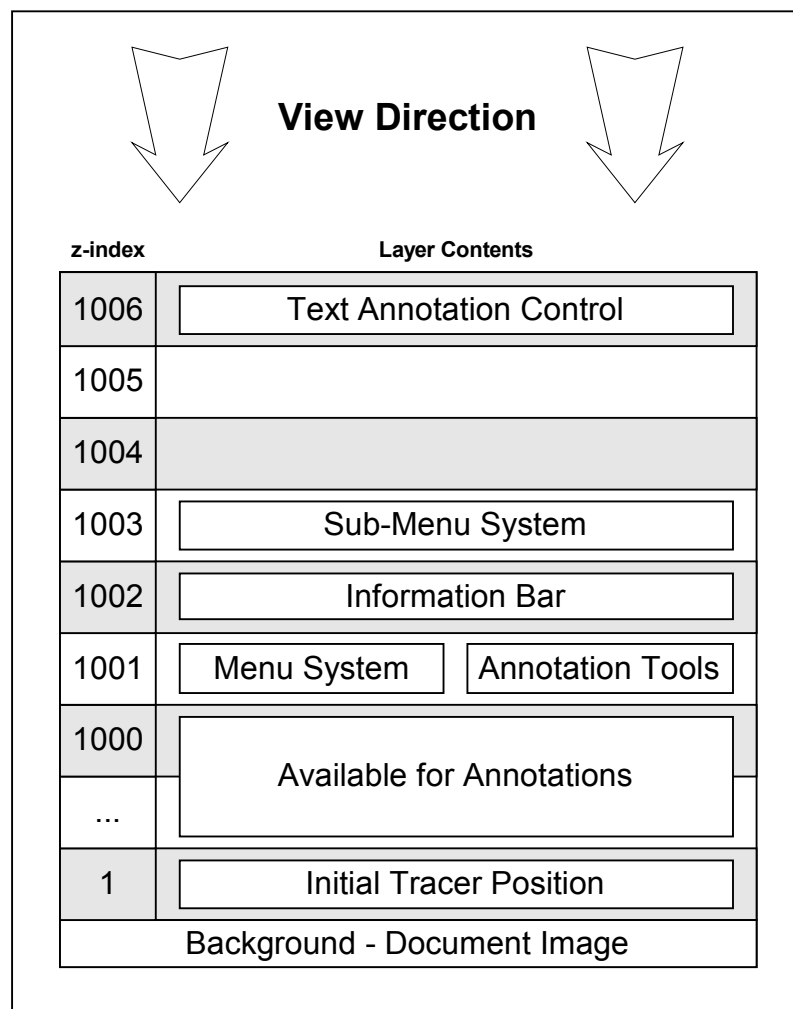


Figure 8: Z-Index Layers

### 4.3.3. Annotate Mode

Annotation Mode was very similar to Read Mode, with the addition of tools to add new annotations. This mode was only available to users with the necessary document permissions. Figure 9 shows this mode. Section 4.4 has a detailed description of this mode, its use and the technical details behind its implementation.

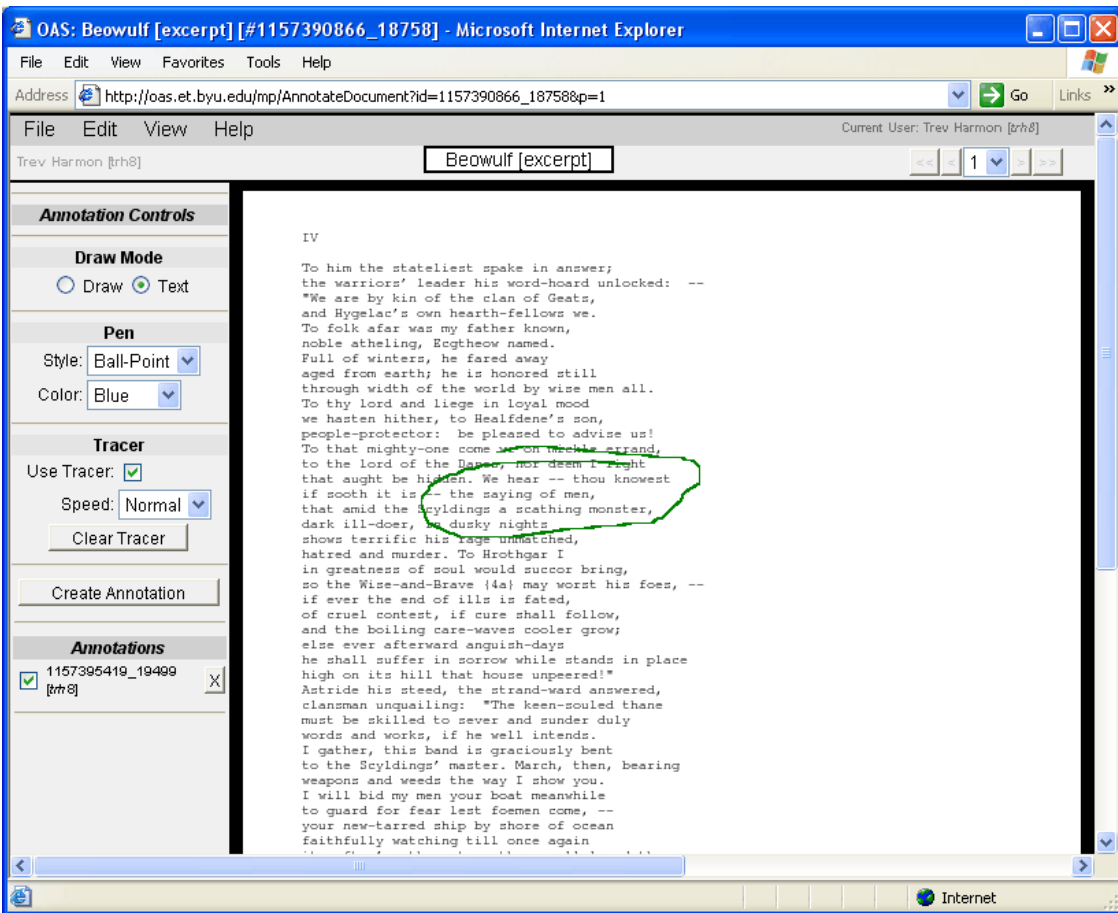


Figure 9: Document Annotate Mode

#### 4.3.4. Original Document

This view mode allows the user to download a copy of the original document in its original file format. No annotations added by *OAS* are present in the downloaded copy, as it is the original.

#### 4.4. Annotating Documents

One of the main goals of *OAS* was allowing annotation of documents via a standards-compliant web browser without the need of any additional applets, plug-ins, etc. This section explains the technical details of how this was accomplished and notes some of the limitations in the chosen approach. All techniques discussed in this section rely heavily on *DHTML*.

##### 4.4.1. Draw Annotation Mode

When in Draw Annotation Mode, the user can use a free-form drawing tool to add handwritten or drawn annotations to the document. These annotations are added using either a Tablet PC stylus or the mouse, which does not give as good of results because of the limited dexterity of a mouse compared to a stylus. However, this supports universal access to *OAS*. The user can select from a several different pen sizes and a number of different colors. While drawing the annotation, a tracer follows the path the user has drawn, as described in Section 4.4.3.

*JavaScript* is used to capture the mouse events in the browser. The recording of coordinates is active while the left mouse button is depressed (or the stylus equivalent), recording the cursor movement coordinates as a single path. A single annotation can be

comprised of multiple paths using different pen sizes and colors. All this information is sent to the *OAS* server for processing when the user clicks the button to create the annotation.

When the *OAS* server receives this information, it uses it to create a new image using the *ImageMagick* graphic library<sup>12</sup>. Because the pen size can vary, the image is resized and offset to insure proper placement and reduce clipping along the edges. Updated placement information and other annotation details are then stored in the database and the client is forced to reload the document and its annotations to display the new annotation.

#### **4.4.2. Text Annotation Mode**

Text Annotation Mode allows the user to add a plain-text annotation to the document. When this mode is selected, the tracer, which is described in Section 4.4.3, traces a box enclosing all points that have been drawn. If nothing has been drawn, a system-defined annotation box in the document's upper-left corner is used.

When the user clicks on the button to create a new annotation, a dialog opens that allows the user to type or paste the annotation text, as shown in Figure 10. Basic controls for font size, face and colors are also available. However, unlike Draw Annotation Mode, which could have multiple colors and pen sizes, only one set of attributes is allowed because they affect the entire annotation.

---

<sup>12</sup> Available at <http://www.imagemagick.org/>

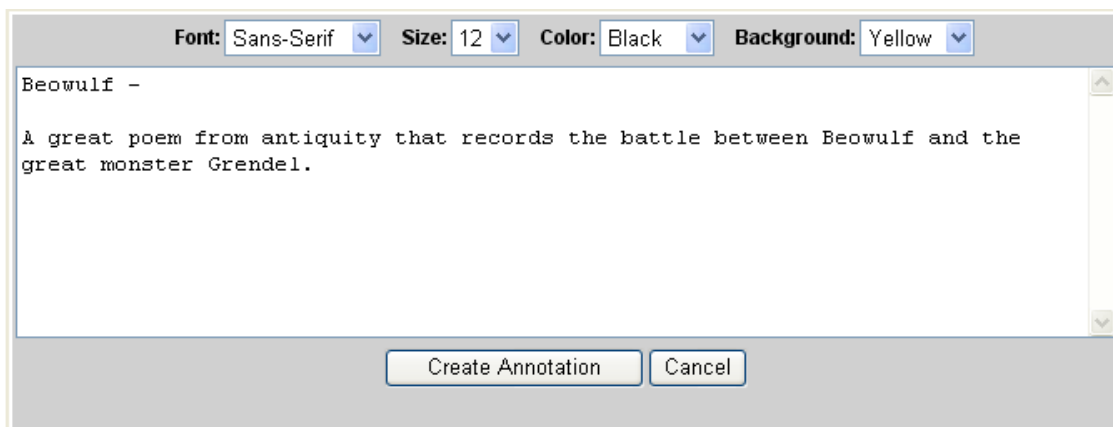


Figure 10: Text Annotation Creation Dialog

When a new text annotation is submitted to *OAS*, its attributes are stored in the database. The system then forces the client to refresh the document's view, which then displayed all annotations, including the new one, as shown in Figure 11.

Text annotations can be edited by the annotation owner, document owner, or moderator. When the user has rights, the annotation's ID appears as a hyper-link in the sidebar, instead of plain-text. Selecting this link brings up a dialog similar to the Annotation Creation Dialog, allowing the user to make any desired changes. However, the size and position of the annotation can not be changed, short of deleting and recreating the annotation.

If the user has rights, the user may also delete annotations by clicking the *X* button next to the annotation's ID. A dialog asking for confirmation is shown before any final action because deletion could not be undone.

One limitation of Text Mode is it currently only accepts plain-text. Rich-text and *HTML* are not allowed. This is due to the complexity of insuring non-plain-text



annotations do not interfere with the page's *HTML* code. Therefore, *HTML* markup characters are translated to their *HTML* entity equivalent (e.g., '<' becomes &lt;).

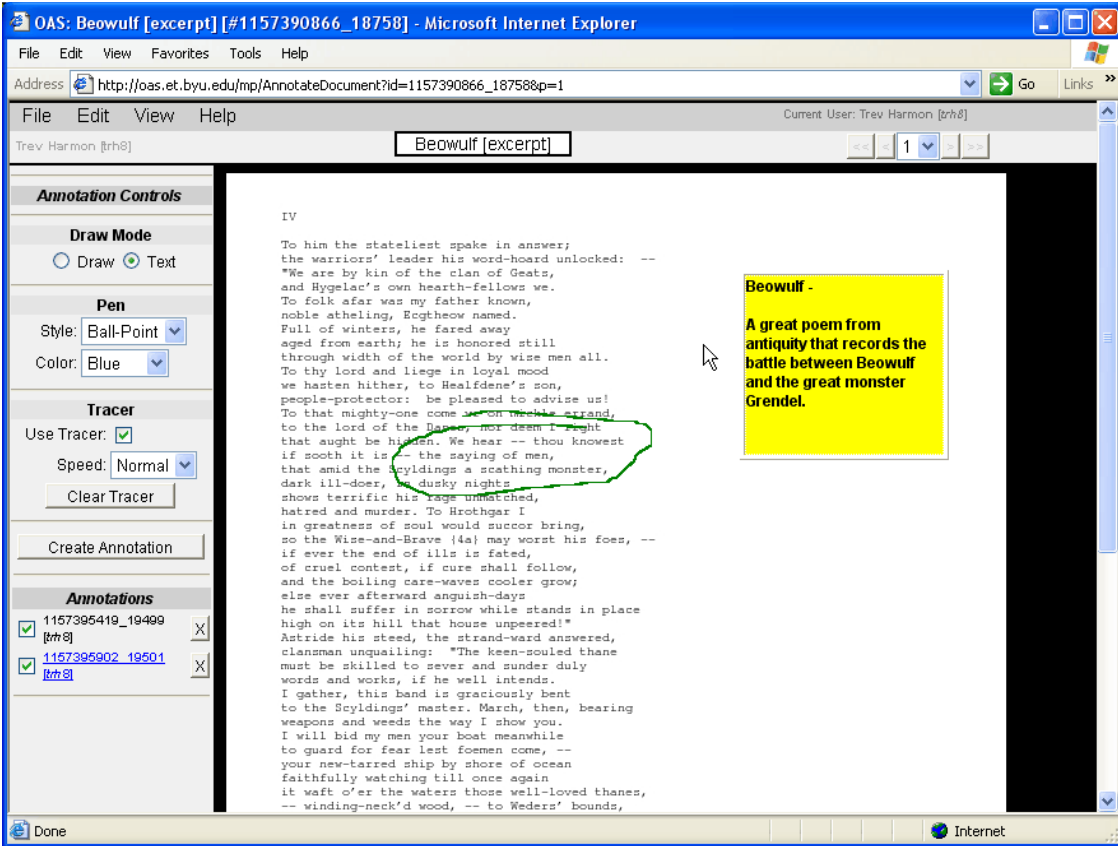


Figure 11: Document Annotation View with Text Annotation

#### 4.4.3. The Tracer

One of the most difficult problems in the design of the *OAS* user interface was how to give appropriate feedback to the user while creating annotations. The use of a tracer was the solution that worked the best, while meeting the system design criteria listed in Section 1.2. Alternate approaches and their pros and cons are discussed in more detail in Section 4.4.4.

The tracer was simply a small 10x10 pixel image, shown in Figure 12, containing an orange ball contained in a *HTML* `<div>` tag that is dynamically moved around the page using *DHTML* based on the web-based path animation process described by Rouyer (Rouyer 1998, 205-219). This color was chosen with the assumption that most documents are black text on a white background. However, this is not always the case, especially because *OAS* accepts images as documents. A future improvement to the system would be to change the tracer image color to match the current pen color.

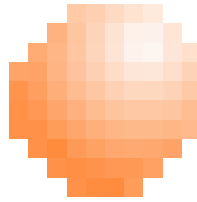


Figure 12: Tracer Image (Enlarged 10x)

When trace data, created by clicking and dragging the mouse in the document viewing area, exists, the tracer continually traces the paths defined by the data. In order to create the tracing action, a *JavaScript* timer is set to fire on an interval measured in milliseconds. This event then moves the tracer's `<div>` tag and the image it contains from the current to the next set of coordinates in the trace data path. A default interval was chosen that offers a compromise between system responsiveness and free-form annotation granularity. Following is a discussion of why this was necessary.

The web browser can only fire so many events within a given period of time. This is determined by the client's hardware (CPU, bus, memory, etc.), operating system, system load and web browser software. These events are used to not only trace the data,

but also do the actual data collection. Shortening the interval between events used to trace the existing data lowers the number of data collection events that can occur. However, if the tracer moves too slowly, the quality of the feedback is reduced.

Consequently, the interface was constructed to allow the user to adjust the speed of the tracer by choosing from three preset values. Alternatively, the user can completely turn off the tracer. However, this also removes the feedback it provides. But, turning off the tracer was necessary for one web browser, *Safari*, to function correctly due to peculiarities in its events “bubbling” structure.

Simply put, each client system has a multiple GUI layers in which events can occur. In this case, these layers are the operating-system-level, browser-level, and document-level. Each of these levels has its own events. The ordering and precedence rules define the “bubbling” structure, where some events on one level can affect those on another. *Safari* does not allow the document-level events (e.g., *mouseMove*, *onClick*, etc.) override the browser-level events (i.e., *drag-and-drop*). Therefore, when the user tried to click and drag the cursor, the browser-level events fired, masking the document-level events. Consequently, the data collection events, which are document-level, were not occurring, resulting in an empty trace path data set. This was not an issue in other web browser implementations.

Another issue with the tracer was its effect on handwriting. Stylus-based systems such as Tablet PCs and those with drawing pads, have enough control over the cursor position to create handwritten text while in Draw Annotation Mode. However, the lack of continual visual feedback can be disconcerting, making it difficult to write effectively. In

addition, if the writing was done faster than the web browsers event model could handle, data points were lost and the handwriting looks very poor.

#### 4.4.4. Alternative Real-time Display Methods

As has been discussed, the tracer method has several drawbacks. However, it was found to be the best of several approaches with the available technology. This section describes some of the alternatives that were explored.

##### 4.4.4.1. Client-side

One approach is to have the client render a temporary representation of the annotation before submitting it to the server for final rendering. This was tested by dynamically creating a series of *HTML* `<div>` tags containing an image positioned along the trace path, effectively creating a visible path. To lessen the number of images needed, images were placed end-to-end, resulting in a path resembling a string of pearls, which is not a perfect representation of what users would expect of electronic ink. A static `<div>` tag set was created in the document. As coordinates were added to the trace path, the contents of this `<div>` tag set were recursively replaced with the following:

$$old\_content</div><div>new\_content$$

This effectively created a new `<div>` tag set for each new image needed to render the path, thus creating a visible path.

While this approach should theoretically have worked, it experienced technical difficulties rather quickly. Both *Firefox* and *Internet Explorer* would stop responding to all user input after the creation of anywhere from nine to fifteen data points, and the user

would need to kill the entire web browser process via the operating system in order to regain control. As an annotation can have several hundred to a few thousand data points, this approach was deemed unusable.

#### **4.4.4.2. Server-side and AJAX**

Another approach is to have the client dynamically access the server via *AJAX* or some other similar technology and have the server dynamically render the image as it was drawn. Original testing of the server's raw image rendering capability suggested the combination of this, network lag and general server overhead required to do this in a full deployment of the system would be prohibitive. Basically, the server could not continually render the necessary images and effectively deliver them over the network (especially via modem). Based on these results, this approach was not implemented.

#### **4.4.4.3. VRML and SVG**

Using the Virtual Reality Modeling Language (*VRML*) or Scalable Vector Graphics (*SVG*) formats would seem a logical choice for the system, as they both can simply describe vector graphics, which is essentially what is being created when the user is in Draw Annotation Mode. However, these did not have sufficient web browser support to meet the criteria for this thesis. With those web browsers that did support one or both of these, implementation varied. Often a third-party plug-in was required in order to render these files. This contradicted the premise of needing only a standards-compliant web browser to access *OAS*.

Other problems with this approach existed, as well. The plug-ins often could not be contained within a larger *HTML* document, except in some cases where an *HTML iframe* could be utilized. However, this causes the annotation to be completely opaque, obscuring what is behind it and severely limiting the usefulness of the annotations. Consequently, this approach was deemed unusable, as well.

#### **4.5. System Administration**

*OAS* provides a number of tools to allow users to administer rights to their documents and administrators to maintain the entire system. These tools are all accessible via web interface. This removes the need for administrators to make changes directly to the file system or database. Extensions to the system should developed using the *API* to maintain this convention.

##### **4.5.1. Users**

While not desirable in many environments, users were allowed to create their own user account for testing purposes. It can be disabled by simply removing or updating the service, as described in Section 4.5.4.

In addition, an administrative service is available that allowed the addition, modification, and removal of system users. The service for administration of user accounts is shown in Figure 13. This service also allows the administrator to control the users' group memberships.

Users are allowed to update their own personal information, as well. Figure 14 shows the form used to accomplish this. All user information can be updated with the

exception of the *user\_id*. A placeholder is used for the password fields, as the database does not contain the original password, only the encrypted version of it. The password is updated in the database if anything other than the placeholder is received by *OAS* and both password field values match. This will only occur when the user has submitted a new password for their account.

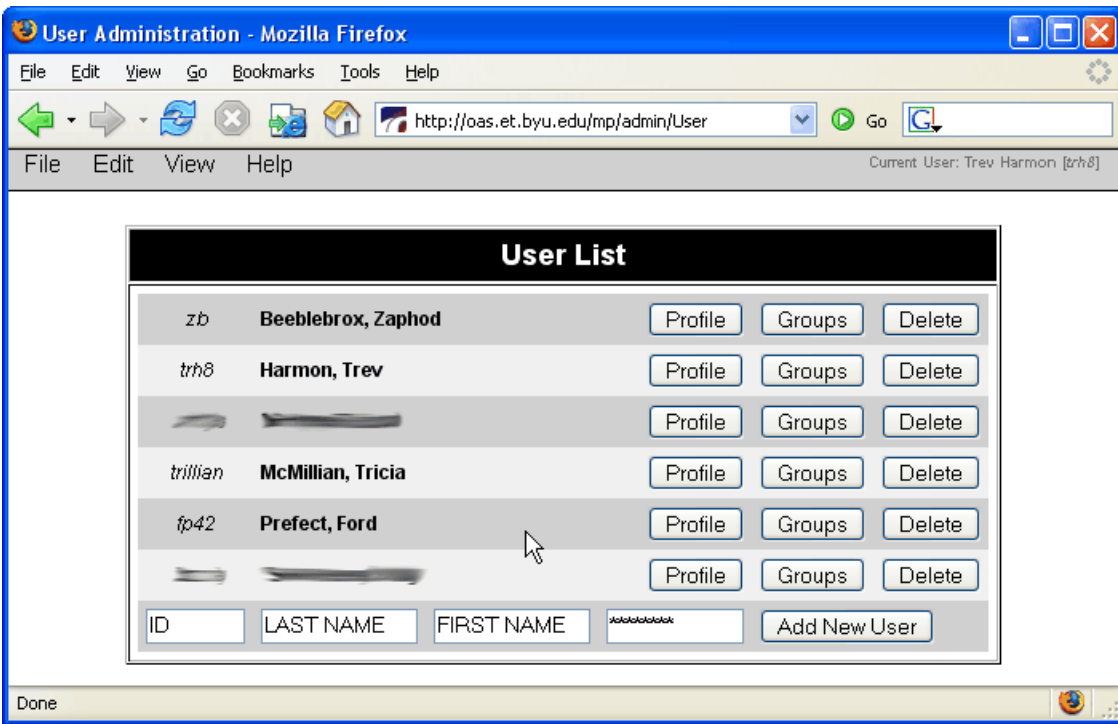


Figure 13: User Administration

#### 4.5.2. Groups

All users are allowed to create groups. By default, the user who created a group is the group owner, who has access to control group membership. However, the current group owner can give group ownership to another user. If the group owner does this, they

are demoted to a normal user within the group. Group ownership transfers are permanent. Once given, group ownership can not be taken back by the original owner.

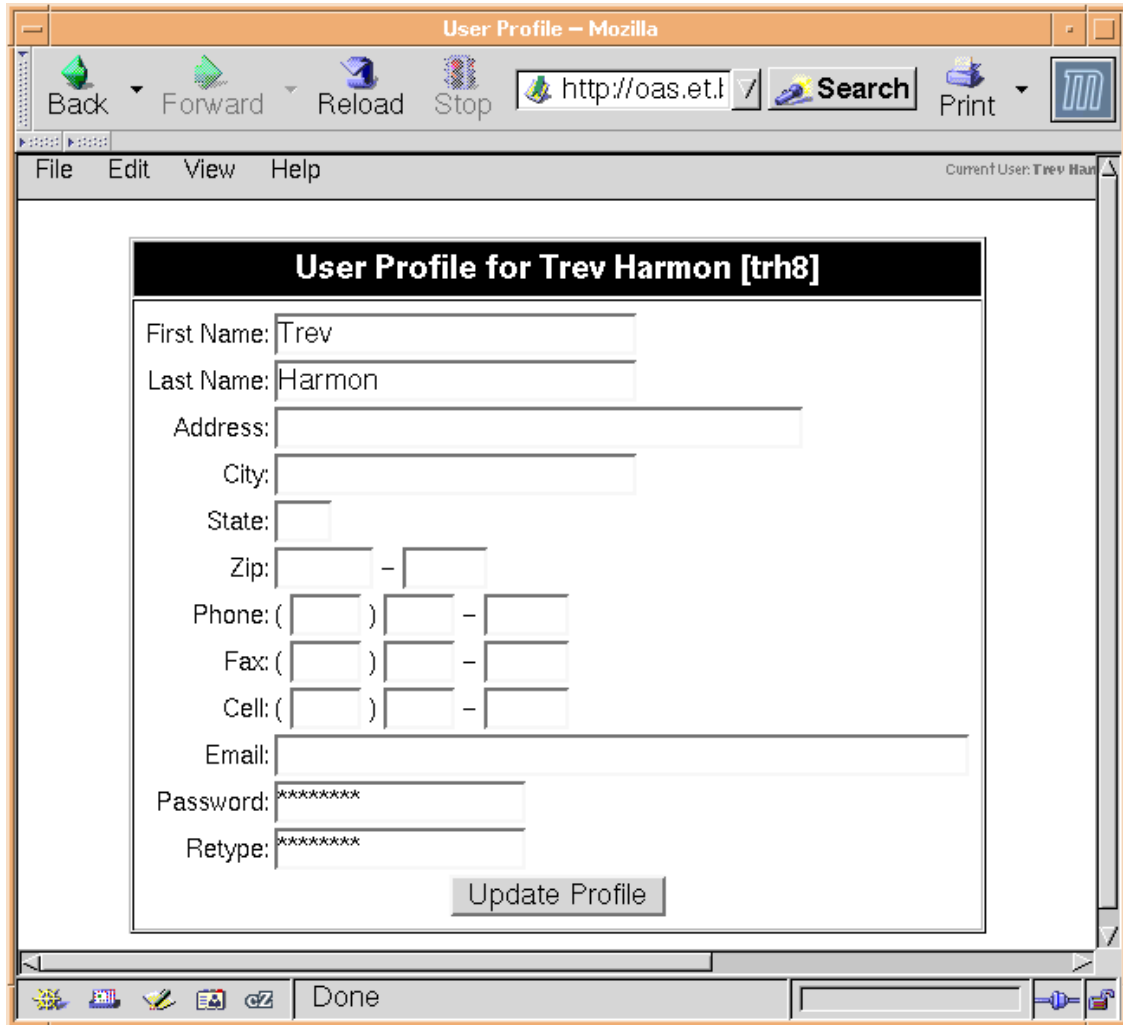


Figure 14: User Profile

By design, all user created groups have lower-case names, with upper-case names reserved for system-defined groups. A special *ADMIN* group exists for those who are administrators on the system. All administrators can modify the group membership of this group. However, no user “owns” it.



Shown in Figure 15, the Group Manager lists the user's group memberships and all groups the user owned. In this case, the user who is logged into the system is an administrator. Consequently, all groups in *OAS* are listed. Administrative users can make changes to any and all groups, even though they may not be the actual owner of the group in question.

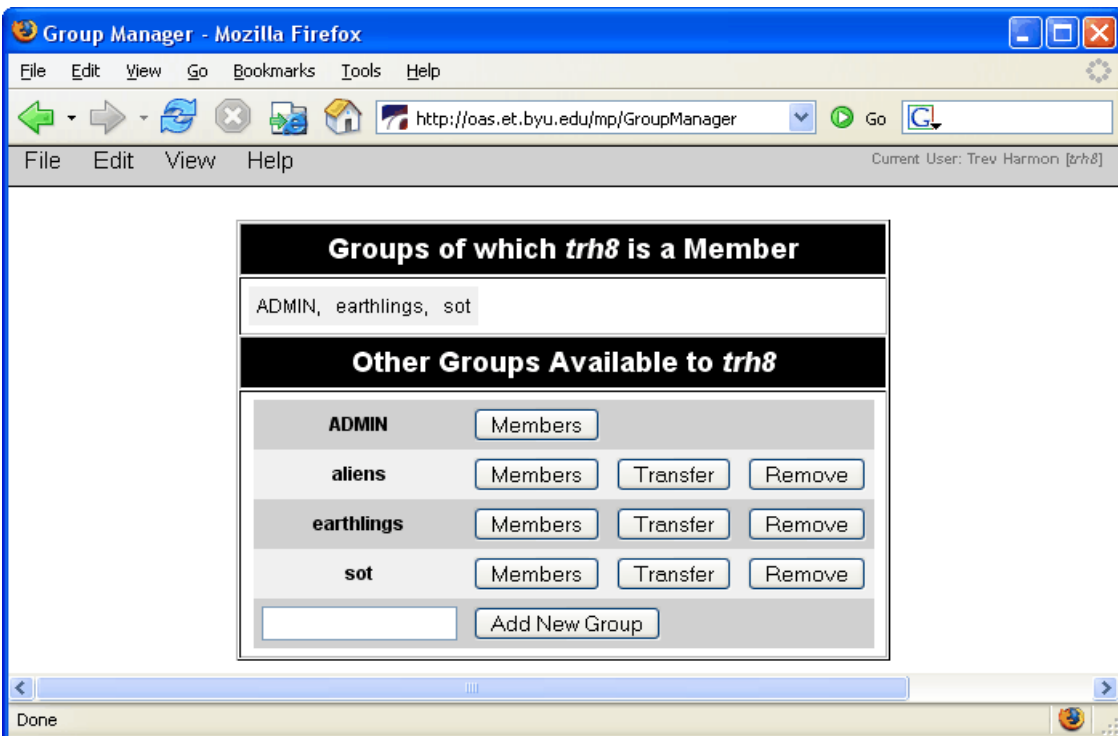


Figure 15: Group Manager

Groups can be given document permissions, as described in Sections 4.1.3 and 4.5.3. Document permissions are additive, meaning all members of the group receive these permissions in addition to the permissions they already have. Groups can not be used to restrict permissions already held by a user, only to give users new permissions.

Groups can be used in a number of useful ways. Because all users are allowed to create their own groups, groups can be used to form study groups, discussion forums, classes, accreditation teams, etc. Since group ownership can be transferred, a secretary could create a class group and then simply transfer ownership to the instructor.

The group model used by *OAS* is very flexible. Through creative use of this model, a very wide range of collaboration and annotation scenarios can be constructed. For example, in a large class, a professor may have several grading teaching assistants, and creates a group for them. When a student submits an assignment, the document ownership is transferred to the professor, who then gives the teaching assistant group read and annotate document permissions for the submitted document. Each teaching assistant can make annotations on the document. The professor, who has the moderator document permission, because he is the document owner, can see which teaching assistant made which annotation.

#### **4.5.3. Documents**

Document administration is fairly simple. All valid users are allowed to create documents through the menu system. This process is described in detail in Section 4.2.

Once created, the document owner have all administrative rights over their document. The owner can delete the document and all related annotations through the menu system, specify which users and groups had access to the document and their associated rights, and give ownership of the document to another user. Figure 16 shows the form for accomplishing these tasks.

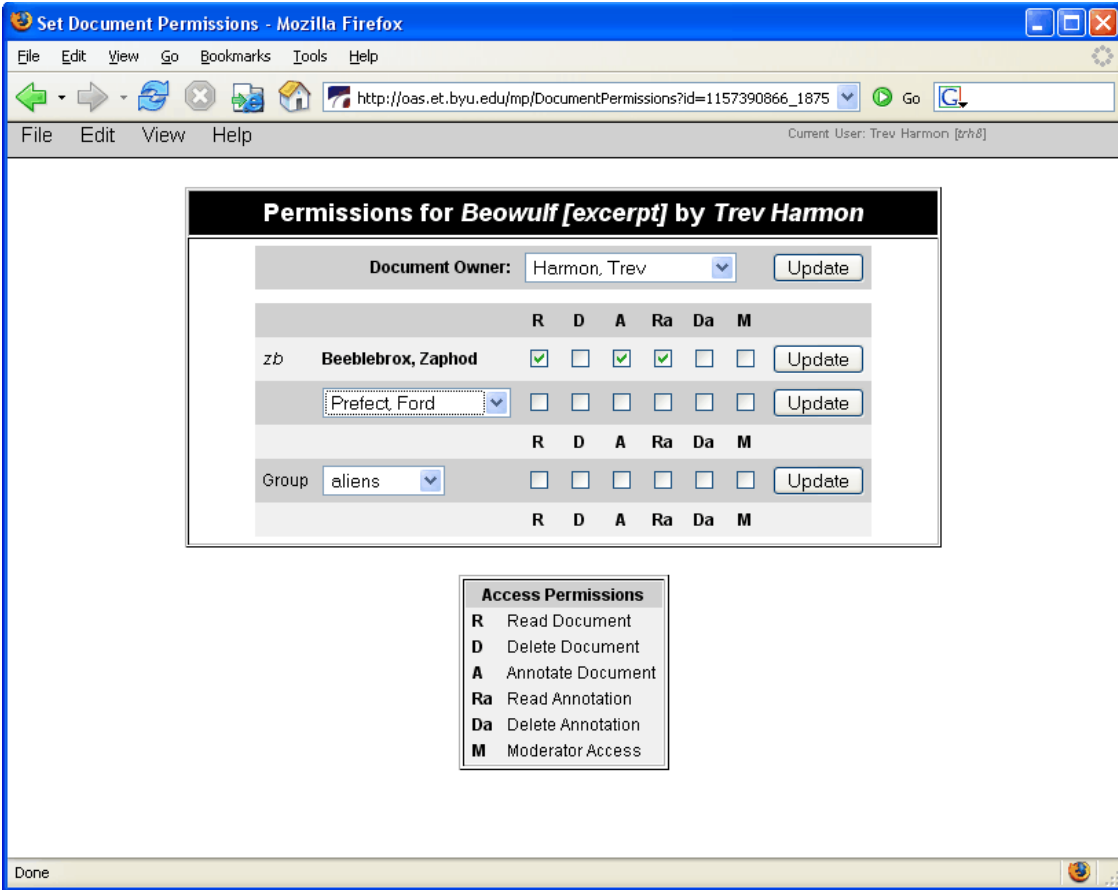


Figure 16: Document Permissions

#### 4.5.4. Services

Users with administrative rights can view, add, and update *OAS* services. The option, which appears in the menu system for an authorized user, lists all currently available services, a short description and the service's URL, as shown in Figure 17.

Service administration was mainly a reference resource, as removing or dramatically changing services could render parts of the system inoperable. However, it would be necessary to add new or updated services in the case of an extension to *OAS*. In short, it provided access to the database table used by the *API* to look up service locations for use in the menu system or by other services.

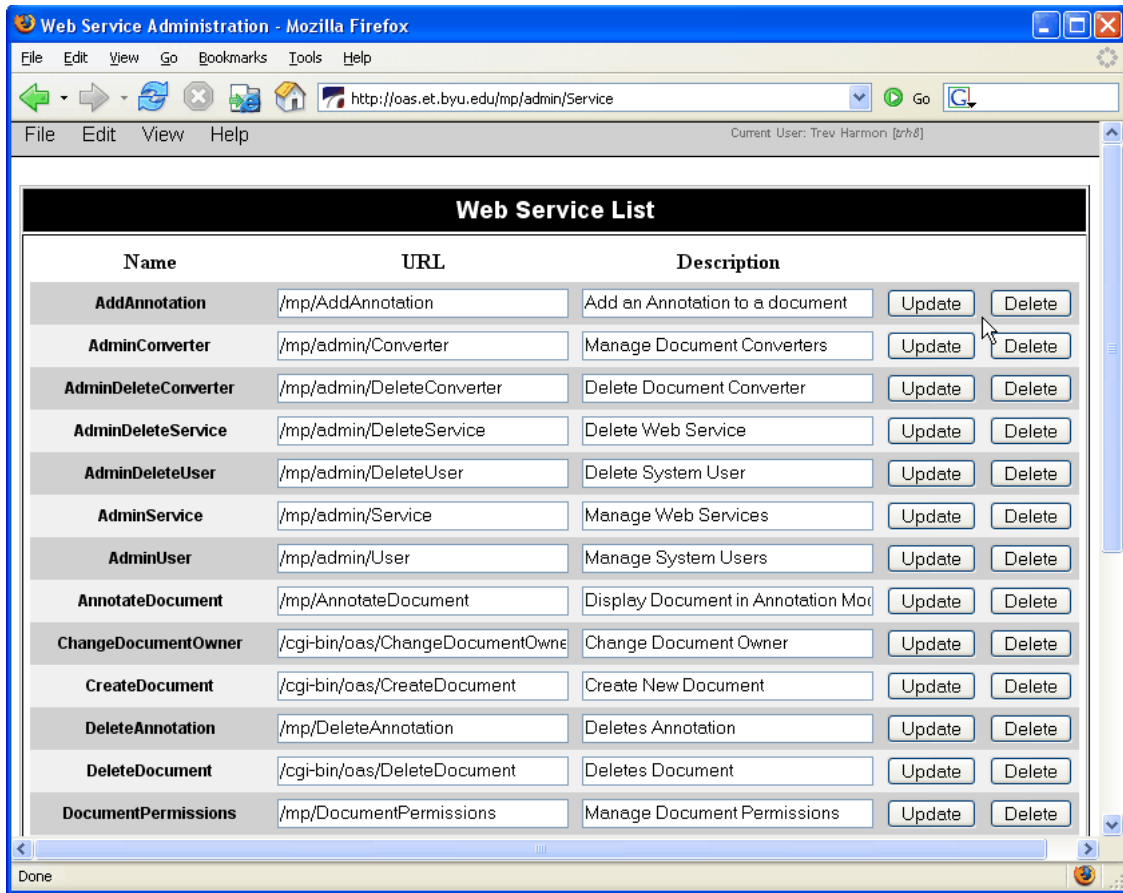


Figure 17: Web Service Administration

#### 4.5.5. Document Converters

Users with administrative rights have a menu option allowing them to add, update, or delete document converters. Document converters and their use in the document acceptance process are discussed in detail in Section 4.2.

Figure 18 shows the converter administration service. It contains the file extension used to match the converter to the file format and the image format that results from the conversion. In addition the full path to the converter executable and a short description are also included.

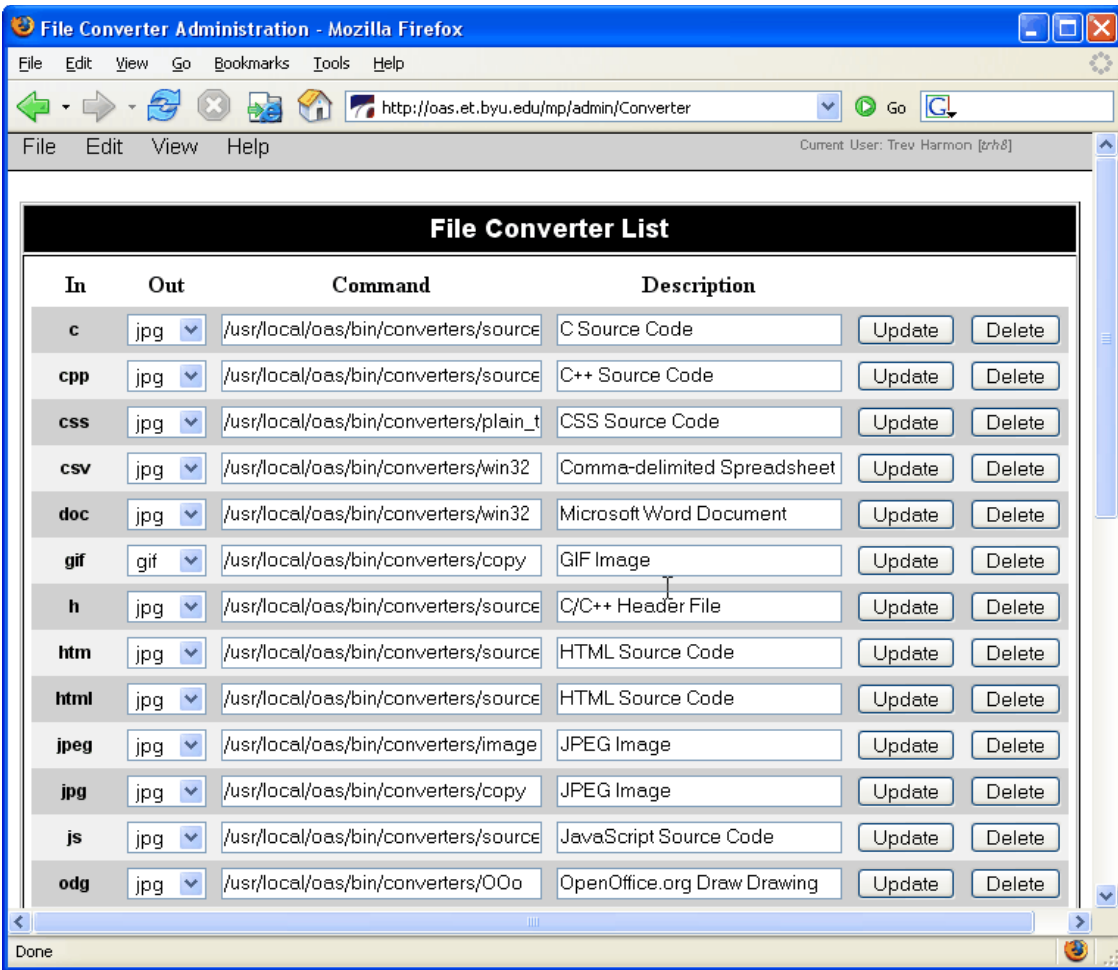


Figure 18: File Format Converter Administration

Additional converters can be added through this administrative tool, as well.

Appendix D contains the source code for all the converters currently used by *OAS*.

#### 4.6. Software API

*OAS* was developed around a central *API*. The reasons for doing this are manifold, including modular code design and simplification of system extension. This was especially true because of the complex database operations performed by *OAS*. If each

individual part of the system had performed its operations directly on the database, continued support would have been difficult, especially if an upgrade required a change to the underlying database structure. With an *API*, changes only need to be made in the *API* code.

The *API* was developed using *Perl 5* modules, as described in Section 3.2.4.1.

This section takes a closer look at the actual *API* and its application in the system. While the actual *API* documentation is found in Appendix A, each *API* module is briefly discussed in this section. Source code for each module is available in Appendix B.

#### **4.6.1. Annotation.pm**

This module handles the creation, management, and deletions of annotations, both in the file system and database. The module also has a convenience function *html*, which creates the *HTML* code necessary for the annotation to be displayed in a *HTML* document, regardless of the annotation type.

#### **4.6.2. Auth.pm**

This module handles all user authentication used by the *OAS*, with the exception of the communication level authentication done by *Apache*, as described in Section 4.1.2. This module is used by many services to determine document permission rights held by the user. It also handles group related security as well.

### 4.6.3. CGI.pm

This module modifies the error reporting actions of the *Perl 5* standard distribution's *CGI.pm* module, which is always used for the *CGI* services in *OAS*. In addition, it provides functions for creating standardized *HTML* headers and footers.

### 4.6.4. Converter.pm

This module manages the different converters in the database. In addition, it contains a helper function used to do the actual document conversion. Consequently, *OAS* services do not need to know anything about the underlying operating system or its call procedures in order to convert documents.

### 4.6.5. DBI.pm

This module is a wrapper class for the *Perl 5* standard distribution's *DBI.pm* module. The interface contains all the connection information necessary for the database, meaning other *API* modules and services do not need to contain this information. In addition, the interface institutes automated connection sharing, which is very important in a large system such as *OAS*. Most of the *API* modules need to establish at least one connections to the database. If each need created a unique connection, the connection table in the *DBMS* would soon be filled. The method used by the interface guaranteed all services strictly utilizing this interface only have a single connection to the database, thus reducing overhead.

#### **4.6.6. Document.pm**

One of the largest in *OAS*, this module manages documents not only in the database, but also on the file system. It also provides wrapper functions for other *API* modules such as *Pages.pm* to improve code readability.

#### **4.6.7. Image.pm**

This module is a wrapper class for the *Image::Magick* module, which is not part of the standard *Perl 5* distribution package. This interface hides many of the details of image manipulation and handles some of the idiosyncrasies in required operation order found in the *Image::Magick* module.

#### **4.6.8. Menu.pm**

This module handles the creation of system menus, controls, tool bars and certain dialogs. It is the one most likely to be changed in a system extension. It is tightly tied to a number of external *JavaScript* files described in Appendix E.

#### **4.6.9. Page.pm**

This module handles page information in the database. Pages are subordinates of documents, but necessary due to the fact that each page is translated into its own image file. Internally, annotations are actually attached to a page in a document, as opposed to the document as a whole.



#### 4.6.10. Person.pm

This module handles user information in the database. This includes not only user profile information, but also the login credentials for the user. See Section 4.1.2 for more information on these credentials and how they are used by the system.

#### 4.6.11. Service.pm

This module manages service information in the database. See Section 3.2.6 for information on services.

#### 4.6.12. System.pm

This module controls the global system information in the database. This includes information on file system paths and system URLs. It also has a number of functions for converting between paths and URLs. Many *API* modules require this module in order to work correctly.

#### 4.6.13. Utils.pm

More of a function library, this module provides a number of useful helper functions that do not fit elsewhere within the *API*. One of these functions, *untaint*, requires special attention when used during development, as it could create serious security problems if used incorrectly.

## 4.7. Browser Support

*OAS* was successfully tested on a number of different browsers and operating systems. Table 4 lists the web browsers shown to be compatible with *OAS*.

Table 4: Supported Web Browsers

Operating System	Web Browser	Supported?
Windows XP Home	Internet Explorer 6.0	Yes
	Firefox 1.0	Yes
	Opera 9.0	Yes
Windows XP Tablet PC Edition 2005	Internet Explorer 7.0 RC1	Yes
	Firefox 1.5	Yes
	Mozilla 1.7.3	Yes
Mac OS X Jaguar	Safari 1.3	Yes (no tracer)
Linux Fedora Core 5	Firefox 1.5	Yes
	Opera 9.00	Yes
	Mozilla 1.7.13	Yes
	Konquerer 3.5.3	Yes
	“Epiphany” GNOME Web Browser 2.14.2.1	Yes

As can be seen by the chart, support was not perfect for all browsers. For example, Safari required the tracer be turned off in order for annotation creation to work correctly. The reason for this is discussed in Section 4.4.3. While certain web browsers did have certain quirks that required retooling of *OAS*, in general, few changes needed to be made to support specific web browsers. This is to be expected, as they all purport to support the published web standards.

With the inclusion of *Internet Explorer* and *Firefox*, the web browsers responsible for the large majority of 2006's web traffic are covered (Jupitermedia Corporation 2006).

However, since several alternative standards-compliant web browsers were available, they were also included in the testing.

#### 4.8. Load Testing

This is the area that had the most disappointing results. Five *Ubuntu Linux* systems were used as clients to perform load testing on *OAS*. Each created five simultaneous instances of the test script described in Section 3.3.2. These then connected to the *OAS* server over the university's network. This simulated a continuous load of twenty-five concurrent users on the system. Because these were virtual users being controlled by scripts, the natural delay associated with human interaction was not present. Consequently, this test represented an effective load much larger than twenty-five concurrent human users.

While the test was running, a large number of entries began appearing in *Apache*'s error log. This was unexpected, as *OAS* had been continually monitored during development to insure no errors were being produced while in a single user environment. Most errors were caused by undefined variable values generally populated either by database calls or by *Apache* when it created the handler's environment.

Figure 19 shows a simplified version of the *Apache* request cycle used in *OAS* (Bekman and Cholet 2003, 728). Requests sent to the server were handled by *Apache*. During the processing, the connections to the database are required for two steps, authentication and the processing of the request by the handler. The database errors experienced during testing were only occurring during the handler phase, as the clients were being given permission to access the resource (i.e., the credentials were being

accepted and validated). However, failures were also occurring during the environment setup. This suggested the errors are occurring as a result of the corruption of the *Apache* child process being used to handle the requests. This is supported by the fact that restarting *Apache* would temporarily correct the problem.

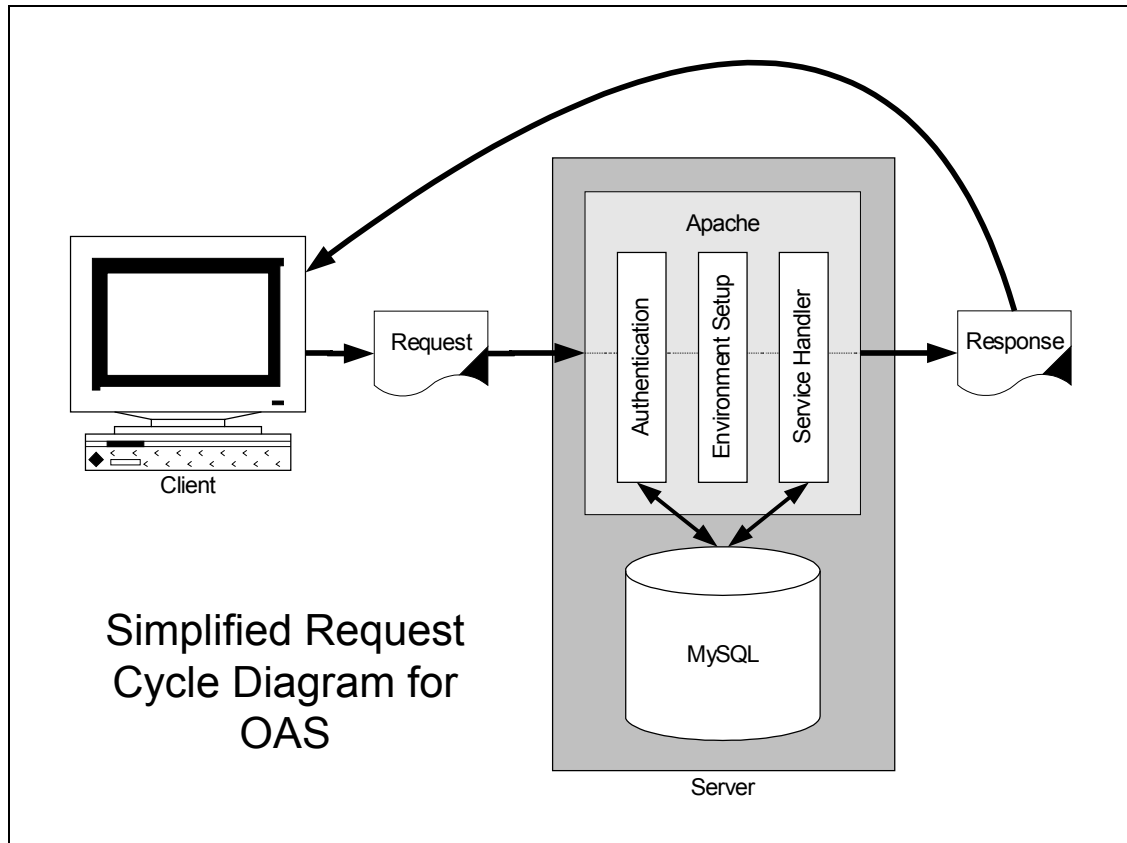


Figure 19: Simplified Request Cycle Diagram for *OAS*

Attempts were made to tweak the settings in the *httpd.conf* file to force *Apache* to recycle its child processes at a faster rate. However, this had no noticeable effect on the problem. Eventually, it was determined the maximum number of concurrent users was three. Occasionally, a fourth could be temporarily added, but not generally.

Because of the evidence, it was determined this problem is not a product of hardware performance. This suggests there is a problem in the underlying structure of the system itself.

One possible solution would be to move this particular service from an *Apache* handler to a normal *CGI* script. While this would fix the symptoms experienced during the load testing, it would not correct the underlying problem. In addition, the performance gains of *mod\_perl2* would be lost by such a course.

Another avenue of research is in the interaction between the *DBMS*. While this interaction does not appear to address all of the symptoms exhibited, a majority of the problems experienced affected the *DBMS* interaction. The variable holding the database connection is persistent across multiple evocations of the handler within the *Apache* child process. It is possible this is causing part of the problem. The entire *DBMS* sub-system could be retooled to take advantage of the *Apache::DBI* module. In a worst-case scenario, pooling of database connections could be disabled. This would require major review and possible rewriting of all modules in the system, as they all assume connections to the database are shared. Most should support a transparent change to the *OAS::DBI* module to implement this approach, but regression testing would need to be done to assure this is the case. Otherwise, unreleased connections could cause a resource leak, leading to a failure in *OAS*.

## 5. Conclusions

Overall, *OAS* was able to meet all of its original design goals. The resulting system is an on-line annotation system that provides free-form and textual annotation functionality through a standards-compliant web browser. However, the inability of the tracer used for free-form annotations to give appropriate user feedback severely limits the actual usability of *OAS*. In addition to the annotated document, *OAS* provides access to the original document, as well. Access in *OAS* is multi-tiered, allowing users fine control over who access to their documents.

However, during the development, other considerations were found, which form the basis of future research discussed later in this chapter. Each of the design goals specified by the thesis research questions in Section 1.2 will be discussed, noting how well the design goal was met and what other considerations need to be addressed.

### 5.1. Web Browsers

*OAS* has shown that it is possible to implement an on-line annotation system that operates successfully on a standards-compliant web browser. However, even though the web browsers are standards-compliant per the definition used for this thesis, their actual compliance varies.

Also, even though layout styles were prescribed by the style sheets, web browsers' occasionally varied on their actual rendering, which caused unexpected artifacts in the user-interface. Several documented browser bugs were also encountered. In some versions of *Firefox*, if a `<select>` tag has a style applied, it is subject to an “endless growing” effect (van der Blonk 2006). In *Internet Explorer 6*, “select elements ignore their and other elements' CSS z-index positioning property ” (Eberhardt 2004 and Microsoft Corporation 2006).

All in all, it appeared standards bodies and web browser developers have been coming to a consensus over the last few years. This has lead to web browsers supporting a unified DOM and feature set, significantly reducing the amount of work needed to make web systems perform successfully across many different web browsers and operating systems. While not all issues have been addressed, enough agreement exists to make *OAS* cross-browser compatible. It is expected that continuing development in web browser technology will further the interoperability of web browsers.

## 5.2. File Formats

With the large number of file formats accepted by *OAS*, as listed in Appendix K, it is currently usable within a number of process domains. In addition, *OAS* was designed to provide a simple method for adding new converters in the future.

*OAS* uses a variety of techniques to accomplish the conversion of the twenty-six file formats. While this does not represent all the techniques that could be needed, it does provide a framework for future development. Extending *OAS* to allow for additional file formats may require the use of additional techniques to create acceptable conversions.

### **5.3. Contextually Anchored Annotations**

Contextually anchoring annotations in dynamic file formats has been shown to be a very difficult task. Consequently, the solution was to convert all documents to a static format, thereby allowing all annotations to be anchored statically, thus providing contextual relevance to the annotations. Since the original document is available within *OAS*, this approach is acceptable in most circumstances.

### **5.4. Original Document Archive**

It is very important to keep a copy of the original document for a number of reasons. One reason is a pristine copy may be needed at a later period, especially if the user wishes to use the document for a purpose where the annotations would be problematic. Another reason is the fact that *OAS* converts all documents to a static format so annotations can be contextually anchored. This process can not easily be reversed, thus necessitating access to the original.

In order to address these issues, *OAS* simply saves a copy of the original document in the file system. Hereby, the user always has access to a copy of the document in the original file format.

### **5.5. Access Control**

Access control plays an important part in the usefulness of *OAS*. The multi-tier approach allows for a level of control that is not found in many other annotation systems. The ability the user has to create their own access groups and assign access rights and



permissions is a powerful tool that can be used to customize *OAS* for a number of different applications and process domains.

Another strong point of the access control functionality in *OAS* is the integration of the web server and *DBMS*. This allows for a unified login and provides the underlying mechanism for multiple layers of security. The web server controls direct access to the file system, while *OAS* services add additional internal security checks. It also provides for easy user administration, as all credentials are stored in a single location.

System access and security controls were successfully implemented in *OAS*. In addition, the multi-tier access control paradigm provides a powerful tool for both the end-user and administrator. Many different applications can be simulated simply by ingenious use of this approach.

## **5.6. System Performance**

*OAS* performed at an acceptable level with up to three concurrent users. However, with additional users the system failed spectacularly. This failure was apparently caused by the corruption of the *Apache* child processes, as opposed to hardware performance issues. *OAS* is therefore more of a proof of concept than a full-blown annotation product. Additional research is needed to determine exactly what changes need to be made to the system to make it more stable in a real-world environment.

The other consideration was the client system performance. Because data collection for annotation is handled by the client's web browser, the client's event management structure becomes an issue. Consequently, client systems with faster

processors, more RAM, and better event models can create cleaner annotations than those with lesser hardware specifications and poorer software design.

As hardware and software continue to improve, this problem is expected to become less of an issue.

### **5.7. Other Considerations**

While not one of the main research points for this thesis, usability was considered in the design of *OAS*. In general, *OAS* design follows good usability practices. However, there is one area where *OAS* has a major usability problem – providing appropriate user feedback for free-form annotations (i.e., the user cannot continually see what is being drawn in real-time).

Several different solutions were explored for this problem, but, in the end, the approach used was chosen because it was the only one that worked, not because it gave the best feedback. This became apparent when a user attempted to use the system to handwrite annotations using a stylus on a Tablet PC or equivalent. A cognitive disconnect was caused when the user was unable to easily see what they were writing in real-time.

This is a problem that should be addressed in future research, as correcting it will greatly increase the usability of *OAS*.

### **5.8. Future Research and Improvements**

This thesis shows the basic technologies are available for an on-line annotation system that accepts multiple file formats. Future research should focus on increasing

system performance and usability, improving security, and extending the system to meet the specialized needs of the different process domains.

### **5.8.1. System Performance**

An area that would benefit from additional research is that of performance. While *OAS* has undergone optimization, greater response times are possible, both in the server- and client-side programming. While optimizing for speed, a balance between code maintainability and speed needs to be considered, as well. In addition, the performance issues discussed in Section 4.8 need to be addressed.

### **5.8.2. Usability**

Better user feedback is needed for free-form annotations. In fact, the entire system would benefit from a complete usability study. A number of the user interfaces are rather clumsy and could be improved. Also, a full *Help* system is also needed for real-world deployment of *OAS*. The stubs for such a system already exist within *OAS*.

### **5.8.3. User Authentication**

As discussed in Section 4.1.2, the authentication method used by *OAS* has a number of issues. This section offers two additional techniques that could be used for the system, along with their advantages and disadvantages.

### **5.8.3.1. HTTPS**

The single largest security problem with the current approach is credentials are sent in plain-text over the network, as it is trivial to capture these credentials if one has access to the network data stream. The simplest solution to this would be to run the entire transaction over *HTTPS*, thereby encrypting the transmission of the credentials.

The drawback to this solution is *HTTPS* takes longer to transmit data. Since *OAS* deals with comparably large image files, this overhead could cause problems with the user experience. This is especially true since many of the scripts and handlers send a specific “no-cache” header to the client. This issue would need to be addressed by future research, and a balance struck between system performance and security.

### **5.8.3.2. Cookie-based Authentication**

Instead of sending the credentials with each transaction, some on-line systems opt for a different approach. While not inherently more secure, a client is given a session cookie to send with each subsequent request once it is authenticated. This session cookie is then authenticated instead of the credentials for each transaction session. This approach still has the problem of possible eavesdropping and requires a special handler be written for static content. However, it does provide for the possibility of a time-out, which would log the user out of the system after a specified period of inactivity.

### **5.8.4. OAS Extensions**

The system designed for this thesis is just a basic system, providing the tools for a number of different process domains, but not strongly specializing in any of them.

Further development could be done to take the basic system and extend it in such a way as to address the needs of specific process domains. Additional modules could be added for handling tasks specific to those process domains.

#### **5.8.4.1. Examples**

For example, the system could be converted to a full-blown academic CMS with the addition of assignment, grading, class and department modules, along with the supporting user and administrative services. Email notification services could also be added. Support for student or professional portfolios could be added to *OAS*.

#### **5.8.4.2. Process for Extending OAS**

*OAS* was designed to be easy to extend in the future, hence the use of an *API*. The *API* is meant simply to provide the basic system functionality and enough hooks to extend *OAS* without needing a major rewrite. However, a successful extension of *OAS* needs to proceed through a series of steps. The basic steps are as follows, though all may not be needed in some cases:

1. Create necessary database tables.
2. Create and test associated *Perl 5* modules and their interfaces for the new database tables.
3. Create and test new services utilizing the new functionality.
4. Add services to *OAS*.
5. Update *Menu.pm* to include access points to the new services.

System extensions can provide new functionality to *OAS* geared towards its specific process domain. By keeping all parts of *OAS* compartmentalized through the use of an extensible *API*, the basic system can be customized for its given environment and deployment goals.

### **5.9. A Final Word**

Overall, *OAS* was a success. It proved it is possible to implement an on-line annotation system that provides free-form and textual annotation functionality through a standards-compliant web browser. The implementation is not perfect in all respects, as the system's usability is severely limited by the ineffective user feedback provided by the tracer while doing free-form annotations. This one issue is a major roadblock in the deployment of *OAS* into a live environment. However, *OAS* shows yet another aspect of the myriad exciting possibilities of the Internet and its continually evolving technologies.



## Selected Bibliography

- Agosti, M., N. Ferro, and N. Orio. "Annotating illuminated manuscripts: an effective tool for research and education." In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries, Denver, Colo., USA, June 7-11, 2005*, by ACM Press, 121-130. New York, N.Y., 2005. Available from <http://doi.acm.org/10.1145/1065385.1065412>.
- Appan, P., B. Shevade, H. Sundaram, and D. Birchfield. "Interfaces for networked media exploration and collaborative annotation." In *Proceedings of the 10th International Conference on Intelligent User Interfaces, San Diego, Calif., USA, January 10-13, 2005*, by ACM Press, 106-113. New York, N.Y., 2005. Available from <http://doi.acm.org/10.1145/1040830.1040860>.
- Barger, D. and T. Moscovich. "Reflowing digital ink annotations." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Ft. Lauderdale, Fla., USA, April 5-10, 2003*, by ACM Press, 385-393. New York, N.Y., 2003. Available from <http://doi.acm.org/10.1145/642611.642678>.
- Bateman, J., R. Henschel, and J. Delin. "A brief introduction to the GeM annotation schema for complex document layout." In *Proceedings of the 2nd Workshop on NLP and XML, International Conference on Computational Linguistics*, by the Association for Computational Linguistics, 17 1-8. Morristown, NJ, 2002.
- Bauer, M. D. *Building Secure Servers with Linux*. Sebastopol, Calif.: O'Reilly & Associates, Inc., 2003.
- Bekman, S. and E. Cholet. *Practical mod\_perl*. Sebastopol, Calif.: O'Reilly & Associates, Inc., May 2003.
- Bottoni, P., R. Civica, S. Levialdi, L. Orso, E. Panizzi, and R. Trinchese. "MADCOW: A multimedia digital annotation system." In *Proceedings of the Working Conference on Advanced Visual Interfaces, Gallipoli, Italy, May 25-28, 2004*, by ACM Press, 55-62. New York, N.Y., 2004. Available from <http://doi.acm.org/10.1145/989863.989870>.



- Bottoni, P., S. Levialdi, A. Labella, E. Panizzi, R. Trinchese, and L. Gigli. "MADCOW: A visual interface for annotating web pages." In *Proceedings of the Working Conference on Advanced Visual Interfaces, Venezia, Italy, May 23-26, 2006*, by ACM Press, 314-317. New York, N.Y., 2006. Available from <http://doi.acm.org/10.1145/1133265.1133331>.
- Brush, A. B. "Annotating digital documents: anchoring, educational use, and notification". In *CHI '02 Extended Abstracts on Human Factors in Computing Systems, Minneapolis, Minn., USA, April 20-25, 2002*, by ACM Press, 542-543. New York, N.Y., 2002. Available from <http://doi.acm.org/10.1145/506443.506472>.
- Brush, A. B., D. Barger, J. Grudin, and A. Gupta. "Notification for shared annotation of digital documents." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves, Minneapolis, Minn., USA, April 20-25, 2002*, by ACM Press, 89-96. New York, N.Y., 2002. Available from <http://doi.acm.org/10.1145/503376.503393>.
- Brush, A. J., D. Barger, A. Gupta, and J. J. Cadiz. "Robust annotation positioning in digital documents." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Seattle, Wash., USA*, by ACM Press, 285-292. New York, N.Y., 2001. Available from <http://doi.acm.org/10.1145/365024.365117>.
- Bulterman, D. C. "Using SMIL to encode interactive, peer-level multimedia annotations". In *Proceedings of the 2003 ACM Symposium on Document Engineering, Grenoble, France, November 20-22, 2003*, by ACM Press, 32-41. New York, N.Y., 2003. Available from <http://doi.acm.org/10.1145/958220.958228>.
- Cadiz, J. J., A. Gupta, and J. Grudin. "Using web annotations for asynchronous collaboration around documents." In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, Philadelphia, Pa., USA*, by ACM Press, 309-318. New York, N.Y., 2000. Available from <http://doi.acm.org/10.1145/358916.359002>.
- Carter, S., E. Churchill, L. Denoue, J. Helfman, and L. Nelson. "Digital Graffiti: public annotation of multimedia content." In *CHI '04 Extended Abstracts on Human Factors in Computing Systems, Vienna, Austria, April 24-29, 2004*, by ACM Press, 1207-1210. New York, N.Y., 2004. Available from <http://doi.acm.org/10.1145/985921.986025>.

- Cimiano, P., S. Handschuh, and S. Staab. "Towards the self-annotating web." In *Proceedings of the 13th International Conference on World Wide Web, New York, N.Y., USA, May 17-20, 2004*, by ACM Press, 462-471. New York, N.Y., 2004. Available from <http://doi.acm.org/10.1145/988672.988735>.
- Clyde, L. A. "Electronic whiteboards." *Teacher Librarian: The Journal for School Library Professionals* 32, no. 2 (December 2004): 43-44. Available from [http://www.teacherlibrarian.com/tltoolkit/info\\_tech/info\\_tech\\_32\\_2.html](http://www.teacherlibrarian.com/tltoolkit/info_tech/info_tech_32_2.html); Accessed September 14, 2006.
- Collis, B. and W. De Boer. "Teachers as learners: embedded tools for implementing a CMS." *TechTrends* 48 no. 6 (2004):7-12.
- Di Giacomo, M. "MySQL: lessons learned on a digital library." *IEEE Software* 22, issue 3 (May-June 2005):10-13.
- Dmitriev, P. A., N. Eiron, M. Fontoura, and E. Shekita. "Using annotations in enterprise search." In *Proceedings of the 15th International Conference on World Wide Web, Edinburgh, Scotland, May 23-26, 2006*, by ACM Press, 811-817. New York, N.Y., 2006. Available from <http://doi.acm.org/10.1145/1135777.1135900>.
- Documet, N. "REST based authentication." June 18, 2006. Available from <http://nanodocumet.homedns.org/rest/>; Accessed August 5, 2006.
- Dominus, M. -J. "Perl: Not just for web programming." *IEEE Software* 15, issue 1 (January-February 1998):69-74.
- Eberhardt, R. "Internet Explorer SELECT element bugs, test suite." September 24, 2004. Available at <http://throbs.net/web/articles/IE-SELECT-bugs/>; Accessed October 6, 2006.
- Gabrielli, S. and A. Law. "Annotation in the wild: Benefits of linking paper to digital media." In *CHI '03 Extended Abstracts on Human Factors in Computing Systems, Ft. Lauderdale, Fla., USA, April 5-10, 2003*, by ACM Press, 890-891. New York, N.Y., 2003. Available from <http://doi.acm.org/10.1145/765891.766052>.
- Gehring, E. F. "Why aren't course-management systems penetrating faster?" In *Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition, 2003*, by American Society for Engineering Education, Session 2158, 2003.

- Golovchinsky, G. and L. Denoue. "Moving markup: repositioning freeform annotations." In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology, Paris, France, October 27-30, 2002*, by ACM Press, 21-30. New York, N.Y., 2002. Available from <http://doi.acm.org/10.1145/571985.571989>.
- Handschuh, S. and S. Staab. "Authoring and annotation of web pages in CREAM." In *Proceedings of the 11th International Conference on World Wide Web, Honolulu, Hawaii, USA, May 7 - 11, 2002*, by ACM Press, 462-473. New York, N.Y., 2002. Available from <http://doi.acm.org/10.1145/511446.511506>.
- Handschuh, S., S. Staab, and A. Maedche. "CREAM: creating relational metadata with a component-based, ontology-driven annotation framework." In *Proceedings of the 1st International Conference on Knowledge Capture, Victoria, British Columbia, Canada, October 22-23, 2001*, by ACM Press, 76-83. New York, N.Y., 2001. Available from <http://doi.acm.org/10.1145/500737.500752>.
- Hansen, F. A. 2006. "Ubiquitous annotation systems: technologies and challenges." In *Proceedings of the Seventeenth Conference on Hypertext and Hypermedia, Odense, Denmark, August 22-25, 2006*, by ACM Press, 121-132. New York, N.Y., 2006. Available from <http://doi.acm.org/10.1145/1149941.1149967>.
- Harmon, T., R. Helps, and M. Bailey. "Electronic assessment, evaluation and archiving of online student assignments." In *Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition, Portland, Ore., USA, 2005*, by American Society for Engineering Education, 2005.
- Hernández, L. O. and M. Pegah. "WebDAV: What it is, what it does, why you need it." In *Proceedings of the 31st Annual ACM SIGUCCS Conference on User Services, San Antonio, Tex., USA, September 21-24, 2003*, by ACM Press, 249-254. New York, N.Y., 2003. Available from <http://doi.acm.org/10.1145/947469.947535>.
- Holmquist, L. E., J. Sanneblad, and L. Gaye. "Total recall: in-place viewing of captured whiteboard annotations." In *CHI '03 Extended Abstracts on Human Factors in Computing Systems, Ft. Lauderdale, Fla., USA, April 5-10, 2003*, by ACM Press, 980-981. New York, N.Y., 2003. Available from <http://doi.acm.org/10.1145/765891.766105>.
- Hong, L., E. H. Chi, and S. K. Card. "Annotating 3D electronic books." In *CHI '05 Extended Abstracts on Human Factors in Computing Systems, Portland, Ore., USA, April 2-7, 2005*, by ACM Press, 1463-1466. New York, N.Y., 2005. Available from <http://doi.acm.org/10.1145/1056808.1056942>.

- Jacquin, T., O. Fambon, and B. Chidlovskii. "A web-based document harmonization and annotation chain: from PDF to RDF." In *Proceedings of the 2005 ACM Symposium on Document Engineering, Bristol, United Kingdom, November 2-4, 2005*, by ACM Press, 225-226. New York, N.Y., 2005. Available from <http://doi.acm.org/10.1145/1096601.1096656>.
- Jupitermedia Corporation. "TheCounter.com - The Affordable Web Site Analysis Tool." Available from <http://www.thecounter.com/stats/>; Accessed September 1, 2006.
- Kahan, J. and M. Koivunen. "Annotea: an open RDF infrastructure for shared Web annotations." In *Proceedings of the 10th International Conference on World Wide Web, Hong Kong, Hong Kong, May 1-5, 2001*, by ACM Press, 623-632. New York, N.Y., 2001. Available from <http://doi.acm.org/10.1145/371920.372166>.
- Kim, S., M. Slater., and E. J. Whitehead. "WebDAV-based hypertext annotation and trail system". In *Proceedings of the Fifteenth ACM Conference on Hypertext and Hypermedia, Santa Cruz, Calif., USA, August 9-13, 2004*, by ACM Press, 87-88. New York, N.Y., 2004. Available from <http://doi.acm.org/10.1145/1012807.1012832>.
- Koivunen, M. -R, "Annotea project." *W3C*. October 10, 2005. Available from <http://www.w3.org/2001/Annotea/>; Accessed September 20, 2006.
- Le Hégarret, P., R. Whitmer, and L. Wood. "W3C Document Object Model." *W3C*. January 19, 2005. Revised June 12, 2006. Available at <http://www.w3.org/DOM/>; Accessed October 9, 2006.
- Leslie, S. "Open source course management systems." EdTechPost Wiki. Available from <http://www.edtechpost.ca/pmwiki/pmwiki.php/EdTechPost/OpenSourceCourseManagementSystems>; Accessed October 3, 2006.
- Maher, T. "A minimal Perl dialect for UNIX & Linux people." *OSCON 2005* [August 1-5, 2005]. Speech Slides. Available at <http://minimalperl.com/consultix/publications/slides.html>; Accessed September 14, 2006.
- Marshall, C. C. and A. B. Brush. "Exploring the relationship between personal and public annotations." In *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries, Tuscon, Ariz., USA, June 7-11, 2004*, by ACM Press, 349-357. New York, N.Y., 2004. Available from <http://doi.acm.org/10.1145/996350.996432>.

- Marshall, C. C. and A. B. Brush. "From personal to shared annotations." In *CHI '02 Extended Abstracts on Human Factors in Computing Systems, Minneapolis, Minn., USA, April 20-25, 2002*, by ACM Press, 812-813. New York, N.Y., 2002. Available from <http://doi.acm.org/10.1145/506443.506610>.
- Mock, K. 2004. "Teaching with Tablet PC's." In *Journal of Computing Sciences in Colleges* 20, no. 2 (Dec. 2004):17-27. Consortium for Computing Sciences in Colleges, USA.
- Microsoft Corporation. "INFO: How the z-index attribute works for HTML elements." Microsoft article ID 177378. Rev. 3.0. 11 May 2006. Available at <http://support.microsoft.com/default.aspx?scid=kb;en-us;177378&Product=ieip>; accessed October 6, 2006.
- Olsen, D. R., T. Taufer, and J. A. Fails. "ScreenCrayons: Annotating anything." In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology, Santa Fe, N.Mex., USA, October 24-27, 2004*, by ACM Press, 165-174. New York, N.Y., 2004. Available from <http://doi.acm.org/10.1145/1029632.1029663>.
- Plimmer, B. and P. Mason. "A pen-based paperless environment for annotating and marking student assignments." In *Proceedings of the 50th Conference on User Interfaces 2006*, Hobart, Australia, January 16-19, 2006, W. Piekarski, ed., by Australian Computer Society, 50:37-44. Darlinghurst, Australia, 2006.
- Popyack, J. L., and N. Herrmann. "Electronic grading: When the tablet is mightier than the pen." *Syllabus* 16, no. 6 (January 2003): 18-20.
- Rouyer, J. *Dynamic HTML Web Magic*. Indianapolis, Ind.: New Riders, 1998.
- Sannomiya, T., T. Amagasa, M. Yoshikawa, and S. Uemura. "A framework for sharing personal annotations on web resources using XML." In *Proceedings of the Workshop on Information Technology For Virtual Enterprises*, Queensland, Australia, January 29-30, 2001, by IEEE Computer Society, 13:40-48. Washington, D.C. 2001.
- Sastry, C. R., D. P. Lewis, and A. Pizano. "Webtour: A system to record and playback dynamic multimedia annotations on web document content." In *Proceedings of the Seventh ACM International Conference on Multimedia (Part 2)*, Orlando, Fla., USA, October 30-November 05, 1999, by ACM Press, 175-178. New York, N.Y., 1999. Available from <http://doi.acm.org/10.1145/319878.319925>.

- Schilit, B. N., G. Golovchinsky, and M. N. Price. "Beyond paper: supporting active reading with free form digital ink annotations." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Los Angeles, Calif., USA, April 18-23, 1998*, C. Karat, A. Lund, J. Coutaz, and J. Karat, eds., by ACM Press/Addison-Wesley Publishing Co., 249-256. New York, N.Y., 1998. Available from <http://doi.acm.org/10.1145/274644.274680>.
- Segawa, O. "Web annotation sharing using P2P." In *Proceedings of the 15th International Conference on World Wide Web, Edinburgh, Scotland, May 23-26, 2006*, by ACM Press, 851-852. New York, N.Y., 2006. Available from <http://doi.acm.org/10.1145/1135777.1135910>.
- Sevasti, A. and B. Christos. "Using Java to implement a multimedia annotation environment for young children." In *Proceedings of the Eighth ACM international Conference on Multimedia, Marina del Rey, Calif., USA*, by ACM Press, 187-194. New York, N.Y., 2000. Available from <http://doi.acm.org/10.1145/354384.354465>.
- Shevade, B. and H. Sundaram. "Vidya: An experiential annotation system." In *Proceedings of the 2003 ACM SIGMM Workshop on Experiential Telepresence, Berkeley, Calif., USA, 2003*, by ACM Press, 91-98. New York, N.Y., 2003. Available from <http://doi.acm.org/10.1145/982484.982502>.
- Siever, E., S. Spainhour, and N. Patwardhan. *Perl in a Nutshell*. Sebastopol, Calif.: O'Reilly & Associates, Inc., 1999.
- Swick, R., E. Prud'hommeaux, M. -R. Koivunen, J. Kahan. "Annotea protocols." *W3C*. December 19, 2002. Draft Rev. 1.314. Available from <http://www.w3.org/2002/12/AnnoteaProtocol-20021219>; Accessed September 20, 2006.
- Takeda, T. and D. Suthers. "Online workspaces for annotation and discussion of documents." in *Proceedings of the 2002 International Conference on Computers in Education, December 3-6, 2002*, by IEEE. 2 1294-1298. 2002.
- Timmins, S. J. "Tablet PC: Blackboard to the web." In *Proceedings of the 32nd Annual ACM SIGUCCS Conference on User Services, Baltimore, Md., USA, October 10-13, 2004*, by ACM Press, 296-300. New York, N.Y., 2004. Available from <http://doi.acm.org/10.1145/1027802.1027870>.
- van der Blonk, M. "Firefox and the endless growing select (impossible resize)." February 9, 2006. Available at <http://michiell.wordpress.com/2006/02/09/firefox-and-the-endless-growing-select-impossible-resize/>; Accessed October 6, 2006.



- Vatton, I. "Welcome to Amaya." *W3C*. August 30, 2006. Available from <http://www.w3.org/Amaya/>; Accessed September 20, 2006.
- Volkmer, T., J. R. Smith, and A. Natsev. "A web-based system for collaborative annotation of large image and video collections: an evaluation and user study." In *Proceedings of the 13th Annual ACM international Conference on Multimedia, Hilton, Singapore, November 6-11, 2005*, by ACM Press, 892-901. New York, N.Y., 2005. Available from <http://doi.acm.org/10.1145/1101149.1101341>.
- Wall, K., S. Higgins, and H. Smith. "The visual helps me understand the complicated things': Pupil views of teaching and learning with interactive whiteboards." In *British Journal of Educational Technology* 26, no. 5 (S 2005):851-67. Blackwell Publishers, United Kingdom. 2005.
- Wall, L. "Natural language principles in Perl." Available from <http://www.wall.org/~larry/natural.html>; Accessed January 19, 2006.
- Wang, C. and G. Chen. "Extending e-books with annotation, online support and assessment mechanisms to increase efficiency of learning." In *Proceedings of the 9th Annual SIGCSE Conference on innovation and Technology in Computer Science Education, Leeds, United Kingdom, June 28-30, 2004*, by ACM Press, 132-136. New York, N.Y., 2004. Available from <http://doi.acm.org/10.1145/1007996.1008032>.
- WaSP. "Frequently asked questions (FAQ) – The Web Standards Project." Available from <http://www.webstandards.org/learn/faq/>; Accessed October 7, 2006.
- Weng, C. and J. H. Gennari. "Asynchronous collaborative writing through annotations." In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, Chicago, Ill., USA, November 6-10, 2004*, by ACM Press, 578-581. New York, N.Y., 2004. Available from <http://doi.acm.org/10.1145/1031607.1031705>.
- West, J. V. "Using Ink in Microsoft Word." May 28, 2003. Available from [http://www.microsoft.com/windowsxp/using/tabletpc/learnmore/vanwest\\_03may28inkword.msp](http://www.microsoft.com/windowsxp/using/tabletpc/learnmore/vanwest_03may28inkword.msp); Accessed September 20, 2006.
- Willis, C. L. and L. Miertschin. "Tablet PC's as instructional tools or the pen is mightier than the 'board!'" In *Proceedings of the 5th Conference on information Technology Education, Salt Lake City, Utah, USA, October 28-30, 2004*, by ACM Press, 153-159. New York, N.Y., 2004. Available from <http://doi.acm.org/10.1145/1029533.1029572>.

- Wolfe, J. L. "Effects of annotations on student readers and writers." In *Proceedings of the Fifth ACM Conference on Digital Libraries, San Antonio, Tex., USA, June 2-7, 2000*, by ACM Press, 19-26. New York, N.Y., 2000. Available from <http://doi.acm.org/10.1145/336597.336620>.
- Wu, X., L. Zhang, and Y. Yu. "Exploring social annotations for the semantic web." In *Proceedings of the 15th international Conference on World Wide Web, Edinburgh, Scotland, May 23-26, 2006*, by ACM Press, 417-426. New York, N.Y., 2006. Available from <http://doi.acm.org/10.1145/1135777.1135839>.
- Zellweger, P. T., N. O. Bouvin, H. Jehøj, and J. D. Mackinlay. "Fluid annotations in an open world." In *Proceedings of the Twelfth ACM Conference on Hypertext and Hypermedia, Århus, Denmark, August 14-18, 2001*, by ACM Press, 9-18. New York, N.Y., 2001. Available from <http://doi.acm.org/10.1145/504216.504224>.
- Zheng, Q., K. Booth, and J. McGrenere. "Co-authoring with structured annotations." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Montréal, Québec, Canada, April 22-27, 2006*, R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, eds., by ACM Press, 131-140. New York, N.Y., 2006. Available from <http://doi.acm.org/10.1145/1124772.1124794>.





## LICENSING



## **Licensing**

Except where noted, this thesis' text is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 2.5 license. Source code not marked as being under a different license are licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 2.5 license. This license does not override any previous licenses or copyrights attached to source code, which are still in effect.

### **Attribution-NonCommercial-NoDerivs 2.5**

Creative Commons Legal Code  
Attribution-NonCommercial-NoDerivs 2.5

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

#### *License*

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN

## CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

### 1. Definitions

- a. "Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
- b. "Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.
- c. "Licensor" means the individual or entity that offers the Work under the terms of this License.
- d. "Original Author" means the individual or entity who created the Work.
- e. "Work" means the copyrightable work of authorship offered under the terms of this License.
- f. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- b. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Derivative Works. All rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Sections 4(d) and 4(e).

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

- a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(c), as requested.
- b. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.
- c. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work, You must keep intact all copyright notices for the Work and provide,

reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; and to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

- d. For the avoidance of doubt, where the Work is a musical composition:
- i. Performance Royalties Under Blanket Licenses. Licensor reserves the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work if that performance is primarily intended for or directed toward commercial advantage or private monetary compensation.
  - ii. Mechanical Rights and Statutory Royalties. Licensor reserves the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions), if Your distribution of such cover version is primarily intended for or directed toward commercial advantage or private monetary compensation.
- e. Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor reserves the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions), if Your public digital performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

## 5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS

OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

#### 6. Limitation on Liability.

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 7. Termination

- a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
- b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

#### 8. Miscellaneous

- a. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
- b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of



this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

- c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
- d. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, neither party will use the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time.

Creative Commons may be contacted at <http://creativecommons.org/>.

#### **Attribution-NonCommercial-ShareAlike 2.5**

Creative Commons Legal Code  
Attribution-NonCommercial-ShareAlike 2.5

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

## *License*

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

### 1. Definitions

1. "Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
2. "Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.
3. "Licensor" means the individual or entity that offers the Work under the terms of this License.
4. "Original Author" means the individual or entity who created the Work.
5. "Work" means the copyrightable work of authorship offered under the terms of this License.
6. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who

has received express permission from the Licensor to exercise rights under this License despite a previous violation.

7. "License Elements" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, Noncommercial, ShareAlike.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

1. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
2. to create and reproduce Derivative Works;
3. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
4. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works;

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Sections 4(e) and 4(f).

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

1. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may

not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(d), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(d), as requested.

2. You may distribute, publicly display, publicly perform, or publicly digitally perform a Derivative Work only under the terms of this License, a later version of this License with the same License Elements as this License, or a Creative Commons iCommons license that contains the same License Elements as this License (e.g. Attribution-NonCommercial-ShareAlike 2.5 Japan). You must include a copy of, or the Uniform Resource Identifier for, this License or other license specified in the previous sentence with every copy or phonorecord of each Derivative Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Derivative Works that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder, and You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Derivative Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Derivative Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Derivative Work itself to be made subject to the terms of this License.
3. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.
4. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity,

journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. For the avoidance of doubt, where the Work is a musical composition:
  1. Performance Royalties Under Blanket Licenses. Licensor reserves the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work if that performance is primarily intended for or directed toward commercial advantage or private monetary compensation.
  2. Mechanical Rights and Statutory Royalties. Licensor reserves the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions), if Your distribution of such cover version is primarily intended for or directed toward commercial advantage or private monetary compensation.
6. Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor reserves the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions), if Your public digital performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

#### 5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS

OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

#### 6. Limitation on Liability.

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 7. Termination

1. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
2. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

#### 8. Miscellaneous

1. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
2. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

3. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
4. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
5. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, neither party will use the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time.

Creative Commons may be contacted at <http://creativecommons.org/>.

## APPENDICES





## Appendix A – API Documentation

The *API* documentation contains a section for each of the Perl 5 modules developed for the system. This documentation is derived from the *POD* documentation found in each module and converted using *pod2text*. Other *POD* converters are available for other output file formats.

### A.1. OAS::Annotation.pm

#### NAME

OAS::Annotation - Perl extension database interface for an OAS annotation object

#### SYNOPSIS

```
use OAS::Annotation;
```

#### DESCRIPTION

This object module provides a front-end interface for the OAS page object, as well as additional helper functions. The module handles all the data verification and database interfacing for the object.

#### Non-Standard Required Modules

```
OAS::System  
OAS::DBI  
OAS::Document  
OAS::Page  
OAS::Person  
OAS::Utils
```

#### Multi-Threading

Basic support for multi-threaded applications is included in the module. However, this has not been fully tested, and should be considered EXPERIMENTAL at best.

#### Exported

None by default.

#### Export OK

```
annotation_list  
annotation_owner  
annotation_encode  
annotation_unencode  
is_valid_annotation_id  
is_valid_author
```

```

is_valid_owner
is_valid_page
is_valid_type
is_valid_width
is_valid_height
is_valid_x_pos
is_valid_y_pos
is_valid_timestamp
is_valid_font
is_valid_pitch
is_valid_color
is_valid_background
is_valid_image
is_valid_text
untaint_annotation_id
untaint_author
untaint_owner
untaint_page
untaint_type
untaint_width
untaint_height
untaint_x_pos
untaint_y_pos
untaint_timestamp
untaint_font
untaint_pitch
untaint_color
untaint_background
untaint_image
untaint_text

```

#### Export Tags

```

check => [ is_valid_annotation_id is_valid_author is_valid_owner
           is_valid_page          is_valid_type   is_valid_width
           is_valid_height        is_valid_x_pos  is_valid_y_pos
           is_valid_timestamp     is_valid_font is_valid_pitch
           is_valid_color         is_valid_background is_valid_image
           is_valid_text          ]

untaint => [ untaint_annotation_id untaint_author untaint_owner
            untaint_page          untaint_type   untaint_width
            untaint_height        untaint_x_pos  untaint_y_pos
            untaint_timestamp     untaint_font untaint_pitch
            untaint_color         untaint_background untaint_image
            untaint_text          ]

```

#### Event Handlers

##### DESTROY

Disconnects the object from the database when the object is either undef'd, over-written or drops out of scope. The parent object's DESTROY method is also called.

#### Functions

##### annotation\_list (\$\$)

\$: Document ID

\$: Page Number

Returns a hash reference containing all associated annotations if successful, else undef

The hash reference is multi-dimensional. The first level is keyed using the Timestamp and ID in order to allow for easy sorting from oldest to youngest. The second level is keyed using the columns from the database table. If an error occurs, undef is returned.

##### annotation\_owner (\$)

\$: ID to look up owner for

Returns the ID of the Person who owns the annotation

`annotation_encode ($)`  
`$`: Text to encode for inclusion in an annotation  
Returns encoded text

`annotation_unencode ($)`  
`$`: Encoded text to change back to normal text  
Returns unencoded text

`is_valid_id ($)`  
`$`: ID to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_annotation_id ($)`  
`$`: ID to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_author ($)`  
`$`: AUTHOR to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_owner ($)`  
`$`: OWNER to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_page ($)`  
`$`: PAGE to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_type ($)`  
`$`: TYPE to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies that TYPE is defined and either equal to \*Image\* or \*Text\*.

`is_valid_width ($)`  
`$`: WIDTH to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_height ($)`  
`$`: HEIGHT to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_x_pos ($)`  
`$`: X\_POS to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_y_pos ($)`  
 \$: Y\_POS to be verified  
 Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_timestamp ($)`  
 \$: TIMESTAMP to be verified  
 Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_font ($)`  
 \$: FONT to be verified  
 Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_pitch ($)`  
 \$: FONT PITCH to be verified  
 Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_color ($)`  
 \$: FONT COLOR to be verified  
 Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_background ($)`  
 \$: BACKGROUND COLOR to be verified  
 Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_image ($)`  
 \$: IMAGE FILENAME to be verified  
 Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_text ($)`  
 \$: TEXT FILENAME to be verified  
 Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`untaint_id ($)`  
 \$: Value to be untainted  
 Returns untainted value if value is of the correct format, else undef

This function calls `is_valid_id` to verify the format of the value.

untaint\_annotation\_id (\$)
   
\$: Value to be untainted
   
Returns untainted value if value is of the correct format, else undef
   
 This function calls is\_valid\_id to verify the format of the value.

untaint\_author (\$)
   
\$: Value to be untainted
   
Returns untainted value if value is of the correct format, else undef
   
 This function calls is\_valid\_ to verify the format of the value.

untaint\_owner (\$)
   
\$: Value to be untainted
   
Returns untainted value if value is of the correct format, else undef
   
 This function calls is\_valid\_ to verify the format of the value.

untaint\_page (\$)
   
\$: Value to be untainted
   
Returns untainted value if value is of the correct format, else undef
   
 This function calls is\_valid\_page to verify the format of the value.

untaint\_type (\$)
   
\$: Value to be untainted
   
Returns untainted value if value is of the correct format, else undef
   
 This function calls is\_valid\_type to verify the format of the value.

untaint\_width (\$)
   
\$: Value to be untainted
   
Returns untainted value if value is of the correct format, else undef
   
 This function calls is\_valid\_width to verify the format of the value.

untaint\_height (\$)
   
\$: Value to be untainted
   
Returns untainted value if value is of the correct format, else undef
   
 This function calls is\_valid\_height to verify the format of the value.

untaint\_x\_pos (\$)
   
\$: Value to be untainted
   
Returns untainted value if value is of the correct format, else undef
   
 This function calls is\_valid\_x\_pos to verify the format of the value.

untaint\_y\_pos (\$)
   
\$: Value to be untainted
   
Returns untainted value if value is of the correct format, else undef
   
 This function calls is\_valid\_y\_pos to verify the format of the value.

untaint\_timestamp (\$)
   
\$: Value to be untainted
   
Returns untainted value if value is of the correct format, else undef
   
 This function calls is\_valid\_timestamp to verify the format of the value.

untaint\_font (\$)
   
\$: Value to be untainted
   
Returns untainted value if value is of the correct format, else undef
   
 This function calls is\_valid\_font to verify the format of the value.

```

untaint_pitch ($)
  $: Value to be untainted
  Returns untainted value if value is of the correct format, else undef

  This function calls is_valid_pitch to verify the format of the value.

untaint_color ($)
  $: Value to be untainted
  Returns untainted value if value is of the correct format, else undef

  This function calls is_valid_color to verify the format of the value.

untaint_background ($)
  $: Value to be untainted
  Returns untainted value if value is of the correct format, else undef

  This function calls is_valid_background to verify the format of the value.

untaint_image ($)
  $: Value to be untainted
  Returns untainted value if value is of the correct format, else undef

  This function calls is_valid_image to verify the format of the value.

untaint_text ($)
  $: Value to be untainted
  Returns untainted value if value is of the correct format, else undef

  This function calls is_valid_text to verify the format of the value.

OAS::Annotation Object - Data Control Methods
new ( )
  Returns a blessed reference to a new OAS::Annotation object

  Creates a new, empty OAS::Annotation object.

Load ($$)
  $: Document ID of the page to load from the database
  $: Page number to load from the database
  Returns an 'evaluate-true' if successful, else undef

  If a non-fatal error occurs, error_message can be checked for a description
  of the error.

Save (;$)
  $: Forces save if evaluates true
  Returns an 'evaluate-true' if successful, else undef

  Commits changes to the database. If a non-fatal error occurs, error_message
  can be checked for a description of the error.

Delete ( )
  Attempts to delete the object from the database. The deletion may fail if
  database foreign-key restraints do not permit it. If a non-fatal error
  occurs, error_message can be checked for a description of the error. This
  function DOES NOT deallocate the object, but instead resets it to an empty
  state, just as if it had been invoked through new.

clear_error ( )
  Clears the last error string stored in the object.

_reset_defaults ( ) [Internal]
  Resets all members to default value (undef)

OAS::Annotation Object - Display Methods
html ( )
  Returns HTML for annotation

```

OAS::Annotation Object - Data Access Methods

id (;\$)

\$: New ID for Annotation

Returns current Annotation ID if no error occurred, else undef

Called with no arguments, the function returns the current Annotation ID. If an argument is supplied, it attempts to set the Annotation ID to this new value, if it has a valid format. If the parameter is an empty string '', the Annotation ID is set to undef, which is then returned.

author (;\$)

\$: New Author

Returns current Author if no error occurred, else undef

Called with no arguments, the function returns the current Author. If an argument is supplied, it attempts to set the Author to this new value, if it has a valid format. If the parameter is an empty string '', the Author is set to undef, which is then returned.

owner (;\$)

\$: New Owner

Returns current Owner if no error occurred, else undef

Called with no arguments, the function returns the current Owner. If an argument is supplied, it attempts to set the Owner to this new value, if it has a valid format. If the parameter is an empty string '', the Owner is set to undef, which is then returned.

document\_id (;\$)

\$: New Document ID

Returns current Document ID if no error occurred, else undef

Called with no arguments, the function returns the current Document ID. If an argument is supplied, it attempts to set the Document ID to this new value, if it has a valid format. If the parameter is an empty string '', the Document ID is set to undef, which is then returned.

page (;\$)

\$: New Page number

Returns current Page number if no error occurred, else undef

Called with no arguments, the function returns the current Page number. If an argument is supplied, it attempts to set the Page number to this new value, if it has a valid format. If the parameter is an empty string '', the Page number is set to undef, which is then returned.

type (;\$)

\$: New Type

Returns current Type if no error occurred, else undef

Called with no arguments, the function returns the current Type. If an argument is supplied, it attempts to set the Type to this new value, if it has a valid format. If the parameter is an empty string '', the Type is set to undef, which is then returned.

width (;\$)

\$: New Width

Returns current Width if no error occurred, else undef

Called with no arguments, the function returns the current Width. If an argument is supplied, it attempts to set the Width to this new value, if it has a valid format. If the parameter is an empty string '', the Width is set to undef, which is then returned.

height (;\$)

\$: New Height

Returns current Height if no error occurred, else undef



Called with no arguments, the function returns the current Height. If an argument is supplied, it attempts to set the Height to this new value, if it has a valid format. If the parameter is an empty string '', the Height is set to undef, which is then returned.

```
x_pos (;$)
  $: New X-Position of the upper-left corner of the annotation
  Returns current X-Position if no error occurred, else undef
```

Called with no arguments, the function returns the current X-Position. If an argument is supplied, it attempts to set the X-Position to this new value, if it has a valid format. If the parameter is an empty string '', the X-Position is set to undef, which is then returned.

```
y_pos (;$)
  $: New Y-Position of the upper-left corner of the annotation
  Returns current Y-Position if no error occurred, else undef
```

Called with no arguments, the function returns the current Y-Position. If an argument is supplied, it attempts to set the Y-Position to this new value, if it has a valid format. If the parameter is an empty string '', the Y-Position is set to undef, which is then returned.

```
timestamp (;$)
  $: New Timestamp
  Returns current Timestamp if no error occurred, else undef
```

Called with no arguments, the function returns the current Timestamp. If an argument is supplied, it attempts to set the Timestamp to this new value, if it has a valid format. If the parameter is an empty string '', the Timestamp is set to undef, which is then returned.

```
font (;$)
  $: New Font type/family/name
  Returns current Font if no error occurred, else undef
```

Called with no arguments, the function returns the current Font. If an argument is supplied, it attempts to set the Font to this new value, if it has a valid format. If the parameter is an empty string '', the Font is set to undef, which is then returned.

```
pitch (;$)
  $: New Font Pitch
  Returns current Font Pitch if no error occurred, else undef
```

Called with no arguments, the function returns the current Font Pitch. If an argument is supplied, it attempts to set the Font Pitch to this new value, if it has a valid format. If the parameter is an empty string '', the Font Pitch is set to undef, which is then returned.

```
color (;$)
  $: New Font Color
  Returns current Font Color if no error occurred, else undef
```

Called with no arguments, the function returns the current Font Color. If an argument is supplied, it attempts to set the Font Color to this new value, if it has a valid format. If the parameter is an empty string '', the Font Color is set to undef, which is then returned.

```
background (;$)
  $: New Background Color
  Returns current Background Color if no error occurred, else undef
```

Called with no arguments, the function returns the current Background Color. If an argument is supplied, it attempts to set the Background Color to this new value, if it has a valid format. If the parameter is an empty string '', the Background Color is set to undef, which is then returned.

image (;\$)  
 \$: New Image Filename  
 Returns current Image Filename if no error occurred, else undef

Called with no arguments, the function returns the current Image Filename.  
 If an argument is supplied, it attempts to set the Image Filename to this new value, if it has a valid format. If the parameter is an empty string '', the Image Filename is set to undef, which is then returned.

text (;\$)  
 \$: New Text Filename of file containing text of annotation  
 Returns current Text Filename if no error occurred, else undef

Called with no arguments, the function returns the current Text Filename.  
 If an argument is supplied, it attempts to set the Text Filename to this new value, if it has a valid format. If the parameter is an empty string '', the Text Filename is set to undef, which is then returned.

error\_message ( )  
 Returns the current error message string

OAS::Page Object - SQL Schema  
 The following SQL was optimized for MySQL 4.1.14

```
CREATE TABLE annotation (
  id          CHAR      ( 16) ASCII          NOT NULL,
  author      CHAR      ( 10) ASCII          NOT NULL,
  owner       CHAR      ( 10) ASCII          NOT NULL,
  document_id CHAR      ( 16) ASCII          NOT NULL,
  page        INT                NOT NULL,
  type        ENUM          ('Image', 'Text') NOT NULL,
  width       INT                NOT NULL,
  height      INT                NOT NULL,
  x_pos       INT                NOT NULL,
  y_pos       INT                NOT NULL,
  timestamp   BIGINT   ( 10)          NOT NULL,
  font        CHAR      ( 20) ASCII          DEFAULT NULL,
  pitch       INT          ( 2)          DEFAULT NULL,
  color       CHAR      ( 16) ASCII          DEFAULT NULL,
  background  CHAR      ( 16) ASCII          DEFAULT NULL,
  image       VARCHAR (256) ASCII          DEFAULT NULL,
  text        VARCHAR (256) ASCII          DEFAULT NULL,
  PRIMARY KEY (id),
  INDEX       (author),
  INDEX       (owner),
  INDEX       (document_id),
  INDEX       (page),
  FOREIGN KEY (author)
  REFERENCES person(id)
  ON UPDATE  CASCADE
  ON DELETE  RESTRICT,
  INDEX       (owner),
  FOREIGN KEY (owner)
  REFERENCES person(id)
  ON UPDATE  CASCADE
  ON DELETE  RESTRICT,
  FOREIGN KEY (document_id, page)
  REFERENCES page(id, page)
  ON UPDATE  CASCADE
  ON DELETE  CASCADE
) TYPE=INNODB;
```

SEE ALSO  
 AUTHOR  
 Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE  
 Copyright (c) 2006 by Trev Harmon

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.5 or, at your option, any later version of Perl 5 you may have available.

## A.2. OAS::Auth.pm

### NAME

OAS::Auth - Perl extension providing authentication services for OAS

### SYNOPSIS

```
use OAS::Auth;
```

### DESCRIPTION

This module provides different functions for implementing authentication features for OAS scripts and programs.

### Non-Standard Required Modules

```
OAS::DBI
OAS::Person
OAS::Document
```

### Exported

None by default.

### Export OK

```
ADMIN_GROUP
ACL_READ_ACCESS
ACL_DELETE_ACCESS
ACL_ANNOTATE_ACCESS
ACL_READ_ANNOTATION_ACCESS
ACL_DELETE_ANNOTATION_ACCESS
ACL_MODERATE_ACCESS
ACL_ALL_ACCESS
current_user
is_user
is_admin
list_all_users
is_valid_group
create_group
remove_group
list_groups
list_users_groups
group_owner
is_group_member
add_user_to_group
remove_user_from_group
list_group_membership
is_valid_acl_access
grant_user_access
grant_group_access
revoke_user_access
revoke_group_access
has_access
has_user_access
has_group_access
list_access
list_user_access
list_group_access
has_read_access
has_delete_access
has_annotate_access
has_read_annotation_access
has_delete_annotation_access
has_moderate_access
who_has_user_access
```

```

    who_has_group_access
    list_accessible_documents
    write_htaccess
    error_message
    clear_error

Export Tags
  user => [ current_user          is_user
            is_admin             list_all_users          ]

  group => [ ADMIN_GROUP
            is_valid_group       create_group
            remove_group         list_groups
            list_users_groups    group_owner
            is_group_member      add_user_to_group
            remove_user_from_group list_group_membership    ]

  acl  => [ ACL_READ_ACCESS      ACL_DELETE_ACCESS
            ACL_ANNOTATE_ACCESS  ACL_READ_ANNOTATION_ACCESS
            ACL_DELETE_ANNOTATION_ACCESS ACL_MODERATE_ACCESS
            ACL_ALL_ACCESS
            is_valid_acl_access  grant_user_access
            grant_group_access  revoke_user_access
            revoke_group_access has_access
            has_user_access     has_group_access
            list_access         list_user_access
            list_group_access   has_read_access
            has_delete_access   has_annotate_access
            has_read_annotation_access has_delete_annotation_access
            has_moderate_access  who_has_user_access
            who_has_group_access list_accessible_documents
            write_htaccess      ]

  error => [ error_message      clear_error          ]

Constants
  ADMIN_GROUP
    Name of the built-in administration group

  ACL_READ_ACCESS
    String version of document read access permission

  ACL_DELETE_ACCESS
    String version of document delete access permission

  ACL_ANNOTATE_ACCESS
    String version of document annotate access permission

  ACL_READ_ANNOTATION_ACCESS
    String version of annotation read access permission

  ACL_DELETE_ANNOTATION_ACCESS
    String version of annotation delete access permission

  ACL_MODERATE_ACCESS
    String version of document moderator access permission

  ACL_ALL_ACCESS
    Array containing string versions of all access permissions

Event Handlers
  END
    Disconnects from the database.

User Functions
  current_user ( )
    Returns the current user per the web authorization

```

```

is_user ($)
    $: ID of person to check
    Returns an 'evaluate-true' or 'evaluate-false'

This function checks to see if the ID has basic access to the system. If a
non-fatal error occurs, this returns undef. Check error_message for a
verbose error message.

is_admin ($)
    $: ID of person to check
    Returns an 'evaluate-true' or 'evaluate-false'

This function checks to see if the ID has administrative access to the
system. If a non-fatal error occurs, this returns undef. Check
error_message for a verbose error message.

list_all_users ( )
    Returns an reference to an array of hashes containing all IDs, LAST_NAMES and
    FIRST_NAMES in the system

The list is sorted by LAST_NAME then FIRST_NAME. If a non-fatal error
occurs, this returns undef. Check error_message for a verbose error
message.

Group Functions
is_valid_group ($)
    $: Group name to check
    Returns an 'evaluate-true' or 'evaluate-false'

This function checks to see if a given group name exists in the database. If
a non-fatal error occurs, this returns undef. Check error_message for a
verbose error message.

create_group ($;$)
    $: Group name to create
    $: Person ID of the owner
    Returns an 'evaluate-true' or 'evaluate-false'

This function creates a new group. If a non-fatal error occurs, this returns
undef. Check error_message for a verbose error message.

remove_group ($)
    $: Group name to remove
    Returns an 'evaluate-true' or 'evaluate-false'

This function removes a group. If a non-fatal error occurs, this returns
undef. Check error_message for a verbose error message.

list_groups (;$)
    $: ID of Owner
    returns a reference to an array containing the Group Names owned by the Owner

If no owner is supplied, all groups are returned. If a non-fatal error
occurs, this returns undef. Check error_message for a verbose error
message.

list_users_groups ($)
    $: ID of Person
    Returns sorted array containing the names of the groups the Person is a member of

If a non-fatal error occurs, this returns undef. Check error_message for a
verbose error message.

group_owner ($;$)
    $: Group to find owner for
    $: Person ID of the new owner of the group
    Returns Person ID of the group owner, if successful, else I<undef>

```

If the new owner parameter is present, the function attempts to set the new owner before doing the lookup. Either way, the current owner is returned. If a non-fatal error occurs, this returns undef. Check error\_message for a verbose error message.

```
is_group_member ($$)
  $: ID of person to check
  $: Name of Group to check
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function checks to see if the ID is in the given Group. If a non-fatal error occurs, this returns undef. Check error\_message for a verbose error message.

```
add_user_to_group ($$)
  $: ID of the person receiving privilege
  $: Name of Group
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function puts ID into the Group noted. If an error occurs, the function returns undef. In this case, check error\_message for a description of the error.

```
remove_user_from_group ($$)
  $: ID of the person losing privilege
  $: Name of Group
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function removes ID from the Group noted. If an error occurs, the function returns undef. In this case, check error\_message for a description of the error.

```
_group_membership_control ($$$) [Internal]
  $: ID of the person whose access is changing
  $: Name of Group
  $: 'evaluate-true' to grant or 'evaluate-false' to revoke
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function handles the actual changes to the database for add\_user\_to\_group and remove\_user\_from\_group. If an error occurs, the function returns undef. In this case, check error\_message for a description of the error.

```
list_group_membership (;$$)
  $: Name of Group
  $: SQL Sort String
  Returns a reference to an array of hashes containing all IDs, LAST_NAMES and FIRST_NAMES in the Privilege Group
```

If the Privilege Group is not provided, all system users are displayed. If no SQL Sort String is supplied, the results are sorted by ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

#### ACL Functions

```
is_valid_acl_access ($)
  $: Access privilege to check
  Returns an 'evaluate-true' or 'evaluate-false'
```

If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

```
grant_user_access ($$$)
  $: ID of person
  $: ID of document
  $: Access privilege to grant
  Returns an 'evaluate-true' or 'evaluate-false'
```

If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

```
grant_group_access ($$$)
  $: Name of Group
  $: ID of document
  $: Access privilege to grant
  Returns an 'evaluate-true' or 'evaluate-false'
```

If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

```
revoke_user_access ($$$)
  $: ID of person
  $: ID of document
  $: Access privilege to revoke
  Returns an 'evaluate-true' or 'evaluate-false'
```

If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

```
revoke_group_access ($$$)
  $: Name of Group
  $: ID of document
  $: Access privilege to revoke
  Returns an 'evaluate-true' or 'evaluate-false'
```

If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

```
has_access ($$$)
  $: ID of person to check
  $: ID of document to check
  $: Access privilege to check
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function determines if the Person ID has the given access to the Document ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

```
has_user_access ($$$)
  $: ID of person to check
  $: ID of document to check
  $: Access privilege to check
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function determines if the Person ID has the given access to the Document ID, based only on the user's Person ID. This function is used by has\_access to determine the overall access privilege of the Person ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

```
has_group_access ($$$)
  $: ID of person to check
  $: ID of document to check
  $: Access privilege to check
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function determines if the Person ID has the given access to the Document ID, based only on group membership. All groups that the Person ID is a member of are checked. This function is used by has\_access to determine the overall access privilege of the Person ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

list\_access (\$\$)  
 \$: ID of person to check  
 \$: ID of document to check  
 Returns an array containing the access privilege names

This function determines what access privileges the Person ID has to the given Document ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

list\_user\_access (\$\$)  
 \$: ID of person to check  
 \$: ID of document to check  
 Returns an array containing the access privilege names

This function determines what access privileges the Person ID has to the given Document ID, based only on the user's Person ID. This function is used by list\_access to determine the overall access privilege of the Person ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

list\_group\_access (\$\$)  
 \$: ID of person to check  
 \$: ID of document to check  
 Returns an array containing the access privilege names

This function determines what access privileges the Person ID has to the given Document ID, based only on group membership. All groups that the Person ID is a member of are checked. This function is used by list\_access to determine the overall access privilege of the Person ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

has\_read\_access (\$\$)  
 \$: ID of person to check  
 \$: ID of document to check  
 Returns an 'evaluate-true' or 'evaluate-false'

This function determines if the Person ID has Document Read Access to the Document ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

has\_delete\_access (\$\$)  
 \$: ID of person to check  
 \$: ID of document to check  
 Returns an 'evaluate-true' or 'evaluate-false'

This function determines if the Person ID has Document Delete Access to the Document ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

has\_annotate\_access (\$\$)  
 \$: ID of person to check  
 \$: ID of document to check  
 Returns an 'evaluate-true' or 'evaluate-false'

This function determines if the Person ID has Document Annotate Access to the Document ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

has\_read\_annotation\_access (\$\$)  
 \$: ID of person to check  
 \$: ID of document to check  
 Returns an 'evaluate-true' or 'evaluate-false'

This function determines if the Person ID has Annotation Read Access to the Document ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.



```
has_delete_annotation_access ($$)
  $: ID of person to check
  $: ID of document to check
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function determines if the Person ID has Annotation Delete Access to the Document ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

```
has_moderate_access ($$)
  $: ID of person to check
  $: ID of document to check
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function determines if the Person ID has Document Moderator Access to the Document ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

```
who_has_user_access ($;$)
  $: ID of document to check
  $: Access level to check
  Returns a reference to a hash of hashes
```

This returns a reference to a hash of hashes. The first-level key is the Person ID and the second-level is access level. The value is only a placeholder and can be ignored. If the second-level key exists, then that Person ID has that access. If no access level is specified, all Person IDs with any access are returned. This 1 show ADMIN users unless they have access independent of their administrator status.

```
who_has_group_access ($;$)
  $: ID of document to check
  $: Access level to check
  Returns a reference to a hash of hashes
```

This returns a reference to a hash of hashes. The first-level key is the Group Name and the second-level is access level. The value is only a placeholder and can be ignored. If the second-level key exists, then that the Group has that access. If no access level is specified, all Groups with any access are returned.

```
list_accessible_docs ($;$)
  $: ID of person to check
  $: SQL Sort String
  Returns a reference to an array of hashes containing some Document columns
```

If no SQL Sort String is supplied, the results are sorted by ID. The columns returned are \*id\*, \*title\*, \*owner\*, \*author\* and \*timestamp\*. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

```
_build_accessible_doc_list ($$;$) [Internal]
  $: ID of the person to build the list for
  $: SQL statement to use to get list
  $: 'evaluate-true' or 'evaluate-false' of whether to run in 'admin mode'
  Returns a reference to either an array or hash containing all Document columns requested
```

This is an internal worker function for list\_accessible\_docs. If it is in 'admin mode', a reference to an array in the same format that is used by list\_accessible\_docs is returned. Otherwise, a hash is built containing all the information that will need to be parsed by list\_accessible\_docs. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

```

_check_acl_override ($$) [Internal]
  $: ID of the person to check
  $: ID of the document to check
  Returns 'evaluate-true' or 'evaluate-false'

```

This function determines whether to override the user's default access because the user is either the owner of the document or is a system administrator. It is used by a number of other functions in this library. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

```

write_htaccess
  $: ID of document
  Returns 'evaluate-true' or 'evaluate-false'

```

This function writes the \*.htaccess\* file for the given Document ID. If a non-fatal error occurs, the function returns undef. Check error\_message for a verbose error message.

#### Error Control Functions

```

error_message ( )
  Returns the last generated error message

```

```

clear_error ( )
  This function clears last generated error message.

```

#### OAS::Auth - SQL Schema

The following SQL was optimized for MySQL 4.1.14

```

CREATE TABLE groups (
  name          CHAR (32) ASCII      NOT NULL,
  owner         CHAR (10) ASCII DEFAULT NULL,
  PRIMARY KEY  (name),
  FOREIGN KEY  (owner)
    REFERENCES person(id)
    ON UPDATE  CASCADE
    ON DELETE  CASCADE
) TYPE=INNODB;

CREATE TABLE group_membership (
  person_id    CHAR (10) ASCII NOT NULL,
  group_name   CHAR (32) ASCII NOT NULL,
  PRIMARY KEY  (person_id, group_name),
  FOREIGN KEY  (person_id)
    REFERENCES person(id)
    ON UPDATE  CASCADE
    ON DELETE  CASCADE,
  FOREIGN KEY  (group_name)
    REFERENCES groups(name)
    ON UPDATE  CASCADE
    ON DELETE  CASCADE
) TYPE=INNODB;

CREATE TABLE acl_user (
  person_id    CHAR (10) ASCII      NOT NULL,
  document_id  CHAR (16) ASCII      NOT NULL,
  acl          ENUM ('R','D','A','Ra','Da','M') NOT NULL,
  FOREIGN KEY  (person_id)
    REFERENCES person(id)
    ON UPDATE  CASCADE
    ON DELETE  CASCADE,
  FOREIGN KEY  (document_id)
    REFERENCES document(id)
    ON UPDATE  CASCADE
    ON DELETE  CASCADE
) TYPE=INNODB;

```

```

CREATE TABLE acl_group (
  group_name CHAR (32) ASCII NOT NULL,
  document_id CHAR (16) ASCII NOT NULL,
  acl ENUM ('R','D','A','Ra','Da','M') NOT NULL,
  FOREIGN KEY (group_name)
    REFERENCES groups(name)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  FOREIGN KEY (document_id)
    REFERENCES document(id)
    ON UPDATE CASCADE
    ON DELETE CASCADE
) TYPE=INNODB;

```

SEE ALSO

AUTHOR

Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE

Copyright (c) 2005, 2006 by Trev Harmon

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.5 or, at your option, any later version of Perl 5 you may have available.

### A.3. OAS::CGI.pm

NAME

OAS::CGI - Perl extension providing CGI/HTML services for OAS

SYNOPSIS

```
use OAS::CGI;
```

DESCRIPTION

This module provides different functions for implementing CGI features for OAS scripts and programs.

Non-Standard Required Modules

```
CGI::Carp
```

Exported

```
html_header
html_footer
```

Export OK

```
html_header
html_footer
close_window_form
```

Export Tags

```
all => [ html_header html_footer close_window_form ]
```

Constants

Event Handlers

HTML Functions

```
html_header (;$$$$)
  $: Title for new page
  $: URL for the CSS file to use for the new page
  $: Additional items for the <head> section
  $: Additional attributes for the <body> tag
  Returns HTML for a standard HTML header section
```

```
html_footer ()
```

```
Returns HTML for a standard HTML footer section
```

```
close_window_form (;$)
    $: Optional text for button label
    Returns HTML form containing a single button that will close the
    web browser containing it.
```

SEE ALSO

AUTHOR

Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE

Copyright (c) 2006 by Trev Harmon

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.5 or, at your option, any later version of Perl 5 you may have available.

## A.4. OAS::Converter.pm

NAME

OAS::Converter - Perl extension database interface for an OAS converter object

SYNOPSIS

```
use OAS::Converter;
```

DESCRIPTION

This object module provides a front-end interface for the OAS converter object, as well as additional helper functions. The module handles all the data verification and database interfacing for the object.

Non-Standard Required Modules

```
OAS::DBI
OAS::Utils
```

Multi-Threading

Basic support for multi-threaded applications is included in the module. However, this has not been fully tested, and should be considered EXPERIMENTAL at best.

Exported

None by default.

Export OK

```
convert
conversion_fail
list_converters
is_valid_extension
is_valid_command
is_valid_description
is_valid_resolution
untaint_extension
untaint_command
untaint_description
untaint_resolution
DEFAULT_RESOLUTION
CONVERSION_ERROR_NONE
CONVERSION_ERROR_INTERNAL
CONVERSION_ERROR_FILENAME
CONVERSION_ERROR_PATH
CONVERSION_ERROR_ID
CONVERSION_ERROR_DATABASE
CONVERSION_ERROR_EXTERNAL
CONVERSION_ERROR_FILESYSTEM
CONVERSION_ERROR_NETWORK
CONVERSION_ERROR_UNKNOWN
```

```

Export Tags
  check      => [ is_valid_extension      is_valid_command
                 is_valid_description   is_valid_resolution ]
  untaint    => [ untaint_extension      untaint_command
                 untaint_description   untaint_resolution ]
  resolution => [ is_valid_resolution    untaint_resolution
                 DEFAULT_RESOLUTION    ]
  errors     => [ conversion_fail
                 CONVERSION_ERROR_NONE  CONVERSION_ERROR_INTERNAL
                 CONVERSION_ERROR_FILENAME CONVERSION_ERROR_PATH
                 CONVERSION_ERROR_ID     CONVERSION_ERROR_DATABASE
                 CONVERSION_ERROR_EXTERNAL CONVERSION_ERROR_FILESYSTEM
                 CONVERSION_ERROR_NETWORK CONVERSION_ERROR_UNKNOWN ]

Constants
Resolution
  DEFAULT_RESOLUTION

Conversion Error Codes
CONVERSION_ERROR_NONE
CONVERSION_ERROR_INTERNAL
CONVERSION_ERROR_FILENAME
CONVERSION_ERROR_PATH
CONVERSION_ERROR_ID
CONVERSION_ERROR_DATABASE
CONVERSION_ERROR_EXTERNAL
CONVERSION_ERROR_FILESYSTEM
CONVERSION_ERROR_NETWORK
CONVERSION_ERROR_UNKNOWN

Event Handlers
DESTROY
  Disconnects the object from the database when the object is either undef'd,
  over-written or drops out of scope. The parent object's DESTROY method is
  also called.

Functions
convert ($$$;$)
  $: Filename (including path) of file to be converted
  $: Path in which to place the converted file (must be terminated by a '/' character)
  $: Document ID of the document being converted
  $: Quality (dpi) of the resulting graphic
  Returns the Document ID if successful, else I<undef>

  This function executes the conversion on Filename based on Filename's file
  extension. The resulting file will be placed in Path with the new file
  extension. The converted files are also added to the database as the pages
  for the supplied Document ID.

conversion_fail ($$)
  $: Message to display
  $: Error code of failure

  This function handles the production of an error during the conversion
  process.

list_converters ( )
  Returns a hash reference containing the list of all file converters

  The first-level key for the reference is the 'in_ext'. The second level keys
  are the columns from the database. If an error occurs, undef is returned.

is_valid_extension ($)
  $: Extension to be verified
  Returns an 'evaluate-true' or 'evaluate-false'

  This function only verifies the format. It makes no claim that the value is
  'valid' either in the system or reality.

```

```

is_valid_command ($)
    $: Command to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is
'valid' either in the system or reality. It doesn't really do any sanity
check; so be careful.

is_valid_description ($)
    $: Description to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is
'valid' either in the system or reality. It doesn't really do any sanity
check; so be careful.

is_valid_resolution ($)
    $: Resolution to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is
'valid' either in the system or reality. It doesn't really do any sanity
check; so be careful.

untaint_extension ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

This function calls is_valid_extension to verify the format of the value.

untaint_command ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

This function calls is_valid_command to verify the format of the value.

untaint_description ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

This function calls is_valid_description to verify the format of the value.

untaint_resolution ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

This function calls is_valid_resolution to verify the format of the value.

OAS::Converter Object - Data Control Methods
new ( )
    Returns a blessed reference to a new OAS::Converter object

Creates a new, empty OAS::Converter object.

Load ($)
    $: Input Extension to load from the database
    Returns an 'evaluate-true' if successful, else undef

If a non-fatal error occurs, error_message can be checked for a description
of the error.

Save (;$)
    $: Forces save if evaluates true
    Returns an 'evaluate-true' if successful, else undef

Commits changes to the database. If a non-fatal error occurs, error_message
can be checked for a description of the error.

```

```

Delete ( )
Attempts to delete the object from the database. The deletion may fail if
database foreign-key restraints do not permit it. If a non-fatal error
occurs, error_message can be checked for a description of the error. This
function DOES NOT deallocate the object, but instead resets it to an empty
state, just as if it had been invoked through new.

clear_error ( )
Clears the last error string stored in the object.

_reset_defaults ( ) [Internal]
Resets all members to default value (undef)

OAS::Converter Object - Data Access Methods
in (;$)
$: New Input Extension
Returns current Input Extension if no error occurred, else undef

Called with no arguments, the function returns the current Input Extension.
If an argument is supplied, it attempts to set the Input Extension to this
new value, if it has a valid format. If the parameter is an empty string '',
the Input Extension is set to undef, which is then returned.

out (;$)
$: New Output Extension
Returns current Output Extension if no error occurred, else undef

Called with no arguments, the function returns the current Output
Extension. If an argument is supplied, it attempts to set the Output
Extension to this new value, if it has a valid format. If the parameter is
an empty string '', the Output Extension is set to undef, which is then
returned.

command (;$)
$: New Conversion Command-line
Returns current Conversion Command-line if no error occurred, else undef

Called with no arguments, the function returns the current Conversion
Command-line. If an argument is supplied, it attempts to set the Conversion
Command-line to this new value, if it has a valid format. If the parameter
is an empty string '', the Conversion Command-line is set to undef, which is
then returned.

The Command may contain no spaces, switches or other parameters. The input
filename and output filename will be passed as the first and second
argument, respectively. If you need to have switches or other parameters on
the command-line, write a wrapper script to be placed here.

description (;$)
$: New Description of File Format
Returns current Description if no error occurred, else undef

Called with no arguments, the function returns the current Description. If
an argument is supplied, it attempts to set the Description to this new
value, if it has a valid format. If the parameter is an empty string '', the
Description is set to undef, which is then returned.

error_message ( )
Returns the current error message string

OAS::Converter Object - SQL Schema
The following SQL was optimized for MySQL 4.1.14

CREATE TABLE converter (
in_ext          CHAR ( 8) NOT NULL,
out_ext         CHAR ( 8) NOT NULL,
command        VARCHAR (256) NOT NULL,
description    VARCHAR (256) NOT NULL,

```

```
PRIMARY KEY (in_ext)
) TYPE=INNODB;
```

SEE ALSO

AUTHOR

Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE

Copyright (c) 2006 by Trev Harmon

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.5 or, at your option, any later version of Perl 5 you may have available.

## A.5. OAS::DBI.pm

NAME

OAS::DBI - Perl extension database interface for OAS

SYNOPSIS

```
use OAS::DBI;
```

DESCRIPTION

This module provides a front-end database interface for OAS. It is used to marshal and consolidate the database connections that are made by OAS. The module keeps a hash of all the current, unique database connections and a count representing active connections. When the count reaches 0, the module closes the database connection.

Non-Standard Required Modules

Exported

```
SAVE_NOT_REQUIRED
FORCE_SAVE
```

The functions `connect` and `disconnect` are not exported so as to not conflict with internal Perl functions of the same name. Consequently, these must be called as `OAS::DBI::connect` and `OAS::DBI::disconnect`. Also, `last_insert_id` must be called as `OAS::DBI::last_insert_id` as well.

Multi-Threading

The majority of this module should be thread-safe. However, this has not been fully tested, and should be considered EXPERIMENTAL at best. The function `last_insert_id` is NOT thread-safe, nor can it be made so under the current DBI implementation.

Constants

DBI\_DEFAULT\_DB\_NAME [Internal]

The database name to use if one is not supplied to connect or disconnect

DBI\_DEFAULT\_DB\_HOST [Internal]

The database hostname to use if one is not supplied to connect or disconnect

DBI\_DEFAULT\_DB\_PORT [Internal]

The database port to use if one is not supplied to connect or disconnect

DBI\_DEFAULT\_DB\_USER [Internal]

The database username to use if one is not supplied to connect or disconnect

DBI\_DEFAULT\_DB\_PASSWORD [Internal]

The database password to use if one is not supplied to connect or disconnect

CONNECTION\_NAME\_DELIMITER [Internal]

The delimiter used for building a connection name used by `_parts_to_connection_name` and `_connection_name_to_parts`



#### SAVE\_NOT\_REQUIRED

This is a constant return value used by many OAS objects that indicates that the save method did not execute because it was not necessary (i.e. no data had changed since the load). This value evaluates \*true\*.

#### FORCE\_SAVE

This constant is used by many OAS object save methods. If the save method receives this parameter, it forces a save to occur, overriding the conditions that would normally return a "SAVE\_NOT\_REQUIRED".

#### Event Handlers

##### END

This gracefully disconnects any remaining database connections when a script exits.

#### Functions

##### connect (;\$\$\$\$)

\$: Username  
\$: Password  
\$: Database name  
\$: Hostname  
\$: Port  
Returns database handle

All information that is not provided by the parameters is filled in by the "Constants" in the module. If this connection already exists, the function increases the connection count and returns the associated DBI-compliant database handle. Otherwise, the function attempts to connect to the database using the supplied parameters. If the connection is successful, it stores the new connection with a count of 1 and returns the database handle. If any non-fatal error occurs, undef is returned. In this case, error\_message can be checked for a verbose error message;

##### disconnect (;\$\$\$\$)

\$: Username  
\$: Password  
\$: Database name  
\$: Hostname  
\$: Port  
Returns an 'evaluate-true' or 'evaluate-false'

All information that is not provided by the parameters is filled in by the "Constants" in the module. The function decrements the count for the connection denoted by the parameters. The connection is closed when this count reaches zero, the actual connection to the database is closed and an 'evaluate-true' is returned. If any non-fatal error occurs, undef is returned. In this case, error\_message can be checked for a verbose error message;

##### last\_insert\_id (\$)

\$: Database handle  
Returns the ID from the last insert into an auto\_increment column

The ID is from the last insert into an auto\_increment column on the database handle, \*regardless of the table\*. Care should be taken to insure that the ID received is from the desired table, and not from a previous transaction. Careful checking of return values should handle this properly. If any non-fatal error occurs, undef is returned. In this case, error\_message can be checked for a verbose error message;

##### \_parts\_to\_connection\_name (\$\$\$\$) [Internal]

\$: Username  
\$: Password  
\$: Database name  
\$: Hostname  
\$: Port  
Returns connection name

This function takes the individual connection descriptors and builds a single connection name that is used internally to identify unique database connections.

```
_connection_name_to_parts ($) [Internal]
  $: Connection name
  Returns an array consisting of (username, password, database name, hostname, port)
```

This function breaks a single connection name into its individual parts.

```
error_message ( )
  Returns the current error message string
```

```
clear_error_message ( )
  Clears the current error message string
```

SEE ALSO

AUTHOR

Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE

Copyright (c) 2005 by Trev Harmon

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.5 or, at your option, any later version of Perl 5 you may have available.

## A.6. OAS::Document.pm

NAME

OAS::Document - Perl extension database interface for an OAS document object

SYNOPSIS

```
use OAS::Document;
```

DESCRIPTION

This object module provides a front-end interface for the OAS document object, as well as additional helper functions. The module handles all the data verification and database interfacing for the object.

Non-Standard Required Modules

```
OAS::System
OAS::DBI
OAS::Person
OAS::Utils
```

Multi-Threading

Basic support for multi-threaded applications is included in the module. However, this has not been fully tested, and should be considered EXPERIMENTAL at best.

Exported

None by default.

Export OK

```
create_new_document_id
document_id_to_path
document_id_to_annotation_path
document_id_to_image_path
document_owner
is_valid_document_id
is_valid_author
is_valid_owner
is_valid_title
is_valid_timestamp
```

```

    is_valid_original
    untaint_document_id
    untaint_author
    untaint_owner
    untaint_title
    untaint_timestamp
    untaint_original

Export Tags
    document => [ create_new_document_id    document_id_to_path
document_id_to_annotation_path
                  document_id_to_image_path document_owner
]

    check    => [ is_valid_document_id    is_valid_author    is_valid_owner
                  is_valid_title         is_valid_timestamp is_valid_original
]

    untaint  => [ untaint_document_id     untaint_author     untaint_owner
                  untaint_title          untaint_timestamp untaint_original
]

Event Handlers
DESTROY
    Disconnects the object from the database when the object is either undef'd,
    over-written or drops out of scope. The parent object's DESTROY method is
    also called.

Functions
create_new_document_id ( )
    Returns a unique Document ID

document_id_to_path ($)
    $: ID to be converted to a path
    Returns the real path that should store the files associated with the ID

document_id_to_annotation_path ($)
    $: ID to be converted to a path
    Returns the real path that should store the annotation files associated with the ID

document_id_to_image_path ($)
    $: ID to be converted to a path
    Returns the real path that should store the images files associated with the ID

document_owner ($)
    $: ID to look up owner for
    Returns the ID of the Person who owns the document

is_valid_id ($)
    $: ID to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

    This function only verifies the format. It makes no claim that the value is
    'valid' either in the system or reality.

is_valid_document_id ($)
    $: ID to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

    This function only verifies the format. It makes no claim that the value is
    'valid' either in the system or reality.

is_valid_author ($)
    $: Author (Person ID) to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

    This function only verifies the format. It makes no claim that the value is
    'valid' either in the system or reality.

```

is\_valid\_owner (\$)  
 \$: Owner (Person ID) to be verified  
 Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

is\_valid\_title (\$)  
 \$: Title to be verified  
 Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

is\_valid\_timestamp (\$)  
 \$: Time to be verified  
 Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

is\_valid\_original (\$)  
 \$: Original to be verified  
 Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

untaint\_id (\$)  
 \$: Value to be untainted  
 Returns untainted value if value is of the correct format, else undef

This function calls is\_valid\_document\_id to verify the format of the value.

untaint\_document\_id (\$)  
 \$: Value to be untainted  
 Returns untainted value if value is of the correct format, else undef

This function calls is\_valid\_document\_id to verify the format of the value.

untaint\_author (\$)  
 \$: Value to be untainted  
 Returns untainted value if value is of the correct format, else undef

This function calls is\_valid\_author to verify the format of the value.

untaint\_owner (\$)  
 \$: Value to be untainted  
 Returns untainted value if value is of the correct format, else undef

This function calls is\_valid\_owner to verify the format of the value.

untaint\_title (\$)  
 \$: Value to be untainted  
 Returns untainted value if value is of the correct format, else undef

This function calls is\_valid\_title to verify the format of the value.

untaint\_timestamp (\$)  
 \$: Value to be untainted  
 Returns untainted value if value is of the correct format, else undef

This function calls is\_valid\_timestamp to verify the format of the value.

```

untaint_original ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_original to verify the format of the value.

OAS::Document Object - Data Control Methods
new ( )
    Returns a blessed reference to a new OAS::Document object

    Creates a new, empty OAS::Document object.

Load ($)
    $: ID of the document to load from the database
    Returns an 'evaluate-true' if successful, else undef

    If a non-fatal error occurs, error_message can be checked for a description
    of the error.

Save (;$)
    $: Forces save if evaluates true
    Returns an 'evaluate-true' if successful, else undef

    Commits changes to the database. If a non-fatal error occurs, error_message
    can be checked for a description of the error.

Delete ( )
    Attempts to delete the object from the database. The deletion may fail if
    database foreign-key restraints do not permit it. If a non-fatal error
    occurs, error_message can be checked for a description of the error. This
    function DOES NOT deallocate the object, but instead resets it to an empty
    state, just as if it had been invoked through new.

clear_error ( )
    Clears the last error string stored in the object.

_reset_defaults ( ) [Internal]
    Resets all members to default value (undef)

OAS::Document Object - Page Control Methods
total_pages ( )
    Returns the number of pages in the document

is_page ($)
    $: Page number to check
    Returns 'evaluate-true' or 'evaluate-false'

    This function determines whether or not the given Page number is valid for
    the document.

add_page ( )
    Returns the page number of the new page

    This function adds a new page to the end of the document.

insert_page ($)
    $: Page number to insert before
    Returns the page number of the new page

    This function inserts a new page into a document at the position specified.

delete_page ($)
    $: Page number to delete from document
    Returns 'evaluate-true' if successful, else 'evaluate-false'

page_number ($)
    $: Page number to check
    Returns the page number

```

This is mainly a debugging function, as the parameter and return value should be the same.

```
page_image ($;$)
  $: Page number
  $: New Image
  Returns the Image for the given page number
```

If an error occurs, the function returns undef. Check error\_message for a description of the error.

```
page_height ($;$)
  $: Page number
  $: New Height
  Returns the Height for the given page number
```

If an error occurs, the function returns undef. Check error\_message for a description of the error.

```
page_width ($;$)
  $: Page number
  $: New Width
  Returns the Width for the given page number
```

If an error occurs, the function returns undef. Check error\_message for a description of the error.

#### OAS::Document Object - Data Access Methods

```
id (;$)
  $: New Document ID
  Returns current Document ID if no error occurred, else undef
```

Called with no arguments, the function returns the current Document ID. If an argument is supplied, it attempts to set the Document ID to this new value, if it has a valid format. If the parameter is an empty string '', the Document ID is set to undef, which is then returned.

```
author (;$)
  $: New Author (Person ID)
  Returns current Author if no error occurred, else undef
```

Called with no arguments, the function returns the current Author. If an argument is supplied, it attempts to set the Author to this new value, if it has a valid format. If the parameter is an empty string '', the Author is set to undef, which is then returned.

```
owner (;$)
  $: New Owner (Person ID)
  Returns current Owner if no error occurred, else undef
```

Called with no arguments, the function returns the current Owner. If an argument is supplied, it attempts to set the Owner to this new value, if it has a valid format. If the parameter is an empty string '', the Owner is set to undef, which is then returned.

```
title (;$)
  $: New Title
  Returns current Title if no error occurred, else undef
```

Called with no arguments, the function returns the current Title. If an argument is supplied, it attempts to set the Title to this new value, if it has a valid format. If the parameter is an empty string '', the Title is set to undef, which is then returned.

```
timestamp (;$)
  $: New Unix Timestamp (seconds since Epoch)
  Returns current Time if no error occurred, else undef
```

Called with no arguments, the function returns the current Timestamp. If an argument is supplied, it attempts to set the Timestamp to this new value, if it has a valid format. If the parameter is an empty string '', the Timestamp is set to undef, which is then returned.

```
original (;$)
    $: New Original Filename
    Returns current Original Filename if no error occurred, else undef
```

Called with no arguments, the function returns the current Original Filename. If an argument is supplied, it attempts to set the Original Filename to this new value, if it has a valid format. If the parameter is an empty string '', the Original Filename is set to undef, which is then returned.

```
error_message ( )
    Returns the current error message string
```

OAS::Document Object - SQL Schema  
The following SQL was optimized for MySQL 4.1.14

```
CREATE TABLE document (
  id          CHAR      ( 16) ASCII NOT NULL,
  author      CHAR      ( 10) ASCII NOT NULL,
  owner       CHAR      ( 10) ASCII NOT NULL,
  title       VARCHAR   (128) ASCII NOT NULL,
  timestamp   BIGINT    ( 10)      NOT NULL,
  original    VARCHAR   (256) ASCII NOT NULL,
  PRIMARY KEY (id),
  INDEX       (author),
  FOREIGN KEY (author)
    REFERENCES person(id)
    ON UPDATE CASCADE
    ON DELETE RESTRICT,
  INDEX       (owner),
  FOREIGN KEY (owner)
    REFERENCES person(id)
    ON UPDATE CASCADE
    ON DELETE RESTRICT
) TYPE=INNODB;
```

SEE ALSO

AUTHOR  
Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE

Copyright (c) 2005,2006 by Trev Harmon

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.5 or, at your option, any later version of Perl 5 you may have available.

## A.7. OAS::Image.pm

NAME

OAS::Image - Perl object for creating images for the OAS System

SYNOPSIS

```
use OAS::Image;
```

DESCRIPTION

This object module provides a front-end interface for creating images for the OAS system.

Non-Standard Required Modules  
 OAS::Carp  
 Image::Magick

Multi-Threading  
 No support for multi-threaded applications is included in the module.  
 However, due to its nature, it should be thread-safe. With this in mind ,it  
 should be considered EXPERIMENTAL at best.

Exported  
 None by default

Export OK  
 None

OAS::Image Object - Object Methods

create ( )  
 Returns a blessed reference to a new OAS::Image object

Creates a new, empty OAS::Image object.

\_init ( ) [Internal]  
 Returns a blessed reference to the OAS::Image object

This function sets all of the default values in the object.

bezier (@)  
 @: A series of (x,y) coordinates  
 Returns 'evaluate-true' if successful, else 'evaluate-false'

The function draws a bezier curve on the image. The data points are sent in  
 as a flat array. For example, the array would be in the format of:

[x1, y1, x2, y2, x3, y3, x4, y4, ... ]

line (\$\$\$\$)  
 \$: X1  
 \$: Y1  
 \$: X2  
 \$: Y2  
 Returns 'evaluate-true' if successful, else 'evaluate-false'

The function draws a line on the image from (X1,Y1) to (X2,Y2).

rectangle (\$\$\$\$)  
 \$: X1  
 \$: Y1  
 \$: X2  
 \$: Y2  
 Returns 'evaluate-true' if successful, else 'evaluate-false'

The function draws a rectangle on the image with opposing corners (X1,Y1)  
 and (X2,Y2).

point (\$\$)  
 \$: X  
 \$: Y  
 Returns 'evaluate-true' if successful, else 'evaluate-false'

The function draws a point on the image at (X,Y).

size (\$\$)  
 \$: Width of image  
 \$: Height of image  
 Returns the array (Width, Height) if successful, else undef

This function resets the entire image to the new size. All previous drawing  
 data will be lost.



```

_size ($$) [Internal]
    $: Width of image
    $: Height of image
    Returns the array (Width, Height) if successful, else undef

This function resets the entire image to the new size. All previous drawing
data will be lost.

format ($)
    $: New Format for image
    Returns the Format

filename ($)
    $: New Filename
    Returns the Filename

write ( )
    Returns 'evaluate-true' if successful, else 'evaluate-false'

This function writes the image out to the file specified by filename.

pen_color ($)
    $: New Pen Color
    Returns the Pen Color

pen_weight ($)
    $: New Pen Weight
    Returns the Pen Weight

SEE ALSO
AUTHOR
    Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE
    Copyright (c) 2006 by Trev Harmon

This library is free software; you can redistribute it and/or modify it
under the same terms as Perl itself, either Perl version 5.8.5 or, at your
option, any later version of Perl 5 you may have available.

```

## A.8. OAS::Menu.pm

```

NAME
    OAS::Menu - Perl extension providing the OAS menu system

SYNOPSIS
    use OAS::Menu;

DESCRIPTION
    This module provides the menu systems for OAS.

Non-Standard Required Modules
    OAS
    OAS::Annotation
    OAS::Auth
    OAS::Document
    OAS::Person

Exported
    MENUBAR_HEIGHT
    INFOBAR_HEIGHT
    SIDEBAR_WIDTH
    BORDER_SIZE
    menubar

```

```

Export OK
  MENUBAR_HEIGHT
  INFOBAR_HEIGHT
  SIDEBAR_WIDTH
  BORDER_SIZE
  menubar
  infobar
  sidebar
  annotation_text

Export Tags
  all => [ MAIN_MENU_BAR TOOLBAR_HEIGHT
           SIDEBAR_WIDTH BORDER_SIZE
           menubar       infobar
           sidebar       annotation_text ]

Constants
ENABLED [Internal]
  This constant is used to denote that a menu item is enabled.

DISABLED [Internal]
  This constant is used to denote that a menu item is not available to the
  user at this time.

MENUBAR_HEIGHT
  This is the height (pixels) of the main menu bar.

INFOBAR_HEIGHT
  This is the height (pixels) of the information bar.

SIDEBAR_WIDTH
  This is the width (pixels) of the side toolbar.

BORDER_SIZE
  This is the size (pixels) of the border around the page graphic.

Functions
menubar (;$$)
  $: OAS::Document Object reference of document to display
  $: Page number of document to display
  Returns the HTML code for the main menu bar

  The returned HTML code contains the necessary JavaScript includes to make
  the menu system work.

_menu_javascript ($$) [Internal]
  $: OAS::Document Object reference
  $: Page number of document to display
  Returns the HTML code to load the menu JavaScript

  This function is used by menubar.

_menu_headings ($$) [Internal]
  $: Current User's ID (OAS::Person Object ID)
  $: Width to make the menu bar
  Returns the HTML code for the table headings

  This function is used by menubar.

```

```
_file_menu ($$$$$%) [Internal]
$: Current User's ID (OAS::Person Object ID)
$: OAS::Document Object reference
$: Page number of document to display
$: Boolean of whether the current user is the owner of the document
$: Boolean of whether the current user is a system administrator
$: Hash reference created by OAS::Service::list_services
%: ACL hash with keys as the current user's permissions
Returns the HTML code for the File Menu
```

This function is used by menubar.

```
_edit_menu ($$$$$%) [Internal]
$: Current User ID's (OAS::Person Object ID)
$: OAS::Document Object reference
$: Page number of document to display
$: Boolean of whether the current user is the owner of the document
$: Boolean of whether the current user is a system administrator
$: Hash reference created by OAS::Service::list_services
%: ACL hash with keys as the current user's permissions
Returns the HTML code for the Edit Menu
```

This function is used by menubar.

```
_view_menu ($$$$$%) [Internal]
$: Current User ID's (OAS::Person Object ID)
$: OAS::Document Object reference
$: Page number of document to display
$: Boolean of whether the current user is the owner of the document
$: Boolean of whether the current user is a system administrator
$: Hash reference created by OAS::Service::list_services
%: ACL hash with keys as the current user's permissions
Returns the HTML code for the View Menu
```

This function is used by menubar.

```
_help_menu () [Internal]
Returns the HTML code for the Help Menu
```

This function is used by menubar.

```
_add_menu_item ($;$) [Internal]
$: Status of the menu
$: String containing text for menu item
$: JavaScript code for the onClick event
Returns the HTML code for a new menu item
```

This function is used by menubar. The Status should be ENABLED or DISABLED. The function will override the provided onClick JavaScript if the menu is DISABLED.

```
info_bar ($$)
$: OAS::Document Object reference of the current document
$: Page number of document to display
Returns the HTML code for the information bar
```

This information bar should only be used in conjunction with menubar.

```
_nav_button ($$$$) [Internal]
$: Document ID for current document
$: Current Page number
$: New Page number that button will jump to
$: Text to display on the button
$: Name of the form that controls this action
Returns the HTML code for a navigational button
```

This internal function is used by toolbar. It creates navigational form buttons that enable/disable based on the current and new page numbers.

```
sidebar ($$;$@)
    $: OAS::Document Object reference of the current document
    $: Page number of document to display
    $: Boolean for whether to show annotation tools
    @: List of Annotation IDs to display
    Returns the HTML code for the annotation sidebar
```

This information bar should only be used in conjunction with menubar and infobar

```
_annotation_tools ($$) [Internal]
    $: OAS::Document Object reference of the current document
    $: Page number of document to display
    Returns the HTML code for the annotation tools
```

```
annotation_text ($$;$$$%)
    $: OAS::Document Object reference of the current document
    $: Page number of document to display
    $: Content to place in the the text window
    $: CSS2 style to apply to annotation text control window <div>
    $: Code to execute when the 'Cancel' button is pushed
    %: Hash containing the settings for each of the pull-down menus
    Returns the HTML code for the annotation text control window
```

This toolbar should only be used in conjunction with menubar. Valid keys for the Settings Hash include: \*font\*, \*pitch\*, \*color\* and \*background\*.

SEE ALSO

AUTHOR

Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE

Copyright (c) 2006 by Trev Harmon

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.5 or, at your option, any later version of Perl 5 you may have available.

## A.9. OAS::Page.pm

NAME

OAS::Page - Perl extension database interface for an OAS page object

SYNOPSIS

```
use OAS::Page;
```

DESCRIPTION

This object module provides a front-end interface for the OAS page object, as well as additional helper functions. The module handles all the data verification and database interfacing for the object.

Non-Standard Required Modules

```
OAS::DBI
OAS::Document
```

Multi-Threading

Basic support for multi-threaded applications is included in the module. However, this has not been fully tested, and should be considered EXPERIMENTAL at best.

Exported

None by default.

#### Export OK

```
is_valid_page_id
is_valid_page_num
is_valid_image
is_valid_height
is_valid_width
untaint_page_id
untaint_page_num
untaint_image
untaint_height
untaint_width
```

#### Export Tags

```
check => [ is_valid_page_id is_valid_page_num is_valid_image
           is_valid_height is_valid_width ]
untaint => [ untaint_page_id untaint_page_num untaint_image
            untaint_height untaint_width ]
```

#### Event Handlers

##### DESTROY

Disconnects the object from the database when the object is either undef'd, over-written or drops out of scope. The parent object's DESTROY method is also called.

#### Functions

##### is\_valid\_id (\$)

\$: ID to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

##### is\_valid\_page\_id (\$)

\$: ID to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

##### is\_valid\_page\_num (\$)

\$: Page to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

##### is\_valid\_image (\$)

\$: Image filename to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

##### is\_valid\_height (\$)

\$: Image height to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

##### is\_valid\_width (\$)

\$: Image width to be verified  
Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

```

untaint_id ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_id to verify the format of the value.

untaint_page_id ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_id to verify the format of the value.

untaint_page_num ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_page to verify the format of the value.

untaint_image ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_image to verify the format of the value.

untaint_height ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_height to verify the format of the value.

untaint_width ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_width to verify the format of the value.

OAS::Page Object - Data Control Methods
new ( )
    Returns a blessed reference to a new OAS::Page object

    Creates a new, empty OAS::Page object.

Load ($$)
    $: Document ID of the page to load from the database
    $: Page number to load from the database
    Returns an 'evaluate-true' if successful, else undef

    If a non-fatal error occurs, error_message can be checked for a description
    of the error.

Save (;$)
    $: Forces save if evaluates true
    Returns an 'evaluate-true' if successful, else undef

    Commits changes to the database. If a non-fatal error occurs, error_message
    can be checked for a description of the error.

Delete ( )
    Attempts to delete the object from the database. The deletion may fail if
    database foreign-key restraints do not permit it. If a non-fatal error
    occurs, error_message can be checked for a description of the error. This
    function DOES NOT deallocate the object, but instead resets it to an empty
    state, just as if it had been invoked through new.

clear_error ( )
    Clears the last error string stored in the object.

```

```

_reset_defaults ( ) [Internal]
  Resets all members to default value (undef)

OAS::Page Object - Data Access Methods
id (;$)
  $: New ID for Page
  Returns current Page ID if no error occurred, else undef

  Called with no arguments, the function returns the current Page ID. If an
  argument is supplied, it attempts to set the Page ID to this new value, if
  it has a valid format. If the parameter is an empty string '', the Page ID
  is set to undef, which is then returned.

page (;$)
  $: New Page number
  Returns current Page number if no error occurred, else undef

  Called with no arguments, the function returns the current Page number. If
  an argument is supplied, it attempts to set the Page number to this new
  value, if it has a valid format. If the parameter is an empty string '', the
  Page number is set to undef, which is then returned.

image (;$)
  $: New Image filename
  Returns current Image filename if no error occurred, else undef

  Called with no arguments, the function returns the current Image filename.
  If an argument is supplied, it attempts to set the Image filename to this
  new value, if it has a valid format. If the parameter is an empty string '',
  the Image filename is set to undef, which is then returned.

height (;$)
  $: New Image height
  Returns current Image height if no error occurred, else undef

  Called with no arguments, the function returns the current Image height. If
  an argument is supplied, it attempts to set the Image height to this new
  value, if it has a valid format. If the parameter is an empty string '', the
  Image height is set to undef, which is then returned.

width (;$)
  $: New Image width
  Returns current Image width if no error occurred, else undef

  Called with no arguments, the function returns the current Image width. If
  an argument is supplied, it attempts to set the Image width to this new
  value, if it has a valid format. If the parameter is an empty string '', the
  Image width is set to undef, which is then returned.

error_message ( )
  Returns the current error message string

OAS::Page Object - SQL Schema
The following SQL was optimized for MySQL 4.1.14

CREATE TABLE page (
  id      CHAR      ( 16) ASCII NOT NULL,
  page    INT        NOT NULL,
  image   VARCHAR   (256) ASCII NOT NULL,
  height  INT        NOT NULL,
  width   INT        NOT NULL,
  PRIMARY KEY (id, page),
  FOREIGN KEY (id)
    REFERENCES document(id)
    ON UPDATE CASCADE
    ON DELETE CASCADE
) TYPE=INNODB;

```

SEE ALSO

AUTHOR

Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE

Copyright (c) 2006 by Trev Harmon

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.5 or, at your option, any later version of Perl 5 you may have available.

## A.10. OAS::Person.pm

NAME

OAS::Person - Perl extension database interface for an OAS person object

SYNOPSIS

```
use OAS::Person;
```

DESCRIPTION

This object module provides a front-end interface for the OAS person object, as well as additional helper functions. The module handles all the data verification and database interfacing for the object.

Non-Standard Required Modules

```
OAS::DBI
OAS::Utils
```

Multi-Threading

Basic support for multi-threaded applications is included in the module. However, this has not been fully tested, and should be considered EXPERIMENTAL at best.

Exported

None by default.

Export OK

```
is_valid_person_id
is_valid_id
is_valid_first_name
is_valid_last_name
is_valid_address
is_valid_city
is_valid_state
is_valid_zip
is_valid_phone
is_valid_fax
is_valid_mobile
is_valid_email
is_valid_password
untaint_person_id
untaint_id
untaint_first_name
untaint_last_name
untaint_address
untaint_city
untaint_state
untaint_zip
untaint_phone
untaint_fax
untaint_mobile
untaint_email
untaint_password
```



```

Export Tags
  check => [ is_valid_person_id is_valid_first_name is_valid_last_name
             is_valid_address  is_valid_city      is_valid_state
             is_valid_zip      is_valid_phone    is_valid_fax
             is_valid_mobile    is_valid_email    is_valid_password ]

  untaint => [ untaint_person_id  untaint_first_name  untaint_last_name
               untaint_address    untaint_city      untaint_state
               untaint_zip        untaint_phone    untaint_fax
               untaint_mobile      untaint_email    untaint_password ]

Constants
  DEFAULT_AREA_CODE
  Default area code for the system to use if one is not supplied for the
  person's phone, fax or mobile numbers.

Event Handlers
  DESTROY
  Disconnects the object from the database when the object is either undef'd,
  over-written or drops out of scope. The parent object's DESTROY method is
  also called.

Validation Functions
  is_valid_person_id ($)
    $: ID to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

    This function only verifies the format. It makes no claim that the value is
    'valid' either in the system or reality.

  is_valid_id ($)
    $: ID to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

    This function only verifies the format. It makes no claim that the value is
    'valid' either in the system or reality.

  is_valid_first_name ($)
    $: Name to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

    This function only verifies the format. It makes no claim that the value is
    'valid' either in the system or reality.

  is_valid_last_name ($)
    $: Name to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

    This function only verifies the format. It makes no claim that the value is
    'valid' either in the system or reality.

  is_valid_address ($)
    $: Address to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

    This function only verifies the format. It makes no claim that the value is
    'valid' either in the system or reality.

  is_valid_city ($)
    $: City to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

    This function only verifies the format. It makes no claim that the value is
    'valid' either in the system or reality.

  is_valid_state ($)
    $: U.S. 2-character state abbreviation to be verified
    Returns an 'evaluate-true' or 'evaluate-false'

```

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

```
is_valid_zip ($)
  $: U.S. 5/9-digit zip code to be verified
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

```
is_valid_phone ($)
  $: U.S. 10-digit (digits only!) phone number to be verified
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

```
is_valid_fax ($)
  $: U.S. 10-digit (digits only!) phone number to be verified
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

```
is_valid_mobile ($)
  $: U.S. 10-digit (digits only!) phone number to be verified
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

```
is_valid_email ($)
  $: Email (user@domain) to be verified
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

```
is_valid_password ($)
  $: Encrypted password to be verified
  Returns an 'evaluate-true' or 'evaluate-false'
```

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

#### Untaint Functions

```
untaint_person_id ($)
  $: Value to be untainted
  Returns untainted value if value is of the correct format, else undef
```

This function calls is\_valid\_person\_id to verify the format of the value.

```
untaint_id ($)
  $: Value to be untainted
  Returns untainted value if value is of the correct format, else undef
```

This function calls is\_valid\_person\_id to verify the format of the value.

```
untaint_first_name ($)
  $: Value to be untainted
  Returns untainted value if value is of the correct format, else undef
```

This function calls is\_valid\_first\_name to verify the format of the value.

```

untaint_last_name ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_last_name to verify the format of the value.

untaint_address ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_address to verify the format of the value.

untaint_city ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_city to verify the format of the value.

untaint_state ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_state to verify the format of the value.

untaint_zip ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_zip to verify the format of the value.

untaint_phone ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_phone to verify the format of the value.

untaint_fax ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_fax to verify the format of the value.

untaint_mobile ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_mobile to verify the format of the value.

untaint_email ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_email to verify the format of the value.

untaint_password ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_password to verify the format of the value.

OAS::Person Object - Data Control Methods
new ( )
    Returns a blessed reference to a new OAS::Person object

    Creates a new, empty OAS::Person object.

```

Load (\$)  
 \$: ID of the person to load from the database  
 Returns an 'evaluate-true' if successful, else undef

If a non-fatal error occurs, error\_message can be checked for a description of the error.

Save (;\$)  
 \$: Forces save if evaluates true  
 Returns an 'evaluate-true' if successful, else undef

Commits changes to the database. If a non-fatal error occurs, error\_message can be checked for a description of the error.

Delete ( )  
 Attempts to delete the object from the database. The deletion may fail if database foreign-key restraints do not permit it. If a non-fatal error occurs, error\_message can be checked for a description of the error. This function DOES NOT deallocate the object, but instead resets it to an empty state, just as if it had been invoked through new.

clear\_error ( )  
 Clears the last error string stored in the object.

\_reset\_defaults ( ) [Internal]  
 Resets all members to default value (undef)

OAS::Person Object - Data Access Methods

id (;\$)  
 \$: New Person ID  
 Returns current Person ID if no error occurred, else undef

Called with no arguments, the function returns the current Person ID. If an argument is supplied, it attempts to set the Person ID to this new value, if it has a valid format. If the parameter is an empty string '', the Person ID is set to undef, which is then returned.

first\_name (;\$)  
 \$: New First Name  
 Returns current First Name if no error occurred, else undef

Called with no arguments, the function returns the current First Name. If an argument is supplied, it attempts to set the First Name to this new value, if it has a valid format. If the parameter is an empty string '', the First Name is set to undef, which is then returned.

last\_name (;\$)  
 \$: New Last Name  
 Returns current Last Name if no error occurred, else undef

Called with no arguments, the function returns the current Last Name. If an argument is supplied, it attempts to set the Last Name to this new value, if it has a valid format. If the parameter is an empty string '', the Last Name is set to undef, which is then returned.

address (;\$)  
 \$: New Address  
 Returns current Address if no error occurred, else undef

Called with no arguments, the function returns the current Address. If an argument is supplied, it attempts to set the Address to this new value, if it has a valid format. If the parameter is an empty string '', the Address is set to undef, which is then returned.

city (;\$)  
 \$: New City  
 Returns current City if no error occurred, else undef

Called with no arguments, the function returns the current City. If an argument is supplied, it attempts to set the City to this new value, if it has a valid format. If the parameter is an empty string '', the City is set to undef, which is then returned.

state (;\$)

\$: New U.S. State 2-character abbreviation  
Returns current U.S. State if no error occurred, else undef

Called with no arguments, the function returns the current U.S. State. If an argument is supplied, it attempts to set the U.S. State to this new value, if it has a valid format. If the parameter is an empty string '', the U.S. State is set to undef, which is then returned.

zip (;\$)

\$: New U.S. 5/9-digit Zip Code  
Returns current U.S. Zip Code if no error occurred, else undef

Called with no arguments, the function returns the current U.S. Zip Code. If an argument is supplied, it attempts to set the U.S. Zip Code to this new value, if it has a valid format. If the parameter is an empty string '', the U.S. Zip Code is set to undef, which is then returned.

phone (;\$)

\$: New 7/10-digit Phone Number (digits only!)  
Returns current Phone Number if no error occurred, else undef

Called with no arguments, the function returns the current Phone Number. If an argument is supplied, it attempts to set the Phone Number to this new value, if it has a valid format. If the parameter is an empty string '', the Phone Number is set to undef, which is then returned. A 7-digit number will be padded to a 10-digit number using the DEFAULT\_AREA\_CODE.

fax (;\$)

\$: New 7/10-digit Fax Number (digits only!)  
Returns current Fax Number if no error occurred, else undef

Called with no arguments, the function returns the current Fax Number. If an argument is supplied, it attempts to set the Fax Number to this new value, if it has a valid format. If the parameter is an empty string '', the Fax Number is set to undef, which is then returned. A 7-digit number will be padded to a 10-digit number using the DEFAULT\_AREA\_CODE.

mobile (;\$)

\$: New 7/10-digit Mobile/Cell Number (digits only!)  
Returns current Mobile/Cell Number if no error occurred, else undef

Called with no arguments, the function returns the current Mobile/Cell Number. If an argument is supplied, it attempts to set the Mobile/Cell Number to this new value, if it has a valid format. If the parameter is an empty string '', the Mobile/Cell Number is set to undef, which is then returned. A 7-digit number will be padded to a 10-digit number using the DEFAULT\_AREA\_CODE.

email (;\$)

\$: New Email Address (user@domain)  
Returns current Email Address if no error occurred, else undef

Called with no arguments, the function returns the current Email Address. If an argument is supplied, it attempts to set the Email Address to this new value, if it has a valid format. If the parameter is an empty string '', the Email Address is set to undef, which is then returned.

password (;\$)

\$: New, unencrypted password  
Returns encrypted password if no error occurred, else undef

Called with no arguments, the function returns the current encrypted password. If an argument is supplied, it attempts to set the Password to the encrypted version new value, if it has a valid format. If the parameter is an empty string '', the Password is set to undef, which is then returned.

```
error_message ( )  
    Returns the current error message string
```

OAS::Person Object - SQL Schema  
The following SQL was optimized for MySQL 4.1.14

```
CREATE TABLE person (  
    id          CHAR      (10) ASCII      NOT NULL,  
    first_name  VARCHAR   (24)           NOT NULL,  
    last_name   VARCHAR   (48)           NOT NULL,  
    address     VARCHAR   (48)           DEFAULT NULL,  
    city        VARCHAR   (24)           DEFAULT NULL,  
    state       CHAR      ( 2) ASCII     DEFAULT NULL,  
    zip         BIGINT    ( 9) UNSIGNED  DEFAULT NULL,  
    phone       BIGINT    (10) UNSIGNED  DEFAULT NULL,  
    fax         BIGINT    (10) UNSIGNED  DEFAULT NULL,  
    mobile      BIGINT    (10) UNSIGNED  DEFAULT NULL,  
    email       VARCHAR   (64)           DEFAULT NULL,  
    password    CHAR      (20)           NOT NULL,  
    PRIMARY KEY (id),  
    INDEX       (last_name)  
) TYPE=INNODB;
```

SEE ALSO

AUTHOR

Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE

Copyright (c) 2005 by Trev Harmon

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.5 or, at your option, any later version of Perl 5 you may have available.

## A.11. OAS::Service.pm

NAME

OAS::Service - Perl extension database interface for an OAS web service object

SYNOPSIS

```
use OAS::Service;
```

DESCRIPTION

This object module provides a front-end interface for the OAS web service object, as well as additional helper functions. The module handles all the data verification and database interfacing for the object.

Non-Standard Required Modules

```
OAS::DBI  
OAS::Utils
```

Multi-Threading

Basic support for multi-threaded applications is included in the module. However, this has not been fully tested, and should be considered EXPERIMENTAL at best.

Exported

None by default.

```

Export OK
  list_services
  service_url
  is_valid_name
  is_valid_url
  is_valid_description
  untaint_name
  untaint_url
  untaint_description

Export Tags
  check => [ is_valid_name      is_valid_url
             is_valid_description ]
  untaint => [ untaint_name    untaint_url
              untaint_description ]

Event Handlers
DESTROY
  Disconnects the object from the database when the object is either undef'd,
  over-written or drops out of scope. The parent object's DESTROY method is
  also called.

Functions
list_services ( )
  Returns a hash reference containing the list of all web services

  The first-level key for the reference is the 'name'. The second level keys
  are the columns from the database. If an error occurs, undef is returned.

service_url ($)
  $: Name of the web service to look up
  Returns the url of the web service

  If an error occurs or if the web service name is not valid, undef is
  returned.

is_valid_name ($)
  $: Name to be verified
  Returns an 'evaluate-true' or 'evaluate-false'

  This function only verifies the format. It makes no claim that the value is
  'valid' either in the system or reality.

is_valid_url ($)
  $: URL to be verified
  Returns an 'evaluate-true' or 'evaluate-false'

  This function only verifies the format. It makes no claim that the value is
  'valid' either in the system or reality.

is_valid_description ($)
  $: Description to be verified
  Returns an 'evaluate-true' or 'evaluate-false'

  This function only verifies the format. It makes no claim that the value is
  'valid' either in the system or reality.

untaint_name ($)
  $: Value to be untainted
  Returns untainted value if value is of the correct format, else undef

  This function calls is_valid_name to verify the format of the value.

```

```

untaint_url ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_url to verify the format of the value.

untaint_description ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_description to verify the format of the value.

OAS::Service Object - Data Control Methods
new ( )
    Returns a blessed reference to a new OAS::Service object

    Creates a new, empty OAS::Service object.

Load ($)
    $: Input Name to load from the database
    Returns an 'evaluate-true' if successful, else undef

    If a non-fatal error occurs, error_message can be checked for a description
    of the error.

Save (;$)
    $: Forces save if evaluates true
    Returns an 'evaluate-true' if successful, else undef

    Commits changes to the database. If a non-fatal error occurs, error_message
    can be checked for a description of the error.

Delete ( )
    Attempts to delete the object from the database. The deletion may fail if
    database foreign-key restraints do not permit it. If a non-fatal error
    occurs, error_message can be checked for a description of the error. This
    function DOES NOT deallocate the object, but instead resets it to an empty
    state, just as if it had been invoked through new.

clear_error ( )
    Clears the last error string stored in the object.

_reset_defaults ( ) [Internal]
    Resets all members to default value (undef)

OAS::Service Object - Data Access Methods
name (;$)
    $: New Name
    Returns current Name if no error occurred, else undef

    Called with no arguments, the function returns the current Name. If an
    argument is supplied, it attempts to set the Name to this new value, if it
    has a valid format. If the parameter is an empty string '', the Name is set
    to undef, which is then returned.

url (;$)
    $: New URL
    Returns current URL if no error occurred, else undef

    Called with no arguments, the function returns the current URL. If an
    argument is supplied, it attempts to set the URL to this new value, if it
    has a valid format. If the parameter is an empty string '', the URL is set
    to undef, which is then returned.

description (;$)
    $: New Description
    Returns current Description if no error occurred, else undef

```



Called with no arguments, the function returns the current Description. If an argument is supplied, it attempts to set the Description to this new value, if it has a valid format. If the parameter is an empty string '', the Description is set to undef, which is then returned.

```
error_message ( )  
    Returns the current error message string
```

OAS::Service Object - SQL Schema  
The following SQL was optimized for MySQL 4.1.14

```
CREATE TABLE service (  
    name          CHAR ( 36)      NOT NULL,  
    url           VARCHAR (128)   NOT NULL,  
    description   VARCHAR (256)  DEFAULT NULL,  
    PRIMARY KEY (name)  
) TYPE=INNODB;
```

SEE ALSO

AUTHOR

Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE

Copyright (c) 2006 by Trev Harmon

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.5 or, at your option, any later version of Perl 5 you may have available.

## A.12. OAS::System.pm

NAME

OAS::System - Perl extension database interface for an OAS system

SYNOPSIS

```
use OAS::System;
```

DESCRIPTION

This object module provides a front-end interface for the OAS system, as well as additional helper functions. The module handles all the data verification and database interfacing for the object.

Non-Standard Required Modules

```
OAS::DBI  
OAS::Utils
```

Multi-Threading

Basic support for multi-threaded applications is included in the module. However, this has not been fully tested, and should be considered EXPERIMENTAL at best.

Exported

None by default

Export OK

```
set_path  
path_to_url  
is_valid_configuration  
is_valid_home_dir  
is_valid_www_dir  
is_valid_www_url  
is_valid_perl_dir  
untaint_configuration  
untaint_home_dir  
untaint_www_dir
```

```

untaint_www_url
untaint_perl_dir

Export Tags
  check => [ is_valid_configuration is_valid_home_dir
             is_valid_www_dir       is_valid_www_url
             is_valid_perl_dir       ]

  untaint => [ untaint_configuration untaint_home_dir
              untaint_www_dir       untaint_www_url
              untaint_perl_dir       ]

Event Handlers
DESTROY
  Disconnects the object from the database when the object is either undef'd,
  over-written or drops out of scope. The parent object's DESTROY method is
  also called.

Functions
set_path ( )
  Returns new value of $ENV{PATH}

  This function sets the path environment variable for use with taint-checks.
  It is called automatically when this module is compiled.

path_to_url ($)
  $: Path to convert to a URL
  Returns the converted URL if successful, else I<undef>

  This function first attempts to map to the WWW Directory for the current
  configuration. If this fails, it tries the CGI-BIN Directory. If the mapping
  fails, undef is returned.

is_valid_configuration ($)
  $: Configuration name to be verified
  Returns an 'evaluate-true' or 'evaluate-false'

  This function only verifies the format. It makes no claim that the value is
  'valid' either in the system or reality.

is_valid_home_dir ($)
  $: Path to base install of OAS to be verified
  Returns an 'evaluate-true' or 'evaluate-false'

  This function only verifies the format. It makes no claim that the value is
  'valid' either in the system or reality.

is_valid_www_dir ($)
  $: Path to WWW root to be verified
  Returns an 'evaluate-true' or 'evaluate-false'

  This function only verifies the format. It makes no claim that the value is
  'valid' either in the system or reality.

is_valid_www_url ($)
  $: WWW base url to be verified
  Returns an 'evaluate-true' or 'evaluate-false'

  This function only verifies the format. It makes no claim that the value is
  'valid' either in the system or reality.

is_valid_perl_dir ($)
  $: Perl Module directory to be verified
  Returns an 'evaluate-true' or 'evaluate-false'

  This function only verifies the format. It makes no claim that the value is
  'valid' either in the system or reality.

```

```

untaint_configuration ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_configuration to verify the format of the
    value.

untaint_home_dir ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_home_url to verify the format of the value.

untaint_www_dir ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_www_url to verify the format of the value.

untaint_www_url ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_www_url to verify the format of the value.

untaint_perl_dir ($)
    $: Value to be untainted
    Returns untainted value if value is of the correct format, else undef

    This function calls is_valid_perl_dir to verify the format of the value.

```

OAS Object - Data Control Methods

```

new ( )
    Returns a blessed reference to a new OAS object

    Creates a new, empty OAS object.

Load ($)
    $: Configuration name of the settings to load from the database
    Returns an 'evaluate-true' if successful, else undef

    If a non-fatal error occurs, error_message can be checked for a description
    of the error.

Save (;$)
    $: Forces save if evaluates true
    Returns an 'evaluate-true' if successful, else undef

    Commits changes to the database. If a non-fatal error occurs, error_message
    can be checked for a description of the error.

Delete ( )
    Attempts to delete the object from the database. If a non-fatal error
    occurs, error_message can be checked for a description of the error. This
    function DOES NOT deallocate the object, but instead resets it to an empty
    state, just as if it had been invoked through new.

clear_error ( )
    Clears the last error string stored in the object.

_reset_defaults ( ) [Internal]
    Resets all members to default value (undef)

```

OAS Object - Data Access Methods

```

configuration (;$)
    $: New Configuration name
    Returns current Configuration Name if no error occurred, else undef

```

Called with no arguments, the function returns the current Configuration name. If an argument is supplied, it attempts to set the Configuration name to this new value, if it has a valid format. If the parameter is an empty string '', the Configuration name is set to undef, which is then returned.

home\_dir (;\$)

\_\$: New path to OAS install directory

Returns current path to OAS install directory if no error occurred, else undef

Called with no arguments, the function returns the current path to OAS install directory. If an argument is supplied, it attempts to set the path to OAS install directory to this new value, if it has a valid format. A trailing '/' is added, if necessary. If the parameter is an empty string '', the path to OAS install directory is set to undef, which is then returned.

www\_dir (;\$)

\_\$: New path to WWW root

Returns current path to WWW root if no error occurred, else undef

Called with no arguments, the function returns the current path to WWW root. If an argument is supplied, it attempts to set the path to WWW root to this new value, if it has a valid format. A trailing '/' is added, if necessary. If the parameter is an empty string '', the path to WWW root is set to undef, which is then returned.

www\_url (;\$)

\_\$: New WWW base URL

Returns current WWW base URL if no error occurred, else undef

Called with no arguments, the function returns the current WWW base URL. If an argument is supplied, it attempts to set the WWW base URL to this new value, if it has a valid format. A trailing '/' is added, if necessary. If the parameter is an empty string '', the WWW base URL is set to undef, which is then returned.

perl\_dir (;\$)

\_\$: New Perl Module directory

Returns current Perl Module directory if no error occurred, else undef

Called with no arguments, the function returns the current Perl Module directory. If an argument is supplied, it attempts to set the Perl Module directory to this new value, if it has a valid format. A trailing '/' is added, if necessary. If the parameter is an empty string '', the Perl Module directory is set to undef, which is then returned.

error\_message ( )

Returns the current error message string

OAS Object - SQL Schema

The following SQL was optimized for MySQL 4.1.14

```
CREATE TABLE system (
  configuration CHAR ( 36) ASCII NOT NULL,
  home_dir VARCHAR (256) ASCII NOT NULL,
  www_dir VARCHAR (256) ASCII NOT NULL,
  www_url VARCHAR (256) ASCII NOT NULL,
  perl_dir VARCHAR (256) ASCII NOT NULL,
  PRIMARY KEY (configuration)
) TYPE=INNODB;
```

SEE ALSO

AUTHOR

Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE

Copyright (c) 2005,2006 by Trev Harmon

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.5 or, at your option, any later version of Perl 5 you may have available.

## A.13. OAS::Utils.pm

### NAME

OAS::Utils - Perl extension providing OAS utility functions

### SYNOPSIS

```
use OAS::Utils;
```

### DESCRIPTION

This module provides different utility modules for OAS.

#### Exported

None by default.

#### Export OK

```
prepad0
postpad0
untaint
file_name
file_extension
file_path
is_valid_path
is_valid_filename
```

#### Export Tags

```
padding => [ prepad0 postpad0 ]
dangerous => [ untaint ]
files => [ file_name file_extension file_path is_valid_path is_valid_filename ]
```

### Padding Functions

These functions are for padding scalars to fit a certain width

#### prepad0 (\$\$)

```
$: Scalar item to pad
$: Minimum length of padded scalar
Returns a scalar front-padded with 0's
```

If input scalar's length is greater than the provided length, the input scalar is returned as-is (i.e. no padding is done).

#### postpad0 (\$\$)

```
$: Scalar item to pad
$: Minimum length of padded scalar
Returns a scalar back-padded with 0's
```

If input scalar's length is greater than the provided length, the input scalar is returned as-is (i.e. no padding is done).

### Dangerous Functions

These functions are such that they can cause problems (generally security holes) if they are not used correctly. They should only be used with a full understanding of what they do and affect.

#### untaint (\$)

```
$: Scalar to be untainted
Returns an untainted copy of the input scalar
```

Untaint does not check the data, save to verify that it has data to work on. It then proceeds to merrily (and blindly) untaint the supplied data, which can be very dangerous if the data has not already been checked for validity prior to calling untaint.

## File Functions

`file_name ($)`

    \$: Path to get filename from  
    Returns the filename

If no filename is present, '' is returned. If an error occurs, undef is returned.

`file_extension ($)`

    \$: Filename to get extension from  
    Returns the file extension

If no extension is present, '' is returned. If an error occurs, undef is returned.

`file_path ($)`

    \$: Full path to get path from  
    Returns the path

If a path is passed in without any file, it must be terminated by a '/' character. If no path is present, '' is returned. If an error occurs, undef is returned.

`is_valid_path ($)`

    \$: Path to be verified  
    Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

`is_valid_filename ($)`

    \$: Filename to be verified  
    Returns an 'evaluate-true' or 'evaluate-false'

This function only verifies the format. It makes no claim that the value is 'valid' either in the system or reality.

SEE ALSO

AUTHOR

Trev Harmon, <trev@byu.edu>

COPYRIGHT & LICENSE

Copyright (c) 2005, 2006 by Trev Harmon

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.5 or, at your option, any later version of Perl 5 you may have available.



## Appendix B – API Source Code

This appendix includes the source code for all of the *API Perl 5* modules. Because Appendix A contains the documentation for these modules, the *POD* sections have been removed from the source code in this appendix.

### B.1. OAS::Annotation.pm

```
package OAS::Annotation;
require 5.6.0;
our $VERSION = 1.0.0;
our $DATE    = '07 Aug 2006';

use strict;
use warnings;

use Carp;
use Thread;
use OAS::System qw(path_to_url);
use OAS::DBI;
use OAS::Document qw(is_valid_document_id untaint_document_id);
use OAS::Page qw(is_valid_page_num untaint_page_num);
use OAS::Person qw(is_valid_person_id untaint_person_id);
use OAS::Utils qw(untaint);

use Exporter;
our @ISA = qw(Exporter);
our @EXPORT = qw();
our @EXPORT_OK = qw(
    annotation_list      annotation_owner      annotation_encode
    annotation_unencode
    is_valid_annotation_id is_valid_author      is_valid_owner
    is_valid_page         is_valid_type        is_valid_width
    is_valid_height       is_valid_x_pos       is_valid_y_pos
    is_valid_timestamp    is_valid_font        is_valid_pitch
    is_valid_color        is_valid_background is_valid_image
    is_valid_text
    untaint_annotation_id untaint_author      untaint_owner
    untaint_page          untaint_type        untaint_width
    untaint_height       untaint_x_pos       untaint_y_pos
    untaint_timestamp    untaint_font        untaint_pitch
    untaint_color        untaint_background  untaint_image
    untaint_text
);
our %EXPORT_TAGS = (
    check => [qw( is_valid_annotation_id is_valid_author      is_valid_owner
```



```

        is_valid_page          is_valid_type          is_valid_width
        is_valid_height       is_valid_x_pos         is_valid_y_pos
        is_valid_timestamp    is_valid_font         is_valid_pitch
        is_valid_color        is_valid_background  is_valid_image
        is_valid_text
    untaint => [qw( untaint_annotation_id  untaint_author    untaint_owner
                  untaint_page          untaint_type      untaint_width
                  untaint_height       untaint_x_pos     untaint_y_pos
                  untaint_timestamp    untaint_font     untaint_pitch
                  untaint_color        untaint_background untaint_image
                  untaint_text
    )];

```

```

=====
# EVENT HANDLERS
#
sub DESTROY
{
    # Disconnect from database on exit if needed
    my $self = shift;
    OAS::DBI::disconnect() if (defined $self->{DBH});
    $self->SUPER::DESTROY if $self->can("SUPER::DESTROY");
}
=====

```

```

-----
# FUNCTIONS
#
sub annotation_list ($$)
{
    my ($doc, $page) = @_;
    return unless defined($doc) and defined($page);
    my $result = undef;

    my $dbh = OAS::DBI::connect;
    my $sth = $dbh->prepare(qq(
        SELECT *
        FROM annotation
        WHERE document_id = '$doc'
        AND page = $page
    )) or return;
    $sth->execute or return;
    while (my $href = $sth->fetchrow_hashref) {
        foreach my $key (keys %{$href}) {
            my $skey = $href->{'timestamp'} . '-' . $href->{'id'};
            $result->{$skey}{$key} = $href->{$key};
        }
    }
    OAS::DBI::disconnect;
    return $result;
}
#
#

```

```

-----
#
sub annotation_owner ($)
{
    my $id = $_[0];
    return unless &is_valid_annotation_id($id);

    my $result = undef;
    my $dbh = OAS::DBI::connect or return;

    my $sth = $dbh->prepare("SELECT owner FROM annotation WHERE id = '$id'");
    $sth->execute;
}

```

```

my @row;
$result = $row[0] if (@row = $sth->fetchrow_array);

OAS::DBI::disconnect;
return $result;
}
#
#-----
#
sub annotation_encode ($)
{
    my $text = (defined $_[0]) ? $_[0] : '';
    $text =~ s/&/&amp;/g;
    $text =~ s/</&lt;/g;
    $text =~ s/>/&gt;/g;
    $text =~ s/\r?\n/<br \/>\n/g;
    $text =~ s/\t/ /g;
    return $text;
}
#
#-----
#
sub annotation_unencode ($)
{
    my $text = (defined $_[0]) ? $_[0] : '';
    $text =~ s/<br\s*\>//g;
    $text =~ s/<.+?>//gs;
    $text =~ s/&gt;/>/g;
    $text =~ s/&lt;/</g;
    $text =~ s/&amp;/&/g;
    return $text;
}
#
#-----
#
sub is_valid_id ($)
{
    return is_valid_document_id $_[0];
}
#
#-----
#
sub is_valid_annotation_id ($)
{
    return is_valid_id $_[0];
}
#
#-----
#
sub is_valid_author ($)
{
    return is_valid_person_id $_[0];
}
#
#-----
#
sub is_valid_owner ($)
{
    return is_valid_person_id $_[0];
}
#
#-----
#
sub is_valid_page ($)
{
    return is_valid_page_num $_[0];
}
#

```

```

#-----
#
sub is_valid_type ($)
{
    return defined($_[0]) && (('Image' eq $_[0]) || ('Text' eq $_[0]));
}
#
#-----
#
sub is_valid_width ($)
{
    local $_ = $_[0] || '';
    return /^ \d+ $/x;
}
#
#-----
#
sub is_valid_height ($)
{
    local $_ = $_[0] || '';
    return /^ \d+ $/x;
}
#
#-----
#
sub is_valid_timestamp ($)
{
    local $_ = $_[0] || '';
    return /^ \d{10} $/x;
}
#
#-----
#
sub is_valid_font ($)
{
    local $_ = $_[0] || '';
    return /^ [\w,\-]{1,20} $/x;
}
#
#-----
#
sub is_valid_pitch ($) {
    local $_ = $_[0] || '';
    return /^ \d{1,2} $/x;
}
#
#-----
#
sub is_valid_color ($)
{
    local $_ = $_[0] || '';
    return /^ \w{1,16} $/x;
}
#
#-----
#
sub is_valid_background ($) {
    local $_ = $_[0] || '';
    return /^ \w{1,16} $/x;
}
#
#-----
#
sub is_valid_image ($)
{
    local $_ = $_[0] || '';
    return /^ [\w\-\.\300-\366\370-\377\/]? $/x;
}

```

```

#
#-----
#
sub is_valid_text ($) {
    local $_ = $_[0] || '';
    return /^\w\-\.\300-\366\370-\377\/]+? $/x;
}
#
#-----
#
sub is_valid_x_pos ($)
{
    local $_ = defined($_[0]) ? $_[0] : '';
    return /^\d+ $/x;
}
#
#-----
#
sub is_valid_y_pos ($)
{
    local $_ = defined($_[0]) ? $_[0] : '';
    return /^\d+ $/x;
}
#
#-----
#
sub untaint_id ($)
{
    return untaint_document_id $_[0];
}
#
#-----
#
sub untaint_annotation_id ($)
{
    return untaint_id $_[0];
}
#
#-----
#
sub untaint_author ($)
{
    return untaint_person_id $_[0];
}
#
#-----
#
sub untaint_owner ($)
{
    return untaint_person_id $_[0];
}
#
#-----
#
sub untaint_page ($)
{
    return untaint_page_num $_[0];
}
#
#-----
#
sub untaint_type ($)
{
    return (is_valid_type $_[0]) ? untaint $_[0] : undef;
}
#
#-----

```

```

#
sub untaint_width ($)
{
    return (is_valid_width $_[0]) ? untaint $_[0] : undef;
}
#
-----
#
sub untaint_height ($)
{
    return (is_valid_height $_[0]) ? untaint $_[0] : undef;
}
#
-----
#
sub untaint_x_pos ($)
{
    return (is_valid_x_pos $_[0]) ? untaint $_[0] : undef;
}
#
-----
#
sub untaint_y_pos ($)
{
    return (is_valid_y_pos $_[0]) ? untaint $_[0] : undef;
}
#
-----
#
sub untaint_timestamp ($)
{
    return (is_valid_timestamp $_[0]) ? untaint $_[0] : undef;
}
#
-----
#
sub untaint_font ($)
{
    return (is_valid_font $_[0]) ? untaint $_[0] : undef;
}
#
-----
#
sub untaint_pitch ($)
{
    return (is_valid_pitch $_[0]) ? untaint $_[0] : undef;
}
#
-----
#
sub untaint_color ($)
{
    return (is_valid_color $_[0]) ? untaint $_[0] : undef;
}
#
-----
#
sub untaint_background ($)
{
    return (is_valid_background $_[0]) ? untaint $_[0] : undef;
}
#
-----
#
sub untaint_image ($)
{
    return (is_valid_image $_[0] ) ? untaint $_[0] : undef;
}

```

```

#
#-----
#
sub untaint_text ($)
{
    return (is_valid_text $_[0]) ? untaint $_[0] : undef;
}
#
#-----

#-----
# CONSTRUCTORS & INITIALIZERS
#
sub new #()#
{
    my $class = shift;
    $class = ref $class || $class;
    my $self = bless {}, $class;
    $self->_reset_defaults;
    $self->clear_error;
    $self->{ CHANGED } = 1;
    $self->{ DBH } = undef;
    return $self;
}
#
#-----
#
sub _reset_defaults ()
{
    my $self = shift;
    foreach (qw(
        id author owner document_id page
        type width height x_pos y_pos
        timestamp font pitch color
        background image text
    )) {
        $self->{$_} = undef;
    }
}
#
#-----

#-----
# DATA CONTROL METHODS
#
sub Load : locked method #($$)#
{
    my $self = shift;
    my $id = $_[0];
    my $page = $_[1];
    my $href = undef;
    my $key = undef;

    # Verify parameters
    if ( not defined $id ) { croak 'Missing Page ID for load'; }
    elsif ( not is_valid_id $id ) { croak 'Invalid Page ID for load'; }

    # Connect to database
    $self->{DBH} = OAS::DBI::connect unless $self->{DBH};

    # Query database for needed information
    unless ($href =
        $self->{DBH}->selectrow_hashref("SELECT * FROM annotation WHERE id='$id'"))
    {

```

```

        $self->{ERROR} = $self->{DBH}->errstr;
        return;
    }
    # Reset object values to default
    $self->_reset_defaults;
    $self->clear_error;
    $self->{CHANGED} = 0;

    # Store the results in the object
    foreach $key qw(
        id author owner document_id page
        type width height x_pos y_pos
        timestamp font pitch color
        background image text
    ) {
        $self->$key($href->{$key}) if defined $href->{$key};
    }

    return 1;
}
#
#-----
#
sub Save : locked #()#
{
    my $self = shift;
    my $key = undef;

    # Check for forced save
    $self->{CHANGED} = 1 if $_[0];

    # Verify needed columns
    if ( not $self->{id} ) {
        $self->{ERROR} = 'Missing ID for save' ; return;
    } elsif ( not $self->{author} ) {
        $self->{ERROR} = 'Missing AUTHOR for save' ; return;
    } elsif ( not $self->{owner} ) {
        $self->{ERROR} = 'Missing OWNER for save' ; return;
    } elsif ( not $self->{document_id} ) {
        $self->{ERROR} = 'Missing DOCUMENT_ID for save' ; return;
    } elsif ( not $self->{page} ) {
        $self->{ERROR} = 'Missing PAGE for save' ; return;
    } elsif ( not $self->{type} ) {
        $self->{ERROR} = 'Missing TYPE for save' ; return;
    } elsif ( not $self->{height} ) {
        $self->{ERROR} = 'Missing HEIGHT for save' ; return;
    } elsif ( not $self->{width} ) {
        $self->{ERROR} = 'Missing WIDTH for save' ; return;
    } elsif ( not $self->{timestamp} ) {
        $self->{ERROR} = 'Missing TIMESTAMP for save' ; return;
    } elsif ( not defined $self->{ x_pos } ) {
        $self->{ERROR} = 'Missing X_POS for save';
        return;
    } elsif ( not defined $self->{ y_pos } ) {
        $self->{ERROR} = 'Missing Y_POS for save';
        return; }
    return SAVE_NOT_REQUIRED unless $self->{CHANGED};

    # Connect to database
    $self->{DBH} = OAS::DBI::connect() unless $self->{DBH};

    # Gather & prepare data for query
    my $id = $self->{DBH}->quote( $self->{ id } );
    my $author = $self->{DBH}->quote( $self->{ author } );
    my $owner = $self->{DBH}->quote( $self->{ owner } );
    my $document_id = $self->{DBH}->quote( $self->{ document_id } );
    my $type = $self->{DBH}->quote( $self->{ type } );
    my $font = $self->{DBH}->quote( $self->{ font } );

```

```

my $color      = $self->{DBH}->quote( $self->{ color      } );
my $background = $self->{DBH}->quote( $self->{ background } );
my $image      = $self->{DBH}->quote( $self->{ image      } );
my $text       = $self->{DBH}->quote( $self->{ text       } );
my $page       = ( $self->{ page       } ) ? int $self->{ page       } : 'NULL';
my $height     = ( $self->{ height    } ) ? int $self->{ height    } : 'NULL';
my $width      = ( $self->{ width     } ) ? int $self->{ width     } : 'NULL';
my $x_pos      = ( $self->{ x_pos     } ) ? int $self->{ x_pos     } : 0;
my $y_pos      = ( $self->{ y_pos     } ) ? int $self->{ y_pos     } : 0;
my $timestamp  = ( $self->{ timestamp } ) ? int $self->{ timestamp } : 'NULL';
my $pitch      = ( $self->{ pitch     } ) ? int $self->{ pitch     } : 'NULL';

# Build SQL query
my $query;
($query = <<__QUERY_END__>> =~ s/(?:\^\s+)|\s+(?==)|\n/ /gm;
    id          = $id,
    author      = $author,
    owner       = $owner,
    document_id = $document_id,
    page        = $page,
    type        = $type,
    width       = $width,
    height      = $height,
    x_pos       = $x_pos,
    y_pos       = $y_pos,
    timestamp   = $timestamp,
    font        = $font,
    pitch       = $pitch,
    color       = $color,
    background  = $background,
    image       = $image,
    text        = $text
__QUERY_END__
)
$query = "INSERT INTO annotation SET $query ON DUPLICATE KEY UPDATE $query";

# Execute query
my $sth = $self->{DBH}->prepare($query);
unless ($sth->execute) {
    $self->{ERROR} = $sth->errstr;
    return;
}

# Final items
$self->{CHANGED} = 0;

return 1;
}
#
#-----
#
sub Delete : locked #()#
{
    my $self = shift;
    my $sth = undef;

    # Delete from database
    if (defined $self->{id}) {
        $self->{DBH} = OAS::DBI::connect() unless $self->{DBH};
        if (!$sth = $self->{DBH}->prepare(
            "DELETE FROM annotation WHERE id='$self->{id}'"
        )) {
            $self->{ERROR} = $self->{DBH}->errstr;
            return;
        }
        unless (defined $sth->execute) {
            $self->{ERROR} = $sth->errstr;
            return;
        }
    }
}

```



```

        # Delete from filesystem
        unlink $self->{ 'filename' } if defined $self->{ 'filename' };
        unlink $self->{ 'image' } if defined $self->{ 'image' };
    }

    # Clear object contents
    $self->_reset_defaults;
    $self->clear_error;
    $self->{CHANGED} = 0;

    return 1;
}
#
#-----
#
sub clear_error #()#
{
    my $self = shift;
    $self->{ERROR} = '';
}
#
#-----

#-----
# DISPLAY METHODS
#
sub html #()#
{
    my $self = shift;
    my $html = "<!-- Start Annotation #${self->{id}} -->\n";

    # Create HTML code for given mode
    if ('Image' eq $self->{type}) {
        # Image Annotation
        my $src = path_to_url $self->{image};
        $html .= qq|
            {id}}"
                width="${self->{width}}" height="${self->{height}}" border="0"
                onmousedown = "return false;" onmouseup = "return false;"
                onmousemove = "return false;" onmouseenter = "return false;"
                onmouseleave = "return false;" onmouseover = "return false;"
                onmouseout = "return false;" onclick = "return false;"
                ondblclick = "return false;"
            />
        |;
    } elsif ('Text' eq $self->{type}) {
        # Text Annotation
        my $text = '';
        if (open ATEXT, $self->{text}) {
            local $/ = undef;
            $text = <ATEXT>;
            close ATEXT;
            $text = annotation_encode $text;
        } else {
            $text = "<!-- ERROR: Could not open annotation's text file -->"
        }
    }
    my $style = 'font-weight:bold;'
        . "font-family:${self->{font}};"
        . "font-size:${self->{pitch}}px;"
        . "color:${self->{color}};"
        . "background:${self->{background}};"
        . "width:${self->{width}}px;"
        . "height:${self->{height}}px;"
        . 'margin:0px 0px 0px 0px;'
        . 'overflow:auto;';
}

```

```

    $html .= qq|
        <table border="1" width="100%" height="100%">
        <tbody>
        <tr>
            <td background="/images/_$self->{background}.gif" valign="top"
                align="left"><p style="$style">$text</p></td>
        </tr>
        </tbody>
        </table>
    |;
} else {
    $self->{ERROR} = 'Invalid Mode for HTML render!';
    $html         .= "<!-- ERROR: $self->{ERROR} -->\n";
};

$html .= "<!-- End Annotation #${self->{id}} -->\n";
$html =~ s/^\t+//gm;
$html =~ s/[\r\n]+/\n/g;

return $html;
}
#
#-----

#-----
# DATA ACCESS METHODS
#
sub id #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ id      } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_id $data) {
            $self->{ERROR} = 'Invalid ID!';
            return;
        }
        $self->{ id      } = untaint_id $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{id};
}
#
#-----
#
sub author #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ author } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_author $data) {
            $self->{ERROR} = 'Invalid AUTHOR!';
            return;
        }
    }
}

```

```

        $self->{ author } = untaint_author $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{author};
}
#
#-----
#
sub owner #(;)$#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ owner } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_owner $data) {
            $self->{ERROR} = 'Invalid OWNER';
            return;
        }
        $self->{ owner } = untaint_owner $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{owner};
}
#
#-----
#
sub document_id #(;)$#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ document_id } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_document_id $data) {
            $self->{ERROR} = 'Invalid IMAGE';
            return;
        }
        $self->{ document_id } = untaint_document_id $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{document_id};
}
#
#-----
#
sub page #(;)$#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ page } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_page $data) {
            $self->{ERROR} = 'Invalid PAGE';
            return;
        }
    }
}

```

```

        $self->{ page      } = untaint_page $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{page};
}
#
#-----
#
sub type #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ type      } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_type $data) {
            $self->{ERROR} = 'Invalid TYPE';
            return;
        }
        $self->{ type      } = untaint_type $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{type};
}
#
#-----
#
sub width #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ width      } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_width $data) {
            $self->{ERROR} = 'Invalid WIDTH';
            return;
        }
        $self->{ width      } = untaint_width $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{width};
}
#
#-----
#
sub height #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ height      } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_height $data) {
            $self->{ERROR} = 'Invalid HEIGHT';
            return;
        }
    }
}

```

```

        $self->{ height } = untaint_height $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{height};
}
#
#-----
#
sub x_pos #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ x_pos } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_x_pos $data) {
            $self->{ERROR} = 'Invalid X_POS';
            return;
        }
        $self->{ x_pos } = untaint_x_pos $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{x_pos};
}
#
#-----
#
sub y_pos #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ y_pos } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_y_pos $data) {
            $self->{ERROR} = 'Invalid Y_POS';
            return;
        }
        $self->{ y_pos } = untaint_y_pos $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{y_pos};
}
#
#-----
#
sub timestamp #($#)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ timestamp } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_timestamp $data) {
            $self->{ERROR} = 'Invalid TIMESTAMP';
            return;
        }
    }
}

```

```

        $self->{ timestamp } = untaint_timestamp $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{timestamp};
}
#
#-----
#
sub font #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ font } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_font $data) {
            $self->{ERROR} = 'Invalid FONT';
            return;
        }
        $self->{ font } = untaint_font $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{font};
}
#
#-----
#
sub pitch #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ pitch } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_pitch $data) {
            $self->{ERROR} = 'Invalid PITCH';
            return;
        }
        $self->{ pitch } = untaint_pitch $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{pitch};
}
#
#-----
#
sub color #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ color } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_color $data) {
            $self->{ERROR} = 'Invalid COLOR';
            return;
        }
    }
}

```

```

        $self->{ color } = untaint_color $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{color};
}
#
#-----
#
sub background #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ background } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_background $data) {
            $self->{ERROR} = 'Invalid BACKGROUND';
            return;
        }
        $self->{ background } = untaint_background $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{background};
}
#
#-----
#
sub image #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ image } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_image $data) {
            $self->{ERROR} = 'Invalid IMAGE';
            return;
        }
        $self->{ image } = untaint_image $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{image};
}
#
#-----
#
sub text #($)#
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ text } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_text $data) {
            $self->{ERROR} = 'Invalid TEXT';
            return;
        }
    }
}

```

```

        $self->{ text } = untaint_text $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{text};
}
#
#-----
#
#
sub error_message #()#
{
    my $self = shift;
    return $self->{ERROR};
}
#
#-----
1;

```

## B.2. OAS::Auth.pm

```

package OAS::Auth;
require 5.8.5;
our $VERSION = 1.0.0;
our $DATE = '05 Aug 2006';

use strict;
use warnings;

use Carp;
use OAS::DBI;
use OAS::Person qw(is_valid_person_id);
use OAS::Document qw(is_valid_document_id document_owner document_id_to_path);

use Exporter;
our @ISA = qw(Exporter);
our @EXPORT = qw();
our @EXPORT_OK = qw{
    ADMIN_GROUP          ACL_READ_ACCESS          ACL_DELETE_ACCESS
    ACL_ANNOTATE_ACCESS  ACL_READ_ANNOTATION_ACCESS ACL_DELETE_ANNOTATION_ACCESS
    ACL_MODERATE_ACCESS  ACL_ALL_ACCESS
    current_user         is_user                 is_admin
    list_all_users       is_valid_group          create_group
    remove_group         list_groups             list_users_groups
    group_owner          is_group_member        add_user_to_group
    remove_user_from_group list_group_membership is_valid_acl_access
    grant_user_access    grant_group_access     revoke_user_access
    revoke_group_access  has_access             has_user_access
    has_group_access     list_access            list_user_access
    list_group_access    has_read_access        has_delete_access
    has_annotate_access  has_read_annotation_access has_delete_annotation_access
    has_moderate_access  who_has_user_access    who_has_group_access
    list_accessible_documents write_htaccess     error_message
    clear_error
};
our %EXPORT_TAGS = (
    user => [qw( current_user         is_user
                 is_admin             list_all_users
                 )],
    group => [qw( ADMIN_GROUP
                 is_valid_group
                 remove_group
                 list_users_groups
                 group_owner
                 is_group_member
                 add_user_to_group
                 remove_user_from_group
                 list_group_membership
                 )],
    acl => [qw( ACL_READ_ACCESS
                ACL_DELETE_ACCESS
                ACL_ANNOTATE_ACCESS
                ACL_READ_ANNOTATION_ACCESS
                )],
);

```



```

ACL_DELETE_ANNOTATION_ACCESS ACL_MODERATE_ACCESS
ACL_ALL_ACCESS
is_valid_acl_access          grant_user_access
grant_group_access           revoke_user_access
revoke_group_access          has_access
has_user_access              has_group_access
list_access                   list_user_access
list_group_access            has_read_access
has_delete_access            has_annotate_access
has_read_annotation_access   has_delete_annotation_access
has_moderate_access          who_has_user_access
who_has_group_access         list_accessible_documents
write_htaccess                )]],
error => [qw( error_message  clear_error  )]]
);

```

```

=====
# CONSTANTS
#
use constant ADMIN_GROUP => 'ADMIN';
#-----
use constant ACL_READ_ACCESS          => 'R';
use constant ACL_DELETE_ACCESS        => 'D';
use constant ACL_ANNOTATE_ACCESS      => 'A';
use constant ACL_READ_ANNOTATION_ACCESS => 'Ra';
use constant ACL_DELETE_ANNOTATION_ACCESS => 'Da';
use constant ACL_MODERATE_ACCESS      => 'M';
use constant ACL_ALL_ACCESS           => (ACL_READ_ACCESS,
                                         ACL_DELETE_ACCESS,
                                         ACL_ANNOTATE_ACCESS,
                                         ACL_READ_ANNOTATION_ACCESS,
                                         ACL_DELETE_ANNOTATION_ACCESS,
                                         ACL_MODERATE_ACCESS);
#
=====

```

```

=====
# VARIABLES
#
my $Dbh    = undef;
my $Error = '';
#
=====

```

```

=====
# EVENT HANDLERS
#
END
{
    OAS::DBI::disconnect if defined $Dbh;
}
#
=====

```

```

#-----
# USER FUNCTIONS
#

```

```

sub current_user ()
{
    return (exists $ENV{'REMOTE_USER'}) ? $ENV{'REMOTE_USER'} : '';
}
#
#-----
#
sub is_user ($)
{
    my $id = $_[0];
    my $href = undef;

    # Validate parameters
    unless ($id) {
        $Error = 'Missing ID for user lookup'; return;
    }
    unless ( is_valid_person_id $id ) {
        $Error = 'Invalid ID for user lookup'; return;
    }

    # Query database for needed information
    $Dbh = OAS::DBI::connect unless $Dbh;
    unless ($href = $Dbh->selectrow_hashref("SELECT id FROM person WHERE id='$id'")) {
        $Error = $Dbh->errstr;
        return;
    }

    return (exists $href->{id});
}
#
#-----
#
sub is_admin ($)
{
    return is_group_member($_[0], ADMIN_GROUP);
}
#
#-----
#
sub list_all_users ()
{
    my $result = [];

    # Prepare and execute query
    $Dbh = OAS::DBI::connect unless $Dbh;
    my $sth;
    unless ($sth = $Dbh->prepare(qq(
        SELECT id, last_name, first_name
        FROM person
        ORDER BY last_name, first_name
    ))) {
        $Error = $Dbh->errstr;
        return;
    }
    unless (defined $sth->execute) {
        $Error = $sth->errstr;
        return;
    }

    # Record information from database
    my $c = 0;
    while (my $href = $sth->fetchrow_hashref) {
        foreach my $key (sort keys %{$href}) {
            $result->[$c][$key] = $href->{$key};
        }
    }
}

```

```

        $c++;
    }

    return $result;
}
#
#-----

#-----
# GROUP FUNCTIONS
#
sub is_valid_group ($)
{
    my $group = $_[0];
    my $href = undef;

    $Dbh = OAS::DBI::connect unless $Dbh;

    # Query database for needed information
    unless ($href =
        $Dbh->selectrow_hashref("SELECT name FROM groups WHERE name='$group'"))
    {
        $Error = $Dbh->errstr;
        return;
    }

    return (exists $href->{name});
}
#
#-----
#
sub create_group ($;$)
{
    my ($name, $owner) = @_;
    return unless (defined($name) and $name);

    # Standardize group name
    $name =~ s/\W/_/g;
    $name = lc substr $name, 0, 32;

    # Do database call
    $Dbh = OAS::DBI::connect unless $Dbh;
    my $sql = 'INSERT INTO groups'
        . " SET name = '$name'";
    $sql .= " , owner = '$owner'" if defined $owner;
    unless ($Dbh->do($sql)) {
        $Error = $Dbh->errstr;
        return;
    }

    # Put owner in group
    add_user_to_group($owner, $name);

    return 1;
}
#
#-----
#
sub remove_group ($)
{
    my $name = $_[0];
    return unless defined $name;

    $Dbh = OAS::DBI::connect unless $Dbh;

```

```

        unless ($Dbh->do("DELETE FROM groups WHERE name='$name'")) {
            $Error = $Dbh->errstr;
            return;
        }

        return 1;
    }
#
#-----
#
sub list_groups (;$)
{
    my $owner = $_[0];

    my ($href, $sth, @result);
    $Dbh = OAS::DBI::connect unless $Dbh;

    my $sql = 'SELECT name'
        . ' FROM groups';
    $sql .= " WHERE owner = '$owner'" if defined $owner;
    $sql .= ' ORDER BY name';
    unless ($sth = $Dbh->prepare($sql)) {
        $Error = $Dbh->errstr;
        return;
    }
    unless ($sth->execute) {
        $Error = $Dbh->errstr;
        return;
    }
    while ($href = $sth->fetchrow_hashref) {
        push @result, $href->{name};
    }
    return \@result;
}
#
#-----
#
sub list_users_groups ($)
{
    my $user = $_[0];
    my ($href, $sth, @result);
    $Dbh = OAS::DBI::connect unless $Dbh;

    my $sql = qq|
        SELECT group_name
          FROM group_membership
         WHERE person_id = '$user'
    |;
    unless ($sth = $Dbh->prepare($sql)) {
        $Error = $Dbh->errstr;
        return;
    }
    unless ($sth->execute) {
        $Error = $Dbh->errstr;
        return;
    }
    while ($href = $sth->fetchrow_hashref) {
        push @result, $href->{group_name};
    }
    return (@result) ? sort @result : ();
}
#
#-----
#

```

```

sub group_owner ($;$)
{
    my ($name, $new_owner) = @_;
    my $result = undef;
    return unless defined $name;

    my ($dbh, $sth, @row);
    $dbh = OAS::DBI::connect unless $dbh;

    # Set new owner, if necessary
    if (is_valid_person_id $new_owner) {
        $dbh->do("UPDATE groups SET owner='$new_owner' WHERE name='$name'");
        add_user_to_group($new_owner, $name);
    }

    # Get owner
    $sth = $dbh->prepare("SELECT owner FROM groups WHERE name='$name'");
    $sth->execute;
    $result = $row[0] if (@row = $sth->fetchrow_array);

    return $result;
}
#
#-----
#
sub is_group_member ($$)
{
    my $id      = $_[0];
    my $group   = $_[1];
    my $href    = undef;

    # Validate parameters
    unless ($id) {
        $Error = 'Missing ID for user lookup'; return;
    }
    unless (is_valid_person_id $id) {
        $Error = 'Invalid ID for user lookup'; return;
    }
    unless ($group) {
        $Error = 'Missing GROUP for user lookup'; return;
    }
    unless (is_valid_group $group) {
        $Error = 'Invalid GROUP for user lookup'; return;
    }

    # The group owner is always a member of the group
    my $owner = &group_owner($group);
    return 1 if (defined($owner) and ($id eq $owner));

    # Query database for needed information
    unless ($href = $dbh->selectrow_hashref(qq(
        SELECT person_id
        FROM group_membership
        WHERE person_id = '$id'
        AND group_name = '$group'
    ))) {
        $Error = $dbh->errstr;
        return;
    }

    return (exists $href->{person_id});
}
#
#-----
#

```

```

sub add_user_to_group ($$)
{
    return _group_membership_control(@_,1);
}
#
#-----
#
sub remove_user_from_group ($$)
{
    my ($user, $group) = @_;
    return 0 if ( ( ADMIN_GROUP ne $group
                  and ( $user eq group_owner($group) ) ) );
    return _group_membership_control(@_,0);
}
#
#-----
#
sub _group_membership_control ($$$)
{
    my ($id, $group, $mode) = @_;

    # Validate parameters
    unless (defined $id) {
        $Error = 'Missing ID for privilege control'; return;
    }
    unless (is_valid_person_id $id) {
        $Error = 'Invalid ID for privilege control'; return;
    }
    unless (defined $group) {
        $Error = 'Missing GROUP for privilege control'; return;
    }
    unless (is_valid_group $group) {
        $Error = 'Invalid GROUP for privilege control'; return;
    }
    unless (defined $mode) {
        $Error = 'Missing MODE for privilege control'; return;
    }

    # Perform database operation
    my $sql = '';
    if ($mode) {
        # Grant privilege
        $sql = qq(
            INSERT
            INTO group_membership
            SET person_id = '$id',
              group_name = '$group'
            ON DUPLICATE KEY
            UPDATE person_id = '$id',
              group_name = '$group'
        );
    } else {
        # Revoke privilege
        $sql = qq(
            DELETE
            FROM group_membership
            WHERE person_id = '$id'
              AND group_name = '$group'
        );
    }
    unless ($Dbh->do($sql)) {
        $Error = $Dbh->errstr;
        return;
    }

    return 1;
}
#

```

```

#-----
#
sub list_group_membership (;$$)
{
    my $group = $_[0];
    my $sort = $_[1] || '';
    my $result = [];

    my $sql = 'SELECT person_id';
    $sql .= ", $sort" if $sort;
    $sql .= ' FROM group_membership';
    $sql .= ', person' if $sort;
    $sql .= ' WHERE' if $sort or $group;
    $sql .= " group_name = '$group'" if $group;
    $sql .= ' AND' if $sort and $group;
    $sql .= ' person.id = group_membership.person_id' if $sort;
    $sql .= " ORDER BY $sort" if $sort;

    # Prepare and execute query
    $Dbh = OAS::DBI::connect unless $Dbh;
    my $sth;
    unless ( $sth = $Dbh->prepare($sql) ) { $Error = $Dbh->errstr; return; }
    unless ( defined $sth->execute ) { $Error = $sth->errstr; return; }

    # Record information from database
    my $c = 0;
    while (my $href = $sth->fetchrow_hashref) {
        $result->[$c]['id'] = $href->{'person_id'};
        foreach my $key (sort keys %{$href}) {
            $result->[$c][$key] = $href->{$key};
        }
        $c++;
    }

    return $result;
}
#-----

```

```

#-----
# ACL FUNCTIONS
#
sub is_valid_acl_access ($)
{
    my $access = $_[0];
    return unless defined($access) and $access =~ /^[A-Z]a?$/;
    my $result = 0;
    foreach my $acl (ACL_ALL_ACCESS) {
        if ($access eq $acl) {
            $result++;
            last;
        }
    }
    return $result;
}
#-----

```

```

#
sub grant_user_access ($$$)
{
    my ($user, $doc, $acl) = @_;
    return unless ( is_valid_person_id $user )
        and ( is_valid_document_id $doc )
        and ( is_valid_acl_access $acl );
}

```



```

# Connect to database and perform operation
$Dbh = OAS::DBI::connect unless $Dbh;
unless ($Dbh->do(qq(
    INSERT INTO acl_user
        SET person_id = '$user',
            document_id = '$doc',
            acl = '$acl'
))) {
    $Error = $Dbh->errstr;
    return;
}

return 1;
}
#
#-----
#
sub grant_group_access ($$$)
{
    my ($group, $doc, $acl) = @_;
    return unless ( defined $group )
        and ( is_valid_document_id $doc )
        and ( is_valid_acl_access $acl );

    # Connect to database and perform operation
    $Dbh = OAS::DBI::connect unless $Dbh;
    unless ($Dbh->do(qq(
        INSERT INTO acl_group
            SET group_name = '$group',
                document_id = '$doc',
                acl = '$acl'
        ))) {
        $Error = $Dbh->errstr;
        return;
    }

    return 1;
}
#
#-----
#
sub revoke_user_access ($$$)
{
    my ($user, $doc, $acl) = @_;
    return unless ( is_valid_person_id $user )
        and ( is_valid_document_id $doc )
        and ( is_valid_acl_access $acl );

    # Connect to database and perform operation
    $Dbh = OAS::DBI::connect unless $Dbh;
    unless ($Dbh->do(qq(
        DELETE FROM acl_user
            WHERE person_id = '$user'
                AND document_id = '$doc'
                AND acl = '$acl'
        ))) {
        $Error = $Dbh->errstr;
        return;
    }

    return 1;
}
#
#-----
#

```



```

sub revoke_group_access ($$$)
{
    my ($group, $doc, $acl) = @_;
    return unless ( defined $group )
        and ( is_valid_document_id $doc )
        and ( is_valid_acl_access $acl );

    # Connect to database and perform operation
    $Dbh = OAS::DBI::connect unless $Dbh;
    unless ($Dbh->do(qq(
        DELETE FROM acl_group
        WHERE group_name = '$group'
        AND document_id = '$doc'
        AND acl = '$acl'
    ))) {
        $Error = $Dbh->errstr;
        return;
    }

    return 1;
}
#
#-----
#
sub has_access ($$$)
{
    return unless ( is_valid_person_id $_[0] )
        and ( is_valid_document_id $_[1] )
        and ( is_valid_acl_access $_[2] );
    return (has_user_access(@_) or has_group_access(@_));
}
#
#-----
#
sub has_user_access ($$$)
{
    my ($user, $doc, $acl) = @_;
    return unless ( is_valid_person_id $user )
        and ( is_valid_document_id $doc )
        and ( is_valid_acl_access $acl );
    return 1 if _check_acl_override($user, $doc);

    # Query database for needed information
    $Dbh = OAS::DBI::connect unless $Dbh;
    my $href = undef;
    unless ($href = $Dbh->selectrow_hashref(qq(
        SELECT person_id
        FROM acl_user
        WHERE person_id = '$user'
        AND document_id = '$doc'
        AND acl = '$acl'
    ))) {
        $Error = $Dbh->errstr;
        return;
    }

    return (exists $href->{person_id});
}
#
#-----
#
sub has_group_access ($$$)
{
    my ($user, $doc, $acl) = @_;
    return unless ( is_valid_person_id $user )
        and ( is_valid_document_id $doc )
        and ( is_valid_acl_access $acl );
}

```

```

# Query database for needed information
$Dbh = OAS::DBI::connect unless $Dbh;
my $href = undef;
unless ($href = $Dbh->selectrow_hashref(qq(
    SELECT acl_group.group_name
    FROM acl_group, group_membership
    WHERE group_membership.group_name = acl_group.group_name
    AND group_membership.person_id = '$user'
    AND acl_group.document_id = '$doc'
    AND acl_group.acl = '$acl'
))) {
    $Error = $Dbh->errstr;
    return;
}

return (exists $href->{group_name});
}
#-----
#
#
sub list_access ($$)
{
    my @user_acl = &list_user_access(@_);
    my @group_acl = &list_group_access(@_);
    my %hash;
    foreach my $acl (@user_acl, @group_acl) { $hash{$acl}++ };
    return sort keys %hash;
}
#-----
#
#
sub list_user_access ($$)
{
    my ($user, $doc) = @_;
    return unless ( is_valid_person_id $user )
        and ( is_valid_document_id $doc );
    return ACL_ALL_ACCESS if _check_acl_override($user, $doc);

    # Query database for needed information
    $Dbh = OAS::DBI::connect unless $Dbh;
    my $sth = $Dbh->prepare(qq(
        SELECT acl
        FROM acl_user
        WHERE person_id = '$user'
        AND document_id = '$doc'
    ));
    $sth->execute;
    my (@row, %result);
    $result{$row[0]}++ while (@row = $sth->fetchrow_array);

    return sort keys %result;
}
#-----
#
#
sub list_group_access ($$)
{
    my ($user, $doc) = @_;
    return unless ( is_valid_person_id $user )
        and ( is_valid_document_id $doc );

    # Query database for needed information
    $Dbh = OAS::DBI::connect unless $Dbh;

```

```

my $sth = $Dbh->prepare(qq(
    SELECT acl_group.acl
    FROM acl_group, group_membership
    WHERE group_membership.group_name = acl_group.group_name
    AND group_membership.person_id = '$user'
    AND acl_group.document_id = '$doc'
));
$sth->execute;
my (@row, %result);
$result{$row[0]}++ while (@row = $sth->fetchrow_array);

return sort keys %result;
}
#-----
#
sub has_read_access ($$)
{
    return &has_access(@_, ACL_READ_ACCESS);
}
#-----
#
sub has_delete_access ($$)
{
    return &has_access(@_, ACL_DELETE_ACCESS);
}
#-----
#
sub has_annotate_access ($$)
{
    return &has_access(@_, ACL_ANNOTATE_ACCESS);
}
#-----
#
sub has_read_annotation_access ($$)
{
    return &has_access(@_, ACL_READ_ANNOTATION_ACCESS);
}
#-----
#
sub has_delete_annotation_access ($$)
{
    return &has_access(@_, ACL_DELETE_ANNOTATION_ACCESS);
}
#-----
#
sub has_moderate_access ($$)
{
    return &has_access(@_, ACL_MODERATE_ACCESS);
}
#-----
#
sub who_has_user_access ($;$)
{
    my $id      = $_[0];
    my $access = $_[1];
    return unless is_valid_document_id($id);
    my $result;

    # Set Owner Permissions
    my $owner = document_owner $id;
    map { $result->{$owner}{$_}++ } ACL_ALL_ACCESS;
}

```

```

# Query database for needed information
my $sql = qq|
    SELECT person_id, acl
    FROM acl_user
    WHERE document_id = '$id'
|;
$sql .= "AND acl = '$access'" if defined $access;
$dbh = OAS::DBI::connect unless $dbh;
my $sth = $dbh->prepare($sql);
$sth->execute;
while (my $href = $sth->fetchrow_hashref) {
    $result->{$href->{'person_id'}}{$href->{'acl'}}++;
}

return $result;
}
#
#-----
#
sub who_has_group_access ($;$)
{
    my $id = $_[0];
    my $access = $_[1];
    return unless is_valid_document_id($id);
    my $result;

    # Query database for needed information
    my $sql = qq|
        SELECT group_name, acl
        FROM acl_group
        WHERE document_id = '$id'
    |;
    $sql .= "AND acl = '$access'" if defined $access;
    $dbh = OAS::DBI::connect unless $dbh;
    my $sth = $dbh->prepare($sql);
    $sth->execute;
    while (my $href = $sth->fetchrow_hashref) {
        $result->{$href->{'group_name'}}{$href->{'acl'}}++;
    }
    delete $result->{'ADMIN_GROUP'} if exists $result->{'ADMIN_GROUP'};

    return $result;
}
#
#-----
#
sub list_accessible_documents ($;$)
{
    my $user = $_[0];
    my $sort = (defined $_[1]) ? $_[1] : 'timestamp';
    return unless is_valid_person_id($user);

    my ($result, $sql);
    $dbh = OAS::DBI::connect unless $dbh;
    if (is_admin $user) {
        # User is an administrator
        # - Gather all documents
        $sql = qq|
            SELECT id, title, owner, author, timestamp
            FROM document
            ORDER BY $sort
        |;
        $result = _build_accessible_doc_list($user, $sql, 1);
    } else {
        # User is a normal user
        # - Gather documents user is the owner of
    }
}

```

```

$sql = qq|
    SELECT id, title, owner, author, timestamp
    FROM document
    WHERE owner = '$user'
|;
my $set1 = _build_accessible_doc_list($user, $sql);

# - Gather documents user has user access to
$sql = qq|
    SELECT document.id,
           document.title,
           document.owner,
           document.author,
           document.timestamp
    FROM document, acl_user
    WHERE document.id = acl_user.document_id
          AND acl_user.person_id = '$user'
|;
my $set2 = _build_accessible_doc_list($user, $sql);

# - Gather documents user has group access to
$sql = qq|
    SELECT document.id,
           document.title,
           document.owner,
           document.author,
           document.timestamp
    FROM document, acl_group, group_membership
    WHERE document.id = acl_group.document_id
          AND acl_group.group_name = group_membership.group_name
          AND group_membership.person_id = '$user'
|;
my $set3 = _build_accessible_doc_list($user, $sql);

# Build combined, sorted list
my %hash;
my $desc = ($sort =~ / DESC$/) ? 1 : 0;
           $sort =~ s/ DESC$//;
foreach my $set ($set1, $set2, $set3) {
    foreach my $key (keys %{$set}) {
        $hash{"$set->{$key}{$sort}$key"} = $set->{$key};
    }
}
my $count = 0;
if ($desc) {
    foreach my $key (reverse sort keys %hash) {
        $result->[$count++] = $hash{$key};
    }
} else {
    foreach my $key (sort keys %hash) {
        $result->[$count++] = $hash{$key};
    }
}

return $result;
}
#
#-----
#
sub _build_accessible_doc_list ($;$;$)
{
    my ($user, $sql, $is_admin) = @_;
    return unless defined($sql) and is_valid_person_id($user);
    $is_admin = (defined $is_admin) ? $is_admin : 0;

    my ($sth, $href, $result);
    my $count = 0;

```

```

$Dbh = OAS::DBI::connect unless $Dbh;
$sth = $Dbh->prepare($sql) or return;
$sth->execute;
while ($href = $sth->fetchrow_hashref) {
    if ($is_admin) {
        # Build final array if administrator
        foreach my $key (keys %{$href}) {
            $result->[$count]{$key} = $href->{$key};
        }
        $result->[$count++]{'acl'} = [ACL_ALL_ACCESS];
    } else {
        # Build temp hash if not administrator
        foreach my $key (keys %{$href}) {
            $result->{$href->{'id'}}{$key} = $href->{$key};
        }
        $result->{$href->{'id'}}{'acl'} =
            [&list_access($user, $href->{'id'})];
    }
}
return $result;
}
#
#-----
#
sub _check_acl_override ($$)
{
    return ((is_admin $_[0]) or ($_[0] eq document_owner $_[1]));
}
#
#-----
#
sub write_htaccess ($)
{
    my $id = $_[0];
    return unless is_valid_document_id $id;

    my $text = qq|
        AuthName "OAS - Online Annotation System"
        AuthType Basic
        AuthMySQLHost localhost
        AuthMySQLDB oas
        AuthMySQLUser oas
        AuthMySQLPassword oas
        AuthMySQLUserTable person
        AuthMySQLNameField id
        AuthMySQLPasswordField password
        AuthMySQLGroupTable "person, group_membership"
        AuthMySQLGroupCondition "person.id = group_membership.person_id"
        AuthMySQLGroupField group_name
        AuthMySQLEnable On
        AuthMySQLPwEncryption crypt
    |;

    # Set User Requires
    my $user_list = '';
    foreach my $user (sort keys %{who_has_user_access $id, ACL_READ_ACCESS}) {
        $user_list .= " $user";
    }
    $text .= "Require user$user_list\n" if $user_list;

    # Set Group Requires
    my $group_list = '';
    $text .= "Require group ' . ADMIN_GROUP . "\n";
    foreach my $group (sort keys %{who_has_group_access $id, ACL_READ_ACCESS}) {
        $group_list .= " $group";
    }
    $text .= "Require group$group_list\n" if $group_list;
    $text .= "Satisfy any\n";
}

```

```

# Clean file before write
$text =~ s/[\r\n]+/\n/g;
$text =~ s/\t//g;

my $file = document_id_to_path($id) . '.htaccess';
open(HTACCESS, ">$file") or return;
print HTACCESS $text;
close HTACCESS;
chmod 0600, $file;
}
#
#-----

#-----
# ERROR CONTROL FUNCTIONS
#
sub error_message ()
{
    return $Error;
}
#
#-----
#
sub clear_error ()
{
    $Error = '';
}
#
#-----

1;

```

### B.3. OAS::CGI.pm

```

package OAS::CGI;
require 5.8.5;
our $VERSION = 1.0.0;
our $DATE    = '04 Aug 2006';

use strict;
use warnings;
use CGI;
use CGI::Carp qw(fatalsToBrowser);

use Exporter;
our @ISA      = qw(Exporter);
our @EXPORT  = qw{
    html_header
    html_footer
};
our @EXPORT_OK = qw{
    html_header
    html_footer
    close_window_form
};
our %EXPORT_TAGS = (
    all => [qw(html_header html_footer close_window_form)]
);

#-----
# HTML FUNCTIONS
#

```

```

sub html_header (;$$$$)
{
    my $title = (defined $_[0]) ? $_[0] : '';
    my $css    = (defined $_[1])
        ? qq|<link rel="stylesheet" type="text/css" href="$_[1]" />|
        : '';
    my $head   = (defined $_[2]) ? $_[2] : '';
    my $body   = (defined $_[3]) ? $_[3] : '';
    my $html   = qq|<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
        "http://www.w3.org/TR/html4/loose.dtd">
        <html>
        <head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <meta http-equiv="Pragma" content="no-cache" />
        <meta http-equiv="Expires" content="-1" />
        $css
        $head
        <title>$title</title>
        </head>
        <body marginwidth="0" marginheight="0" $body>

    |;
    $html =~ s/\t+//g;
    $html =~ s/[\r\n]+ ?[\r\n]+/\n/g;
    return $html;
}
#
#-----
#
sub html_footer ()
{
    my $html .= qq|
        </body>
        <head>
        <meta http-equiv="Pragma" content="no-cache" />
        <meta http-equiv="Expires" content="-1" />
        </head>
        </html>

    |;
    $html =~ s/\t+//g;
    $html =~ s/[\r\n]+ ?[\r\n]+/\n/g;
    return $html;
}
#
#-----
#
sub close_window_form (;$)
{
    my $label = (defined $_[0]) ? $_[0] : 'Close Window';

    return qq|
        <form name="close_window" onSubmit="self.close()">
        <input type="SUBMIT" name="submit" value="$label" />
        </form>

    |;
}
#
#-----
1;

```

## B.4. OAS::Converter.pm

```

package OAS::Converter;
require 5.8.5;

```



```

our $VERSION = 1.0.0;
our $DATE    = '07 Aug 2006';

use strict;
use warnings;

use Carp;
use Thread;
use OAS::DBI;
use OAS::Utils qw(untaint :files);
use OAS::Document qw(is_valid_document_id);

use Exporter;
our @ISA = qw(Exporter);
our @EXPORT = qw();
our @EXPORT_OK = qw(
    DEFAULT_RESOLUTION
    CONVERSION_ERROR_NONE      CONVERSION_ERROR_INTERNAL
    CONVERSION_ERROR_FILENAME  CONVERSION_ERROR_PATH
    CONVERSION_ERROR_ID        CONVERSION_ERROR_DATABASE
    CONVERSION_ERROR_EXTERNAL  CONVERSION_ERROR_FILESYSTEM
    CONVERSION_ERROR_NETWORK   CONVERSION_ERROR_UNKNOWN
    convert                     conversion_fail
    list_converters
    is_valid_extension          is_valid_command
    is_valid_description        is_valid_resolution
    untaint_extension          untaint_command
    untaint_description        untaint_resolution
);
our %EXPORT_TAGS = (
    check      => [qw( is_valid_extension          is_valid_command
                       is_valid_description        is_valid_resolution
                       untaint_extension          untaint_command
                       untaint_description        untaint_resolution
                       resolution => [qw( is_valid_resolution
                                           DEFAULT_RESOLUTION
                                           )],
    errors     => [qw( conversion_fail
                       CONVERSION_ERROR_NONE      CONVERSION_ERROR_INTERNAL
                       CONVERSION_ERROR_FILENAME  CONVERSION_ERROR_PATH
                       CONVERSION_ERROR_ID        CONVERSION_ERROR_DATABASE
                       CONVERSION_ERROR_EXTERNAL  CONVERSION_ERROR_FILESYSTEM
                       CONVERSION_ERROR_NETWORK   CONVERSION_ERROR_UNKNOWN
    )];

#=====
# EVENT HANDLERS
#
sub DESTROY
{
    # Disconnect from database on exit if needed
    my $self = shift;
    OAS::DBI::disconnect() if (defined $self->{DBH});
    $self->SUPER::DESTROY if $self->can("SUPER::DESTROY");
}
#
#=====

#=====
# CONSTANTS
#
use constant DEFAULT_RESOLUTION          => 72;
use constant CONVERSION_ERROR_NONE      => 0;
use constant CONVERSION_ERROR_INTERNAL  => 10;
use constant CONVERSION_ERROR_FILENAME  => 11;
use constant CONVERSION_ERROR_PATH      => 12;

```

```

use constant CONVERSION_ERROR_ID          => 13;
use constant CONVERSION_ERROR_DATABASE   => 14;
use constant CONVERSION_ERROR_EXTERNAL   => 50;
use constant CONVERSION_ERROR_FILESYSTEM => 51;
use constant CONVERSION_ERROR_NETWORK    => 52;
use constant CONVERSION_ERROR_UNKNOWN    => 255;
#
#=====

#-----
# FUNCTIONS
#
sub convert ($$$;$)
{
    my ($in_filename, $out_path, $id, $quality) = @_;

    # Verify Information
    return unless defined( $in_filename )
        and defined( $out_path )
        and defined( $id );

    my $in_path = file_path      $in_filename;
    my $in_file = file_name      $in_filename;
    my $in_ext  = file_extension $in_file;

    return unless is_valid_filename( $in_file )
        and is_valid_path( $in_path )
        and is_valid_path( $out_path )
        and is_valid_document_id( $id );

    $quality = DEFAULT_RESOLUTION unless $quality =~ /\d{2,3}$/;

    $in_path = untaint $in_path;
    $in_file = untaint $in_file;
    $in_ext  = untaint $in_ext;
    $out_path = untaint $out_path;
    $id       = untaint $id;
    $quality  = untaint $quality;

    # Check for appropriate conversion
    my $conv = new OAS::Converter;
    return unless defined $in_ext;
    return unless &is_valid_extension($in_ext);
    return unless $conv->Load($in_ext);

    # Do conversion
    my $command = $conv->command;
    my $result = system $command, $in_filename, $out_path, $id, $quality;
    if ($result) {
        warn "Conversion Failed [$result]";
        return;
    }

    return $id;
}
#
#-----
#
sub conversion_fail ($$)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';
    my $error_code = (defined $_[1]) ? $_[1] : CONVERSION_ERROR_UNKNOWN;
    warn "Conversion Failed: $message [$error_code]";
    exit $error_code;
}
#

```

```

#-----
#
sub list_converters ()
{
    my $result = undef;

    my $dbh = OAS::DBI::connect;
    my $sth = $dbh->prepare('SELECT * FROM converter');
    $sth->execute;
    while (my $href = $sth->fetchrow_hashref) {
        foreach my $key (keys %{$href}) {
            $result->{$href->{'in_ext'}}{$key} = $href->{$key};
        }
    }
    OAS::DBI::disconnect;

    return $result;
}
#
#-----
#
sub is_valid_extension ($)
{
    local $_ = $_[0] || '';
    return /^[a-z\d\-\]{1,8} $/x;
}
#
#-----
#
sub is_valid_command ($)
{
    local $_ = $_[0] || '';
    return /^[\\w\\-\\.\\300-\\366\\370-\\377]+ $/x;
}
#
#-----
#
sub is_valid_description ($)
{
    local $_ = $_[0] || '';
    return /^[\\w\\-+\\/'"\\.\\: \\300-\\366\\370-\\377]+ $/x;
}
#
#-----
#
sub is_valid_resolution ($)
{
    local $_ = $_[0] || '';
    return /^[d]{2,3} $/x;
}
#
#-----
#
sub untaint_extension ($)
{
    return (is_valid_extension $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_command ($)
{
    return (is_valid_command $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#

```

```

sub untaint_description ($)
{
    return (is_valid_description $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_resolution ($)
{
    return ( is_valid_resolution $_[0] ) ? untaint $_[0] : undef;
}
#
#-----

#-----
# CONSTRUCTORS & INITIALIZERS
#
sub new #()#
{
    my $class = shift;
    $class = ref $class || $class;
    my $self = bless {}, $class;
    $self->_reset_defaults;
    $self->clear_error;
    $self->{ CHANGED } = 1;
    $self->{ DBH } = undef;
    return $self;
}
#
#-----
#
sub _reset_defaults ()
{
    my $self = shift;
    foreach (qw(in out command description)) {
        $self->{$_} = undef;
    }
}
#
#-----

#-----
# DATA CONTROL METHODS
#
sub Load : locked method #($)#
{
    my $self = shift;
    my $in = $_[0];
    my $href = undef;
    my $key = undef;

    # Verify parameters
    if ( not defined $in ) { croak 'Missing Extension for load'; }
    elsif ( not is_valid_extension $in ) { croak 'Invalid Extension for load'; }

    # Connect to database
    $self->{DBH} = OAS::DBI::connect unless $self->{DBH};

    # Query database for needed information
    unless ($href = $self->{DBH}->selectrow_hashref(
        "SELECT * FROM converter WHERE in_ext='$in'"
    )) {

```

```

        $self->{ERROR} = $self->{DBH}->errstr;
        return;
    }

    # Reset object values to default
    $self->_reset_defaults;
    $self->_clear_error;
    $self->{CHANGED} = 0;

    # Store the results in the object
    $self->in($href->{'in_ext'}) if defined $href->{'in_ext'};
    $self->out($href->{'out_ext'}) if defined $href->{'out_ext'};
    $self->command($href->{'command'}) if defined $href->{'command'};
    $self->description($href->{'description'}) if defined $href->{'description'};

    return 1;
}
#
#-----
#
sub Save : locked #()#
{
    my $self = shift;
    my $key = undef;

    # Check for forced save
    $self->{CHANGED} = 1 if $_[0];

    # Verify needed columns
    if ( not $self->{in} ) {
        $self->{ERROR} = 'Missing IN for save'; return;
    } elsif ( not $self->{out} ) {
        $self->{ERROR} = 'Missing OUT for save'; return;
    } elsif ( not $self->{command} ) {
        $self->{ERROR} = 'Missing COMMAND for save'; return;
    } elsif ( not $self->{desc} ) {
        $self->{ERROR} = 'Missing DESCRIPTION for save'; return;
    }
    return SAVE_NOT_REQUIRED unless $self->{CHANGED};

    # Connect to database
    $self->{DBH} = OAS::DBI::connect unless $self->{DBH};

    # Gather & prepare data for query
    my $in = $self->{DBH}->quote( $self->{ in } );
    my $out = $self->{DBH}->quote( $self->{ out } );
    my $command = $self->{DBH}->quote( $self->{ command } );
    my $desc = $self->{DBH}->quote( $self->{ desc } );

    # Build SQL query
    my $query;
    ($query = <<__QUERY_END__>> =~ s/(?:\^\s+)|\s+(?==)|\n/ /gm;
        in_ext = $in,
        out_ext = $out,
        command = $command,
        description = $desc
    __QUERY_END__
    $query = "INSERT INTO converter SET $query ON DUPLICATE KEY UPDATE $query";

    # Execute query
    my $sth = $self->{DBH}->prepare($query);
    unless ($sth->execute) {
        $self->{ERROR} = $sth->errstr;
        return;
    }
}

```

```

# Final items
$self->{CHANGED} = 0;

return 1;
}
#
-----
#
sub Delete : locked #()#
{
    my $self = shift;
    my $sth = undef;

    # Delete from database
    if (defined $self->{in}) {
        $self->{DBH} = OAS::DBI::connect unless $self->{DBH};
        if (!$sth = $self->{DBH}->prepare(
            "DELETE FROM converter WHERE in_ext='$self->{in}'"
        )) {
            $self->{ERROR} = $self->{DBH}->errstr;
            return;
        }
        unless (defined $sth->execute) {
            $self->{ERROR} = $sth->errstr;
            return;
        }
    }

    # Clear object contents
    $self->_reset_defaults;
    $self->_clear_error;
    $self->{CHANGED} = 0;

    return 1;
}
#
-----
#
sub clear_error #()#
{
    my $self = shift;
    $self->{ERROR} = '';
}
#
-----

#-----
# DATA ACCESS METHODS
#
sub in
{
    my $self = shift;
    if (@_) {
        my $data = lc shift;
        if ('' eq $data) {
            $self->{ in      } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_extension $data) {
            $self->{ERROR} = 'Invalid IN';
            return;
        }
    }
}

```

```

        $self->{ in      } = untaint_extension $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{in};
}
#
#-----
#
sub out
{
    my $self = shift;
    if (@_) {
        my $data = lc shift;
        if ('' eq $data) {
            $self->{ out      } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_extension $data) {
            $self->{ERROR} = 'Invalid OUT';
            return;
        }
        $self->{ out      } = untaint_extension $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{out};
}
#
#-----
#
sub command
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ command } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_command $data) {
            $self->{ERROR} = 'Invalid COMMAND';
            return;
        }
        $self->{ command } = untaint_command $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{command};
}
#
#-----
#
sub description
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ desc      } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_description $data) {
            $self->{ERROR} = 'Invalid DESCRIPTION';
            return;
        }
    }
}

```

```

        $self->{ desc      } = untaint_description $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{desc};
}
#
#-----
#
sub error_message #()#
{
    my $self = shift;
    return $self->{ERROR};
}
#
#-----
1;

```

## B.5. OAS::DBI.pm

```

package OAS::DBI;
require 5.6.0;
our $VERSION = 1.0.0;
our $DATE    = '15 Feb 2006';

use strict;
use warnings;

use Carp;
use DBI;
use Thread;

use Exporter ();
our @ISA      = qw(Exporter);
our @EXPORT  = qw(SAVE_NOT_REQUIRED FORCE_SAVE);

```

```

#=====
# CONSTANTS
#
use constant DBI_DEFAULT_DB_NAME      => 'oas';
use constant DBI_DEFAULT_DB_HOST      => '127.0.0.1';
use constant DBI_DEFAULT_DB_PORT      => 3306;
use constant DBI_DEFAULT_DB_USER      => 'oas';
use constant DBI_DEFAULT_DB_PASSWORD => 'oas';
use constant CONNECTION_NAME_DELIMITER => ':';
#-----
use constant SAVE_NOT_REQUIRED        => -1;
use constant FORCE_SAVE                 => 1;
#
#=====

#=====
# GLOBALS
#
my %Connection = ();
my $Error      = '';
#
#=====

```



```

=====
# EVENT HANDLERS
#
END {
    # At script exit, disconnect any remaining database handles
    my $con;
    foreach $con (keys %Connection) {
        &disconnect(&_connection_name_to_parts($con));
    }
}
#
=====

#-----
# DATABASE FUNCTIONS
#
sub connect : locked method #($$$$)#
{
    my $username = $_[0] || DBI_DEFAULT_DB_USER;
    my $password = $_[1] || DBI_DEFAULT_DB_PASSWORD;
    my $database = $_[2] || DBI_DEFAULT_DB_NAME;
    my $hostname = $_[3] || DBI_DEFAULT_DB_HOST;
    my $port     = $_[4] || DBI_DEFAULT_DB_PORT;
    my $con_name = &_parts_to_connection_name(
        $username,
        $password,
        $database,
        $hostname,
        $port
    );
    return unless $con_name;

    if (exists $Connection{$con_name}{CONNECTION_COUNT}) {
        $Connection{$con_name}{CONNECTION_COUNT}++;
    } else {
        $Connection{$con_name}{CONNECTION_COUNT} = 1;
    }

    # Build new connection if needed
    if (1 == $Connection{$con_name}{CONNECTION_COUNT}) {
        my $datasource = "DBI:mysql:database=$database;host=$hostname;port=$port";

        # Build new connection - catch exceptions and report them
        eval {
            $Connection{$con_name}{DATABASE_HANDLE} =
                DBI->connect($datasource, $username, $password)
        };
        if ($?) {
            delete $Connection{$con_name};
            $Error = $?;
            return;
        }
    }

    return $Connection{$con_name}{DATABASE_HANDLE};
}
#
#-----
#
sub disconnect #($$$$)#
{
    my $username = $_[0] || DBI_DEFAULT_DB_USER;
    my $password = $_[1] || DBI_DEFAULT_DB_PASSWORD;
    my $database = $_[2] || DBI_DEFAULT_DB_NAME;
    my $hostname = $_[3] || DBI_DEFAULT_DB_HOST;
    my $port     = $_[4] || DBI_DEFAULT_DB_PORT;

```

```

my $con_name = &_amp_parts_to_connection_name(
    $username,
    $password,
    $database,
    $hostname,
    $port
);
return unless $con_name;

# Verify connection exists
if (!exists $Connection{$con_name}) {
    $Error = 'Invalid database connection';
    return;
}

# Remove connection and disconnect if needed
$Connection{$con_name}{CONNECTION_COUNT}--;
if (0 == $Connection{$con_name}{CONNECTION_COUNT}) {
    # Disconnecting from database
    $Connection{$con_name}{DATABASE_HANDLE}->disconnect;
    delete $Connection{$con_name};
}

return 1;
}
#-----
#
sub last_insert_id ($)
{
    my $dbh = $_[0];
    return unless $dbh;
    return $dbh->{mysql_insertid};
}
#-----
#
sub _parts_to_connection_name ($$$$)
{
    my ($username, $password, $database, $hostname, $port) = @_;

    # Validate input parameters
    confess 'Missing database username' unless defined $username;
    confess 'Invalid database username' unless $username =~ /^ [\w\ -]+ $/x;
    confess 'Missing database password' unless defined $password;
    confess 'Invalid database password' unless $password =~ /^ [\w\ -]* $/x;
    confess 'Missing database name' unless defined $database;
    confess 'Invalid database name' unless $database =~ /^ [\w\ -]+ $/x;
    confess 'Missing database hostname' unless defined $hostname;
    confess 'Invalid database hostname' unless $hostname =~ /^ [\w\ -\.]+ $/x;
    confess 'Missing database port' unless defined $port;
    confess 'Invalid database port' unless $port =~ /^ \d{1,5} $/x;

    return join CONNECTION_NAME_DELIMITER,
        ($username, $password, $database, $hostname, $port);
}
#-----
#
sub _connection_name_to_parts ($)
{
    my $con_name = $_[0];
    my $delimiter = CONNECTION_NAME_DELIMITER;

    # Validate input parameter
    confess 'Missing connection name' unless $con_name;

```

```

confess 'Invalid connection name' unless ($con_name =~
    /^
        [\w\~]+   $delimiter   # Username
        [\w\~]*   $delimiter   # Password
        [\w\~]+   $delimiter   # Database name
        [\w\~\.]+ $delimiter   # Hostname
        \d{1,5}   # Port
    $/x
);

return split /$delimiter/, $con_name;
}
#
#-----
#
sub error_message ()
{
    return $Error;
}
#
#-----
#
sub clear_error_message ()
{
    $Error = '';
}
#
#-----
1;

```

## B.6. OAS::Document.pm

```

package OAS::Document;
require 5.6.0;
our $VERSION = 1.0.0;
our $DATE    = '07 May 2006';

use strict;
use warnings;

use Carp;
use Thread;
use OAS::System;
use OAS::DBI;
use OAS::Page;
use OAS::Person qw(is_valid_person_id untaint_person_id);
use OAS::Utils  qw(untaint);

use Exporter;
our @ISA = qw(Exporter);
our @EXPORT = qw();
our @EXPORT_OK = qw(
    create_new_document_id    document_id_to_path    document_id_to_annotation_path
    document_id_to_image_path document_owner
    is_valid_document_id      is_valid_author       is_valid_owner
    is_valid_title            is_valid_timestamp    is_valid_original
    untaint_document_id       untaint_author        untaint_owner
    untaint_title             untaint_timestamp     untaint_original
);
our %EXPORT_TAGS = (
    document => [qw( create_new_document_id    document_id_to_path
                     document_id_to_annotation_path document_id_to_image_path
                     document_owner

```

```

        check    => [qw( is_valid_document_id      is_valid_author
                       is_valid_owner           is_valid_title
                       is_valid_timestamp       is_valid_original    )],
        untaint => [qw( untaint_document_id      untaint_id
                       untaint_owner           untaint_title
                       untaint_timestamp       untaint_original  )]
    );

```

```

=====
# EVENT HANDLERS
#
sub DESTROY
{
    # Disconnect from database on exit if needed
    my $self = shift;
    OAS::DBI::disconnect() if (defined $self->{DBH});
    $self->SUPER::DESTROY if $self->can("SUPER::DESTROY");
}
#
=====

```

```

-----
# FUNCTIONS
#
sub create_new_document_id ()
{
    my $filename = '';

    # Create filename
    $filename = $$;
    $filename = '0'.$filename until (5 <= length($filename));
    $filename = time.'_'.$filename;

    return $filename;
}
#
-----

```

```

#
sub document_id_to_path ($)
{
    my $id = $_[0];
    return unless is_valid_document_id($id);
    my $path = $OAS::System::Conf->www_dir . 'd/'
        . substr($id, 0, 3) . '/'
        . substr($id, 3, 1) . '/'
        . substr($id, 4, 1) . '/'
        . substr($id, 5, 1) . '/'
        . substr($id, 6, 1) . '/'
        . substr($id, 7) . '/';
    $path =~ s|//|/|g;
    $path = (OAS::Utils::is_valid_path $path) ? untaint $path : undef;
    return $path;
}
#
-----

```

```

#
sub document_id_to_annotation_path ($)
{
    my $path = document_id_to_path $_[0];
    return (defined $path) ? $path.'a/' : undef;
}
#
-----
#

```

```

sub document_id_to_image_path ($)
{
    my $path = document_id_to_path $_[0];
    return (defined $path) ? $path.'i/' : undef;
}
#
#-----
#
sub document_owner ($)
{
    my $id = $_[0];
    return unless is_valid_document_id($id);

    my $result = undef;
    my $dbh = OAS::DBI::connect or return;

    my $sth = $dbh->prepare("SELECT owner FROM document WHERE id = '$id'");
    $sth->execute;

    my @row;
    $result = $row[0] if (@row = $sth->fetchrow_array);

    OAS::DBI::disconnect;
    return $result;
}
#
#-----
#
sub is_valid_id ($)
{
    local $_ = $_[0] || '';
    return 7^ \d{10} _ \d{5} $/x;
}
#
#-----
#
sub is_valid_document_id ($)
{
    return is_valid_id $_[0];
}
#
#-----
#
sub is_valid_author ($)
{
    return is_valid_person_id $_[0];
}
#
#-----
#
sub is_valid_owner ($)
{
    return is_valid_person_id $_[0];
}
#
#-----
#
sub is_valid_title ($)
{
    local $_ = $_[0] || '';
    return 7^ .+ $/x;
}
#
#-----
#

```

```

sub is_valid_timestamp ($)
{
    local $_ = $_[0] || '';
    return /^\d{10} $/x;
}
#-----
#
sub is_valid_original ($)
{
    local $_ = $_[0] || '';
    return /^\w[\.\300-\366\370-\377]+? $/x;
}
#-----
#
sub untaint_id ($)
{
    return (is_valid_id $_[0]) ? untaint $_[0] : undef;
}
#-----
#
sub untaint_document_id ($)
{
    return untaint_id $_[0];
}
#-----
#
sub untaint_author ($)
{
    return untaint_person_id $_[0];
}
#-----
#
sub untaint_owner ($)
{
    return untaint_person_id $_[0];
}
#-----
#
sub untaint_title ($)
{
    return (is_valid_title $_[0]) ? untaint $_[0] : undef;
}
#-----
#
sub untaint_timestamp ($)
{
    return (is_valid_timestamp $_[0]) ? untaint $_[0] : undef;
}
#-----
#
sub untaint_original ($)
{
    return (is_valid_original $_[0]) ? untaint $_[0] : undef;
}
#-----

```

```

#####
## CONSTRUCTORS & INITIALIZERS #####
#####
sub new #()#
{
    my $class = shift;
    $class = ref $class || $class;
    my $self = bless {}, $class;
    $self->_reset_defaults;
    $self->clear_error;
    $self->{ CHANGED } = 1;
    $self->{ DBH      } = undef;
    return $self;
}
#
#-----
#
sub _reset_defaults ()
{
    my $self = shift;
    foreach (qw(id author owner title timestamp original)) {
        $self->{$_} = undef;
    }
}
#
#-----

#-----
# DATA CONTROL METHODS
#
sub Load : locked method #($)#
{
    my $self = shift;
    my $id   = $_[0];
    my $href = undef;

    # Verify parameters
    if ( !defined $id ) { croak 'Missing Document ID for load'; }
    elsif ( !is_valid_document_id($id) ) { croak 'Invalid Document ID for load'; }

    # Connect to database
    $self->{DBH} = OAS::DBI::connect unless $self->{DBH};

    # Query database for needed information
    unless ($href = $self->{DBH}->selectrow_hashref(
        "SELECT * FROM document WHERE id='$id'"
    )) {
        $self->{ERROR} = $self->{DBH}->errstr;
        return;
    }

    # Reset object values to default
    $self->_reset_defaults;
    $self->clear_error;
    $self->{CHANGED} = 0;

    # Store the results in the object
    foreach my $key qw/id author owner title timestamp original/ {
        $self->{$key}($href->{$key}) if (defined $href->{$key});
    }

    # Load pages from the database
    my $sth;

```

```

unless ($sth = $self->{DBH}->prepare("SELECT * FROM page WHERE id='$id'")) {
    $self->{ERROR} = $self->{DBH}->errstr;
    return;
}
unless ($sth->execute) {
    $self->{ERROR} = $self->{DBH}->errstr;
    return;
}
while (my $href = $sth->fetchrow_hashref) {
    my $new_index = int($href->{'page'}) - 1;
    $self->{PAGE}[$new_index] = new OAS::Page;
    foreach my $key qw/id page image height width/ {
        $self->{PAGE}[$new_index]->$key($href->{$key})
            if defined $href->{$key};
    }
}
$sth->finish();

return 1;
}
#
#-----
#
sub Save : locked #()#
{
    my $self = shift;
    my $key = undef;

    # Check for forced save
    $self->{CHANGED} = 1 if $_[0];

    # Verify needed columns
    if ( !$self->{id} ) {
        $self->{ERROR} = 'Missing ID for save'; return;
    } elsif ( !$self->{author} ) {
        $self->{ERROR} = 'Missing AUTHOR for save'; return;
    } elsif ( !$self->{owner} ) {
        $self->{ERROR} = 'Missing OWNER for save'; return;
    } elsif ( !$self->{title} ) {
        $self->{ERROR} = 'Missing TITLE for save'; return;
    } elsif ( !$self->{timestamp} ) {
        $self->{ERROR} = 'Missing TIMESTAMP for save'; return;
    } elsif ( !$self->{original} ) {
        $self->{ERROR} = 'Missing ORIGINAL for save'; return;
    }
    return SAVE_NOT_REQUIRED unless $self->{CHANGED};

    # Create filesystem directory, if needed
    system 'mkdir', '-p', document_id_to_path( $self->{id} );
    system 'mkdir', '-p', document_id_to_image_path( $self->{id} );
    system 'mkdir', '-p', document_id_to_annotation_path( $self->{id} );

    # Connect to database
    $self->{DBH} = OAS::DBI::connect() unless $self->{DBH};

    # Gather & prepare data for query
    my $id = $self->{DBH}->quote( $self->{ id } );
    my $author = $self->{DBH}->quote( $self->{ author } );
    my $owner = $self->{DBH}->quote( $self->{ owner } );
    my $title = $self->{DBH}->quote( $self->{ title } );
    my $timestamp = $self->{DBH}->quote( $self->{ timestamp } );
    my $original = $self->{DBH}->quote( $self->{ original } );

    # Build SQL query
    my $query;
    ($query = <<__QUERY_END__>> =~ s/(?:\^\s+)|\s+(?==)|\n/ /gm;
        id = $id,
        author = $author,

```



```

        owner      = $owner,
        title      = $title,
        timestamp  = $timestamp,
        original   = $original
__QUERY_END__
$query = "INSERT INTO document SET $query ON DUPLICATE KEY UPDATE $query";

# Execute query
my $sth = $self->{DBH}->prepare($query);
unless ($sth->execute) {
    $self->{ERROR} = $sth->errstr;
    return;
}

# Save pages
my $total_pages = $self->total_pages;
foreach my $page_num (0..$total_pages-1) {
    # Save pages
    unless ($self->{PAGE}[$page_num]->Save) {
        $self->{ERROR} = 'Failed to save page '
            . ($page_num + 1) . ': '
            . $self->{PAGE}[$page_num]->error_message;
        return;
    }
}

# Delete all pages above current threshold
unless ($self->{DBH}->do(
    "DELETE FROM page WHERE id=$id AND page > $total_pages"
)) {
    $self->{ERROR} = $sth->errstr;
    return;
}

# Final items
$self->{CHANGED} = 0;

return 1;
}
#
#-----
#
sub Delete : locked #()#
{
    my $self = shift;
    my $sth = undef;

    if (defined $self->{id}) {

        # Delete from database
        $self->{DBH} = OAS::DBI::connect() unless $self->{DBH};
        my $id      = $self->{DBH}->quote($self->{id});
        if (!$sth = $self->{DBH}->prepare("DELETE FROM document WHERE id=$id")) {
            $self->{ERROR} = $self->{DBH}->errstr;
            return;
        }
        unless (defined $sth->execute) {
            $self->{ERROR} = $sth->errstr;
            return;
        }

        # Delete from filesystem
        system('/bin/rm', '-Rf', document_id_to_path($self->{id}))
            if is_valid_document_id $self->{id};
    }

    # Clear object contents
    $self->_reset_defaults;
}

```

```

        $self->clear_error;
        $self->{CHANGED} = 0;

        return 1;
    }
#-----
#
#
sub clear_error #()#
{
    my $self = shift;
    $self->{ERROR} = '';
}
#-----

#-----
# PAGE CONTROL METHODS
#
sub total_pages #()#
{
    my $self = shift;
    return 0 unless defined $self->{PAGE};
    return scalar @{$self->{PAGE}};
}
#-----
#
sub is_page #($)#
{
    my $self      = shift;
    my $page_num = $_[0];

    # Verify validity of page to be deleted
    unless (defined $page_num) {
        $self->{ERROR} = 'Missing page number';
        return 0;
    }
    unless ( ( $page_num =~ /\^d+$/ ) and
              ( $page_num > 0           ) and
              ( $page_num <= $self->total_pages)
            ) {
        $self->{ERROR} = 'Invalid page number';
        return 0;
    }
    return 1;
}
#-----
#
#
sub add_page #()#
{
    my $self      = shift;
    my $page_num = $self->total_pages;

    # Add page
    $self->{PAGE}[$page_num] = new OAS::Page;
    unless (defined $self->{PAGE}[$page_num]) {
        $self->{ERROR} = 'Could not create new page';
        return;
    }
    $self->{PAGE}[$page_num]->id($self->id);
    $self->{PAGE}[$page_num]->page($page_num+1);
}

```

```

        # Return new page number
        return $page_num + 1;
    }
#
#-----
#
sub insert_page #($)#
{
    my $self      = shift;
    my $page_num = $_[0];

    # Verify validity of page to be inserted
    return unless $self->is_page($page_num--);

    # Add page
    splice @{$self->{PAGE}}, $page_num, 0, new OAS::Page;
    unless (defined $self->{PAGE}[$page_num]) {
        $self->{ERROR} = 'Could not create new page';
        return;
    }
    $self->{PAGE}[$page_num]->id($self->id);
    $self->{PAGE}[$page_num]->page($page_num+1);

    # Renumber the pages
    foreach my $page_num (0..$self->total_pages-1) {
        $self->{PAGE}[$page_num]->page($page_num+1);
    }

    return $page_num + 1;
}
#
#-----
#
sub delete_page #($)#
{
    my $self      = shift;
    my $page_num = $_[0];

    # Verify validity of page to be deleted
    return unless $self->is_page($page_num--);

    splice @{$self->{PAGE}}, $page_num, 1;

    # Renumber the pages
    foreach my $page_num (0..$self->total_pages-1) {
        $self->{PAGE}[$page_num]->page($page_num+1);
    }

    return 1;
}
#
#-----
#
sub page_number #($)#
{
    my $self      = shift;
    my $page_num = shift;

    # Verify validity of page to be deleted
    return unless $self->is_page($page_num--);

    return $self->{PAGE}[$page_num]->page();
}
#
#-----
#

```

```

sub page_image #($;$)#
{
    my $self      = shift;
    my $page_num = shift;
    my $result    = undef;

    # Verify validity of page
    return unless $self->is_page($page_num--);

    # Process request and bubble-up the error message if it exists
    $result = $self->{PAGE}[$page_num]->image(@_);
    $self->{ERROR} = $self->{PAGE}[$page_num]->error_message unless defined $result;
    return $result;
}
#
#-----
#
sub page_height #($;$)#
{
    my $self      = shift;
    my $page_num = shift;
    my $result    = undef;

    # Verify validity of page
    return unless $self->is_page($page_num--);

    # Process request and bubble-up the error message if it exists
    $result = $self->{PAGE}[$page_num]->height(@_);
    $self->{ERROR} = $self->{PAGE}[$page_num]->error_message unless defined $result;
    return $result;
}
#
#-----
#
sub page_width #($;$)#
{
    my $self      = shift;
    my $page_num = shift;
    my $result    = undef;

    # Verify validity of page
    return unless $self->is_page($page_num--);

    # Process request and bubble-up the error message if it exists
    $result = $self->{PAGE}[$page_num]->width(@_);
    $self->{ERROR} = $self->{PAGE}[$page_num]->error_message unless defined $result;
    return $result;
}
#
#-----

#-----
# DATA ACCESS METHODS
#
sub id
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ id } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
    }
}

```

```

        unless (is_valid_document_id $data) {
            $self->{ERROR} = 'Invalid ID';
            return;
        }
        $self->{ id      } = untaint_document_id $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{id};
}
#
#-----
#
#
sub author
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ author  } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        $data = lc $data;
        unless (is_valid_author $data) {
            $self->{ERROR} = 'Invalid AUTHOR';
            return;
        }
        $self->{ author  } = untaint_author $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{author};
}
#
#-----
#
#
sub owner
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ owner  } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        $data = lc $data;
        unless (is_valid_owner $data) {
            $self->{ERROR} = 'Invalid OWNER';
            return;
        }
        $self->{ owner  } = untaint_owner $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{owner};
}
#
#-----
#
#
sub title
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ title  } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
    }
}

```

```

        unless (is_valid_title $data) {
            $self->{ERROR} = 'Invalid TITLE';
            return;
        }
        $self->{ title } = untaint_title $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{title};
}
#
#-----
#
sub timestamp
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ timestamp } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_timestamp $data) {
            $self->{ERROR} = 'Invalid TIMESTAMP';
            return;
        }
        $self->{ timestamp } = untaint_timestamp $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{timestamp};
}
#
#-----
#
sub original
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ original } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_original $data) {
            $self->{ERROR} = 'Invalid ORIGINAL';
            return;
        }
        $self->{ original } = untaint_original $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{original};
}
#
#-----
#
sub error_message #()#
{
    my $self = shift;
    return $self->{ERROR};
}
#
#-----
1;

```

## B.7. OAS::Image.pm

```
package OAS::Image;
our $VERSION = 1.0.0;
our $DATE    = '03 Aug 2006';

use strict;
use Carp;
use Image::Magick;

#=====
# GLOBALS
#
my %Default = (
    width      => 640,
    height     => 480,
    format     => 'gif',
    color      => 'transparent',
    resolution => 72,
    pen_weight => 2,
    pen_color  => 'black',
    filename   => ''
);
#
#=====

#-----
# OAS::Image Object
#
sub create #()#
{
    my $class = shift;
    $class = ref $class || $class;
    my $self = { @_ };
    bless $self, $class;
    $self->_init;
    return $self;
}
#
#-----
#
sub _init #()#
{
    my $self = shift;
    undef $self->{image};
    $self->{image} = new Image::Magick;
    $self->_size($self->{'width'}, $self->{'height'});
    $self->{image}->ReadImage("xc:$Default{color}");
#    $self->format($Default{'type'});
#    $self->resolution($Default{'resolution'});
    $self->pen_weight($self->{'pen_weight'});
    $self->pen_color($self->{'pen_color'});
    $self->filename($Default{'filename'});

    return $self;
}
#
#-----
#
sub bezier #(@)#
{
    my $self = shift;
    my @point = @_;
```

```

my $points = '';
for (0..$#point) {
    croak 'Invalid parameters for bezier' unless
    (
        ($point[$_] =~ /\d+$/) and
        (
            ($_ % 2)
            ? $point[$_] < $self->{'height'}
            : $point[$_] < $self->{'width'}
        )
    );
    $points .= ($_ % 2) ? "$point[$_] " : "$point[$_],";
}
chop $points;

return $self->{image}->Draw(
    strokewidth => $self->{'pen_weight'},
    antialias   => 'true',
    stroke      => $self->{'pen_color'},
    primitive   => 'bezier',
    points      => $points
);
}
#
#-----
#
sub line #($$$$)#
{
    my $self = shift;
    my ($x1, $y1, $x2, $y2) = @_;
    croak 'Invalid parameters for line' unless
    (
        ($x1 =~ /\d+$/) and ($x1 <= $self->{'width' }) and
        ($y1 =~ /\d+$/) and ($y1 <= $self->{'height' }) and
        ($x2 =~ /\d+$/) and ($x2 <= $self->{'width' }) and
        ($y2 =~ /\d+$/) and ($y2 <= $self->{'height' })
    );

    return $self->{image}->Draw(
        strokewidth => $self->{'pen_weight'},
        antialias   => 'true',
        stroke      => $self->{'pen_color'},
        primitive   => 'line',
        points      => "$x1,$y1 $x2,$y2"
    );
}
#
#-----
#
sub rectangle #($$$$)#
{
    my $self = shift;
    my ($x1, $y1, $x2, $y2) = @_;
    croak 'Invalid parameters for rectangle' unless
    (
        ($x1 =~ /\d+$/) and ($x1 <= $self->{'width'}) and
        ($y1 =~ /\d+$/) and ($y1 <= $self->{'height'}) and
        ($x2 =~ /\d+$/) and ($x2 <= $self->{'width'}) and
        ($y2 =~ /\d+$/) and ($y2 <= $self->{'height'})
    );
}

```



```

return $self->{image}->Draw(
    strokewidth => $self->{'pen_weight'},
    antialias    => 'true',
    stroke       => $self->{'pen_color'},
    primitive    => 'rectangle',
    points       => "$x1,$y1 $x2,$y2"
);
}
#
#-----
#
sub point #($$)#
{
    my $self = shift;
    my ($x1, $y) = @_;
    my $x2      = $x1;
    croak 'Invalid parameters for point' unless
    (
        ($x1 =~ /\d+$/) and ($x1 <= $self->{'width'}) and
        ($y  =~ /\d+$/) and ($y  <= $self->{'height'})
    );
    $x1 -= int($self->{'pen_weight'}/2 - .5);
    $x1 = 0 if $x1 < 0;
    $x2 += int($self->{'pen_weight'}/2);
    $x2 += 1 if $x1 == $x2;
    return $self->{image}->Draw(
        strokewidth => $self->{'pen_weight'},
        antialias    => 'true',
        stroke       => $self->{'pen_color'},
        primitive    => 'line',
        points       => "$x1,$y $x2,$y"
    );
}
#
#-----
#
sub size #($$)#
{
    my $self = shift;
    my $x    = (defined $_[0]) ? $_[0] : $Default{'width'};
    my $y    = (defined $_[1]) ? $_[1] : $Default{'height'};
    croak 'Invalid parameters for size' unless
    (
        ($x =~ /\d+$/) and ($y =~ /\d+$/)
    );
    $self->{'width'} = $x;
    $self->{'height'} = $y;
    $self->_init;
    return ($x, $y);
}
#
#-----
#
sub _size #($$)#
{
    my $self = shift;
    my $x    = (defined $_[0]) ? $_[0] : $Default{'width'};
    my $y    = (defined $_[1]) ? $_[1] : $Default{'height'};
    croak 'Invalid parameters for _size' unless
    (
        ($x =~ /\d+$/) and ($y =~ /\d+$/)
    );
    $self->{'width'} = $x;
    $self->{'height'} = $y;
    my $size = $x.'x'.$y;
    my $result = $self->{image}->Set(size=>$size);
}

```

```

        croak $result if $result;
        return ($x, $y);
    }
#-----
#
sub format #($)#
{
    my $self = shift;
    my $format = (defined $_[0]) ? $_[0] : $Default{'format'};
    $self->{'format'} = $format;
#    my $result = $self->{image}->Set(format=>$format);
#    croak $result if $result;
    return $format;
}
#-----
#
sub filename #($)#
{
    my $self = shift;
    my $filename = $_[0];
    croak ('Invalid parameter for filename') unless
    (
        (defined $filename) and ((length $filename) or ('' eq $filename))
    );
    return $self->{'filename'} = $filename;
}
#-----
#
sub write #()#
{
    my $self = shift;
    my $result = $self->{image}->Write(
        filename => $self->{'filename'}
    );
    croak $result if $result;
    return not $result;
}
#-----
#
sub pen_color #($)#
{
    my $self = shift;
    my $color = (defined $_[0]) ? $_[0] : $Default{'pen_color'};
    return $self->{'pen_color'} = $color;
}
#-----
#
sub pen_weight #($)#
{
    my $self = shift;
    my $weight = (defined $_[0]) ? $_[0] : $Default{'pen_weight'};
    return $self->{'pen_weight'} = $weight;
}
#-----
1;

```

## B.8. OAS::Menu.pm

```
package OAS::Menu;
require 5.8.5;
our $VERSION = 1.0.0;
our $DATE    = '10 Aug 2006';

use strict;
use warnings;

use Carp;
use OAS::System qw(path_to_url);
use OAS::Annotation qw(annotation_owner annotation_list);
use OAS::Auth qw(:acl current_user is_admin);
use OAS::Document qw(is_valid_document_id document_id_to_path);
use OAS::Person;
use OAS::Service;

use Exporter ();
our @ISA      = qw(Exporter);
our @EXPORT  = qw(
    MENUBAR_HEIGHT
    INFOBAR_HEIGHT
    SIDEBAR_WIDTH
    BORDER_SIZE
    menubar
);
our @EXPORT_OK = qw(
    MENUBAR_HEIGHT
    INFOBAR_HEIGHT
    SIDEBAR_WIDTH
    BORDER_SIZE
    menubar
    infobar
    sidebar
    annotation_text
);
our %EXPORT_TAGS = (
    all => [qw( MENUBAR_HEIGHT INFOBAR_HEIGHT
                SIDEBAR_WIDTH  BORDER_SIZE
                menubar        infobar
                sidebar        annotation_text )]
);

#####
# CONSTANTS
#
use constant ENABLED => 1;
use constant DISABLED => 0;
#-----
use constant MENUBAR_HEIGHT => 23;
use constant INFOBAR_HEIGHT => 24;
use constant SIDEBAR_WIDTH => 160;
use constant BORDER_SIZE => 10;
#
#####

#-----
# FUNCTIONS
#
```

```

sub menubar (;$)
{
    my $doc      = ( defined $_[0]      ) ? $_[0] : undef;
    my $page     = ( defined $_[1]     ) ? $_[1] : '';
    $page       = ( $page =~ /^\/d+$/ ) ? $page : undef;
    my $user     = current_user;
    my $is_owner = 0;
    my $is_admin = is_admin $user;
    my $html     = '';
    my $width    = 1;
    my %acl      = ();
    if (defined $doc) {
        $is_owner = ($user eq $doc->owner);
        # Setup Permission Hash
        map { $acl{$_}++ } list_access $user, $doc->id;
        $width = $doc->page_width($page) + SIDEBAR_WIDTH + (2 * BORDER_SIZE);
    }
    my $services = OAS::Service::list_services;

    $html = &_menu_headings($user, $width)
        . &_file_menu($user, $doc, $page, $is_owner, $is_admin, $services, %acl)
        . &_edit_menu($user, $doc, $page, $is_owner, $is_admin, $services, %acl)
        . &_view_menu($user, $doc, $page, $is_owner, $is_admin, $services, %acl)
        . &_help_menu
        . &_menu_javascript($doc, $page);

    $html =~ s/\t//g;
    $html =~ s/\n+/\n/g;
    return $html;
};
#
#-----
#
sub _menu_javascript ($$)
{
    my ($doc, $page) = @_;
    my $width = (defined $doc)
        ? $doc->page_width($page) + SIDEBAR_WIDTH + (2 * BORDER_SIZE) : 0;
    my $height = MENUBAR_HEIGHT . 'px';
    my $html = qq|
        <script type="text/javascript" src="/js/menu.js"></script>
        <script type="text/javascript" src="/js/div_manager.js"></script>
        <script type="text/javascript">
            if (window_width() > $width) {
                div_width('menuSystem', window_width() + 'px');
            } else {
                div_width('menuSystem', '$width' + 'px');
            }
            var menu_names = new Array(4);
            var fileMenu = 0;
            var editMenu = 1;
            var viewMenu = 2;
            var helpMenu = 3;
            menu_names[fileMenu] = 'fileMenu';
            menu_names[editMenu] = 'editMenu';
            menu_names[viewMenu] = 'viewMenu';
            menu_names[helpMenu] = 'helpMenu';
            create_menu(menu_names);
        </script>
    |;

    return $html;
}
#
#-----
#
sub _menu_headings ($$)
{

```

```

my ($user, $width) = @_ ;
$width = (defined $width) ? $width : 1;

my $person = new OAS::Person;
$person->Load($user);
my $name = '<strong>'
    . $person->first_name . '&nbsp;.' . $person->last_name
    . "&nbsp;[<em>$user</em>]</strong>";

qq|
<div id="menuSystem"><table border="0" cellspacing="0" cellpadding="0">
  <tbody>
    <tr>
      <td align="left"><a
        class="menuHead" href=""
        onClick="return toggle_menu(fileMenu)">File</a><a
        class="menuHead" href=""
        onClick="return toggle_menu(editMenu)">Edit</a><a
        class="menuHead" href=""
        onClick="return toggle_menu(viewMenu)">View</a><a
        class="menuHead" href=""
        onClick="return toggle_menu(helpMenu)">Help</a>
      </td>
      <td align="left"><span class="userLoggedIn"
        >Current&nbsp;User:&nbsp;&nbsp;$name</span></td>
    </tr>
    <tr>
      <td colspan="2"></td>
    </tr>
  </tbody>
</table>
</div>

|;
}
#
#-----
#
sub _file_menu ($$$$$%)
{
    my ($user, $doc, $page, $is_owner, $is_admin, $services, %acl) = @_ ;
    my $doc_id = (defined $doc) ? $doc->id : '';
    my $service = '';

    # File Menu
    my $html = qq|<div id="fileMenu" class="submenuSystem">\n|;
    $service = $services->{CreateDocument}{url};
    $html .= &_add_menu_item(
        ENABLED,
        'New',
        "return helper_app('$service', 450, 250, fileMenu)"
    );
    $service = $services->{OpenDocument}{url};
    $html .= &_add_menu_item(
        ENABLED,
        'Open',
        "return helper_app('$service?ob=timestamp%20DESC', 450, 250, fileMenu)"
    );
    $service = $services->{DeleteDocument}{url};
    $html .= &_add_menu_item(
        ( defined($doc) and
          ($is_owner or $is_admin or exists($acl{ &ACL_DELETE_ACCESS }))),
        'Delete',
        "return helper_app('$service?id=$doc_id', 400, 75, fileMenu)"
    );
    $service = $OAS::System::Conf->www_url . 'd/';
}

```

```

$html .= &_add_menu_item(
    (defined $doc),
    'Close',
    "top.location.href='$service'; return false"
);
$service = $services->{Logout}{url};
$html .= &_add_menu_item(
    ENABLED,
    'Exit/Logout',
    "top.location.href='$service'; return false"
);
$html .= "</div>\n";

return $html;
}
#
#-----
#
sub _edit_menu ($$$$$%)
{
    my ($user, $doc, $page, $is_owner, $is_admin, $services, %acl) = @_;
    my $doc_id = (defined $doc) ? $doc->id : '';
    my $service = '';

    my $html = qq|<div id="editMenu" class="submenuSystem">\n|;
    $service = $services->{DocumentPermissions}{url};
    $html .= &_add_menu_item(
        (defined($doc) and ($is_owner or $is_admin)),
        'Permissions',
        "top.location.href='$service?id=$doc_id'; return false"
    );
    $service = $services->{UserProfile}{url};
    $html .= &_add_menu_item(
        ENABLED,
        'User Profile',
        "top.location.href='$service?id=$user'; return false"
    );
    $service = $services->{GroupManager}{url};
    $html .= &_add_menu_item(
        ENABLED,
        'Groups',
        "top.location.href='$service'; return false"
    );
    if (is_admin $user) {
        $html .= " <hr>\n";
        $service = $services->{AdminUser}{url};
        $html .= &_add_menu_item(
            ENABLED,
            'Users',
            "top.location.href='$service'; return false"
        );
        $service = $services->{AdminConverter}{url};
        $html .= &_add_menu_item(
            ENABLED,
            'File Converters',
            "top.location.href='$service'; return false"
        );
        $service = $services->{AdminService}{url};
        $html .= &_add_menu_item(
            ENABLED,
            'Web Services',
            "top.location.href='$service'; return false"
        );
    }
    $html .= "</div>\n";

    return $html;
}

```

```

#
#-----
#
sub _view_menu ($$$$$%)
{
    my ($user, $doc, $page, $is_owner, $is_admin, $services, %acl) = @_;
    my $service = '';
    my $html    = qq|<div id="viewMenu" class="submenuSystem">\n|;

    if (defined $doc) {
        my $id = $doc->id;

        $service = $services->{ShowDocument}{url};
        $html    .= &_add_menu_item(
            exists($acl{&ACL_READ_ACCESS}),
            'Display',
            "top.location.href='$service?id=$id&p=$page'; return false"
        );
        $service = $services->{ReadDocument}{url};
        $html    .= &_add_menu_item(
            ( exists($acl{ &ACL_READ_ACCESS          }) and
              exists($acl{ &ACL_READ_ANNOTATION_ACCESS }) ),
            'Read',
            "top.location.href='$service?id=$id&p=$page'; return false;"
        );
        $service = $services->{AnnotateDocument}{url};
        $html    .= &_add_menu_item(
            ( exists($acl{ &ACL_READ_ACCESS          }) and
              exists($acl{ &ACL_ANNOTATE_ACCESS }) ),
            'Annotate',
            "top.location.href='$service?id=$id&p=$page'; return false"
        );
        $service = path_to_url(document_id_to_path($id) . $doc->original);
        $html    .= &_add_menu_item(
            exists($acl{&ACL_READ_ACCESS}),
            'Original',
            "top.location.href='$service'; return false;"
        );
    } else {
        $html .= qq| <a class="menuItem" href="">&nbsp;</a>\n|;
    }

    $html .= "</div>\n";

    return $html;
}
#
#-----
#
sub _help_menu ()
{
    my $service = '';
    my $html    = qq|<div id="helpMenu" class="submenuSystem">\n|;

    $html .= &_add_menu_item(
        ENABLED,
        'Help',
        ''
    );

    $service = OAS::Service::service_url 'FileFormats';
    $html    .= &_add_menu_item(
        ENABLED,
        'File Formats',
        "return helper_app('$service', 450, 250, helpMenu)"
    );
}

```

```

$html .= &_add_menu_item(
    ENABLED,
    'About',
    ''
);
$html .= "</div>\n";

return $html;
}
#
#-----
#
sub _add_menu_item ($$;$)
{
    my $status = (defined $_[0]) ? 1 && $_[0] : DISABLED;
    my $tag    = (defined $_[1]) ? $_[1] : '';
    $tag      =~ s/ /&nbsp;/g;
    my $onClick = (defined $_[2]) ? $_[2] : 'return false';
    my $class   = ($status) ? 'menuItem' : 'menuDisabled';
    $onClick    = ($status) ? $onClick : 'return false';
    my $url     = (exists $ENV{REQUEST_URI})
        ? $ENV{REQUEST_URI}
        : '';

    return qq| <a class="$class" href="$url" onclick="$onClick;">$tag</a>\n|;
}
#
#-----
#
sub infobar ($$)
{
    my ($doc, $page) = @_;
    return unless defined $doc;
    $page = 1 unless $page =~ /\d+$/;

    my $user = current_user;
    my $id   = $doc->id;
    my $prev = (0 == $page - 1) ? 1 : $page - 1;
    my $last = $doc->total_pages;
    my $next = ($last < $page + 1) ? $last : $page + 1;
    my $width = $doc->page_width($page) + SIDEBAR_WIDTH + (2 * BORDER_SIZE);
    my $mb_h  = MENUBAR_HEIGHT;
    my $ib_h  = INFOBAR_HEIGHT . 'px';
    my $person = new OAS::Person;
    $person->Load($doc->author);
    my $title = $doc->title;
    my $author = $person->first_name . ' '
        . $person->last_name . ' ['
        . $person->id . ']' ;

    my $html = qq|
        <div id="menuInfobar">
            <table border="0" cellspacing="0" cellpadding="0">
                <tbody>
                    <tr>
                        <td>
                            <table border="0" cellspacing="0" cellpadding="0" width="770">
                                <tbody>
                                    <tr>
                                        <td valign="top">
                                            <span class="infoAuthor">$author</span>
                                        </td>
                                        <td valign="top" align="center">
                                            <span class="infoTitle">$title</span>
                                        </td>
                                        <td valign="top" align="right">
                                            <table border="0" cellspacing="1" cellpadding="0">
                                                <tbody>

```



```

        <tr>
| . &_nav_button($id, $page, 1, '<<', 'navButtonHome')
. &_nav_button($id, $page, $prev, '<', 'navButtonPgUp')
. qq|
        <td class="pageNavToolbar">
        <form name="navButtonP" method="GET">
        <input type="HIDDEN" name="id" value="$id" />
        <select name="p" onChange="this.form.submit();">
|;
foreach my $p (1..$last) {
    my $selected = ($p == $page) ? ' SELECTED' : '';
    $html .= qq|                <option value="$p"$selected>$p</option>\n|;
}
$html .= qq|
        </select></td>
        </form>
| . &_nav_button($id, $page, $next, '>', 'navButtonPgDown')
. &_nav_button($id, $page, $last, '>>', 'navButtonEnd')
. qq|
        </tr>
        </tbody>
        </table>
    </td>
    </tr>
    </tbody>
    </table>
    </td>
    </tr>
    <tr><td></td></tr>
    </tbody>
    </table>
</div>
<script type="text/javascript">
    if (window_width() > $width) {
        div_width('menuInfobar', window_width() + 'px');
    } else {
        div_width('menuInfobar', '$width' + 'px');
    };
    document.onkeydown = keyDown;
    function keyDown(e) {
        var keynum = (window.event) ? e.keyCode : e.which;
        switch (keynum) {
            case 33:
                document.navButtonPgUp.submit();
                return false;
                break;
            case 34:
                document.navButtonPgDown.submit();
                return false;
                break;
            case 35:
                document.navButtonEnd.submit();
                return false;
                break;
            case 36:
                document.navButtonHome.submit();
                return false;
                break;
        }
    }
</script>
|;
$html =~ s/\t//g;
$html =~ s/\n+/\n/g;
return $html;
}
#

```

```

#-----
#
sub _nav_button ($$$$)
{
    my ($id, $page, $new_page, $tag, $name) = @_;
    my $class = ($page == $new_page) ? 'pageNavDisabled' : 'pageNavToolbar';
    my $onClick = ($page == $new_page) ? 'onClick="return false"; ' : '';
    qq|
        <td class="pageNavToolbar">
        <form name="$name" id="$name" method="GET">
            <input type="HIDDEN" name="id" value="$id" />
            <input type="HIDDEN" name="p" value="$new_page" />
            <input class="$class" type="SUBMIT" value="$tag" $onClick/>
        </td>
        </form>
    |;
}
#-----
#
sub sidebar ($$;$@)
{
    my ($doc, $page, $annotate, @annotations) = @_;
    return unless defined $doc;
    $page = 1 unless $page =~ /\^d+$/;
    $annotate = 0 unless ((defined $annotate) and $annotate);
    my $user = current_user;
    my $id = $doc->id;
    my $height = $doc->page_height($page) + 20;
    my $width = SIDEBAR_WIDTH;
    my $table_w = $width - 2;
    my $bars_h = MENUBAR_HEIGHT + INFOBAR_HEIGHT;
    my $services = OAS::Service::list_services;

    # Build Sidebar
    my $html = qq|
        <!-- SIDEBAR CODE START -->
        <script type="text/javascript" src="/js/cookie.js"></script>
        <script type="text/javascript" src="/js/annotation.js"></script>
        <script type="text/javascript">
            read_cookies();
            function toggle_annotation(annotation_id)
            {
                var checkbox = document.getElementById('af' + annotation_id.substr(1));
                checkbox.checked ? div_show(annotation_id) : div_hide(annotation_id);
            }
        </script>
        <div id="menuSidebar">
        <table border="0" cellspacing="0" cellpadding="0">
        <tbody>
        <tr>
        <td></td>
        <td align="center" valign="top">
        <table border="0" cellspacing="0" cellpadding="0">
        <tbody>
    |;
    if ($annotate) {
        $html .= qq|
            <tr>
            <td>
        |;
        $html .= &_annotation_tools($id, $page, $table_w);
        $html .= qq|
            </td>
        </tr>
    |;
    }
}

```

```

$html .= qq|
    <tr>
        <td align="center" valign="top">
            <table border="0" cellspacing="1" cellpadding="1" width="$table_w">
                <tbody>
                    <form name="annotations" onSubmit="return false">
                        <tr><td colspan="3"><hr /></td></tr>
                        <tr>
                            <td class="sideControlTitle" colspan="3">
                                Annotations
                            </td>
                        </tr>
                    </tr>
                </tbody>
            </table>
        </td>
    </tr>
|;

# Add Annotations
my $is_moderator = has_moderate_access $user, $id;
my $annotation = new OAS::Annotation;
foreach my $a_id (sort @annotations) {

    $annotation->Load($a_id);
    my $is_owner = ($user eq $annotation->owner);
    my $a_title = ($is_owner or $is_moderator)
        ? $a_id . ' [<em>' . $annotation->author . '</em>]</a>'
        : $a_id;
    my $service = '';

    # Annotation Header
    my $img = $annotation->image;
    $service = $services->{EditAnnotation}{url};
    my $link = (($is_owner or $is_moderator)
        and not (defined($img)
        and length($img)))
        ? &_add_menu_item(
            ENABLED,
            $a_title,
            "return helper_app('$service?id=$a_id', 625, 275, fileMenu)"
        )
        : $a_title;
    $link =~ s/class="\w+?"/class="sideControlAnnotation"/;
    $link =~ s/&nbsp;/ /g;
    $html .= qq|
        <tr>
            <td class="sideControlCenter">
                <input type="CHECKBOX" id="af$a_id" name="af$a_id" CHECKED
                    onClick="toggle_annotation('$a_id')"
                    onChange="toggle_annotation('$a_id')" />
            </td>
            <td class="sideControlAnnotation">$link</td>
            <td class="sideControlCenter">
        </tr>
    |;

    # Annotation Delete Button
    if ($annotate and ($is_owner or $is_moderator)) {
        $service = $services->{DeleteAnnotation}{url};
        $html .= qq|
            <input class="pageNavToolbar" type="BUTTON"
                name="da$a_id" value="X" onClick=
                "return helper_app('$service?id=$a_id', 400, 75, fileMenu);"
            >
        |;
    }

    # Annotation Footer
    $html .= qq|
        </td>
    </tr>
|;
}

```

```

# Finish Sidebar
$html .= qq|
    <tr><td colspan="3"><hr /></td></tr>
    </form>
    </tbody>
    </table>
    </td>
    </tr>
    </tbody>
    </table>
    </td>
    </tr>
    </tbody>
    </table>
    </div>
    <script type="text/javascript">
    if (window_height() > ($height + $bars_h)) {
        div_height('menuSidebar', (window_height() - $bars_h) + 'px');
    } else {
        div_height('menuSidebar', '$height' + 'px');
    }
    </script>
    <!-- SIDEBAR CODE END -->
|;
$html =~ s/\t//g;
$html =~ s/\n+/\n/g;
return $html;
}
#
#-----
#
sub _annotation_tools ($$$)
{
    my ($id, $page, $table_width) = @_;
    my $service = OAS::Service::service_url 'AddAnnotation';

    qq|
        <table border="0" cellspacing="0" cellpadding="0" width="$table_width">
        <tbody>
        <form name="settings" onSubmit="return false">
        <tr><td><hr /></td></tr>
        <tr>
        <td class="sideControlTitle">
            Annotation Controls
        </td>
        </tr>
        <tr><td><hr /></td></tr>
        <tr>
        <td class="sideControlHeading">Draw Mode</td>
        </tr>
        <tr>
        <td align="center">
        <table border="0" cellspacing="1" cellpadding="1">
        <tbody>
        <tr>
        <td class="sideControlCenter">
            <input type="radio" name="mode" value="DRAW"
            onClick="set_mode(DRAW_MODE)" CHECKED />
            Draw
        </td>
        <td class="sideControlCenter">
            <input type="radio" name="mode" value="TEXT"
            onClick="set_mode(TEXT_MODE)" />
            Text
        </td>
        </tr>
        </tbody>
        </table>
        </td>
        </tr>
        </tbody>
        </table>

```

```

    </td>
  </tr>
<tr><td><hr /></td></tr>
<tr>
  <td class="sideControlHeading">Pen</td>
</tr>
<tr>
  <td align="center">
    <table border="0" cellspacing="1" cellpadding="1">
      <tbody>
        <tr>
          <td class="sideControlRight">
            Style:
          </td>
          <td class="sideControlLeft">
            <select name="pen_size" onChange="set_pen()">
              <option value="1" >Hairline </option>
              <option value="2" SELECTED>Ball-Point</option>
              <option value="4" >Marker </option>
              <option value="8" >Brush </option>
            </select>
          </td>
        </tr>
        <tr>
          <td class="sideControlRight">
            Color:
          </td>
          <td class="sideControlLeft">
            <select name="pen_color" onChange="set_pen()">
              <option value="black" >Black </option>
              <option value="blue" SELECTED>Blue </option>
              <option value="gray" >Gray </option>
              <option value="green" >Green </option>
              <option value="orange" >Orange</option>
              <option value="red" >Red </option>
              <option value="yellow" >Yellow</option>
              <option value="white" >White </option>
            </select>
          </td>
        </tr>
      </tbody>
    </table>
  </td>
<tr><td><hr /></td></tr>
<tr>
  <td class="sideControlHeading">Tracer</td>
</tr>
<tr>
  <td align="center">
    <table border="0" cellspacing="1" cellpadding="1">
      <tbody>
        <tr>
          <td class="sideControlRight">
            Use Tracer:
          </td>
          <td class="sideControlLeft">
            <input type="CHECKBOX" name="use_tracer" CHECKED
              onchange="set_cookies();" />
          </td>
        </tr>
        <tr>
          <td class="sideControlRight">
            Speed:
          </td>
          <td class="sideControlLeft">
            <select name="trace_speed">
              <option value="10" >Fast </option>
              <option value="25" SELECTED>Normal</option>
            </select>
          </td>
        </tr>
      </tbody>
    </table>
  </td>
</tr>

```

```

        <option value="100"          >Slow </option>
    </select>
</td>
</tr>
<tr>
<td class="sideControlCenter" colspan="2" align="center">
    <input class="pageNavToolBar" type="SUBMIT" value="Clear Tracer"
        onClick="clear_path(); return false;" />
</td>
</tr>
</tbody>
</table>
</td>
</tr>
</form>
<tr><td><hr /></td></tr>
<form name="add_annotation" method="POST"
action="$service" onSubmit="return add_annotation_form_submit();">
<input type="HIDDEN" name="id" value="$id" />
<input type="HIDDEN" name="p" value="$page" />
<input type="HIDDEN" name="mode" value="" />
<input type="HIDDEN" name="x_min" value="" />
<input type="HIDDEN" name="y_min" value="" />
<input type="HIDDEN" name="x_max" value="" />
<input type="HIDDEN" name="y_max" value="" />
<input type="HIDDEN" name="x_data" value="" />
<input type="HIDDEN" name="y_data" value="" />
<input type="HIDDEN" name="x_offset" value="" />
<input type="HIDDEN" name="y_offset" value="" />
<input type="HIDDEN" name="text" value="" />
<input type="HIDDEN" name="font" value="" />
<input type="HIDDEN" name="font_pitch" value="" />
<input type="HIDDEN" name="font_color" value="" />
<input type="HIDDEN" name="background" value="" />
<input type="HIDDEN" name="referrer" value="" />
<tr>
<td align="center">
<table border="0" cellspacing="1" cellpadding="1">
<tbody>
<tr>
<td class="sideControlRight">
<input class="pageNavToolBar" type="SUBMIT"
value="Create Annotation" />
</td>
</tr>
</tbody>
</table>
</td>
</tr>
</form>
</tbody>
</table>
|;
}
#
#-----
#
sub annotation_text ($$;$$$)
{
    my ($doc, $page, $annotation, $style, $code) = @_;
    return unless defined $doc;

    # Setup variables
    $page = 1 unless $page =~ /\d+$/;
    my $x_pos = (SIDEBAR_WIDTH + BORDER_SIZE + 1) . 'px';
    my $y_pos = (MENUBAR_HEIGHT + INFOBAR_HEIGHT + BORDER_SIZE + 1) . 'px';

```

```

$style = (defined $style)
? $style
: 'position:absolute;'
. 'background:#D0D0D0;'
. "top:$y_pos;"
. "left:$x_pos;"
. 'width:600px;'
. 'height:250px;'
. 'border-style:outset;'
. 'z-index:1006;'
. 'visibility:hidden;';

$code = (defined $code)
? $code
: qq|div_hide('annotationText'); clear_path(); blocking=false; return true;|;

# Setup settings
my %settings = (
    font      => 'sans-serif',
    pitch     => 12,
    color     => 'black',
    background => 'yellow',
);
if (defined $annotation) {
    $settings{'font'} = $annotation->font
        if defined $annotation->font;
    $settings{'pitch'} = $annotation->pitch
        if defined $annotation->pitch;
    $settings{'color'} = $annotation->color
        if defined $annotation->color;
    $settings{'background'} = $annotation->background
        if defined $annotation->background;
}

# Setup Content
my $content = '';
if ((defined $annotation) and (open ATEXT, $annotation->text)) {
    local $/ = undef;
    $content = <ATEXT>;
    close ATEXT;
}

# Build HTML
my $html = qq|
<!-- Start Annotation Text Input Screen -->
<div id="annotationText" style="$style">
<table border="0" cellspacing="0" cellpadding="2" width="100%">
<form name="annotation_text_control" onSubmit="return false;">
<tbody>
<tr>
<td align="center">
<table border="0" cellspacing="0" cellpadding="1">
<tbody>
<tr>
<td class="annotationTextControl" align="right">Font:</td>
<td align="left">
<select name="font" onChange="set_pen();">
|;
# Setup Fonts Types
foreach my $font (
    'serif',
    'sans-serif',
    'monospace',
    'times,serif',
    'arial,sans-serif',
    'courier,monospace'
) {
    my $selected = ($font eq $settings{'font'}) ? ' SELECTED' : '';
    my $name     = ucfirst $font;

```

```

        $name      =~ s/^(.+?),.+$/\1/;
        $name      =~ s/-(\w)\/'-'.uc($1)/ge;

        $html .= qq|          <option value="$font"$selected>$name</option>\n|;
    }
    $html .= qq|
        </select>
        <td class="annotationTextControl" align="right">Size:</td>
        <td align="left">
            <select name="pitch" onChange="set_pen();" >
|;
# Setup Font Pitch
foreach my $pitch qw(8 9 10 11 12 14 16 18 20 22 24 26 28 36 48 72) {
    my $selected = ($pitch == $settings{'pitch'}) ? ' SELECTED' : '';

        $html .= qq|          <option value="$pitch"$selected>$pitch</option>\n|;
    }
    $html .= qq|
        </select>
        <td class="annotationTextControl" align="right">Color:</td>
        <td align="left">
            <select name="color" onChange="set_pen();" >
|;
# Setup Font Color
foreach my $color qw(black blue red green yellow orange gray white) {
    my $selected = ($color eq $settings{'color'}) ? ' SELECTED' : '';
    my $name      = ucfirst $color;

        $html .= qq|          <option value="$color"$selected>$name</option>\n|;
    }
    $html .= qq|
        </select>
        </td>
        <td class="annotationTextControl" align="right">Background:</td>
        <td align="left">
            <select name="background" onChange="set_pen();" >
|;
# Setup Background Color
foreach my $background qw(black blue red green yellow orange gray white) {
    my $selected = ($background eq $settings{'background'}) ? ' SELECTED' : '';
    my $name      = ucfirst $background;

        $html .= qq|          <option
                                value="$background"$selected>$name</option>\n|;
    }
    $html .= qq|
        </select>
        </td>
        </tr>
        </tbody>
        </table>
        </td>
        </tr>
        <tr>
        <td align="center">
            <textarea name="text" cols="60" rows="10"
                wrap="virtual">$content</textarea>
        </td>
        </tr>
        <tr>
        <td align="center">
            <table border="0" cellspacing="0" cellpadding="1">
                <tbody>
                    <tr>
                        <td><input type="SUBMIT" value="Create Annotation"
                            onClick="add_annotation_text_submit(); return true; " />
                        </td>
                        <td><input type="RESET" value="Cancel"

```



```

                onClick="$code" />
            </td>
        </tr>
    </tbody>
</table>
</td>
</tr>
</tbody>
</form>
</table>
</div>
<!-- End Annotation Text Input Screen -->
|;
$html =~ s/\t//g;
$html =~ s/\n+/\n/g;
return $html;
}
#
#-----
1;

```

## B.9. OAS::Page.pm

```

package OAS::Page;
require 5.6.0;
our $VERSION = 1.0.0;
our $DATE    = '03 Aug 2006';

use strict;
use warnings;

use Carp;
use Thread;
use OAS::DBI;
use OAS::Document qw(is_valid_document_id untaint_document_id);
use OAS::Utils    qw(untaint);

use Exporter;
our @ISA = qw(Exporter);
our @EXPORT = qw();
our @EXPORT_OK = qw(
    is_valid_page_id is_valid_page_num is_valid_image
    is_valid_height is_valid_width
    untaint_page_id  untaint_page_num  untaint_image
    untaint_height  untaint_width
);
our %EXPORT_TAGS = (
    check => [qw( is_valid_page_id is_valid_page_num is_valid_image
                  is_valid_height is_valid_width
                  untaint_page_id  untaint_page_num  untaint_image
                  untaint_height  untaint_width
                  )],
    untaint => [qw( untaint_page_id  untaint_page_num  untaint_image
                   untaint_height  untaint_width
                   )]
);

#=====
# EVENT HANDLERS
#
sub DESTROY
{
    # Disconnect from database on exit if needed
    my $self = shift;

```

```

OAS::DBI::disconnect() if (defined $self->{DBH});
$self->SUPER::DESTROY if $self->can("SUPER::DESTROY");
}
#
=====

#-----
# FUNCTIONS
#
sub is_valid_id ($)
{
    return OAS::Document::is_valid_document_id($_[0]);
}
#
#-----
#
sub is_valid_page_id ($)
{
    return is_valid_id $_[0];
}
#
#-----
#
sub is_valid_page_num ($)
{
    local $_ = $_[0] || '';
    return /^ \d+ $/x;
}
#
#-----
#
sub is_valid_image ($)
{
    local $_ = $_[0] || '';
    return /^ [\w\-\.\300-\366\370-\377\/]? $/x;
}
#
#-----
#
sub is_valid_height ($)
{
    local $_ = $_[0] || '';
    return /^ \d+ $/x;
}
#
#-----
#
sub is_valid_width ($)
{
    local $_ = $_[0] || '';
    return /^ \d+ $/x;
}
#
#-----
#
sub untaint_id ($)
{
    return OAS::Document::untaint_document_id($_[0]);
}
#
#-----
#
sub untaint_page_id ($)
{
    return untaint_id $_[0];
}

```

```

#-----
#
sub untaint_page_num ($)
{
    return (is_valid_page_num $_[0]) ? untaint $_[0] : undef;
}
#-----
#
sub untaint_image ($)
{
    return (is_valid_image $_[0]) ? untaint $_[0] : undef;
}
#-----
#
sub untaint_height ($)
{
    return (is_valid_height $_[0] ) ? untaint $_[0] : undef;
}
#-----
#
sub untaint_width ($)
{
    return (is_valid_width $_[0] ) ? untaint $_[0] : undef;
}
#-----

#-----
# CONSTRUCTORS & INITIALIZERS
#
sub new #()#
{
    my $class = shift;
    $class = ref $class || $class;
    my $self = bless {}, $class;
    $self->_reset_defaults;
    $self->clear_error;
    $self->{ CHANGED } = 1;
    $self->{ DBH      } = undef;
    return $self;
}
#-----
#
sub _reset_defaults ()
{
    my $self = shift;
    foreach (qw(id page image height width)) {
        $self->{$_} = undef;
    }
}
#-----

#-----
# DATA CONTROL METHODS
#

```

```

sub Load : locked method #($$)#
{
    my $self = shift;
    my $id   = $_[0];
    my $page = $_[1];
    my $href = undef;
    my $key  = undef;

    # Verify parameters
    if ( not defined $id ) { croak 'Missing Page ID for load' ; }
    elsif ( not defined $page ) { croak 'Missing Page Number for load'; }
    elsif ( not is_valid_id $id ) { croak 'Invalid Page ID for load' ; }
    elsif ( not is_valid_page $page ) { croak 'Invalid Page Number for load'; }

    # Connect to database
    $self->{DBH} = OAS::DBI::connect unless $self->{DBH};

    # Query database for needed information
    unless ($href = $self->{DBH}->selectrow_hashref(
        "SELECT * FROM page WHERE id='$id' AND page=$page"
    )) {
        $self->{ERROR} = $self->{DBH}->errstr;
        return;
    }

    # Reset object values to default
    $self->_reset_defaults;
    $self->clear_error;
    $self->{CHANGED} = 0;

    # Store the results in the object
    foreach $key qw/id page image height width/ {
        $self->{$key}($href->{$key}) if defined $href->{$key};
    }

    return 1;
}
#-----
#
#
sub Save : locked #()#
{
    my $self = shift;
    my $key  = undef;

    # Check for forced save
    $self->{CHANGED} = 1 if $_[0];

    # Verify needed columns
    if ( not $self->{id} ) {
        $self->{ERROR} = 'Missing ID for save'; return;
    } elsif ( not $self->{page} ) {
        $self->{ERROR} = 'Missing PAGE for save'; return;
    } elsif ( not $self->{image} ) {
        $self->{ERROR} = 'Missing IMAGE for save'; return;
    } elsif ( not $self->{height} ) {
        $self->{ERROR} = 'Missing HEIGHT for save'; return;
    } elsif ( not $self->{width} ) {
        $self->{ERROR} = 'Missing WIDTH for save'; return;
    }
    return SAVE_NOT_REQUIRED unless $self->{CHANGED};

    # Connect to database
    $self->{DBH} = OAS::DBI::connect() unless $self->{DBH};

    # Gather & prepare data for query
    my $id   = $self->{DBH}->quote( $self->{ id   } );
    my $image = $self->{DBH}->quote( $self->{ image } );

```

```

my $name = $self->{DBH}->quote( $self->{ name } );
my $page = ( $self->{ page } ) ? int $self->{ page } : 'NULL';
my $height = ( $self->{ height } ) ? int $self->{ height } : 'NULL';
my $width = ( $self->{ width } ) ? int $self->{ width } : 'NULL';

# Build SQL query
my $query;
($query = <<__QUERY_END__>> =~ s/(?:\^\\s+)|\\s+(?==)|\\n/ /gm;
    id = $id,
    page = $page,
    image = $image,
    height = $height,
    width = $width
__QUERY_END__
)
$query = "INSERT INTO page SET $query ON DUPLICATE KEY UPDATE $query";

# Execute query
my $sth = $self->{DBH}->prepare($query);
unless ($sth->execute) {
    $self->{ERROR} = $sth->errstr;
    return;
}

# Final items
$self->{CHANGED} = 0;

return 1;
}
#
#-----
#
sub Delete : locked #()#
{
    my $self = shift;
    my $sth = undef;

    # Delete from database
    if (defined $self->{id}) {
        $self->{DBH} = OAS::DBI::connect() unless $self->{DBH};
        if (!$sth = $self->{DBH}->prepare(
            "DELETE FROM page WHERE id=$self->{id} AND page=$self->{page}"
        )) {
            $self->{ERROR} = $self->{DBH}->errstr;
            return;
        }
        unless (defined $sth->execute) {
            $self->{ERROR} = $sth->errstr;
            return;
        }
    }

    # Clear object contents
    $self->_reset_defaults;
    $self->clear_error;
    $self->{CHANGED} = 0;

    return 1;
}
#
#-----
#
sub clear_error #()#
{
    my $self = shift;
    $self->{ERROR} = '';
}
#
#-----

```

```

#-----
# DATA ACCESS METHODS
#
sub id
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ id } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_id $data) {
            $self->{ERROR} = 'Invalid ID';
            return;
        }
        $self->{ id } = untaint_id $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{id};
}
#
#-----
#
sub page
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ page } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_page_num $data) {
            $self->{ERROR} = 'Invalid PAGE';
            return;
        }
        $self->{ page } = untaint_page_num $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{page};
}
#
#-----
#
sub image
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ image } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_image $data) {
            $self->{ERROR} = 'Invalid IMAGE';
            return;
        }
        $self->{ image } = untaint_image $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{image};
}
#
#-----

```

```

#
sub height
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ height } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_height $data) {
            $self->{ERROR} = 'Invalid HEIGHT';
            return;
        }
        $self->{ height } = untaint_height $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{height};
}
#
#-----
#
sub width
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ width } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_width $data) {
            $self->{ERROR} = 'Invalid WIDTH';
            return;
        }
        $self->{ width } = untaint_width $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{width};
}
#
#-----
#
sub error_message #()#
{
    my $self = shift;
    return $self->{ERROR};
}
#
#-----
1;

```

## B.10. OAS::Person.pm

```

package OAS::Person;
require 5.8.5;
our $VERSION = 1.0.0;
our $DATE = '20 Jul 2006';

use strict;
use warnings;

```

```

use Thread;
use Carp;
use DBI;
use OAS::DBI;
use OAS::Utils qw(untaint);

use Exporter;
our @ISA = qw(Exporter);
our @EXPORT = qw();
our @EXPORT_OK = qw(
    is_valid_person_id is_valid_id is_valid_first_name is_valid_last_name
    is_valid_address is_valid_city is_valid_state is_valid_zip
    is_valid_phone is_valid_fax is_valid_mobile is_valid_email
    is_valid_password
    untaint_person_id untaint_id untaint_first_name untaint_last_name
    untaint_address untaint_city untaint_state untaint_zip
    untaint_phone untaint_fax untaint_mobile untaint_email
    untaint_password
);
our %EXPORT_TAGS = (
    check => [qw( is_valid_person_id is_valid_first_name is_valid_last_name
                  is_valid_address is_valid_city is_valid_state
                  is_valid_zip is_valid_phone is_valid_fax
                  is_valid_mobile is_valid_email is_valid_password )],
    untaint => [qw( untaint_person_id untaint_first_name untaint_last_name
                   untaint_address untaint_city untaint_state
                   untaint_zip untaint_phone untaint_fax
                   untaint_mobile untaint_email untaint_password )]
);

#####
# CONSTANTS
#
use constant DEFAULT_AREA_CODE => 801;
#
#####

#####
# EVENT HANDLERS
#
sub DESTROY
{
    # Disconnect from database on exit if needed
    my $self = shift;
    OAS::DBI::disconnect() if (defined $self->{DBH});
    $self->SUPER::DESTROY if $self->can("SUPER::DESTROY");
}
#
#####

-----
# VALIDATION & UNTAINT FUCTIONS
#
sub is_valid_id ($)
{
    local $_ = $_[0] || '';
    return /^ [a-z\d]{1,10} $/x;
}
#
-----
#

```



```

sub is_valid_person_id ($)
{
    return is_valid_id $_[0];
}
#
#-----
#
sub is_valid_first_name ($)
{
    local $_ = $_[0] || '';
    return /^[a-zA-Z\-\'\.\ \300-\366\370-\377]+$/x;
}
#
#-----
#
sub is_valid_last_name ($)
{
    local $_ = $_[0] || '';
    return /^[a-zA-Z\-\'\.\ \300-\366\370-\377]+$/x;
}
#
#-----
#
sub is_valid_address ($)
{
    local $_ = $_[0] || '';
    return /^[\\w\-\'\.\#\ \300-\366\370-\377]+$/x;
}
#
#-----
#
sub is_valid_city ($)
{
    local $_ = $_[0] || '';
    return /^[a-zA-Z\-\'\.\ \300-\366\370-\377]+$/x;
}
#
#-----
#
sub is_valid_state ($)
{
    local $_ = $_[0] || '';
    return /^[A-Z]{2}$/x;
}
#
#-----
#
sub is_valid_zip ($)
{
    local $_ = $_[0] || '';
    return /^\d{5}(?:\d{4})?$/x;
}
#
#-----
#
sub is_valid_phone ($)
{
    local $_ = $_[0] || '';
    return /^\d{10}$/x;
}
#
#-----
#
sub is_valid_fax ($)
{
    return is_valid_phone $_[0];
}
#

```

```

#-----
#
sub is_valid_mobile ($)
{
    return is_valid_phone $_[0];
}
#
#-----
#
sub is_valid_email ($)
{
    local $_ = $_[0] || '';
    return /^[a-z\d\-\._]+? \@ [a-z\d\-\._]+? \.[a-z]{2,3} $/x;
}
#
#-----
#
sub is_valid_password ($)
{
    local $_ = $_[0] || '';
    return /^\S{1,20} $/x;
}
#
#-----
#
sub untaint_id ($)
{
    return (is_valid_id $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_person_id ($)
{
    return untaint_id $_[0];
}
#
#-----
#
sub untaint_first_name ($)
{
    return (is_valid_first_name $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_last_name ($)
{
    return (is_valid_last_name $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_address ($)
{
    return (is_valid_address $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_city ($)
{
    return (is_valid_city $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#

```

```

sub untaint_state ($)
{
    return (is_valid_state $_[0] ) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_zip ($)
{
    return (is_valid_zip $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_phone ($)
{
    return (is_valid_phone $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_fax ($)
{
    return untaint_phone $_[0];
}
#
#-----
#
sub untaint_mobile ($)
{
    return untaint_phone $_[0];
}
#
#-----
#
sub untaint_email ($)
{
    return (is_valid_email $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_password ($)
{
    return (is_valid_password $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
#-----
# CONSTRUCTORS & INITIALIZERS
#
sub new #()#
{
    my $class = shift;
    $class = ref $class || $class;
    my $self = bless {}, $class;
    $self->_reset_defaults;
    $self->clear_error;
    $self->{ CHANGED } = 1;
    $self->{ DBH } = undef;
    return $self;
}
#
#-----
#
#

```

```

sub _reset_defaults #()#
{
    my $self = shift;
    foreach (qw(
        id first_name last_name address city state
        zip phone fax mobile email password))
    {
        $self->{$_} = undef;
    }
}
#
#-----

#-----
# DATA CONTROL METHODS
#
sub Load : locked method #($)#
{
    my $self = shift;
    my $id   = $_[0];
    my $href = undef;
    my $key  = undef;

    # Verify parameters
    if ( !defined $id ) { croak 'Missing Person ID for load'; }
    elsif ( !is_valid_person_id $id ) { croak 'Invalid Person ID for load'; }

    # Connect to database
    $self->{DBH} = OAS::DBI::connect() unless $self->{DBH};

    # Query database for needed information
    unless ($href = $self->{DBH}->selectrow_hashref(
        "SELECT * FROM person WHERE id='$id'"
    )) {
        $self->{ERROR} = $self->{DBH}->errstr;
        return;
    }

    # Reset object values to default
    $self->_reset_defaults;
    $self->clear_error;
    $self->{CHANGED} = 0;

    # Store the results in the object
    foreach $key qw(
        id first_name last_name address city state
        zip phone fax mobile email/
    ) {
        $self->{$key}($href->{$key}) if defined $href->{$key};
    }
    $self->{password} = $href->{password} if defined $href->{password};

    return 1;
}
#
#-----
#
sub Save : locked #(;$)#
{
    my $self = shift;
    my $key  = undef;

    # Check for forced save
    $self->{CHANGED} = 1 if $_[0];
}

```

```

# Verify needed columns
if ( !defined $self->{id} ) {
    $self->{ERROR} = 'Missing ID for save'; return;
} elsif ( !defined $self->{first_name} ) {
    $self->{ERROR} = 'Missing FIRST_NAME for save'; return;
} elsif ( !defined $self->{last_name} ) {
    $self->{ERROR} = 'Missing LAST_NAME for save'; return;
} elsif ( !defined $self->{password} ) {
    $self->{ERROR} = 'Missing PASSWORD for save'; return;
}
return SAVE_NOT_REQUIRED unless $self->{CHANGED};

# Connect to database
$self->{DBH} = OAS::DBI::connect() unless $self->{DBH};

# Gather & prepare data for query
my $id      = $self->{DBH}->quote( $self->{ id      } );
my $first_name = $self->{DBH}->quote( $self->{ first_name } );
my $last_name  = $self->{DBH}->quote( $self->{ last_name  } );
my $address    = $self->{DBH}->quote( $self->{ address  } );
my $city       = $self->{DBH}->quote( $self->{ city     } );
my $state      = $self->{DBH}->quote( $self->{ state    } );
my $email      = $self->{DBH}->quote( $self->{ email    } );
my $password   = $self->{DBH}->quote( $self->{ password } );
my $zip        = ( defined $self->{ zip      } ) ? int $self->{ zip      } : 'NULL';
my $phone      = ( defined $self->{ phone   } ) ? int $self->{ phone   } : 'NULL';
my $fax        = ( defined $self->{ fax     } ) ? int $self->{ fax     } : 'NULL';
my $mobile     = ( defined $self->{ mobile  } ) ? int $self->{ mobile  } : 'NULL';

# Build SQL query
my $query;
($query = <<__QUERY_END__>> =~ s/(?:\^\s+)|\s+(?==)|\n/ /gm;
    id      = $id,
    first_name = $first_name,
    last_name  = $last_name,
    address    = $address,
    city       = $city,
    state      = $state,
    zip        = $zip,
    phone      = $phone,
    fax        = $fax,
    mobile     = $mobile,
    email      = $email,
    password   = $password
__QUERY_END__
)
$query = "INSERT INTO person SET $query ON DUPLICATE KEY UPDATE $query";

# Execute query
my $sth = $self->{DBH}->prepare($query);
unless ($sth->execute) {
    $self->{ERROR} = $sth->errstr;
    return;
}

# Final items
$self->{CHANGED} = 0;

return 1;
}
#
#-----
#
sub Delete : locked #()#
{
    my $self = shift;
    my $sth  = undef;

```

```

# Delete from database
if ($self->{id}) {
    $self->{DBH} = OAS::DBI::connect() unless $self->{DBH};
    unless ($sth = $self->{DBH}->prepare(
        "DELETE FROM person WHERE id='$self->{id}'"
    )) {
        $self->{ERROR} = $self->{DBH}->errstr;
        return;
    }
    unless (defined $sth->execute) {
        $self->{ERROR} = $sth->errstr;
        return;
    }
}

# Clear object contents
$self->_reset_defaults;
$self->clear_error;
$self->{CHANGED} = 0;

return 1;
}
#
#-----
#
#
sub clear_error #()#
{
    my $self = shift;
    $self->{ERROR} = '';
}
#
#-----

#-----
# DATA ACCESS METHODS
#
sub id
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ id      } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        $data = lc $data;
        unless (is_valid_person_id $data) {
            $self->{ERROR} = 'Invalid ID';
            return;
        }
        $self->{ id      } = untaint_person_id $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{id};
}
#
#-----
#
#
sub first_name
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ first_name } = undef;

```

```

        $self->{ CHANGED      } = 1;
        return;
    }
    unless (is_valid_first_name $data) {
        $self->{ERROR} = 'Invalid FIRST_NAME';
        return;
    }
    $self->{ first_name } = untaint_first_name $data;
    $self->{ CHANGED      } = 1;
}
return $self->{first_name};
}
#
#-----
#
#
sub last_name
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ last_name } = undef;
            $self->{ CHANGED      } = 1;
            return;
        }
        unless (is_valid_last_name $data) {
            $self->{ERROR} = 'Invalid LAST_NAME';
            return;
        }
        $self->{ last_name } = untaint_last_name $data;
        $self->{ CHANGED      } = 1;
    }
    return $self->{last_name};
}
#
#-----
#
#
sub address
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ address } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_address $data) {
            $self->{ERROR} = 'Invalid ADDRESS';
            return;
        }
        $self->{ address } = untaint_address $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{address};
}
#
#-----
#
#
sub city
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ city      } = undef;

```

```

        $self->{ CHANGED } = 1;
        return;
    }
    unless (is_valid_city $data) {
        $self->{ERROR} = 'Invalid CITY';
        return;
    }
    $self->{ city } = untaint_city $data;
    $self->{ CHANGED } = 1;
}
return $self->{city};
}
#
#-----
#
sub state
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ state } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        $data = uc $data;
        unless (is_valid_state $data) {
            $self->{ERROR} = 'Invalid STATE';
            return;
        }
        $self->{ state } = untaint_state $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{state};
}
#
#-----
#
sub zip
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ zip } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_zip $data) {
            $self->{ERROR} = 'Invalid ZIP';
            return;
        }
        $self->{ zip } = untaint_zip $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{zip};
}
#
#-----
#
sub phone
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ phone } = undef;

```



```

        $self->{ CHANGED } = 1;
        return;
    }
    $data = DEFAULT_AREA_CODE . $data unless 10 == length $data;
    unless (is_valid_phone $data) {
        $self->{ERROR} = 'Invalid PHONE';
        return;
    }
    $self->{ phone } = untaint_phone $data;
    $self->{ CHANGED } = 1;
}
return $self->{phone};
}
#
#-----
#
# sub fax
# {
#     my $self = shift;
#     if (@_) {
#         my $data = shift;
#         if ('' eq $data) {
#             $self->{ fax } = undef;
#             $self->{ CHANGED } = 1;
#             return;
#         }
#         $data = DEFAULT_AREA_CODE . $data unless 10 == length $data;
#         unless (is_valid_fax $data) {
#             $self->{ERROR} = 'Invalid FAX';
#             return;
#         }
#         $self->{ fax } = untaint_fax $data;
#         $self->{ CHANGED } = 1;
#     }
#     return $self->{fax};
# }
#
#-----
#
# sub mobile
# {
#     my $self = shift;
#     if (@_) {
#         my $data = shift;
#         if ('' eq $data) {
#             $self->{ mobile } = undef;
#             $self->{ CHANGED } = 1;
#             return;
#         }
#         $data = DEFAULT_AREA_CODE . $data unless 10 == length $data;
#         unless (is_valid_mobile $data) {
#             $self->{ERROR} = 'Invalid MOBILE';
#             return;
#         }
#         $self->{ mobile } = untaint_mobile $data;
#         $self->{ CHANGED } = 1;
#     }
#     return $self->{ mobile };
# }
#
#-----
#
# sub email
# {
#     my $self = shift;
#     if (@_) {
#         my $data = shift;
#         if ('' eq $data) {

```

```

        $self->{ email } = undef;
        $self->{ CHANGED } = 1;
        return;
    }
    $data = lc $data;
    unless (is_valid_email $data) {
        $self->{ERROR} = 'Invalid EMAIL';
        return;
    }
    $self->{ email } = untaint_email $data;
    $self->{ CHANGED } = 1;
}
return $self->{email};
}
#
#-----
#
sub password
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ password } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        my $salt = join '', ('.', '/', 0..9, 'A'..'Z', 'a'..'z')[rand 64, rand 64];
        $data = crypt $data, $salt;
        unless (is_valid_password $data) {
            $self->{ERROR} = 'Invalid PASSWORD';
            return;
        }
        $self->{ password } = untaint_password $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{password};
}
#
#-----
#
sub error_message #()#
{
    my $self = shift;
    return $self->{ERROR};
}
#
#-----
1;

```

## B.11. OAS::Service.pm

```

package OAS::Service;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE = '05 Aug 2006';

use strict;
use Carp;
use Thread;
use OAS::DBI;
use OAS::Utils qw(untaint);
use OAS::Document qw(is_valid_document_id);

```

```

use Exporter;
our @ISA = qw(Exporter);
our @EXPORT = qw();
our @EXPORT_OK = qw(
    list_services      service_url
    is_valid_name      is_valid_url
    is_valid_description
    untaint_name       untaint_url
    untaint_description
);
our %EXPORT_TAGS = (
    check      => [qw( is_valid_name      is_valid_url
                       is_valid_description
                       untaint_name       untaint_url
                       untaint_description
                       )],
    untaint    => [qw( untaint_name       untaint_url
                       untaint_description
                       )]
);

#####
# EVENT HANDLERS
#
sub DESTROY
{
    # Disconnect from database on exit if needed
    my $self = shift;
    OAS::DBI::disconnect() if (defined $self->{DBH});
    $self->SUPER::DESTROY if $self->can("SUPER::DESTROY");
}
#
#####

#-----
# FUNCTIONS
#
sub list_services ()
{
    my $result = undef;

    my $dbh = OAS::DBI::connect;
    my $sth = $dbh->prepare('SELECT * FROM service');
    $sth->execute;
    while (my $href = $sth->fetchrow_hashref) {
        foreach my $key (keys %{$href}) {
            $result->{$href->{'name'}}{$key} = $href->{$key};
        }
    }
    OAS::DBI::disconnect;

    return $result;
}
#
#-----
#
sub service_url ($)
{
    my $name = $_[0];
    return unless $name;
    return unless is_valid_name($name);
    my $url = undef;
    my $dbh = OAS::DBI::connect;
    my $sth = $dbh->prepare("SELECT url FROM service where name='$name'");
    $sth->execute;
}

```

```

        if (my $href = $sth->fetchrow_hashref) {
            $url = $href->{url};
        }
        OAS::DBI::disconnect;

        return $url;
    }
#-----
#
sub is_valid_name ($)
{
    local $_ = $_[0] || '';
    return /^\w{1,36}$/x;
}
#-----
#
sub is_valid_url ($)
{
    local $_ = $_[0] || '';
    return /^\s*\/[\w\-\.\\/\300-\366\370-\377]{1,254}$/x;
}
#-----
#
sub is_valid_description ($)
{
    local $_ = $_[0] || '';
    return /^\s*[\w\-'"\.\:\ \300-\366\370-\377]{1,254}$/x;
}
#-----
#
sub untaint_name ($)
{
    return (is_valid_name $_[0]) ? untaint $_[0] : undef;
}
#-----
#
sub untaint_url ($)
{
    return (is_valid_url $_[0]) ? untaint $_[0] : undef;
}
#-----
#
sub untaint_description ($)
{
    return (is_valid_description $_[0]) ? untaint $_[0] : undef;
}
#-----

#-----
# CONSTRUCTORS & INITIALIZERS
#
sub new #()#
{
    my $class = shift;
    $class = ref $class || $class;
    my $self = bless {}, $class;
    $self->_reset_defaults;
    $self->clear_error;
    $self->{ CHANGED } = 1;
}

```

```

        $self->{ DBH      } = undef;
        return $self;
    }
#
#-----
#
sub _reset_defaults ()
{
    my $self = shift;
    foreach (qw(name url description)) {
        $self->{$_} = undef;
    }
}
#
#-----

#-----
# DATA CONTROL METHODS
#
sub Load : locked method #($)#
{
    my $self = shift;
    my $name = $_[0];
    my $href = undef;
    my $key = undef;

    # Verify parameters
    if ( not defined $name ) { croak 'Missing NAME for load'; }
    elsif ( not is_valid_name $name ) { croak 'Invalid NAME for load'; }

    # Connect to database
    $self->{DBH} = OAS::DBI::connect unless $self->{DBH};

    # Query database for needed information
    unless ($href = $self->{DBH}->selectrow_hashref(
        "SELECT * FROM service WHERE name='$name'"
    )) {
        $self->{ERROR} = $self->{DBH}->errstr;
        return;
    }

    # Reset object values to default
    $self->_reset_defaults;
    $self->clear_error;
    $self->{CHANGED} = 0;

    # Store the results in the object
    $self->name( $href->{ 'name' } ) if defined $href->{ 'name' };
    $self->url( $href->{ 'url' } ) if defined $href->{ 'url' };
    $self->filename( $href->{ 'description' } ) if defined $href->{ 'description' };

    return 1;
}
#
#-----
#
sub Save : locked #()#
{
    my $self = shift;
    my $key = undef;

    # Check for forced save
    $self->{CHANGED} = 1 if $_[0];
}

```

```

# Verify needed columns
if ( not $self->{name} ) {
    $self->{ERROR} = 'Missing NAME for save'; return;
} elsif ( not $self->{url} ) {
    $self->{ERROR} = 'Missing URL for save'; return;
}
return SAVE_NOT_REQUIRED unless $self->{CHANGED};

# Connect to database
$self->{DBH} = OAS::DBI::connect unless $self->{DBH};

# Gather & prepare data for query
my $name      = $self->{DBH}->quote( $self->{ name      } );
my $url       = $self->{DBH}->quote( $self->{ url       } );
my $description = $self->{DBH}->quote( $self->{ description } );

# Build SQL query
my $query;
($query = <<__QUERY_END__>> =~ s/(?:\^\s+)|\s+(?==)|\n/ /gm;
    name      = $name,
    url       = $url,
    description = $description
__QUERY_END__
)
$query = "INSERT INTO service SET $query ON DUPLICATE KEY UPDATE $query";

# Execute query
my $sth = $self->{DBH}->prepare($query);
unless ($sth->execute) {
    $self->{ERROR} = $sth->errstr;
    return;
}

# Final items
$self->{CHANGED} = 0;

return 1;
}
#
#-----
#
sub Delete : locked #()#
{
    my $self = shift;
    my $sth = undef;

    # Delete from database
    if (defined $self->{name}) {
        $self->{DBH} = OAS::DBI::connect unless $self->{DBH};
        if (!$sth = $self->{DBH}->prepare(
            "DELETE FROM service WHERE name='$self->{name}'"
        )) {
            $self->{ERROR} = $self->{DBH}->errstr;
            return;
        }
        unless (defined $sth->execute) {
            $self->{ERROR} = $sth->errstr;
            return;
        }
    }

    # Clear object contents
    $self->_reset_defaults;
    $self->clear_error;
    $self->{CHANGED} = 0;

    return 1;
}
#

```

```

#-----
#
sub clear_error #()#
{
    my $self = shift;
    $self->{ERROR} = '';
}
#
#-----

#-----
# DATA ACCESS METHODS
#
sub name
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ name } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_name $data) {
            $self->{ERROR} = 'Invalid NAME';
            return;
        }
        $self->{ name } = untaint_name $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{name};
}
#
#-----
#
sub url
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ url } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_url $data) {
            $self->{ERROR} = 'Invalid URL';
            return;
        }
        $self->{ url } = untaint_url $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{url};
}
#
#-----
#
sub description
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ description } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
    }
}

```

```

    }
    unless (is_valid_description $data) {
        $self->{ERROR} = 'Invalid DESCRIPTION';
        return;
    }
    $self->{ description } = untaint_description $data;
    $self->{ CHANGED      } = 1;
}
return $self->{description};
}
#
#-----
#
sub error_message #()#
{
    my $self = shift;
    return $self->{ERROR};
}
#
#-----
1;

```

## B.12. OAS::System.pm

```

package OAS::System;
require 5.6.0;
our $VERSION = 1.0.0;
our $DATE    = '04 Aug 2006';

use strict;
use warnings;

use Carp;
use Thread;
use OAS::DBI;
use OAS::Utils qw(untaint);

use Exporter;
our @ISA = qw(Exporter);
our @EXPORT = qw();
our @EXPORT_OK = qw(
    set_path          path_to_url
    is_valid_configuration is_valid_home_dir
    is_valid_www_dir   is_valid_www_url
    untaint_configuration untaint_home_dir
    untaint_www_dir    untaint_www_url
);
our %EXPORT_TAGS = (
    check => [qw( is_valid_configuration is_valid_home_dir
                  is_valid_www_dir       is_valid_www_url )],
    untaint => [qw( untaint_configuration untaint_home_dir
                    untaint_www_dir       untaint_www_url )]
);

#=====
# VARIABLES
#
our $Conf = new OAS::System;
$Conf->Load('default') or croak $Conf->error_message;
#
#=====

```



```

=====
# EVENT HANDLERS
#
sub DESTROY
{
    # Disconnect from database on exit if needed
    my $self = shift;
    OAS::DBI::disconnect() if (defined $self->{DBH});
    $self->SUPER::DESTROY if $self->can("SUPER::DESTROY");
}
#
=====

#-----
# FUNCTIONS
#
sub set_path ()
{
    $ENV{PATH} = '/usr/local/bin:/usr/bin:/bin:/usr/local/oas/bin/converters';
}
set_path;
#
#-----
#
sub path_to_url ($)
{
    my $path = $_[0];
    return unless defined $path;
    my $result = undef;

    my $check = $Conf->www_dir;
    if ($path =~ /^$check/) {
        $result = '/' . substr $path, length($check);
        $result =~ s|//|/|g;
    }

    return $result;
}
#
#-----
#
sub is_valid_configuration ($) {
    local $_ = $_[0] || '';
    return 7^\w+$/;
}
#
#-----
#
sub is_valid_home_dir ($) {
    local $_ = $_[0] || '';
    return 7^ \/[ \w\-\.\.\300-\366\370-\377]{0,254} $/x;
}
#
#-----
#
sub is_valid_www_dir ($) {
    local $_ = $_[0] || '';
    return 7^ \/[ \w\-\.\.\300-\366\370-\377]{0,254} $/x;
}
#
#-----
#
sub is_valid_www_url ($) {
    local $_ = $_[0] || '';
    return 7^ \/[ \w\-\.\.\300-\366\370-\377]{0,254} $/x;
}
}

```

```

#
#-----
#
sub is_valid_perl_dir ($) {
    local $_ = $_[0] || '';
    return 7^ \/ [\w\-\.\.\300-\366\370-\377]{0,254} $/x;
}
#
#-----
#
sub untaint_configuration ($)
{
    return (is_valid_configuration $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_home_dir ($)
{
    return (is_valid_home_dir $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_www_dir ($)
{
    return (is_valid_www_dir $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_www_url ($)
{
    return (is_valid_www_url $_[0]) ? untaint $_[0] : undef;
}
#
#-----
#
sub untaint_perl_dir ($)
{
    return (is_valid_perl_dir $_[0]) ? untaint $_[0] : undef;
}
#
#-----

#-----
# CONSTRUCTORS & INITIALIZERS
#
sub new #()#
{
    my $class = shift;
    $class = ref $class || $class;
    my $self = bless {}, $class;
    $self->_reset_defaults;
    $self->clear_error;
    $self->{ CHANGED } = 1;
    $self->{ DBH } = undef;
    return $self;
}
#
#-----
#

```

```

sub _reset_defaults ()
{
    my $self = shift;
    foreach (qw(configuration home_dir www_dir www_url perl_dir)) {
        $self->{$_} = undef;
    }
}
#
#-----

#-----
# DATA CONTROL METHODS
#
sub Load : locked method #($)#
{
    my $self = shift;
    my $conf = $_[0];
    my $href = undef;
    my $key = undef;

    # Verify parameters
    if ( !defined $conf ) {
        croak 'Missing Configuration for load!';
    } elsif ( !is_valid_configuration($conf) ) {
        croak 'Invalid Configuration for load!';
    }

    # Connect to database
    $self->{DBH} = OAS::DBI::connect unless $self->{DBH};

    # Query database for needed information
    unless ($href = $self->{DBH}->selectrow_hashref(
        "SELECT * FROM system WHERE configuration='$conf'"
    )) {
        $self->{ERROR} = $self->{DBH}->errstr;
        return;
    }

    # Reset object values to default
    $self->_reset_defaults;
    $self->clear_error;
    $self->{CHANGED} = 0;

    # Store the results in the object
    foreach $key qw(configuration home_dir www_dir www_url perl_dir) {
        $self->{$key}($href->{$key}) if defined $href->{$key};
    }

    return 1;
}
#
#-----
#
sub Save : locked #()#
{
    my $self = shift;
    my $key = undef;

    # Check for forced save
    $self->{CHANGED} = 1 if $_[0];

    # Verify needed columns
    if ( !$self->{configuration} ) {
        $self->{ERROR} = 'Missing CONFIGURATION for save!'; return;
    } elsif ( !$self->{home_dir} ) {
        $self->{ERROR} = 'Missing HOME_DIR for save!'; return;
    }
}

```

```

} elsif ( !$self->{www_dir} ) {
    $self->{ERROR} = 'Missing WWW_DIR for save'; return;
} elsif ( !$self->{www_url} ) {
    $self->{ERROR} = 'Missing WWW_URL for save'; return;
} elsif ( !$self->{perl_dir} ) {
    $self->{ERROR} = 'Missing PERL_DIR for save'; return;
}
return SAVE_NOT_REQUIRED unless $self->{CHANGED};

# Connect to database
$self->{DBH} = OAS::DBI::connect() unless $self->{DBH};

# Gather & prepare data for query
my $configuration = $self->{DBH}->quote( $self->{ configuration } );
my $home_dir      = $self->{DBH}->quote( $self->{ home_dir      } );
my $www_dir       = $self->{DBH}->quote( $self->{ www_dir       } );
my $www_url       = $self->{DBH}->quote( $self->{ www_url       } );
my $perl_dir      = $self->{DBH}->quote( $self->{ perl_dir      } );

# Build SQL query
my $query;
($query = <<__QUERY_END__>> =~ s/(?:\^\s+)|\s+(?==)|\n/ /gm;
    configuration = $configuration,
    home_dir      = $home_dir,
    www_dir       = $www_dir,
    www_url       = $www_url,
    perl_dir      = $perl_dir
__QUERY_END__
)
$query = "INSERT INTO system SET $query ON DUPLICATE KEY UPDATE $query";

# Execute query
my $sth = $self->{DBH}->prepare($query);
unless ($sth->execute) {
    $self->{ERROR} = $sth->errstr;
    return;
}

# Final items
$self->{CHANGED} = 0;

return 1;
}
#
#-----
#
sub Delete : locked #()#
{
    my $self = shift;
    my $sth = undef;

    # Delete from database
    if (defined $self->{configuration}) {
        $self->{DBH} = OAS::DBI::connect() unless $self->{DBH};
        if (!( $sth = $self->{DBH}->prepare(
            "DELETE FROM system WHERE id='$self->{configuration}'"
        ))) {
            $self->{ERROR} = $self->{DBH}->errstr;
            return;
        }
        unless (defined $sth->execute) {
            $self->{ERROR} = $sth->errstr;
            return;
        }
    }
}

# Clear object contents
$self->_reset_defaults;

```

```

$self->clear_error;
$self->{CHANGED} = 0;

return 1;
}
#
#-----
#
sub clear_error #()#
{
    my $self = shift;
    $self->{ERROR} = '';
}
#
#-----

#-----
# DATA ACCESS METHODS
#
sub configuration
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ configuration } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_configuration $data) {
            $self->{ERROR} = 'Invalid CONFIGURATION';
            return;
        }
        $self->{ configuration } = untaint_configuration $data;
        $self->{ CHANGED } = 1;
    }
    return $self->{configuration};
}
#
#-----
#
sub home_dir
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ home_dir } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_home_dir $data) {
            $self->{ERROR} = 'Invalid HOME_DIR';
            return;
        }
        $self->{ home_dir } = untaint_home_dir $data . '/';
        $self->{ home_dir } =~ s|//|/|g;
        $self->{ CHANGED } = 1;
    }
    return $self->{home_dir};
}
#
#-----
#

```

```

sub www_dir
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ www_dir } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_www_dir $data) {
            $self->{ERROR} = 'Invalid WWW_DIR';
            return;
        }
        $self->{ www_dir } = untaint_www_dir $data . '/';
        $self->{ www_dir } =~ s|//|/|g;
        $self->{ CHANGED } = 1;
    }
    return $self->{www_dir};
}
#
#-----
#
sub www_url
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ www_url } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_www_url $data) {
            $self->{ERROR} = 'Invalid WWW_URL';
            return;
        }
        $self->{ www_url } = untaint_www_url $data . '/';
        $self->{ www_url } =~ s|//|/|g;
        $self->{ CHANGED } = 1;
    }
    return $self->{www_url};
}
#
#-----
#
sub perl_dir
{
    my $self = shift;
    if (@_) {
        my $data = shift;
        if ('' eq $data) {
            $self->{ perl_dir } = undef;
            $self->{ CHANGED } = 1;
            return;
        }
        unless (is_valid_perl_dir $data) {
            $self->{ERROR} = 'Invalid PERL_DIR';
            return;
        }
        $self->{ perl_dir } = untaint_perl_dir $data . '/';
        $self->{ perl_dir } =~ s|//|/|g;
        $self->{ CHANGED } = 1;
    }
    return $self->{perl_dir};
}
#
#-----

```

```

#
sub error_message #()#
{
    my $self = shift;
    return $self->{ERROR};
}
#
#-----
1;

```

## B.13. OAS::Utils.pm

```

package OAS::Utils;
require 5.6.0;
our $VERSION = 1.0.0;
our $DATE    = '07 Jul 2006';

use strict;
use warnings;

use Carp;

use Exporter ();
our @ISA      = qw(Exporter);
our @EXPORT  = qw();
our @EXPORT_OK = qw(
    untaint
    prepad0      postpad0
    file_name    file_extension file_path
    is_valid_path is_valid_filename
);
our %EXPORT_TAGS = (
    padding => [qw( prepad0 postpad0 )],
    dangerous => [qw( untaint )],
    files => [qw( file_name file_extension file_path
        is_valid_path is_valid_filename )]
);

#-----
# FUNCTIONS
#
sub untaint ($) # EXTREMELY DANGEROUS if used incorrectly (i.e., lacking previous checks)
{
    croak 'Missing parameter for untaint' unless defined $_[0];
    my $data = $_[0];
    $data =~ /^^(.*)$/s;
    return $1;
}
#
#-----
#
sub prepad0 ($$)
{
    croak 'Missing scalar for prepad0' unless defined $_[0];
    croak 'Missing length for prepad0' unless defined $_[1];
    my ($data, $len) = @_;
    $data = "0$data" while length($data) < $len;
    return $data;
}
#
#-----
#

```

```

sub postpad0 ($$)
{
    croak 'Missing scalar for postpad0' unless defined $_[0];
    croak 'Missing length for postpad0' unless defined $_[1];
    my ($data, $len) = @_;
    $data .= '0' while length($data) < $len;
    return $data;
}
#-----
#
sub file_name ($)
{
    return unless defined $_[0];
    my $filename = $_[0];
    return $filename unless $filename =~ /\//;
    return substr $filename, 1 + rindex($filename, '/');
}
#-----
#
sub file_extension ($)
{
    return unless defined $_[0];
    my $extension = $_[0];
    return '' unless $extension =~ /\./;
    return substr $extension, 1 + rindex($extension, '.');
}
#-----
#
sub file_path ($)
{
    return unless defined $_[0];
    my $path = $_[0];
    return '' unless $path =~ /\//;
    return substr $path, 0, 1 + rindex($path, '/');
}
#-----
#
sub is_valid_path ($)
{
    local $_ = $_[0];
    return /^ \/[^\w\-\.\300-\366\370-\377\/]+ \/ $/x;
}
#-----
#
sub is_valid_filename ($)
{
    local $_ = $_[0];
    return /^ [\w\-\.\300-\366\370-\377]+ $/x;
}
#-----
1;

```





## Appendix C – Services Source Code

This appendix contains the source code for the different services that were used by *OAS*. See Section 3.2.6 for more information on services and how they are used.

### C.1. AddAnnotation.pm

```
package OAS::Apache2::AddAnnotation;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE    = '07 Aug 2006';

use strict;
use Apache2::Const      qw(REDIRECT);
use Apache2::RequestRec ();
use APR::Table          ();
use CGI                 ();
use OAS::System         ();
use OAS::CGI            ();
use OAS::Annotation    ();
use OAS::Auth           ();
use OAS::Document      ();
use OAS::Image          ();
use OAS::Service        ();

#####
# MAIN
#
sub handler
{
    my $r = shift;

    my $cgi      = new CGI;
    my $annotation = new OAS::Annotation;
    my ($referrer, $x_min, $y_min, $x_data, $y_data) = _init($r, $cgi, $annotation);
    &create_annotation($r, $cgi, $annotation, $x_min, $y_min, $x_data, $y_data);

    $r->headers_out->set( Location => $referrer );
    return REDIRECT;
}
#
#####

#-----
#
```

```

sub _init ($$$)
{
    my ($r, $cgi, $annotation) = @_;

    my $x_min = undef;
    my $y_min = undef;
    my @x_data = ();
    my @y_data = ();

    # Verify access
    my $author = OAS::Auth::current_user;
    my $owner = (defined $cgi->param('owner'))
        ? $cgi->param('owner')
        : $author;
    my $doc_id = $cgi->param('id');
    error_message($r, "$author does not have annotate access")
        unless OAS::Auth::has_annotate_access $author, $doc_id;

    # Calculate necessary values
    $x_min = int $cgi->param('x_min');
    $y_min = int $cgi->param('y_min');
    my $width = int($cgi->param('x_max')) - $x_min;
    my $height = int($cgi->param('y_max')) - $y_min;

    # Build annotation object
    $annotation->id( OAS::Document::create_new_document_id );
    $annotation->author( $author );
    $annotation->owner( $owner );
    $annotation->document_id( $doc_id );
    $annotation->page( $cgi->param('p') );
    $annotation->width( $width );
    $annotation->height( $height );
    $annotation->x_pos( $x_min );
    $annotation->y_pos( $y_min );
    $annotation->timestamp( $r->request_time );
    $annotation->font( $cgi->param( 'font' ) );
    $annotation->pitch( $cgi->param( 'font_pitch' ) );
    $annotation->color( $cgi->param( 'font_color' ) );
    $annotation->background( $cgi->param( 'background' ) );

    # Gather remaining <form> data
    my $referrer = (defined $cgi->param('referrer'))
        ? $cgi->param('referrer')
        : OAS::Service::service_url('AnnotateDocument');
    my $x_list = ($cgi->param('x_data') =~ /\^([\dA-Z,]+)$/)
        ? $1
        : error_message($r, 'Invalid form data: [x_data]');
    @x_data = split '/',/, $x_list;
    my $y_list = ($cgi->param('y_data') =~ /\^([\dW,]+)$/)
        ? $1
        : error_message($r, 'Invalid form data: [y_data]');
    @y_data = split '/',/, $y_list;

    return ($referrer, $x_min, $y_min, \@x_data, \@y_data);
}

```

```

#
#-----
#

```

```

sub create_annotation ($$$$$)
{
    my ($r, $cgi, $annotation, @coordinate_data) = @_;

    # Set Mode related items
    my $mode = (defined $cgi->param('mode')) ? $cgi->param('mode') : 'DRAW';
    my $filename = undef;
    if ('DRAW' eq $mode) {
        $annotation->type('Image');
        create_image($r, $annotation, @coordinate_data);
    }
}

```

```

} else {
    $annotation->type('Text');
    my $text = $cgi->param('text');
    $text = '&nbsp;' unless defined($text) and length($text);
    $filename = OAS::Document::document_id_to_annotation_path(
        $annotation->document_id
    )
        . 'a' . $annotation->id . '.txt';
    $annotation->text($filename);
    open(ATEXT, ">$filename") or
        error_message($r, "Cannot write annotation data file: $!");
    print ATEXT $text;
    close(ATEXT);
}

error_message($r, $annotation->error_message)
    unless $annotation->Save;
}
#
#-----
#
sub create_image ($$@)
{
    my ($r, $annotation, $x_min, $y_min, $x_data, $y_data) = @_;
    my $image = create OAS::Image;
    my $filename = OAS::Document::document_id_to_image_path($annotation->document_id)
        . 'a' . $annotation->id . '.gif';
    my $src = OAS::System::path_to_url $filename;
    my $drawing = 0;
    my $count = 0;

    # Rectify Image for line sizes
    my ($x_offset, $y_offset) = rectify_image($annotation, $x_data, $y_data);

    # Setup image
    my $width = $annotation->width;
    my $height = $annotation->height;
    $image->size($width, $height);
    $image->filename($annotation->image($filename));

    # Process data arrays
    my ($old_x, $old_y) = ();
    my $array_length = scalar @{$x_data};
    for (my $i=0; $i<$array_length; $i++) {

        # Get new data points
        my $new_x = $x_data->[$i];
        my $new_y = $y_data->[$i];

        # Check type of data points
        if ($new_x =~ /\d+$/) {
            if ($drawing) {
                # New data is coordinates
                $new_x = int($new_x) - $x_min + $x_offset;
                $new_y = int($new_y) - $y_min + $y_offset;
                $new_x = ( $new_x > $width ) ? $width : $new_x;
                $new_y = ( $new_y > $height ) ? $height : $new_y;
                $count++;

                # Draw new line
                $image->line($old_x, $old_y, $new_x, $new_y) if $count > 1;
                $old_x = $new_x;
                $old_y = $new_y;
            } else {
                error_message($r, 'Drawing/Coordinate Mis-match');
            }
        }
    }
}

```

```

    } else {
        # New data is command
        my $cmd = uc $new_x;
        if ('PEN_DOWN' eq $cmd) {
            $drawing = 1;
            $count = 0;
        } elsif ('PEN_UP' eq $cmd) {
            $drawing = 0;
            $image->point($old_x, $old_y) if (1 == $count);
        } elsif ('PEN_COLOR' eq $cmd) {
            $image->pen_color(lc $new_y);
        } elsif ('PEN_WEIGHT' eq $cmd) {
            $image->pen_weight(int $new_y);
        } else {
            error_message($r, 'Unknown Command');
        }
    }
}

# Write image
error_message($r, "Could not write image: $filename")
unless $image->write;
$annotation->image($filename);
}
#
#-----
#
sub rectify_image ($$$)
{
    my ( $annotation, $x_data, $y_data ) = @_;
    my ( $x_offset, $y_offset ) = (0,0);
    my ( $max_x, $max_y, $min_x, $min_y ) = (0,0,562949953421312,562949953421312);

    # Determine new image size & position taking line thickness into account
    my $array_length = scalar @{$x_data};
    my $pen_weight = 0;
    my $max_pen_weight = 0;
    for (my $i=0; $i<$array_length; $i++) {

        # Get new data points
        my $new_x = $x_data->[$i];
        my $new_y = $y_data->[$i];

        # Check type of data points
        if ($new_x =~ /\d+$/) {
            $max_x = ($new_x + $pen_weight > $max_x)
                ? $new_x + $pen_weight
                : $max_x;
            $max_y = ($new_y + $pen_weight > $max_y)
                ? $new_y + $pen_weight
                : $max_y;
            $min_x = ($new_x - $pen_weight < $min_x)
                ? $new_x - $pen_weight
                : $min_x;
            $min_y = ($new_y - $pen_weight < $min_y)
                ? $new_y - $pen_weight
                : $min_y;
        } elsif ('PEN_WEIGHT' eq uc $new_x) {
            $max_pen_weight = ($pen_weight > $max_pen_weight)
                ? $pen_weight
                : $max_pen_weight;
            $pen_weight = int($new_y / 2);
        }
    }
    $min_x = 0 if $min_x < 0;
    $min_y = 0 if $min_y < 0;
}

```

```

# Calculate offset
$x_offset = $annotation->x_pos - $min_x;
$y_offset = $annotation->y_pos - $min_y;
$x_offset = 0 if $x_offset < 0;
$y_offset = 0 if $y_offset < 0;

# Reset annotation data
$annotation->width( $max_x - $min_x + $max_pen_weight + 1 );
$annotation->height( $max_y - $min_y + $max_pen_weight + 1 );
$annotation->x_pos( $min_x );
$annotation->y_pos( $min_y );

return ($x_offset, $y_offset);
}
#
#-----
#
sub error_message ($$)
{
    my ($r, $message) = @_;
    $message = (defined $message) ? $message : '<unknown>';
    $r->content_type('text/html');
    print qq|
        <h1>ERROR</h1>
        <ul>
            <li>$message</li>
        </ul>
    |;
    exit;
}
#
#-----
1;

```

## C.2. AdminConverter.pm

```

package OAS::Apache2::AdminConverter;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE = '05 Aug 2006';

use strict;
use Apache2::Const qw(OK);
use Apache2::RequestRec ();
use CGI ();
use OAS::CGI ();
use OAS::Auth ();
use OAS::Converter ();
use OAS::Menu ();
use OAS::Service ();

#=====
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    print OAS::CGI::html_header('File Converter Administration', '/css/oas.css');
    print OAS::Menu::menubar;

    my $params = _init();

```

```

        if (not defined $params->{s}) { show_converters(          ) }
    elsif ('Add'      eq $params->{s}) { add_converter(    $params      ) }
    elsif ('Update'  eq $params->{s}) { update_converter( $params      ) }
    else              { error_message( 'Invalid state' ) }

    print OAS::CGI::html_footer;
    return OK;
}
#
#=====

#-----
#
sub _init ()
{
    # Verify User
    my $user = OAS::Auth::current_user;
    error_message("$user is not an administrator")
        unless OAS::Auth::is_admin $user;

    # Gather CGI parameters
    my $cgi = new CGI;
    my %params = ();
    map { $params{$_} = $cgi->param($_) } $cgi->param;

    return \%params;
}
#
#-----
#
sub show_converters (;$)
{
    my $message = (defined $_[0]) ? $_[0] : '';
    my $count = 0;

    # Setup Basic Form
    print qq|
        <div id="updateForm">
        <table border="0" width="100%" height="100%"><tbody><tr><td
            align="center" valign="center">

            $message

            <table cellpadding="5" cellspacing="1" border="1">
            <tbody>
            <tr>
            <th class="updateForm">File Converter List</th>
            </tr>
            <tr>
            <td align="center">
            <table cellpadding="5" cellspacing="0" border="0">
            <tbody>
            <tr>
            <th>In</th>
            <th>Out</th>
            <th>Command</th>
            <th>Description</th>
            <th colspan="2">&nbsp;&nbsp;&nbsp;</th>
            </tr>

            |;

    my $file_formats = OAS::Converter::list_converters;
    foreach my $in_ext (sort keys %{$file_formats}) {
        _show_converter_entry($file_formats->{$in_ext}, ++$count);
    }
    my $class = ( $count % 2 ) ? 'editListOdd' : 'editListEven';

```

```

print qq|
<form method="POST">
  <input type="HIDDEN" name="s" value="Add" />
<tr class="$class">
  <td align="center" valign="center">
    <input type="TEXT" name="in_ext" value="" size="4" maxlength="8" />
  </td>
  <td align="center" valign="center">
    <select name='out_ext'>
      <option >gif</option>
      <option SELECTED>jpg</option>
      <option >png</option>
    </select>
  </td>
  <td align="left" valign="center">
    <input type="TEXT" name="command" value="" size="32" maxlength="256" />
  </td>
  <td align="left" valign="center">
    <input type="TEXT" name="description" value="" size="28"
    maxlength="256" />
  </td>
  <td align="center" valign="center" colspan="2">
    <input type="SUBMIT" value="Add File Converter" />
  </td>
</form>
</tr>
</tbody>
</table>
</td>
</tbody>
</table>

</td></tr></tbody></table>
</div>

|;
}
#
#-----
#
sub _show_converter_entry ($$)
{
  my ($href, $count) = @_;
  return unless defined $href;

  my $class = ( ++$count % 2 ) ? 'editListOdd' : 'editListEven';
  my $in     = $href->{ 'in_ext' };
  my $out    = $href->{ 'out_ext' };
  my $cmd    = $href->{ 'command' };
  my $desc   = $href->{ 'description' };
  my $is_gif = ('gif' eq $out) ? ' SELECTED' : '';
  my $is_jpg = ('jpg' eq $out) ? ' SELECTED' : '';
  my $is_png = ('png' eq $out) ? ' SELECTED' : '';
  my $service = OAS::Service::service_url 'AdminDeleteConverter';

  print qq|
  <tr class="$class">
    <td align="center" valign="center">
      <form name="update_$in" method="POST">
        <input type="HIDDEN" name="s" value="Update" />
        <input type="HIDDEN" name="in_ext" value="$in" />
        <strong>$in</strong>
      </td>
    <td align="center" valign="center">
      <select name='out_ext'>
        <option$is_gif>gif</option>
        <option$is_jpg>jpg</option>
        <option$is_png>png</option>
      </select>
    </td>
  </tr>
}

```



```

        </td>
        <td align="left" valign="center">
            <input type="TEXT" name="command" value="$cmd" size="32"
                maxlength="256" />
        </td>
        <td align="left" valign="center">
            <input type="TEXT" name="description" value="$desc" size="28"
                maxlength="256" />
        </td>
        <td align="right" valign="center">
            <input type="SUBMIT" value="Update" />
        </td>
        <td align="left" valign="center">
            <input type="BUTTON" value="Delete"
                onClick="return helper_app('$service?in=$in', 400, 200, fileMenu);" />
        </td>
    </form>
</tr>

|;
}
#
#-----
#
sub add_converter ($)
{
    my $params = $_[0];
    my $converter = new OAS::Converter;
    if ($converter->Load($params->{'in_ext'})) {
        show_converters(
            '<ul><li>File converter for <em>' . $params->{'in_ext'} .
            '</em> already exists</li></ul>');
    } else {
        update_converter($params);
    }
}
#
#-----
#
sub update_converter ($)
{
    my $params = $_[0];
    my $message = '';
    my $converter = new OAS::Converter;
    error_message('Missing Input Extension')
        unless (exists $params->{'in_ext'});
    error_message('Missing Output Extension')
        unless (exists $params->{'out_ext'});
    error_message('Missing Conversion Command')
        unless (exists $params->{'command'});
    error_message('Missing Description')
        unless (exists $params->{'description'});
    my $in = $params->{'in_ext'} ;
    my $out = $params->{'out_ext'} ;
    my $cmd = $params->{'command'} ;
    my $desc = $params->{'description'} ;
    unless (OAS::Converter::is_valid_extension $in) {
        $message .= "<li>Invalid Input Extension</li>\n"
    }
    unless (OAS::Converter::is_valid_extension $out) {
        $message .= "<li>Invalid Output Extension</li>\n"
    }
    unless (OAS::Converter::is_valid_command $cmd) {
        $message .= "<li>Invalid Conversion Command</li>\n"
    }
    unless (OAS::Converter::is_valid_description $desc) {
        $message .= "<li>Invalid Description</li>\n"
    }
}

```

```

unless ($message) {
    $converter->in($in);
    $converter->out($out);
    $converter->command($cmd);
    $converter->description($desc);
    $message .= '<li>Unable to save file converter:'
        . $converter->error_message . "</li>\n"
        unless $converter->Save;
}

if ($message) {
    $message = "<ul>\n$message</ul>";
}

show_converters($message);

}
#
#-----
#
sub error_message ($)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';

    print qq|
        <h1>Error:</h1>
        <ul>
        <li>$message</li>
        </ul>
    | . OAS::CGI::html_footer;
    exit;
}
#
#-----
1;

```

### C.3. AdminDeleteConverter.pm

```

package OAS::Apache2::AdminDeleteConverter;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE = '04 Aug 2006';

use strict;
use Apache2::Const qw(OK);
use Apache2::RequestRec ();
use CGI ();
use OAS::CGI ();
use OAS::Auth ();
use OAS::Converter ();
use OAS::Service ();

#-----
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    print OAS::CGI::html_header('Delete File Converter', '/css/oas.css');
    my $converter = new OAS::Converter;
    if (_init($converter)) {
        delete_converter($converter);
    }
}

```

```

    } else {
        ask_for_confirmation($converter);
    }
    print OAS::CGI::html_footer;
    return OK;
};
#
#=====

#-----
#
#
sub _init ($)
{
    my $converter = $_[0];
    my $cgi       = new CGI;
    my $in_ext    = $cgi->param('in');
    error_message('Invalid Extension')
        unless OAS::Converter::is_valid_extension $in_ext;

    $converter->Load($in_ext) or error_message('Converter does not exist');
    my $user = OAS::Auth::current_user;
    error_message("$user does not have rights to delete this Converter")
        unless OAS::Auth::is_admin $user;

    my $confirmed = $cgi->param('confirmed');
    $confirmed = (defined($confirmed) and ('YES' eq $confirmed)) ? 1 : 0;

    return $confirmed;
}
#
#-----
#
sub ask_for_confirmation ($)
{
    my $in_ext = $_[0]->in;
    print qq|
        <table border="0" width="100% height="100%">
        <tbody>
        <tr>
        <td align="center" valign="center">
        <table cellpadding="1" cellspacing="1" border="0">
        <tbody>
        <tr>
        <td align="center" colspan="2">
        Are you sure you want to delete this File Converter?
        </td>
        </tr>
        <tr>
        <td align="right">

        <form method="POST">
        <input type="HIDDEN" name="in" value="$in_ext" />
        <input type="HIDDEN" name="confirmed" value="YES" />
        <input type="SUBMIT" value="Yes" />
        </form>

        </td>
        <td align="left">

|;
print OAS::CGI::close_window_form('No');
print qq|
        </td>
        </tr>
        </tbody>
        </table>
        </td>

```

```

                </tr>
            </tbody>
        </table>
    |;
}
#
#-----
#
sub delete_converter ($)
{
    my $converter = $_[0];
    my $in_ext    = $converter->in;
    error_message(
        "File Converter for <strong><em>$in_ext</em></strong> cannot be deleted"
    ) unless $converter->Delete;
    my $service = OAS::Service::service_url 'AdminConverter';
    print qq|
        <script type="text/javascript">
            opener.location.href='$service';
            self.close();
        </script>
    |;
}
#
#-----
#
sub error_message ($)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';

    print qq|
        <h1>Error:</h1>
        <ul>
            <li>$message</li>
        </ul>
    | . OAS::CGI::close_window_form . qq|
        </body>
        </html>
    |;

    exit;
}
#
#-----
1;

```

#### C.4. AdminDeleteService.pm

```

package OAS::Apache2::AdminDeleteService;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE    = '05 Aug 2006';

use strict;
use Apache2::Const      qw(OK);
use Apache2::RequestRec ();
use CGI                 ();
use OAS::CGI            ();
use OAS::Auth           ();
use OAS::Service        ();

```

```

=====
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    print OAS::CGI::html_header('Delete Web Service', '/css/oas.css');
    my $service = new OAS::Service;
    if (_init($service)) { delete_service( $service ); }
    else { ask_for_confirmation( $service ); }
    print OAS::CGI::html_footer;
    return OK;
};
#
=====

#-----
#
sub _init ($)
{
    my $service = $_[0];
    my $cgi = new CGI;
    my $name = $cgi->param('name');
    error_message('Invalid Name') unless OAS::Service::is_valid_name $name;

    $service->Load($name) or error_message('Service does not exist');
    my $user = OAS::Auth::current_user;
    error_message("$user does not have rights to delete this Service")
        unless OAS::Auth::is_admin $user;

    my $confirmed = $cgi->param('confirmed');
    $confirmed = (defined($confirmed) and ('YES' eq $confirmed)) ? 1 : 0;

    return $confirmed;
}
#
#-----
#
sub ask_for_confirmation ($)
{
    my $name = $_[0]->name;
    print qq|
        <table border="0" width="100% height="100%">
        <tbody>
        <tr>
        <td align="center" valign="center">
        <table cellpadding="1" cellspacing="1" border="0">
        <tbody>
        <tr>
        <td align="center" colspan="2">
        Are you sure you want to delete this Web Service?
        </td>
        </tr>
        <tr>
        <td align="right">

        <form method="POST">
        <input type="HIDDEN" name="name" value="$name" />
        <input type="HIDDEN" name="confirmed" value="YES" />
        <input type="SUBMIT" value="Yes" />
        </form>

        </td>
        <td align="left">
|;

```

```

print OAS::CGI::close_window_form('No');
print qq|
        </td>
      </tr>
    </tbody>
  </table>
</td>
</tr>
</tbody>
</table>
|;
}
#
#-----
#
sub delete_service ($)
{
  my $service = $_[0];
  my $name    = $service->name;
  error_message("Web Service <strong><em>$name</em></strong> cannot be deleted")
    unless $service->Delete;
  my $redirect = OAS::Service::service_url 'AdminService';
  print qq|
        <script type="text/javascript">
          opener.location.href='$redirect';
          self.close();
        </script>
|;
}
#
#-----
#
sub error_message ($)
{
  my $message = (defined $_[0]) ? $_[0] : '<unknown>';

  print qq|
    <h1>Error:</h1>
    <ul>
      <li>$message</li>
    </ul>
    | . OAS::CGI::close_window_form . qq|
    </body>
    </html>
|;

  exit;
}
#
#-----
1;

```

## C.5. AdminDeleteUser.pm

```

package OAS::Apache2::AdminDeleteUser;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE    = '05 Aug 2006';

use strict;
use Apache2::Const qw(OK);
use Apache2::RequestRec ();
use CGI ();
use OAS::CGI ();

```

```

use OAS::Auth          ();
use OAS::Person        ();
use OAS::Service       ();

#=====
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    print OAS::CGI::html_header('Delete Person', '/css/oas.css');

    my $person = new OAS::Person;
    if (_init($person)) { delete_person( $person ) }
    else { ask_for_confirmation( $person ) }

    print OAS::CGI::html_footer;
    return OK;
}
#
#-----
#
sub _init ($)
{
    my $person = $_[0];
    my $cgi     = new CGI;

    my $id = $cgi->param('id');
    error_message('Invalid User ID') unless OAS::Person::is_valid_person_id $id;

    my $user = OAS::Auth::current_user;
    $person->Load($id) or error_message('User does not exist');
    error_message("$user does not have rights to delete this User")
        unless OAS::Auth::is_admin $user;

    my $confirmed = $cgi->param('confirmed');
    $confirmed = (defined($confirmed) and ('YES' eq $confirmed)) ? 1 : 0;
    return $confirmed;
}
#
#-----
#
sub ask_for_confirmation ($)
{
    my $id = $_[0]->id;
    print qq|
        <table border="0" width="100% height="100%">
        <tbody>
        <tr>
        <td align="center" valign="center">
        <table cellpadding="1" cellspacing="1" border="0">
        <tbody>
        <tr>
        <td align="center" colspan="2">
            Are you sure you want to delete this User?
        </td>
        </tr>
        <tr>
        <td align="right">
            <form method="POST">
            <input type="HIDDEN" name="id" value="$id" />

```

```

        <input type="HIDDEN" name="confirmed" value="YES" />
        <input type="SUBMIT" value="Yes" />
    </form>

    </td>
    <td align="left">
| . OAS::CGI::close_window_form('No') . qq|
    </td>
</tr>
</tbody>
</table>
</td>
</tr>
</tbody>
</table>
|;
}
#
#-----
#
sub delete_person ($)
{
    my $person = $_[0];
    my $id      = $person->id;
    error_message(
        "User <strong><em>$id</em></strong> cannot be deleted - " .
        '<em>User is the owner of documents in the system</em>'
    ) unless $person->Delete;
    my $service = OAS::Service::service_url 'AdminUser';
    print qq|
        <script type="text/javascript">
            opener.location.href='$service';
            self.close();
        </script>
    |;
}
#
#-----
#
sub error_message ($)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';

    print qq|
        <h1>Error:</h1>
        <ul>
            <li>$message</li>
        </ul>
| . OAS::CGI::close_window_form . qq|
        </body>
        </html>
    |;

    exit;
}
#
#-----
1;

```

## C.6. AdminService.pm

```

package OAS::Apache2::AdminService;
require 5.8.0;

```



```

our $VERSION = 1.0.0;
our $DATE    = '05 Aug 2006';

use strict;
use Apache2::Const      qw(OK);
use Apache2::RequestRec ();
use CGI                 ();
use OAS::CGI           ();
use OAS::Auth          ();
use OAS::Menu          ();
use OAS::Service       ();

#=====
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    print OAS::CGI::html_header('Web Service Administration', '/css/oas.css');
    print OAS::Menu::menubar;

    my $params = _init();

    if (not defined $params->{s}) { show_services(          ) }
    elsif ('Add' eq $params->{s}) { add_service( $params    ) }
    elsif ('Update' eq $params->{s}) { update_service( $params    ) }
    else { error_message( 'Invalid state' ) }

    print OAS::CGI::html_footer;
    return OK;
}
#=====

#-----
#
sub _init ()
{
    # Verify User
    my $user = OAS::Auth::current_user;
    error_message("$user is not an administrator")
        unless OAS::Auth::is_admin $user;

    # Gather CGI parameters
    my $cgi = new CGI;
    my %params = ();
    map { $params{$_} = $cgi->param($_) } $cgi->param;

    return \%params;
}
#-----
#
sub show_services (;$)
{
    my $message = (defined $_[0]) ? $_[0] : '';
    my $count = 0;

    # Setup Basic Form

```

```

print qq|
  <div id="updateForm">
    <table border="0" width="100%" height="100%"><tbody><tr><td align="center"
      valign="center">

      $message

      <table cellpadding="5" cellspacing="1" border="1">
        <tbody>
          <tr>
            <th class="updateForm">Web Service List</th>
          </tr>
          <tr>
            <td align="center">
              <table cellpadding="5" cellspacing="0" border="0">
                <tbody>
                  <tr>
                    <th>Name</th>
                    <th>URL</th>
                    <th>Description</th>
                    <th colspan="2">&nbsp;&nbsp;&nbsp;</th>
                  </tr>
                </tbody>
              </table>
            </td>
          </tr>
        </tbody>
      </table>

    </div>

|;

my $services = OAS::Service::list_services;
foreach my $name (sort keys %{$services}) {
  _show_service_entry($services->{$name}, ++$count);
}
my $class = ( $count % 2 ) ? 'editListOdd' : 'editListEven';

print qq|
  <form method="POST">
    <input type="HIDDEN" name="s" value="Add" />
    <tr class="$class">
      <td align="center" valign="center">
        <input type="TEXT" name="name" value="" size="24" maxlength="36" />
      </td>
      <td align="center" valign="center">
        <input type="TEXT" name="url" value="" size="32" maxlength="128" />
      </td>
      <td align="left" valign="center">
        <input type="TEXT" name="desc" value="" size="32" maxlength="256" />
      </td>
      <td align="center" valign="center" colspan="2">
        <input type="SUBMIT" value="Add Web Service" />
      </td>
    </form>
  </tr>
</tbody>
</table>

</td></tr></tbody></table>

</div>

|;
}
#
#-----
#
sub _show_service_entry ($$)
{
  my ($href, $count) = @_;
  return unless defined $href;

  my $class = ( ++$count % 2 ) ? 'editListOdd' : 'editListEven';
  my $name = $href->{ 'name' };
}

```

```

my $url = $href->{ 'url' };
my $desc = (exists($href->{'description'}) and defined($href->{'description'}))
? $href->{'description'}
: '';
my $service = OAS::Service::service_url 'AdminDeleteService';

print qq|
|
|  |

```

```

unless (OAS::Service::is_valid_url $url) {
    $message .= "<li>Invalid URL</li>\n"
}
if ($desc and not OAS::Service::is_valid_description $desc) {
    $message .= "<li>Invalid Description</li>\n"
}

unless ($message) {
    $service->name($name);
    $service->url($url);
    $service->description($desc);
    $message .= '<li>Unable to save web service:'
        . $service->error_message . "</li>\n"
        unless $service->Save;
}

if ($message) {
    $message = "<ul>\n$message</ul>";
}

show_services($message);
}
#
#-----
#
sub error_message ($)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';

    print qq|
        <h1>Error:</h1>
        <ul>
        <li>$message</li>
        </ul>
    | . OAS::CGI::html_footer;
    exit;
}
#
#-----
1;

```

## C.7. AdminUser.pm

```

package OAS::Apache2::AdminUser;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE = '05 Aug 2006';

use strict;
use Apache2::Const qw(OK);
use Apache2::RequestRec ();
use CGI ();
use OAS::CGI ();
use OAS::Auth ();
use OAS::Menu ();
use OAS::Person ();
use OAS::Service ();

#-----
# MAIN
#

```

```

sub handler
{
    my $r = shift;
    $r->content_type('text/html');

    print OAS::CGI::html_header('User Administration', '/css/oas.css');
    print OAS::Menu::menubar;

    my $cgi = new CGI;
    my $state = _init($cgi);

    if (not defined $state) { list_users(                ) }
    elsif ('Group' eq $state) { show_group_membership( $cgi                ) }
    elsif ('Update' eq $state) { update_group_membership( $cgi, $state    ) }
    elsif ('Remove' eq $state) { update_group_membership( $cgi, $state    ) }
    elsif ('Add' eq $state) { add_user( $cgi                ) }
    else { error_message( 'Invalid state' ) }

    print OAS::CGI::html_footer;

    return OK;
};
#
=====

#-----
#
sub _init ($)
{
    my $cgi = $_[0];
    my $user = OAS::Auth::current_user;
    error_message("$user is not an administrator")
        unless OAS::Auth::is_admin $user;
    return $cgi->param('s');
}
#
#-----
#
sub list_users (;$)
{
    my $message = (defined $_[0]) ? $_[0] : '';

    # Setup Basic Form
    print qq|
        <div id="updateForm">
        <table border="0" width="100%" height="100%"><tbody><tr><td align="center"
            valign="center">

            $message

            <table cellpadding="5" cellspacing="1" border="1">
            <tbody>
            <tr>
            <th class="updateForm">User List</th>
            </tr>
            <tr>
            <td align="center">
            <table cellpadding="5" cellspacing="0" border="0">
            <tbody>

    |;

    my $count = 0;
    map { _list_user_entry($_, ++$count) } @({OAS::Auth::list_all_users});
    my $class = ( $count % 2 ) ? 'editListOdd' : 'editListEven';

```

```

print qq|
<form method="POST">
<input type="HIDDEN" name="s" value="Add" />
<tr class="$class">
<td>
<input type="TEXT" name="id" value="ID" size="6" maxlength="10" />
</td>
<td>
<input type="TEXT" name="last_name" value="LAST NAME" size="12"
maxlength="48" />
</td>
<td>
<input type="TEXT" name="first_name" value="FIRST NAME" size="12"
maxlength="24" />
</td>
<td>
<input type="PASSWORD" name="password" value="PASSWORD" size="10"
maxlength="20" />
</td>
<td align="left" colspan="2">
<input type="SUBMIT" value="Add New User" />
</td>
</tr>
</form>
</tbody>
</table>
</td>
</tbody>
</table>

</td></tr></tbody></table>
</div>

|;
}
#
#-----
#
sub _list_user_entry ($$)
{
my ($user, $count) = @_;
return unless defined $user;

my $class = ( ++$count % 2 ) ? 'editListOdd' : 'editListEven';
my $id = $user->{'id'};
my $name = $user->{'last_name'} . ', ' . $user->{'first_name'};

my $up_service = OAS::Service::service_url 'UserProfile';
my $du_service = OAS::Service::service_url 'AdminDeleteUser';

print qq|
<tr class="$class">
<td align="center" valign="center">
<em>$id</em>
</td>
<td align="left" valign="center" colspan="2">
<strong>$name</strong>
</td>
<td align="right" valign="center">
<form method="POST" action="$up_service">
<input type="HIDDEN" name="id" value="$id" />
<input type="SUBMIT" value="Profile" />
</td>
</form>
<td align="center" valign="center">
<form method="POST">
<input type="HIDDEN" name="s" value="Group" />
<input type="HIDDEN" name="id" value="$id" />
<input type="SUBMIT" value="Groups" />

```

```

        </td>
    </form>
    <td align="left" valign="center">
    <form method="POST" action="delete_user.cgi">
        <input type="HIDDEN" name="id" value="$id" />
        <input type="BUTTON" value="Delete" onClick=
            "return helper_app('$du_service?id=$id&s=Delete', 400, 200, fileMenu);"
        />
    </td>
    </form>
</tr>

    |;
}
#
#-----
#
sub add_user ($)
{
    my $cgi = $_[0];

    # Get parameters
    my $error = 0;
    my $message = '';
    my $id = defined($cgi->param('id')) ? lc($cgi->param('id')) : undef;
    my $fname = $cgi->param('first_name');
    my $lname = $cgi->param('last_name');
    my $passwd = $cgi->param('password');

    # Verify information
    unless (defined $id) {
        $message .= "<li>Missing User ID</li>\n"; $error++;
    }
    unless (defined $fname) {
        $message .= "<li>Missing First Name</li>\n"; $error++;
    }
    unless (defined $lname) {
        $message .= "<li>Missing Last Name</li>\n"; $error++;
    }
    unless (defined $passwd) {
        $message .= "<li>Missing Password</li>\n"; $error++;
    }
    unless (OAS::Person::is_valid_person_id $id) {
        $message .= "<li>Invalid User ID</li>\n"; $error++;
    }
    unless (OAS::Person::is_valid_first_name $fname) {
        $message .= "<li>Invalid First Name</li>\n"; $error++;
    }
    unless (OAS::Person::is_valid_last_name $lname) {
        $message .= "<li>Invalid Last Name</li>\n"; $error++;
    }
    unless (OAS::Person::is_valid_password $passwd) {
        $message .= "<li>Invalid Password</li>\n"; $error++;
    }

    # Process Request
    unless ($error) {
        my $user = new OAS::Person;
        if ($user->Load($id)) {
            $message .= "<li>User ID <strong><em>$id</em></strong>"
                . " is already in use</li>\n";
            $error++;
        } else {
            $user->id($id);
            $user->first_name($fname);
            $user->last_name($lname);
            $user->password($passwd);
        }
    }
}

```

```

        unless ($user->Save) {
            $message .= '<li>User Creation Failed: '
                . $user->error_message . "</li>\n";
            $error++;
        }
    }
}
if ($error) {
    $message = "<ul>\n$message</ul>";
}

list_users($message);
}
#
#-----
#
sub show_group_membership ($;$)
{
    my ($cgi, $message) = @_;
    $message = (defined $message) ? $message : '';
    my $user = $cgi->param('id');
    return list_users unless OAS::Person::is_valid_person_id $user;

    # Setup Basic Form
    print qq|
        <div id="updateForm">
        <table border="0" width="100%" height="100%"><tbody><tr><td align="center"
            valign="center">

            $message

            <table cellpadding="5" cellspacing="1" border="1">
            <tbody>
            <tr>
            <th class="updateForm">Group Membership for <em>$user</em></th>
            </tr>
            <tr>
            <td align="center">
                <table cellpadding="5" cellspacing="0" border="0">
                <tbody>

    |;

    my $count = 0;
    my %groups = ();
    map { $groups{$_}++ } @{{OAS::Auth::list_groups(undef)}};
    foreach my $name (sort keys %groups) {
        if (OAS::Auth::is_group_member $user, $name) {
            _list_group_entry($user, $name, ++$count);
            delete $groups{$name};
        }
    }
    my $class = ( $count % 2 ) ? 'editListOdd' : 'editListEven';

    my $remaining_groups = scalar (keys %groups);
    print qq|
        <form method="POST">
        <input type="HIDDEN" name="s" value="Update" />
        <input type="HIDDEN" name="id" value="$user" />
        <tr class="$class">
        <td>
            <select name="group">
    | if $remaining_groups;
    foreach my $name (sort keys %groups) {
        print qq|\t\t\t <option>$name</option>\n|;
    }
    print qq|
        </select>
        </td>

```



```

        <td align="center" colspan="2">
            <input type="SUBMIT" value="Add Group" />
        </td>
    </tr>
</form>
| if $remaining_groups;
print qq|
    </tbody>
</table>
</td>
</tbody>
</table>

    <p>
    <form name="return">
    <input type="SUBMIT" value="Return to User List" />
    </form>
    </p>

</td></tr></tbody></table>
</div>

|;
}
#
#-----
#
sub _list_group_entry ($$$)
{
    my ($user, $name, $count) = @_;
    return unless defined($user) and defined($name);

    my $class = ( ++$count % 2 ) ? 'editListOdd' : 'editListEven';

    print qq|
        <tr class="$class">
            <td align="center" valign="center">
                <strong>$name</strong>
            </td>
            <td align="left" valign="center">
                <form method="POST">
                    <input type="HIDDEN" name="s" value="Remove" />
                    <input type="HIDDEN" name="id" value="$user" />
                    <input type="HIDDEN" name="group" value="$name" />
                    <input type="SUBMIT" value="Remove Group" />
                </td>
            </form>
        </tr>

    |;
}
#
#-----
#
sub update_group_membership ($$)
{
    my ($cgi, $state) = @_;

    # Get parameters
    my $error = 0;
    my $message = '';
    my $user = $cgi->param('id');
    my $name = $cgi->param('group');

    # Verify information
    unless (defined $user) {
        $message .= "<li>Missing User ID</li>\n"; $error++;
    }
}

```

```

unless (defined $name) {
    $message .= "<li>Missing Group Name</li>\n"; $error++;
}
unless (OAS::Person::is_valid_person_id $user) {
    $message .= "<li>Invalid User ID</li>\n"; $error++;
}
unless (OAS::Auth::is_valid_group $name) {
    $message .= "<li>Invalid Group Name</li>\n"; $error++;
}

# Process Request
unless ($error) {
    if ('Update' eq $state) {
        unless (OAS::Auth::add_user_to_group $user, $name) {
            $message .= "<li>Could not add $user to $name</li>\n";
            $error++;
        }
    } else {
        unless (OAS::Auth::remove_user_from_group $user, $name) {
            $message .= "<li>Could not remove $user from $name</li>\n";
            $error++;
        }
    }
}
if ($error) {
    $message = "<ul>\n$message</ul>";
}

show_group_membership($cgi, $message);
}
#
#-----
#
sub error_message ($)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';

    print qq|
        <h1>Error:</h1>
        <ul>
            <li>$message</li>
        </ul>
        |. OAS::CGI::html_footer;
    exit;
}
#
#-----
1;

```

## C.8. AnnotateDocument.pm

```

package OAS::Apache2::AnnotateDocument;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE = '05 Aug 2006';

use strict;
use Apache2::Const qw(OK);
use Apache2::RequestRec ();
use CGI ();
use OAS::System ();
use OAS::CGI ();
use OAS::Annotation ();
use OAS::Document ();

```

```

use OAS::Menu          ();
use OAS::Auth          ();
use OAS::Person        ();

#=====
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');

    my $doc = new OAS::Document;
    my ($page, $title, $user) = _init($doc);
    setup_display($doc, $page, $title, $user);
    print OAS::CGI::html_footer;

    return OK;
}
#
#=====

#-----
#
sub _init ($)
{
    my $doc          = $_[0];
    my $cgi          = new CGI;
    my $is_error     = 0;
    my $error_message = '';
    my $id           = $cgi->param('id');

    # Check document errors
    if (not OAS::Document::is_valid_document_id $id) {
        $is_error++;
        $error_message .= "<li>Invalid Document ID</li>\n";
    } elsif (not $doc->Load($id)) {
        $is_error++;
        $error_message .= "<li>Cannot Load Document #\$id</li>\n";
    }

    # Check user errors
    my $user = OAS::Auth::current_user;
    if (not OAS::Person::is_valid_person_id $user) {
        $is_error++;
        $error_message .= "<li>Invalid User ID</li>\n";
    } elsif (not OAS::Auth::has_read_access $user, $id) {
        $is_error++;
        $error_message .=
            "<li>$user does not have read access to Document $id</li>\n";
    }

    # Display Error, if present, and exit
    if ($is_error) {
        print qq|
            <h1>ERROR</h1>
            <ul>
                $error_message
            </ul>
        |;
        warn $error_message;
        exit;
    }

    # Set Title
    my $title = 'OAS: ' . $doc->title . " [#\$id]";
}

```

```

# Reset Page, if necessary
my $page = 1;
if (defined $cgi->param('p')) {
    my $p = $cgi->param('p');
    $page = $p if $p =~ /\d+$/;
}
$page = $doc->total_pages if $page > $doc->total_pages;

return $page, $title, $user;
}
#
#-----
#
sub setup_display ($$$)
{
    my ($doc, $page, $title, $user) = @_;
    my $src = OAS::System::path_to_url $doc->page_image($page);
    my $offset_x = OAS::Menu::SIDEBAR_WIDTH() + OAS::Menu::BORDER_SIZE();
    my $offset_y = OAS::Menu::MENUBAR_HEIGHT() + OAS::Menu::INFOBAR_HEIGHT()
        + OAS::Menu::BORDER_SIZE();
    my $width = $doc->page_width($page) + $offset_x + OAS::Menu::BORDER_SIZE();
    my $height = $doc->page_height($page) + $offset_y + OAS::Menu::BORDER_SIZE();

    # Header Information
    print OAS::CGI::html_header(
        $title,
        '/css/oas.css',
        qq|
            <style type="text/css">
            body
            {
                background-image: url("$src");
                background-attachment: scroll;
                background-color: black;
                background-position: |.$offset_x.px |.$offset_y.qq|px;
                background-repeat: no-repeat;
            }
            </style>
        |,
        qq|onLoad="document_init($width,$height);"|
    );

    # Top Bars
    print OAS::Menu::menubar $doc, $page;
    print OAS::Menu::infobar $doc, $page;

    # Setup Annotations
    my $annotations = OAS::Annotation::annotation_list $doc->id, $page;
    my $annotation = new OAS::Annotation;
    my $a_path = OAS::Document::document_id_to_annotation_path $doc->id;
    my @a_list = ();

    foreach my $key (sort keys %{$annotations}) {
        if ( ( $user eq $annotations->{$key}{'owner'} ) or
            ( OAS::Auth::has_read_annotation_access($user, $doc->id) ) )
        {
            # Print annotation <div>
            $annotation->Load($annotations->{$key}{'id'});

            my $a_id = $annotation->id;
            my $a_width = $annotation->width + 10;
            my $a_height = $annotation->height + 10;
            my $a_x_pos = $annotation->x_pos + $offset_x;
            my $a_y_pos = $annotation->y_pos + $offset_y;
        }
    }
}

```

```

        my $style = 'position:absolute;'
                . "top:$a_y_pos" . 'px;'
                . "left:$a_x_pos" . 'px;'
                . "width:$a_width" . 'px;'
                . "height:$a_height" . 'px;'
                . 'overflow:hidden;';
my $html = $annotation->html;
print qq|
    <div id="a$a_id" name="a$a_id"
        style="$style">
        $html
    </div>
|;
close ANNOTATION;
push @a_list, $a_id;
    }
}

# Sidebar
print OAS::Menu::sidebar $doc, $page, 1, @a_list;

# Widgets
print qq|
    <div id="tracerWidget"></div>
    <script type="text/javascript">
        DOCUMENT_OFFSET_X = $offset_x;
        DOCUMENT_OFFSET_Y = $offset_y;
    </script>
|;

# Annotation Text Window
print OAS::Menu::annotation_text $doc, $page;

print qq|<script type="text/javascript">update_from_cookies()</script>|;
}
#
#-----
1;

```

## C.9. ChangeDocumentOwner

```

#!/usr/bin/perl -wT
require 5.8.0;
our $VERSION = 0.0.1;
our $DATE = '04 Aug 2006';

use strict;
use CGI                ();
use OAS::CGI           ();
use OAS::Document     ();
use OAS::Auth          ();
use OAS::Service      ();
use OAS::Person       ();

#-----
# MAIN
#
my $cgi = new CGI;
print $cgi->header('text/html');

```

```

print OAS::CGI::html_header('Change Document Owner', '/css/oas.css');
my $doc = new OAS::Document;
my $owner = new OAS::Person;

if (&_init($cgi, $doc, $owner)) { &change_document_owner( $doc, $owner ) }
else { &ask_for_confirmation( $doc, $owner ) }

print OAS::CGI::html_footer;
exit;
#
=====

#-----
#
sub _init ($$$)
{
    my ($cgi, $doc, $owner) = @_;

    my $id = $cgi->param('id');
    my $new_owner = $cgi->param('new_owner');
    error_message('Invalid Document ID')
        unless OAS::Document::is_valid_document_id $id;
    error_message('Invalid Person ID')
        unless OAS::Person::is_valid_person_id $new_owner;

    $doc->Load($id) or error_message('Document does not exist');
    my $user = OAS::Auth::current_user;
    error_message("$user is not the Document's owner")
        unless OAS::Auth::is_admin($user) or ($user eq $doc->owner);

    $owner->Load($new_owner) or error_message('New owner does not exist');

    my $confirmed = $cgi->param('confirmed');
    $confirmed = (defined($confirmed) and ('YES' eq $confirmed)) ? 1 : 0;

    return $confirmed;
}
#
#-----
#
sub ask_for_confirmation ($$)
{
    my ($doc, $owner) = @_;

    my $id = $doc->id;
    my $name = $owner->first_name . '&nbsp;' . $owner->last_name;
    my $new_owner = $owner->id;
    print qq|
        <table border="0" width="100% height="100%">
        <tbody>
        <tr>
        <td align="center" valign="center">
        <table cellpadding="1" cellspacing="1" border="0">
        <tbody>
        <tr>
        <td align="center" colspan="2">
        Are you sure you want to give this document to
        <strong>$name</strong>?
        </td>
        </tr>
        <tr>
        <td align="right">

        <form method="POST">
        <input type="HIDDEN" name="id" value="$id" />
        <input type="HIDDEN" name="new_owner" value="$new_owner" />

```

```

                <input type="HIDDEN" name="confirmed" value="YES" />
                <input type="SUBMIT" value="Yes" />
            </form>

            </td>
            <td align="left">
|;
print OAS::CGI::close_window_form('No');
print qq|
                </td>
                </tr>
            </tbody>
        </table>
    </td>
</tr>
</tbody>
</table>
|;
}
#
#-----
#
sub change_document_owner ($$)
{
    my ($doc, $owner) = @_;
    my $new_owner      = $owner->id;
    my $old_owner      = $doc->owner;
    my $doc_id         = $doc->id;
    error_message('Missing New Owner') unless defined $new_owner;
    error_message('Invalid New Owner') unless $doc->owner($new_owner) and $doc->Save;
    OAS::Auth::grant_user_access(
        $old_owner, $doc_id, OAS::Auth::ACL_READ_ACCESS() );
    OAS::Auth::revoke_user_access(
        $old_owner, $doc_id, OAS::Auth::ACL_DELETE_ACCESS() );
    OAS::Auth::grant_user_access(
        $old_owner, $doc_id, OAS::Auth::ACL_ANNOTATE_ACCESS() );
    OAS::Auth::grant_user_access(
        $old_owner, $doc_id, OAS::Auth::ACL_READ_ANNOTATION_ACCESS() );
    OAS::Auth::revoke_user_access(
        $old_owner, $doc_id, OAS::Auth::ACL_DELETE_ANNOTATION_ACCESS() );
    OAS::Auth::revoke_user_access(
        $old_owner, $doc_id, OAS::Auth::ACL_MODERATE_ACCESS() );
    OAS::Auth::write_htaccess($doc_id);
    my $redirect = OAS::Service::service_url 'ShowDocument';
    print qq|
        <script type="text/javascript">
            opener.location.href='$redirect?id=$doc_id';
            self.close();
        </script>
|;
}
#
#-----
#
sub error_message ($)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';

    print qq|
        <h1>Error:</h1>
        <ul>
            <li>$message</li>
        </ul>
|;
}

```

```

        | . OAS::CGI::close_window_form . qq|
          </body>
          </html>
      |;

      exit;
  }
#
#-----

```

## C.10. CreateDocument

```

#!/usr/bin/perl -wT

use strict;
use CGI                ();
use OAS::CGI           ();
use OAS::Auth          ();
use OAS::Converter     ();
use OAS::Document     ();
use OAS::Person       ();
use OAS::Utils        ();
use OAS::Service      ();
use OAS::System       ();

#-----
# MAIN
#
my $cgi = new CGI;
print $cgi->header('text/html');

print OAS::CGI::html_header('Create New Document')
      . qq|<script src="/js/div_manager.js"></script>\n|;

my $user = OAS::Auth::current_user;
my $state = $cgi->param('s') || '';

('create' eq $state) ? &create_document($cgi, $user)
                    : &new_document($user);

print OAS::CGI::html_footer;
exit;
#
#-----

#-----
#
sub new_document ($;$)
{
    my ($user, $title) = @_;
    $title = (defined $title) ? $title : '';
    my $user_list = OAS::Auth::list_all_users;
    print qq|
        <form name="create_form" enctype="multipart/form-data"
          method="POST" accept-charset="ISO-8859-1">
        <input type="HIDDEN" name="s" value="create" />
        <table>
        <tbody>
        <tr>
        <td valign="top">Title: </td>
        <td valign="top"><input type="TEXT" name="title" value="$title" size="48"
          /></td>

```





```

$original =~ s/[\w\.]//g;
$original = OAS::Utils::untaint $original;

# Put Timer on Screen
print qq|
    <div id="display">
    <table>
    <tbody>
    <tr>
    <td></td>
    <td style="font-size:150%">File Conversion in Progress...</td>
    </tr>
    </tbody>
    </table>
    </div>
|;

# Create document
my $doc = new OAS::Document;
my $id = $doc->id(OAS::Document::create_new_document_id);
$doc->author($user);
$doc->owner($owner);
$doc->title($title);
$doc->timestamp(time);
$doc->original($original);
$doc->Save or fatal_error($doc->error_message);
$id = $doc->id;
if ($owner ne $user) {
    OAS::Auth::grant_user_access $user, $id, OAS::Auth::ACL_READ_ACCESS();
    OAS::Auth::grant_user_access $user, $id, OAS::Auth::ACL_ANNOTATE_ACCESS();
}
OAS::Auth::write_htaccess $id;

# Create files to new directory
my $path = OAS::Document::document_id_to_path $id;
my $ipath = OAS::Document::document_id_to_image_path $id;
my $original_filename = $path . $original;
open (ORIGINAL, ">$original_filename")
    or fatal_error("Unable to create original file $original_filename: $!", $id);
print ORIGINAL while <$handle>;
close ORIGINAL;

# Process is complete - Load in main window and close pop-up
OAS::System::set_path;
if (OAS::Converter::convert $original_filename, $ipath, $id, $quality) {
    my $service = OAS::Service::service_url 'ShowDocument';
    print qq|
    <script type="text/javascript">
    var url = "$service?id=$id";
    opener.location.href=url;
    self.close();
    </script>
|;
} else {
    my $converters = OAS::Converter::list_converters;
    my $extension = lc OAS::Utils::file_extension $original_filename;
    if (exists $converters->{$extension}) {
        fatal_error('Server Error', $id);
    } else {
        fatal_error('File type is not recognized by the system', $id);
    }
}
}
#
#-----
#

```

```

sub fatal_error (;$$)
{
    my $message = (defined $_[0]) ? $_[0] : undef;
    my $id      = (defined $_[1]) ? $_[1] : undef;

    my $result = '<p>File conversion could not be completed';
    if (defined $message) {
        $result .= qq|
            for the following reason:
            <ul><li><tt>$message</tt></li></ul>
        |;
    }
    $result .= "</p>\n";
    if (defined $id) {
        my $doc = new OAS::Document;
        $doc->Delete if $doc->Load($id);
    }
    show_display($result);

    print OAS::CGI::close_window_form, OAS::CGI::html_footer;
    exit;
}
#
#-----
#
sub show_display ($)
{
    my $data = (defined $_[0]) ? $_[0] : '';
    $data =~ s/"\/\\"/g;
    $data =~ s/[\r\n\t]/ /g;
    print qq|
<script type="text/javascript">
div_content("display","$data");
</script>
|;
}
#
#-----

```

## C.11. DeleteAnnotation.pm

```

package OAS::Apache2::DeleteAnnotation;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE    = '05 Aug 2006';

use strict;
use Apache2::Const      qw(OK);
use Apache2::RequestRec ();
use CGI                 ();
use OAS::CGI            ();
use OAS::Annotation    ();
use OAS::Auth           ();
use OAS::Service       ();

#=====
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
}

```

```

print OAS::CGI::html_header('Delete Annotation', '/css/oas.css');
my $annotation = new OAS::Annotation;

if (_init($annotation)) { delete_annotation( $annotation ); }
else { ask_for_confirmation( $annotation ); }

print OAS::CGI::html_footer;

return OK;
}
#
=====

#-----
#
sub _init ()
{
    my $annotation = $_[0];

    my $cgi = new CGI;
    my $user = OAS::Auth::current_user;
    my $id = $cgi->param('id');
    error_message('Invalid Annotation ID')
        unless OAS::Annotation::is_valid_annotation_id $id;

    $annotation->Load($id) or error_message('Annotation does not exist');

    error_message("$user does not have rights to delete this annotation")
        unless ( $user eq $annotation->owner ) or
            ( OAS::Auth::has_delete_annotation_access $user, $annotation->document_id );

    my $confirmed = $cgi->param('confirmed');
    $confirmed = (defined($confirmed) and ('YES' eq $confirmed)) ? 1 : 0;

    return $confirmed;
}
#
#-----
#
sub ask_for_confirmation ($)
{
    my $id = $_[0]->id;
    print qq|
        <table border="0" width="100% height="100%">
        <tbody>
        <tr>
        <td align="center" valign="center">
        <table cellpadding="1" cellspacing="1" border="0">
        <tbody>
        <tr>
        <td align="center" colspan="2">
        Are you sure you want to delete this annotation?
        </td>
        </tr>
        <tr>
        <td align="right">

        <form name="confirmForm" method="POST">
        <input type="HIDDEN" name="id" value="$id" />
        <input type="HIDDEN" name="confirmed" value="YES" />
        <input type="SUBMIT" name="yes" value="Yes" />
        </form>
    </td>
    </tr>
    </tbody>
    </table>
    </td>
    </tr>
    </tbody>
    </table>
    </qq>
}

```

```

                </td>
                <td align="left">
|;
print OAS::CGI::close_window_form('No!');
print qq|
                </td>
                </tr>
            </tbody>
        </table>
    </td>
</tr>
</tbody>
</table>
<script type="text/javascript">
    document.confirmForm.yes.focus();
</script>
|;
}
#
#-----
#
sub delete_annotation ()
{
    my $annotation = $_[0];

    my $id = $annotation->document_id;
    my $page = $annotation->page;
    $annotation->Delete or error_message($annotation->error_message);
    my $service = OAS::Service::service_url 'AnnotateDocument';
    print qq|
        <script type="text/javascript">
            opener.location.href='$service?id=$id&p=$page';
            self.close();
        </script>
|;
}
#
#-----
#
sub error_message ($)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';

    print qq|
        <h1>Error:</h1>
        <ul>
            <li>$message</li>
        </ul>
| . OAS::CGI::close_window_form . qq|
        </body>
    </html>

|;

    exit;
}
#
#-----
1;

```

## C.12. DeleteDocument

```

#!/usr/bin/perl -wT
require 5.8.0;

```

```

our $VERSION = 0.0.1;
our $DATE    = '04 Aug 2006';

use strict;
use CGI      ();
use OAS::CGI ();
use OAS::Document ();
use OAS::Auth ();
use OAS::System ();

#####
# MAIN
#
my $cgi = new CGI;
print $cgi->header('text/html');

print OAS::CGI::html_header('Delete Document', '/css/oas.css');
my $doc = new OAS::Document;
if (&_init($cgi, $doc)) {
    &delete_document($doc);
} else {
    &ask_for_confirmation($doc);
}
print OAS::CGI::html_footer;
exit;
#
#####

#-----
#
sub _init ($)
{
    my ($cgi, $doc) = @_;
    my $id = $cgi->param('id');
    error_message('Invalid Document ID')
        unless OAS::Document::is_valid_document_id $id;

    $doc->Load($id) or error_message('Document does not exist');
    my $user = OAS::Auth::current_user;
    error_message("$user does not have rights to delete this document")
        unless OAS::Auth::has_delete_access $user, $doc->id;

    my $confirmed = $cgi->param('confirmed');
    $confirmed = (defined($confirmed) and ('YES' eq $confirmed)) ? 1 : 0;

    return $confirmed;
}
#
#-----
#
sub ask_for_confirmation ($)
{
    my $doc = $_[0];
    my $id = $doc->id;
    print qq|
        <table border="0" width="100% height="100%">
        <tbody>
        <tr>
        <td align="center" valign="center">
        <table cellpadding="1" cellspacing="1" border="0">
        <tbody>
        <tr>
        <td align="center" colspan="2">
        Are you sure you want to delete this document?
        </td>
    
```

```

        </tr>
        <tr>
            <td align="right">

                <form method="POST">
                    <input type="HIDDEN" name="id" value="$id" />
                    <input type="HIDDEN" name="confirmed" value="YES" />
                    <input type="SUBMIT" value="Yes" />
                </form>

            </td>
            <td align="left">
|;
print OAS::CGI::close_window_form('No');
print qq|
        </td>
        </tr>
        </tbody>
        </table>
        </td>
        </tr>
        </tbody>
        </table>
|;
}
#
#-----
#
sub delete_document ($)
{
    my $doc = $_[0];
    $doc->Delete or error_message($doc->error_message);
    my $redirect = $OAS::System::Conf->www_url . 'd/';
    print qq|
        <script type="text/javascript">
            opener.location.href='$redirect';
            self.close();
        </script>
|;
}
#
#-----
#
sub error_message ($)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';

    print qq|
        <h1>Error:</h1>
        <ul>
            <li>$message</li>
        </ul>
| . OAS::CGI::close_window_form . qq|
        </body>
        </html>
|;
    exit;
}
#
#-----

```

## C.13. DocumentPermissions.pm

```
package OAS::Apache2::DocumentPermissions;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE    = '07 Aug 2006';

use strict;
use Apache2::Const      qw(OK);
use Apache2::RequestRec ();
use CGI                 ();
use OAS::CGI            ();
use OAS::Auth           qw(/^ACL/ /^grant/ /^revoke/);
use OAS::Document      ();
use OAS::Person         ();
use OAS::Menu           ();
use OAS::Service        ();

#=====
# CONSTANTS
#
use constant USER_TYPE => 'user';
use constant GROUP_TYPE => 'group';
#
#=====

#-----
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    print OAS::CGI::html_header('Set Document Permissions','/css/oas.css');

    my $doc = new OAS::Document;
    my $params = _init($doc);
    print OAS::Menu::menubar($doc, 1);

    if (not defined $params->{s}) {
        display_document_permissions($doc);
    } elsif ('Update' eq $params->{s}) {
        update_document_permissions($doc, $params);
    } else {
        error_message('Unknown state');
    }

    print OAS::CGI::html_footer;
    return OK;
}
#
#-----

#-----
#
sub _init ($)
{
    my $doc = $_[0];
    my $cgi = new CGI;
    my %params = ();
    map { $params{$_} = $cgi->param($_) } $cgi->param;
}
```



```

&error_message('Missing or invalid Document ID')
    unless OAS::Document::is_valid_document_id $params{id};
&error_message("Unable to load Document ID #{$params{id}")
    unless $doc->Load($params{id});
my $user = OAS::Auth::current_user;
&error_message("$user does not have rights to edit permissions")
    unless (($user eq $doc->owner) or (OAS::Auth::is_admin $user));

return \%params;
}
#
#-----
#
sub display_document_permissions ($;$)
{
    my ($doc, $message) = @_;
    $message = (defined $message) ? $message : '';

    my $author = new OAS::Person;
    $author->Load($doc->author);
    my $name = $author->first_name . ' ' . $author->last_name;
    my $title = '<em>' . $doc->title . "</em> by <em>$name</em>";
    my $id = $doc->id;
    my $owner = $doc->owner;
    my $person = new OAS::Person;
    my $count = 0;

    # Gather Permission Listings
    my $users = OAS::Auth::who_has_user_access $id;
    my $groups = OAS::Auth::who_has_group_access $id;
    my $user_list = OAS::Auth::list_all_users;
    my $group_list = OAS::Auth::list_groups;

    # Setup Basic Form
    my $service = OAS::Service::service_url 'ChangeDocumentOwner';
    my $helper_app = qq|'$service?id=$id&new_owner=' +|
        . qq| document.newOwnerForm.new_owner.value, 400, 75, fileMenu|;
    print qq|
        <div id="updateForm">
            <table border="0" width="100%" height="100%"><tbody><tr><td align="center"
                valign="center">

                $message

            <table cellpadding="5" cellspacing="1" border="1">
                <tbody>
                <tr>
                <th class="updateForm">Permissions for $title</th>
                </tr>
                <tr>
                <td align="center">
                    <table cellpadding="5" cellspacing="0" border="0">
                        <tbody>
                        <tr class="editListEven">
                            <td>
                                <form method="POST" name="newOwnerForm" id="newOwnerForm"
                                    onsubmit="return helper_app($helper_app);">
                                </td>
                                <td align="right"><strong>Document Owner:</strong></td>
                                <td align="left" colspan="6">
                                    <select name="new_owner">
                                </td>
                            </tr>
                        </tbody>
                    </table>
                </td>
                </tr>
                </tbody>
            </table>
        </div>
    |;
    foreach my $user (@{$user_list}) {
        my $selected = ($user->{'id'} eq $doc->owner) ? ' SELECTED' : '';
        next unless $person->Load($user->{'id'});
        my $name = $person->last_name . ', ' . $person->first_name;
        print qq|\t\t\t\t <option value="$user->{id}"$selected>$name</option>\n|;
    }
}

```

```

print qq|
        </select>
        </td>
        <td align="center"><input type="submit" value="Update" /></td>
    </tr>
</form>
<tr>
    <td colspan="9">
    <form name="update_form" method="POST">
    </td>
    </tr>
|;

print _show_header_bar(++$count);

# Show each user with permissions
foreach my $user (sort keys %{$users}) {
    next if $user eq $owner;
    next unless $person->Load($user);
    my $name = $person->last_name . ', ' . $person->first_name;
    print _show_permissions(
        $id,
        $name,
        $user,
        USER_TYPE,
        ++$count,
        %{$users->{$user}}
    );
}

# Create pull-down with users without permissions
my $html = qq|<select name="user">\n|;
foreach my $user (@{$user_list}) {
    $html .= qq| <option value="$user->{id}">$user->{last_name}, |
        . qq| $user->{first_name}</option>\n|
        if not exists $users->{$user->{'id'}};
}
$html .= "</select>\n";
print _show_permissions($id, $html, '', USER_TYPE, ++$count, ());

# Show each group with permissions
print _show_header_bar(++$count);
foreach my $group (sort keys %{$groups}) {
    print _show_permissions(
        $id,
        $group,
        $group,
        GROUP_TYPE,
        ++$count,
        %{$groups->{$group}}
    );
}

# Create pull-down with groups without permissions
$html = qq|<select name="group">\n|;
foreach my $group (@{$group_list}) {
    next if $group eq OAS::Auth::ADMIN_GROUP();
    $html .= qq| <option>$group</option>\n|
        if not exists $groups->{$group};
}
$html .= "</select>\n";
print _show_permissions($id, $html, '', GROUP_TYPE, ++$count, ());
print _show_header_bar(++$count);

print qq|
        </tbody>
    </table>
    </td>

```

```

        </tbody>
    </table>
</form>

    &nbsp;  <br />

    <table border="1"><tbody><tr><td>
    <table cellspacing="0" cellpadding="2" border="0">
    <tbody>
    <tr class="editListEven">
    <th align="center" colspan="6">Access Permissions</th>
    </tr>
    | . _access_description( 'R', 'Read Document' )
    . _access_description( 'D', 'Delete Document' )
    . _access_description( 'A', 'Annotate Document' )
    . _access_description( 'Ra', 'Read Annotation' )
    . _access_description( 'Da', 'Delete Annotation' )
    . _access_description( 'M', 'Moderator Access' )
    . qq|
    </tbody>
    </table>
    </td></tr></tbody></table>

    </td></tr></tbody></table>
    </div>

|;
}
#
#-----
#
sub _show_header_bar ($)
{
    my $count = ($_[0] + 1) % 2;
    my $class = ($count) ? 'editListOdd' : 'editListEven';
    qq|
        <tr class="$class">
        <td colspan="2"></td>
        <td align="center"><strong>R</strong></td>
        <td align="center"><strong>D</strong></td>
        <td align="center"><strong>A</strong></td>
        <td align="center"><strong>Ra</strong></td>
        <td align="center"><strong>Da</strong></td>
        <td align="center"><strong>M</strong></td>
        <td></td>
        </tr>
    |;
}
#
#-----
#
sub _show_permissions ($$$$%)
{
    my ($doc_id, $name, $id, $type, $count, %acl) = @_;
    return unless ((USER_TYPE eq $type) or (GROUP_TYPE eq $type));

    my $access_R = (exists $acl{ &ACL_READ_ACCESS }) ? ' CHECKED' : '';
    my $access_D = (exists $acl{ &ACL_DELETE_ACCESS }) ? ' CHECKED' : '';
    my $access_A = (exists $acl{ &ACL_ANNOTATE_ACCESS }) ? ' CHECKED' : '';
    my $access_Ra = (exists $acl{ &ACL_READ_ANNOTATION_ACCESS }) ? ' CHECKED' : '';
    my $access_Da = (exists $acl{ &ACL_DELETE_ANNOTATION_ACCESS }) ? ' CHECKED' : '';
    my $access_M = (exists $acl{ &ACL_MODERATE_ACCESS }) ? ' CHECKED' : '';

    my $tag = ( USER_TYPE eq $type ) ? "<em>$id</em>" : 'Group';
    my $class = ( ++$count % 2 ) ? 'editListOdd' : 'editListEven';

    qq|
        <tr class="$class">
        <td>

```

```

        <form method="POST">
            <input type="HIDDEN" name="id" value="$doc_id" />
            <input type="HIDDEN" name="ug" value="$id" />
            <input type="HIDDEN" name="t" value="$type" />
            <input type="HIDDEN" name="s" value="Update" />
            $tag
        </td>
        <td><strong>$name</strong></td>
        <td><input type="CHECKBOX" name="R"$access_R /></td>
        <td><input type="CHECKBOX" name="D"$access_D /></td>
        <td><input type="CHECKBOX" name="A"$access_A /></td>
        <td><input type="CHECKBOX" name="Ra"$access_Ra /></td>
        <td><input type="CHECKBOX" name="Da"$access_Da /></td>
        <td><input type="CHECKBOX" name="M"$access_M /></td>
        <td><input type="SUBMIT" value="Update" /></td>
    </tr>
</form>

|;
}
#
#-----
#
sub _access_description ($$)
{
    my ($acl, $description) = @_ ;
    qq|
        <tr class="editListOdd">
            <td valign="bottom">&nbsp;<strong>$acl</strong>&nbsp;</td>
            <td valign="bottom">&nbsp;$description&nbsp;</td>
        </tr>
    |;
}
#
#-----
#
sub update_document_permissions ($$)
{
    my ($doc, $params) = @_ ;
    my $id = $params->{ 'ug' } || undef;
    $id = $params->{ 'user' } || undef unless defined($id) and length($id);
    $id = $params->{ 'group' } || undef unless defined($id) and length($id);
    my $type = $params->{ 't' } || undef;
    error_message('Invalid Type')
        unless defined($type) and ((USER_TYPE eq $type) or (GROUP_TYPE eq $type));

    # Build permission variables
    my $aclR = (exists $params->{'R' }) ? $params->{'R' } : 0;
    $aclR = ( $aclR ) ? 1 : 0;
    my $aclD = (exists $params->{'D' }) ? $params->{'D' } : 0;
    $aclD = ( $aclD ) ? 1 : 0;
    my $aclA = (exists $params->{'A' }) ? $params->{'A' } : 0;
    $aclA = ( $aclA ) ? 1 : 0;
    my $aclRa = (exists $params->{'Ra'}) ? $params->{'Ra' } : 0;
    $aclRa = ( $aclRa ) ? 1 : 0;
    my $aclDa = (exists $params->{'Da'}) ? $params->{'Da' } : 0;
    $aclDa = ( $aclDa ) ? 1 : 0;
    my $aclM = (exists $params->{'M' }) ? $params->{'M' } : 0;
    $aclM = ( $aclM ) ? 1 : 0;

    # Set permissions
    my $doc_id = $doc->id;
    if (USER_TYPE eq $type) {
        # User
        ($aclR ) ? grant_user_access( $id, $doc_id, ACL_READ_ACCESS)
                : revoke_user_access($id, $doc_id, ACL_READ_ACCESS);
        ($aclD ) ? grant_user_access( $id, $doc_id, ACL_DELETE_ACCESS)
                : revoke_user_access($id, $doc_id, ACL_DELETE_ACCESS);
    }
}

```

```

($aclA ) ? grant_user_access( $id, $doc_id, ACL_ANNOTATE_ACCESS)
          : revoke_user_access($id, $doc_id, ACL_ANNOTATE_ACCESS);
($aclRa) ? grant_user_access( $id, $doc_id, ACL_READ_ANNOTATION_ACCESS)
          : revoke_user_access($id, $doc_id, ACL_READ_ANNOTATION_ACCESS);
($aclDa) ? grant_user_access( $id, $doc_id, ACL_DELETE_ANNOTATION_ACCESS)
          : revoke_user_access($id, $doc_id, ACL_DELETE_ANNOTATION_ACCESS);
($aclM ) ? grant_user_access( $id, $doc_id, ACL_MODERATE_ACCESS)
          : revoke_user_access($id, $doc_id, ACL_MODERATE_ACCESS);
} else {
# Group
($aclR ) ? grant_group_access( $id, $doc_id, ACL_READ_ACCESS)
          : revoke_group_access($id, $doc_id, ACL_READ_ACCESS);
($aclD ) ? grant_group_access( $id, $doc_id, ACL_DELETE_ACCESS)
          : revoke_group_access($id, $doc_id, ACL_DELETE_ACCESS);
($aclA ) ? grant_group_access( $id, $doc_id, ACL_ANNOTATE_ACCESS)
          : revoke_group_access($id, $doc_id, ACL_ANNOTATE_ACCESS);
($aclRa) ? grant_group_access( $id, $doc_id, ACL_READ_ANNOTATION_ACCESS)
          : revoke_group_access($id, $doc_id, ACL_READ_ANNOTATION_ACCESS);
($aclDa) ? grant_group_access( $id, $doc_id, ACL_DELETE_ANNOTATION_ACCESS)
          : revoke_group_access($id, $doc_id, ACL_DELETE_ANNOTATION_ACCESS);
($aclM ) ? grant_group_access( $id, $doc_id, ACL_MODERATE_ACCESS)
          : revoke_group_access($id, $doc_id, ACL_MODERATE_ACCESS);
}

OAS::Auth::write_htaccess($doc_id);
display_document_permissions($doc);
}
#
#-----
#
sub error_message ($)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';

    print qq|
        <h1>Error:</h1>
        <ul>
        <li>$message</li>
        </ul>
        | . OAS::CGI::html_footer;
    exit;
}
#
#-----
1;

```

## C.14. EditAnnotation.pm

```

package OAS::Apache2::EditAnnotation;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE    = '05 Aug 2006';

use strict;
use Apache2::Const      qw(OK);
use Apache2::RequestRec ();
use CGI                 ();
use OAS::CGI            ();
use OAS::Annotation    ();
use OAS::Menu           ();
use OAS::Auth           ();
use OAS::Service       ();

```

```

=====
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    print OAS::CGI::html_header('Edit Annotatin', '/css/oas.css');

    my $cgi      = new CGI;
    my $annotation = new OAS::Annotation;
    my $state    = _init($cgi, $annotation);

    if (not defined $state) { display_edit_window(      $annotation ) }
    elsif ('Update' eq $state) { update_annotation( $cgi, $annotation ) }
    else { error_message(      'Unknown State'  ) }

    print OAS::CGI::html_footer;
    return OK;
}
#
=====

#-----
#
sub _init ()
{
    my ($cgi, $annotation) = @_;

    my $id  = $cgi->param('id');
    my $user = OAS::Auth::current_user;

    error_message('Invalid Annotation ID')
        unless OAS::Annotation::is_valid_annotation_id $id;
    error_message('Cannot load annotation')
        unless $annotation->Load($id);
    error_message("$user does not have rights to edit this annotation")
        unless OAS::Auth::has_annotate_access $user, $annotation->document_id;

    return (defined $cgi->param('s')) ? $cgi->param('s') : undef;
}
#
#-----
#
sub display_edit_window ($)
{
    my $annotation = $_[0];

    if ('Image' eq $annotation->type) {
        # Graphical Annotation
        print qq|
            <table border="0" width="100%">
            <tbody>
            <tr>
            <td align="center">
                <h2>Annotation Cannot Be Changed</h2>
                Annotation can only be deleted.
            </td>
            </tr>
            <tr><td align="center">| . OAS::CGI::close_window_form .
            qq|</td></tr>
            </tbody>
            </table>
        |;
    }
}

```

```

} else {
    # Text Annotation
    print qq|
    <script type="text/javascript">
    function add_annotation_text_submit ()
    {
        document.add_annotation.text.value      =
        document.annotation_text_control.text.value;
        document.add_annotation.font.value      =
        document.annotation_text_control.font.value;
        document.add_annotation.font_pitch.value =
        document.annotation_text_control.pitch.value;
        document.add_annotation.font_color.value =
        document.annotation_text_control.color.value;
        document.add_annotation.background.value =
        document.annotation_text_control.background.value;
        document.add_annotation.submit();
    }
    </script>
    |;
    my $style =
    'background:#D0D0D0;width:600px;height:250px;border-style:outset;';
    print OAS::Menu::annotation_text (
        $annotation->document_id,
        $annotation->page,
        $annotation,
        $style,
        'self.close();'
    );
    my $id = $annotation->id;
    print qq|
    <form name="add_annotation" method="POST">
    <input type="HIDDEN" name="id" value="$id" />
    <input type="HIDDEN" name="s" value="Update" />
    <input type="HIDDEN" name="text" value="" />
    <input type="HIDDEN" name="font" value="" />
    <input type="HIDDEN" name="font_pitch" value="" />
    <input type="HIDDEN" name="font_color" value="" />
    <input type="HIDDEN" name="background" value="" />
    </form>
    |;
}
}
#
#-----
#
sub update_annotation ($$)
{
    my ($cgi, $annotation) = @_;

    # Read in parameters
    my $text      = (defined $cgi->param('text'))
        ? $cgi->param('text')
        : '';
    my $font      = (defined $cgi->param('font'))
        ? $cgi->param('font')
        : 'sans-serif';
    my $pitch     = (defined $cgi->param('font_pitch'))
        ? $cgi->param('font_pitch')
        : 12;
    my $color     = (defined $cgi->param('font_color'))
        ? $cgi->param('font_color')
        : 'black';
    my $background = (defined $cgi->param('background'))
        ? $cgi->param('background')
        : 'yellow';
    $annotation->font($font);
    $annotation->pitch($pitch);
}

```

```

$annotation->color($color);
$annotation->background($background);

# Save Annotation
my $filename = $annotation->text;
open(ANNOTATION, ">$filename") or error_message('Cannot open annotation file');
print ANNOTATION $text;
close ANNOTATION;
$annotation->Save or error_message('Problem saving to database');

# Clean-up
my $id      = $annotation->document_id;
my $page    = $annotation->page;
my $service = OAS::Service::service_url 'AnnotateDocument';
print qq|
    <script type="text/javascript">
        opener.location.href='$service?id=$id&p=$page';
        self.close();
    </script>
|;
}
#
#-----
#
sub error_message ($)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';
    print qq|
        <h1>ERROR</h1>
        <ul>
        <li>$message</li>
        </ul>
    | . OAS::CGI::html_footer;
    exit;
}
#
#-----
1;

```

## C.15. FileFormats.pm

```

package OAS::Apache2::FileFormats;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE    = '04 Aug 2006';

use strict;
use Apache2::Const      qw(OK);
use Apache2::RequestRec ();
use OAS::Converter      ();
use OAS::CGI            ();

sub handler {
    my $r = shift;
    $r->content_type('text/html');

    print OAS::CGI::html_header('Available File Formats')
        . qq|
            <p style="text-align:center;">File Formats Recognized by OAS:</p>
            <div style="width:430px;height:120px;border-style:inset;overflow:auto;">
            <table cellpadding="2" border="0" width="100%">

```



```

        <tbody>
        <tr>
        |;

my $formats = OAS::Converter::list_converters;
foreach my $in_ext (sort keys %{$formats}) {
    my $description = $formats->{$in_ext}{'description'};
    print qq|
        <tr>
        <td align="center" valign="center"
            style="background:black;color:white;font-family:sans-serif;">
            <strong style="font-weight:bold;">$in_ext</strong></td>
        <td align="left" valign="center" style="font-size:85%;">
            $description</td>
        </tr>
        |;
    }

print qq|
    </tbody>
    </table>
    </div>
    |
    . '<div style="text-align:center"><p>'
    . OAS::CGI::close_window_form
    . "</p></div>\n"
    . OAS::CGI::html_footer;

return OK;
}

1;

```

## C.16. GroupManager.pm

```

package OAS::Apache2::GroupManager;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE    = '05 Aug 2006';

use strict;
use Apache2::Const      qw(OK);
use Apache2::RequestRec ();
use CGI                 ();
use OAS::CGI            ();
use OAS::Auth           ();
use OAS::Menu           ();
use OAS::Person         ();

#####
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    print OAS::CGI::html_header('Group Manager', '/css/oas.css');
    print OAS::Menu::menubar;
    my ($params, $groups, $user, $is_admin) = _init();

    if (not defined $params->{s}) {
        show_group_list($groups, $user);
    } elsif ('AddGroup' eq $params->{s}) {
        add_new_group($params, $user);
    }
}

```

```

} elsif ('RemoveGroup' eq $params->{s}) {
    remove_existing_group($params, $user, $is_admin);
} elsif ('Member' eq $params->{s}) {
    show_group_membership($params, $groups, $user, $is_admin);
} elsif ('AddUser' eq $params->{s}) {
    add_user($params, $groups, $user, $is_admin);
} elsif ('RemoveUser' eq $params->{s}) {
    remove_user($params, $groups, $user, $is_admin);
} elsif ('Transfer' eq $params->{s}) {
    transfer_group($params, $user, $is_admin);
} elsif ('Transferred' eq $params->{s}) {
    transfer_group_verified($params, $user, $is_admin);
} else {
    error_message('Invalid state') ;
}

print OAS::CGI::html_footer;
return OK;
};
#
#-----

#-----
#
sub _init ()
{
    # Setup user data
    my $user      = OAS::Auth::current_user;
    my $is_admin = OAS::Auth::is_admin $user;

    # Gather CGI request data
    my $cgi      = new CGI;
    my %params = ();
    map { $params{$_} = $cgi->param($_) } $cgi->param;

    # Gather group list
    my @groups = ($is_admin)
        ? @ { &OAS::Auth::list_groups          }
        : @ { &OAS::Auth::list_groups($user) };

    return (\%params, \@groups, $user, $is_admin);
}
#
#-----
#
sub show_group_list ($;$)
{
    my ($groups, $user, $message) = @_ ;
    $message      = (defined $message) ? $message : '';

    # Get Users Current Group Membership
    my @group_list = OAS::Auth::list_users_groups $user;
    my $group_list = '';
    for (my $group = 0; $group <= $#group_list; $group++) {
        $group_list .= $group_list[$group];
        $group_list .= ',&nbsp;&nbsp;&nbsp;' if $group + 1 <= $#group_list;
    }

    # Setup Basic Form
    print qq|
        <div id="updateForm">
        <table border="0" width="100%" height="100%"><tbody><tr><td align="center"
        valign="center">

            $message

```

```

<table cellpadding="5" cellspacing="1" border="1">
<tbody>
<tr>
<th class="updateForm">Groups of which <em>$user</em> is a Member</th>
</tr>
<tr>
<td>
<table cellpadding="5" cellspacing="0" border="0">
<tbody>
<tr class="editListOdd">
<td>$group_list</td>
</tr>
</tbody>
</table>
</td>
</tr>
<tr>
<th class="updateForm">Other Groups Available to <em>$user</em></th>
</tr>
<tr>
<td align="center">
<table cellpadding="5" cellspacing="0" border="0">
<tbody>
|;

my $count = 0;
map { _list_group_entry($_, ++$count) } @{$groups};
my $class = ( $count % 2 ) ? 'editListOdd' : 'editListEven';

print qq|
<form method="POST">
<input type="HIDDEN" name="s" value="AddGroup" />
<tr class="$class">
<td>
<input type="TEXT" name="name" value="" size="16" maxlength="32" />
</td>
<td align="left" colspan="3">
<input type="SUBMIT" value="Add New Group" />
</td>
</tr>
</form>
</tbody>
</table>
</td>
</tbody>
</table>

</td></tr></tbody></table>
</div>

|;
}
#
#-----
#
sub _list_group_entry ($$)
{
my ($name, $count) = @_;
return unless defined $name;

my $admin = OAS::Auth::ADMIN_GROUP() eq $name;
my $class = ( ++$count % 2 ) ? 'editListOdd' : 'editListEven';

print qq|
<tr class="$class">
<td align="center" valign="center">
<strong>$name</strong>
</td>
<td align="center" valign="center">

```

```

        <form method="POST">
          <input type="HIDDEN" name="s" value="Member" />
          <input type="HIDDEN" name="name" value="$name" />
          <input type="SUBMIT" value="Members" />
        </td>
      </form>
      <td align="center" valign="center">
|;
print qq|
      <form method="POST">
        <input type="HIDDEN" name="s" value="Transfer" />
        <input type="HIDDEN" name="name" value="$name" />
        <input type="SUBMIT" value="Transfer" />
| unless $admin;
print "\t\t </td>\n";
print "\t\t </form>\n" unless $admin;
print qq|\t\t <td align="center" valign="center">\n|;
print qq|
      <form method="POST" onSubmit="return confirm('Delete Group [$name]?')">
        <input type="HIDDEN" name="s" value="RemoveGroup" />
        <input type="HIDDEN" name="name" value="$name" />
        <input type="SUBMIT" value="Remove" />
| unless $admin;
print "\t\t </td>\n";
print "\t\t </form>\n" unless $admin;
print "\t\t </tr>\n";
}
#
#-----
#
sub add_new_group ($$)
{
  my ($params, $user) = @_ ;
  error_message('Missing Group Name') unless exists $params->{name};
  my $message = '';

  # Create Group
  unless (OAS::Auth::create_group $params->{name}, $user) {
    $message = "<ul><li>Unable to create new group</li></ul>\n";
  }

  my (undef, $groups, undef) = _init();
  show_group_list($groups, $user, $message);
}
#
#-----
#
sub remove_existing_group ($$$)
{
  my ($params, $user, $is_admin) = @_ ;

  error_message('Missing Group Name') unless exists $params->{name};;
  my $name = $params->{name};
  error_message('Invalid Group Name') unless OAS::Auth::is_valid_group $name;
  error_message("$user does not have rights to remove $name")
    unless (($user eq OAS::Auth::group_owner($name)) or $is_admin);
  my $message = '';

  # Remove Group
  unless (OAS::Auth::remove_group $name) {
    $message = "<ul><li>Unable to remove group <em>$name</em></li></ul>\n";
  }

  my (undef, $groups, undef) = _init();
  show_group_list($groups, $user, $message);
}
#
#-----

```

```

#
sub show_group_membership ($$$$;$)
{
    my ($params, $groups, $current_user, $is_admin, $message) = @_;
    $message = (defined $message) ? $message : '';
    my $name = $params->{name};
    return
        show_group_list($groups, $current_user) unless OAS::Auth::is_valid_group $name;
    my $count = 0;

    # Setup Basic Form
    print qq|
        <div id="updateForm">
        <table border="0" width="100%" height="100%"><tbody><tr><td align="center"
            valign="center">

            $message

            <table cellpadding="5" cellspacing="1" border="1">
            <tbody>
            <tr>
            <th class="updateForm">Group Membership for <em>$name</em></th>
            </tr>
            <tr>
            <td align="center">
            <table cellpadding="5" cellspacing="0" border="0">
            <tbody>
    |;

    my $user_list = OAS::Auth::list_all_users;
    my %found_users;
    my @members = @(&OAS::Auth::list_group_membership($name));
    my $owner = OAS::Auth::group_owner $name;
    foreach my $user (@members) {
        _list_membership_entry($user->{'id'}, $name, $owner, $is_admin, ++$count);
        $found_users{$user->{'id'}}++;
    }
    my $class = ( $count % 2 ) ? 'editListOdd' : 'editListEven';

    my $remaining_groups = scalar (@{$user_list});
    print qq|
        <form method="POST">
        <input type="HIDDEN" name="s" value="AddUser" />
        <input type="HIDDEN" name="name" value="$name" />
        <tr class="$class">
        <td>
        <select name="id">
    | if $remaining_groups;
    foreach my $user (@{$user_list}) {

        my $id = $user->{'id'};
        next if exists $found_users{$id};

        my $full_name = $user->{'last_name'} . ', ' . $user->{'first_name'};
        print qq|\t\t <option value="$id">$full_name</option>\n|;
    }
    print qq|
        </select>
        </td>
        <td align="center" colspan="2">
        <input type="SUBMIT" value="Add User" />
        </td>
        </tr>
        </form>
    | if $remaining_groups;
    print qq|
        </tbody>
        </table>

```

```

        </td>
      </tbody>
    </table>
  | . _return_button() . qq|
    </td></tr></tbody></table>
  </div>
};

}
#
#-----
#
sub _list_membership_entry ($$$$$)
{
  my ($id, $group, $owner, $is_admin, $count) = @_;
  return unless defined($id) and defined($group) and (defined($owner) or $is_admin);
  $owner = '' unless defined $owner;

  my $user = new OAS::Person;
  $user->Load($id);
  my $class = ( ++$count % 2 ) ? 'editListOdd' : 'editListEven';
  my $name = $user->last_name . ', ' . $user->first_name;
  my $remove = ($owner eq $id)
    ? '<strong>Group Owner</strong>'
    : qq|
      <input type="HIDDEN" name="s" value="RemoveUser" />
      <input type="HIDDEN" name="id" value="$id" />
      <input type="HIDDEN" name="name" value="$group" />
      <input type="SUBMIT" value="Remove" />
    |;

  print qq|
    <tr class="$class">
      <td align="center" valign="center">
        <strong>$name</strong>
      </td>
      <td align="left" valign="center">
        <form method="POST"
          onSubmit="return confirm('Remove [$id] from [$group]?')">
          $remove
        </td>
      </form>
    </tr>
  |;
}
#
#-----
#
sub add_user ($$$$$)
{
  my ($params, $groups, $user, $is_admin) = @_;
  my $name = $params->{ 'name' } || undef;
  my $id = $params->{ 'id' } || undef;
  my $message = '';

  # Verify Input
  if (not defined $name) {
    error_message('Missing Group Name')
  } elsif (not defined $id) {
    error_message('Missing User ID')
  } elsif (not OAS::Auth::is_valid_group $name) {
    error_message('Invalid Group Name')
  } elsif (not OAS::Person::is_valid_person_id $id) {
    error_message('Invalid User ID')
  }
  error_message("$user does not have rights to add $id to $name")
  unless ($is_admin or ($user eq OAS::Auth::group_owner $name));
}

```

```

# Update Group
unless (OAS::Auth::add_user_to_group $id, $name) {
    $message = "<ul><li>Unable to add $id to <em>$name</em></li></ul>\n";
}

show_group_membership($params, $groups, $user, $is_admin, $message);
}
#
#-----
#
sub remove_user ()
{
    my ($params, $groups, $user, $is_admin) = @_ ;
    my $name = $params->{ 'name' } || undef;
    my $id = $params->{ 'id' } || undef;
    my $message = '';

    # Verify Input
    if (not defined $name) {
        error_message('Missing Group Name')
    } elsif (not defined $id) {
        error_message('Missing User ID')
    } elsif (not OAS::Auth::is_valid_group $name) {
        error_message('Invalid Group Name')
    } elsif (not OAS::Person::is_valid_person_id $id) {
        error_message('Invalid User ID')
    }
    error_message("$user does not have rights to remove $id from $name")
        unless ($is_admin or ($user eq OAS::Auth::group_owner $name));

    # Update Group
    unless (OAS::Auth::remove_user_from_group $id, $name) {
        $message = "<ul><li>Unable to remove $id from <em>$name</em></li></ul>\n";
    }

    show_group_membership($params, $groups, $user, $is_admin, $message);
}
#
#-----
#
sub transfer_group ($$$)
{
    my ($params, $user, $is_admin) = @_ ;
    error_message('Missing Group Name') unless exists $params->{name};
    my $name = $params->{name};
    error_message('Invalid Group Name') unless OAS::Auth::is_valid_group $name;
    error_message("$user does not have rights to transfer $name")
        unless (($user eq OAS::Auth::group_owner $name) or $is_admin);

    # Setup Basic Form
    print qq|
<div id="updateForm">
<table border="0" width="100%" height="100%"><tbody><tr><td align="center"
    valign="center">

<table cellpadding="5" cellspacing="1" border="1">
<tbody>
<tr>
<th class="updateForm">Transfer Group <em>$name</em></th>
</tr>
<tr>
<td align="center">
<table cellpadding="5" cellspacing="0" border="0">
<tbody>
<tr class="editListOdd">
<td align="center" colspan="3">
        Transferring a group changes the ownership of that group. Once<br />
        transferred, you will no longer have access to view the group<br />

```

```

        membership, make changes to the group, or remove it.
    </td>
</tr>
<form method="POST">
<input type="HIDDEN" name="s" value="Transferred" />
<input type="HIDDEN" name="name" value="$name" />
<tr class="editListOdd">
    <td align="right">
        New Owner:
    </td>
    <td align="left">
        <select name="owner">
|;
my $user_list = OAS::Auth::list_all_users;
foreach my $u (@{$user_list}) {
    my $id      = $u->{'id'};
    my $selected = ($id eq $user) ? ' SELECTED' : '';
    my $name    = $u->{'last_name'} . ', ' . $u->{'first_name'};
    print qq|\t\t\t\t\t <option value="$id"$selected>$name</option>\n|;
}
print qq|
    </select>
</td>
    <td align="left">
        <input type="SUBMIT" value="Transfer Group" />
    </td>
</tr>
</form>
</tbody>
</table>
</td>
</tbody>
</table>
| . _return_button() . qq|
    </td></tr></tbody></table>
</div>
|;
}
#
#-----
#
sub transfer_group_verified ($$$)
{
    my ($params, $user, $is_admin) = @_;
    my $name      = $params->{'name'} || undef;
    my $owner     = $params->{'owner'} || undef;
    my $message = '';

    # Verify Input
    if (not defined $name) {
        error_message('Missing Group Name')
    } elsif (not defined $owner) {
        error_message('Missing Owner ID')
    } elsif (not OAS::Auth::is_valid_group $name) {
        error_message('Invalid Group Name')
    } elsif (not OAS::Person::is_valid_person_id $owner) {
        error_message('Invalid Owner ID')
    }
    error_message("$user does not have rights to transfer $name")
        unless (($user eq OAS::Auth::group_owner $name) or $is_admin);

    # Transfer Group
    unless ($owner eq OAS::Auth::group_owner($name, $owner)) {
        $message = "<ul><li>Unable to transfer group <em>$name</em></li></ul>\n";
    }
}

```



```

    my (undef, $groups, undef) = _init();
    show_group_list($groups, $user, $message);
}
#
#-----
#
sub _return_button ()
{
    qq|
        <p>
        <form name="return">
        <input type="SUBMIT" value="Return to Group List" />
        </form>
        </p>

    |;
}
#
#-----
#
sub error_message ($)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';

    print qq|
        <h1>Error:</h1>
        <ul>
        <li>$message</li>
        </ul>
        | . OAS::CGI::html_footer;
    exit;
}
#
#-----
1;

```

## C.17. IndexMenu.pm

```

package OAS::Apache2::IndexMenu;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE    = '04 Aug 2006';

use strict;
use Apache2::Const      qw(OK);
use Apache2::RequestRec ();
use OAS::Menu           ();

#=====
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    print OAS::Menu::menubar;
    return OK;
}
#=====
1;

```

## C.18. Logout.pm

The process used for this handler is based heavily on the work of Nano Documet (Documet 2006). It is used here by permission. Consequently, this section is not covered under Creative Commons..

```
package OAS::Apache2::Logout;
require 5.6.0;
our $VERSION = 1.0.0;
our $DATE    = '05 Aug 2006';

# XHTML slightly modified from: http://nanodocumet.homedns.org/rest/
# Copyright 2006, Nano Documet, used with permission

use strict;
use Apache2::Const      qw(OK);
use Apache2::RequestRec ();
use OAS::System         ();

#####
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    my $home = $OAS::System::Conf->www_url;
    my $protect = $home . 'd/';
    my $html = qq|
        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
            "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd ">
        <html xmlns="http://www.w3.org/1999/xhtml">
        <head>
        <title>Logout</title>
        </head>
        <script language="javascript" type="text/javascript">
        try{
            var agt=navigator.userAgent.toLowerCase();
            if (agt.indexOf("msie") != -1) {
                document.execCommand("ClearAuthenticationCache");
            } else {
                var xmlhttp = createXMLObject();
                xmlhttp.open("GET", "$protect", true, "OAS_LogOut_User", "*****");
                xmlhttp.send("");
                xmlhttp.abort();
            }
            window.location = "$home";
        } catch(e) {
            alert("there was an error");
        }
        function createXMLObject() {
            try {
                if (window.XMLHttpRequest) {
                    xmlhttp = new XMLHttpRequest();
                } else if (window.ActiveXObject) {
                    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
                }
            } catch (e) {
                xmlhttp=false
            }
        }
    |;
}
```

```

        return xmlhttp;
    }
</script>
</body>
</html>

|;

$html =~ s/[\r\n]+/\n/g;
$html =~ s/^\t//g;
print $html;
return OK;
}
#
=====

1;

```

## C.19. new\_user.cgi

```

#!/usr/bin/perl -wT

use strict;
use warnings;
use CGI;
use OAS::CGI;
use OAS::Auth qw(is_user);
use OAS::Person qw(:check);

#=====
# GLOBALS
#
my $Cgi      = new CGI;
my $User_ID = $Cgi->param( 'id'      );
my $First_Name = $Cgi->param( 'first_name' );
my $Last_Name = $Cgi->param( 'last_name' );
my $Password = $Cgi->param( 'password' );
#
#=====

#=====
# MAIN
#
if (&new_user) { &add_new_user }
else          { &display_form }
exit;
#
#=====

#-----
#
sub new_user ()
{
    my $result = 0;
    my $state = $Cgi->param('s');
    $result = (defined($state) and ('Add' eq $state));
    if ($result) {
        unless (( defined $User_ID
                    and ( defined $First_Name
                          and ( defined $Last_Name
                                )
                            )
                )
        )
    }
}

```

```

        and ( defined $Password          )
        and ( is_valid_person_id $User_ID )
        and ( is_valid_first_name $First_Name )
        and ( is_valid_last_name $Last_Name )
        and ( is_valid_password $Password ))
    {
        &display_form('Missing or Invalid Values');
        exit;
    }

    if (is_user($User_ID)) {
        &display_form(
            'User ID is already in use. Please choose a different one.');
```

---

```

        exit;
    }
}

return $result;
}
#
#-----
#
sub add_new_user ()
{
    my $user = new OAS::Person;
    $user->id($User_ID);
    $user->first_name($First_Name);
    $user->last_name($Last_Name);
    $user->password($Password);
    unless ($user->Save) {
        &display_form("Unable to create new account: $User_ID");
        exit;
    }
    print $Cgi->redirect("/mp/UserProfile?id=$User_ID");
}
#
#-----
#
sub display_form (;$)
{
    my $message = $_[0];
    $message = "<p><strong>ERROR: <em>$message</em></strong></p>"
        if defined $message;

    my $html = "content-type: text/html\n\n"
        . html_header('OAS: Online Annotation System', '/css/oas.css')
        . qq|
        <form name="welcomeForm" method="POST">
        <input type="HIDDEN" name="s" value="Add" />
        <div id="welcomeScreen">
        <h1 class="welcomeText"><a href="/d/"
        ><u>OAS: Online Annotation System</u></a></h1>

        <table border="0" width="100%"><tbody><tr><td align="center">
        $message
        <p>Please enter your informatin below. User IDs may only contain lowercase
        letters or numbers. Once your account has been created, you will be
        presented with another logon screen.</p>
        <table class="welcomeText" cellspacing="5" cellpadding="5" border="0">
        <tbody>
        <tr>
        <td align="right">User&nbsp;ID:</td>
        <td align="left"><input type="TEXT" name="id" size="10" maxlength="10"
        /></td>
        </tr>
        <tr>
        <td align="right">First&nbsp;Name:</td>
        <td align="left"><input type="TEXT" name="first_name" size="24"

```

```

        maxlength="24" /></td>
</tr>
<tr>
  <td align="right">Last Name:</td>
  <td align="left"><input type="TEXT" name="last_name" size="24"
    maxlength="48" /></td>
</tr>
<tr>
  <td align="right">Password:</td>
  <td align="left"><input type="PASSWORD" name="password" size="20"
    maxlength="20" /></td>
</tr>
<tr>
  <td align="center" colspan="2">
    <input type="SUBMIT" value="Create New User and Request Login" />
  </td>
</tr>
</tbody>
</table>

</td></tr></tbody></table>

</div>
</form>
| . html_footer;

$html =~ s/^\t//gm;
print $html;
}
#
#-----

```

## C.20. OpenDocument.pm

```

package OAS::Apache2::OpenDocument;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE    = '04 Aug 2006';

use strict;
use Apache2::Const      qw(OK);
use Apache2::RequestRec ();
use CGI                 ();
use OAS::CGI            ();
use OAS::Auth           ();
use OAS::Service        ();
use OAS::Utils          ();

#=====
# CONSTANTS
#
use constant DEFAULT_SORT => 'timestamp DESC';
#
#=====

#=====
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    my $cgi = new CGI;
    my $sort = (defined $cgi->param('ob')) ? $cgi->param('ob') : DEFAULT_SORT;

```

```

    build_header($sort);
    build_table($sort);

    build_footer();
    return OK;
};
#
#=====

#-----
#
sub build_header ($)
{
    my $sort = $_[0];
    my $url = OAS::Service::service_url 'ShowDocument';
    print OAS::CGI::html_header('Open Document', '/css/open_document.css')
        . qq|
        <script type="text/javascript">
            function load_document(id)
            {
                var url = "$url?id=" + id;
                opener.location.href=url;
                self.close();
            }
        </script>
        <p style="text-align:center">Select File to Open:</p>
        <div style="width:430px;height:120px;border-style:inset;overflow:auto;">
        <table width="100%">
        <tbody>
        <tr>
|    . sort_header('Date', 'timestamp', $sort)
|    . sort_header('Title', 'title', $sort)
|    . sort_header('Author', 'author', $sort)
        . qq|
            <th> Access </th>
            </tr>
|;
}
#
#-----
#
sub build_table ($)
{
    my $sort = $_[0];
    my $docs = OAS::Auth::list_accessible_documents OAS::Auth::current_user, $sort;

    # Process results
    foreach my $ref (@$docs) {
        my $timestamp = &timestamp($ref->{'timestamp'});
        my $acl       = join ' ', sort @{$ref->{'acl'}};
        my $title     = qq|
            <a class="title" href="javascript:load_document('$ref->{'id'}')"
            onClick="load_document('$ref->{'id'}')" target="_main">
            $ref->{'title'}
            </a>
|;
        print qq|
            <tr>
            <td class="timestamp">$timestamp</td>
            <td class="title">$title</td>
            <td class="author">$ref->{'author'}</td>
            <td class="acl">$acl</td>
            </tr>
|;
    }
}

```

```

#
#-----
#
sub build_footer ()
{
    print qq|
        </tbody>
        </table>
        </div>
        <div style="text-align:center"><p>
    | . OAS::CGI::close_window_form
    | . "</p></div>\n"
    | . OAS::CGI::html_footer;
}
#
#-----
#
sub sort_header ($$$)
{
    my $heading = ( defined $_[0] ) ? $_[0] : '&nbsp;';
    my $sort_col = ( defined $_[1] ) ? $_[1] : DEFAULT_SORT;
    my $a_sort   = ( defined $_[2] ) ? $_[2] : '';
    my $sort     = ( $sort_col eq $a_sort ) ? $sort_col.'%20DESC' : $sort_col;
    my $url      = OAS::Service::service_url 'OpenDocument';

    qq| <th><a class="heading" href="$url?ob=$sort">$heading</a></th>\n|;
}
#
#-----
#
sub timestamp ($)
{
    my $time = (defined $_[0]) ? $_[0] : time;
    my @ta   = localtime $time;
    return sprintf(
        '%d-%s-%d<br />%d:%.2d:%.2d&nbsp;%s',
        $ta[3],
        qw(Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec)[$ta[4]],
        $ta[5] + 1900,
        (0 == $ta[2] % 12) ? 12 : $ta[2] % 12,
        $ta[1],
        $ta[0],
        (12 <= $ta[2]) ? 'PM' : 'AM'
    );
}
#
#-----
1;

```

## C.21. ReadDocument.pm

```

package OAS::Apache2::ReadDocument;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE    = '05 Aug 2006';

use strict;
use Apache2::Const qw(OK);
use Apache2::RequestRec ();
use CGI          ();
use OAS::System  ();
use OAS::CGI     ();
use OAS::Annotation ();
use OAS::Document ();

```

```

use OAS::Menu      ();
use OAS::Auth      ();
use OAS::Person    ();
=====
# MAIN
#

sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    setup_display(_init());
    print OAS::CGI::html_footer;
    return OK;
}
#
=====

#-----
#
sub _init ()
{
    my $is_error      = 0;
    my $error_message = '';
    my $cgi           = new CGI;
    my $doc           = new OAS::Document;
    my $id            = $cgi->param('id');

    # Check document errors
    if (not OAS::Document::is_valid_document_id $id) {
        $is_error++;
        $error_message .= "<li>Invalid Document ID</li>\n";
    } elsif (not $doc->Load($id)) {
        $is_error++;
        $error_message .= "<li>Cannot Load Document #$id</li>\n";
    }

    # Check user errors
    my $user = OAS::Auth::current_user;
    if (not OAS::Person::is_valid_person_id $user) {
        $is_error++;
        $error_message .= "<li>Invalid User ID</li>\n";
    } elsif (not OAS::Auth::has_read_annotation_access $user, $id) {
        $is_error++;
        $error_message .=
            "<li>$user does not have read access to Document $id</li>\n";
    }

    # Display Error, if present, and exit
    if ($is_error) {
        print qq|
            <h1>ERROR</h1>
            <ul>
                $error_message
            </ul>
        |;
        warn $error_message;
        exit;
    }

    # Set Title
    my $title = 'OAS: ' . $doc->title . " [#$id]";

    # Reset Page, if necessary
    my $page = 1;
    if (defined $cgi->param('p')) {

```



```

        my $p      = $cgi->param('p');
        $page = $p if $p =~ /\d+$/;
    }
    $page = $doc->total_pages if $page > $doc->total_pages;

    return ($doc, $page, $title, $user);
}
#-----
#
sub setup_display ($$$)
{
    my ($doc, $page, $title, $user) = @_;

    my $src      = OAS::System::path_to_url $doc->page_image($page);
    my $offset_x = OAS::Menu::SIDEBAR_WIDTH() + OAS::Menu::BORDER_SIZE();
    my $offset_y = OAS::Menu::MENUBAR_HEIGHT() + OAS::Menu::INFOBAR_HEIGHT()
        + OAS::Menu::BORDER_SIZE();
    my $width    = $doc->page_width( $page ) + $offset_x + OAS::Menu::BORDER_SIZE;
    my $height   = $doc->page_height( $page ) + $offset_y + OAS::Menu::BORDER_SIZE;

    # Header Information
    print OAS::CGI::html_header(
        $title,
        '/css/oas.css',
        qq|
            <style type="text/css">
            body
            {
                background-image: url("$src");
                background-attachment: scroll;
                background-color: black;
                background-position: |.$offset_x.'px' |.$offset_y.qq|px;
                background-repeat: no-repeat;
            }
            </style>
        |,
        qq|onLoad="div_show('menuSidebar')"|
    );

    # Top Bars
    print OAS::Menu::menubar $doc, $page;
    print OAS::Menu::infobar $doc, $page;

    # Setup Annotations
    my $annotations = OAS::Annotation::annotation_list $doc->id, $page;
    my $annotation = new OAS::Annotation;
    my $a_path     = OAS::Document::document_id_to_annotation_path $doc->id;
    my @a_list     = ();

    foreach my $key (sort keys %{$annotations}) {
        if ( ( $user eq $annotations->{$key}{'owner'} ) or
            ( OAS::Auth::has_read_annotation_access $user, $doc->id ) )
        {
            # Print annotation <div>
            $annotation->Load($annotations->{$key}{'id'});

            my $a_id      = $annotation->id;
            my $a_width  = $annotation->width + 10;
            my $a_height = $annotation->height + 10;
            my $a_x_pos  = $annotation->x_pos + $offset_x;
            my $a_y_pos  = $annotation->y_pos + $offset_y;
            my $style    = 'position:absolute;'
                . "top:$a_y_pos" . 'px;'
                . "left:$a_x_pos" . 'px;'
                . "width:$a_width" . 'px;'
                . "height:$a_height" . 'px;'
                . 'overflow:auto;';
        }
    }
}

```

```

my $html      = $annotation->html;

print qq|
    <div id="a$a_id" name="a$a_id"
        style="$style">
        $html
    </div>
|;

push @a_list, $a_id;
}
}

# Sidebar
print OAS::Menu::sidebar $doc, $page, 0, @a_list;
}
#
#-----
1;

```

## C.22. ShowDocument.pm

```

package OAS::Apache2::ShowDocument;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE    = '04 Aug 2006';

use strict;
use Apache2::Const      qw(OK);
use Apache2::RequestRec ();
use CGI                 ();
use OAS::CGI            ();
use OAS::Document      ();
use OAS::Menu           ();
use OAS::Auth           ();
use OAS::Person        ();

#=====
# MAIN
#
sub handler
{
    my $r = shift;
    $r->content_type('text/html');
    my $doc = new OAS::Document;
    my ($page, $title) = &_init($doc);
    setup_display($doc, $page, $title);
    display_page($doc, $page, $title);
    print OAS::CGI::html_footer;
    return OK;
}
#
#=====

#-----
#
sub _init ($)
{
    my $doc          = $_[0];
    my $is_error     = 0;
    my $error_message = '';

```

```

my $cgi          = new CGI;
my $id           = $cgi->param('id');
my $user         = OAS::Auth::current_user;
my $page        = 1;
my $title       = '';

# Check document errors
if (not OAS::Document::is_valid_document_id $id) {
    $is_error++;
    $error_message .= "<li>Invalid Document ID</li>\n";
} elsif (not $doc->Load($id)) {
    $is_error++;
    $error_message .= "<li>Cannot Load Document #\$id</li>\n";
}

# Check user errors
if (not OAS::Person::is_valid_person_id $user) {
    $is_error++;
    $error_message .= "<li>Invalid User ID</li>\n";
} elsif (not OAS::Auth::has_read_access $user, $id) {
    $is_error++;
    $error_message .=
        "<li>\$user does not have read access to Document \$id</li>\n";
}

# Display Error, if present, and exit
if ($is_error) {
    print qq|
        <h1>ERROR</h1>
        <ul>
        $error_message
        </ul>
    |;
    warn $error_message;
    exit;
}

# Set Title
$title = 'OAS: ' . $doc->title . " [#\$id]";

# Reset Page, if necessary
if (defined $cgi->param('p')) {
    my $p = $cgi->param('p');
    $page = $p if $p =~ /\d+$/;
}
$page = $doc->total_pages if $page > $doc->total_pages;

return ($page, $title);
}
#
#-----
#
sub setup_display ($$$)
{
    my ($doc, $page, $title) = @_;
    print OAS::CGI::html_header(
        $title,
        '/css/oas.css',
        qq|
            <style type="text/css">
                body
                {
                    background-color: black;
                }
            </style>
        |
    );
    print OAS::Menu::menubar $doc, $page;
}

```

```

        print OAS::Menu::infobar $doc, $page;
    }
    #
    #-----
    #
    sub display_page ($$$)
    {
        my ($doc, $page, $title) = @_ ;

        my $src = OAS::System::path_to_url $doc->page_image($page);
        my $top = ( OAS::Menu::MENUBAR_HEIGHT()
            + OAS::Menu::INFOBAR_HEIGHT()
            + OAS::Menu::BORDER_SIZE() ) . 'px';
        my $left = OAS::Menu::BORDER_SIZE() . 'px';
        print qq|
            <div style="position:absolute;top:$top;left:$left;">
                
            </div>
        |;
    }
    #
    #-----
    1;

```

### C.23. UserProfile.pm

```

package OAS::Apache2::UserProfile;
require 5.8.0;
our $VERSION = 1.0.0;
our $DATE = '05 Aug 2006';

use strict;
use Apache2::Const qw(OK);
use Apache2::RequestRec ();
use CGI ();
use OAS::CGI ();
use OAS::Person ();
use OAS::Auth ();
use OAS::Menu ();

#####
# CONSTANTS
#
use constant DEFAULT_PASSWORD => ' ';
#
#####

=====
# MAIN
#
sub handler ()
{
    my $r = shift;
    $r->content_type('text/html');
    print OAS::CGI::html_header('User Profile','/css/oas.css');
    print OAS::Menu::menubar;
    my $person = new OAS::Person;
    my $params = &_init($person);
    if (not exists $params->{s}) { display_user_profile($person) }
    elsif ('Update' eq $params->{s}) { update_user_profile($person, $params) }
    else { error_message('Unknown state') }
    print OAS::CGI::html_footer;
}

```

```

        return OK;
    }

#
#-----

#-----
#
sub _init ($$)
{
    my ($person) = @_;

    # Get parameters from CGI
    my $cgi      = new CGI;
    my %params = ();
    map { $params{$_} = $cgi->param($_) } $cgi->param;

    # Setup other items
    my $id      = (exists $params{id}) ? $params{id} : undef;
    my $user    = OAS::Auth::current_user;

    # Validate inputs
    error_message('Missing or invalid Person ID')
        unless OAS::Person::is_valid_person_id $id;
    error_message("Unable to load Person ID #$_id")
        unless $person->Load($id);
    error_message("$user does not have rights to edit $id")
        unless (($id eq $user) or (OAS::Auth::is_admin $user));

    return \%params;
}
#
#-----
#
sub display_user_profile ($,$%)
{
    my ($Person, $message, %data) = @_;

    $message = (defined $message) ? $message : '';
    my $id    = $Person->id;
    my $name  = $Person->first_name . ' ' . $Person->last_name;
    my %person;

    foreach my $key qw(
        first_name last_name address city state zip phone fax mobile email
    ) {
        eval qq|\$person{\$key} = \$Person->{$key}|;
    }
    foreach my $key (keys %data) {
        $person{$key} = $data{$key};
    }

    print qq|
<div id="updateForm">
<table border="0" width="100%" height="100%">
  <tbody><tr><td align="center" valign="center">

    $message

    <form name="update_form" method="POST">
<input type="HIDDEN" name="id" value="$id" />
<input type="HIDDEN" name="s" value="Update" />
<table cellpadding="5" cellspacing="1" border="1">
<tbody>
<tr>
  <th class="updateForm">User Profile for $name [$id]</th>
</tr>

```

```

<tr>
  <td align="center">
    <table cellpadding="1" cellspacing="1" border="0">
      <tbody>
| . _build_update_row(
      'First Name',
      'first_name',
      $person{first_name},
      'TEXT',
      24 )
. _build_update_row(
      'Last Name',
      'last_name',
      $person{last_name},
      'TEXT',
      24,
      48 )
. _build_update_row(
      'Address',
      'address',
      $person{address},
      'TEXT',
      36,
      48 )
. _build_update_row(
      'City',
      'city',
      $person{city},
      'TEXT',
      24,
      16 )
. _build_update_row(
      'State',
      'state',
      $person{state},
      'TEXT',
      2 )
. _build_update_zip_row(
      'Zip',
      'zip',
      $person{zip} )
. _build_update_phone_row(
      'Phone',
      'phone',
      $person{phone} )
. _build_update_phone_row(
      'Fax',
      'fax',
      $person{fax} )
. _build_update_phone_row(
      'Cell',
      'mobile',
      $person{mobile} )
. _build_update_row(
      'Email',
      'email',
      $person{email},
      'TEXT',
      48,
      64 )
. _build_update_row(
      'Password',
      'password',
      DEFAULT_PASSWORD,
      'PASSWORD',
      16 )

```

```

        . _build_update_row(
            'Retype',
            'password2',
            DEFAULT_PASSWORD,
            'PASSWORD',
            16 )
    . qq|
        <tr>
        <td align="center" colspan="2">
            | . _input_field('submit', 'Update Profile', 'SUBMIT') . qq|
        </td>
        </tr>
    </tbody>
</table>
</td>
</tbody>
</table>
</form>

</td></tr></tbody></table>
</div>
|;
}
#
#-----
#
sub _input_field ($$;$$$)
{
    my ($name, $value, $type, $size, $max) = @_;
    $value = ( defined $value ) ? $value : '';
    $type = ( defined $type ) ? uc $type : 'TEXT';
    $max = ( defined $max ) ? $max : $size;

    my $size_html = (defined $size) ? qq| size="$size" maxlength="$max"| : '';
    qq|<input type="$type" name="$name" value="$value"$size_html />|;
}
#
#-----
#
sub _build_update_row ($$$;$$$)
{
    my $tag = ucfirst shift;
    my $input = &_input_field(@_);
    qq|
        <tr>
        <td class="updateFormRight"> $tag: </td>
        <td class="updateFormLeft" > $input </td>
        </tr>
    |;
}
#
#-----
#
sub _build_update_zip_row ($$$)
{
    my ($tag, $name, $value) = @_;
    my ($n1, $n2, $v1, $v2);
    $tag = ucfirst $tag;
    $n1 = $name . '1';
    $n2 = $name . '2';

    if ((defined $value) and ($value =~ /\d{5}(\d{4})?$/)) {
        ($v1, $v2) = ($1, $2);
    } else {
        $v1 = $v2 = '';
    }
}

```

```

}

qq|
    <tr>
        <td class="updateFormRight"> $tag: </td>
        <td class="updateFormLeft">
            | . _input_field($n1, $v1, 'TEXT', 5) . qq| -
            | . _input_field($n2, $v2, 'TEXT', 4) . qq|
        </td>
    </tr>
|;
}
#
#-----
#
sub _build_update_phone_row ($$$)
{
    my ($tag, $name, $value) = @_ ;
    my ($n1, $n2, $n3, $v1, $v2, $v3);
    $tag = ucfirst $tag;
    $n1 = $name . '1';
    $n2 = $name . '2';
    $n3 = $name . '3';

    if ((defined $value) and ($value =~ /^(\d{3})(\d{3})(\d{4})$/)) {
        ($v1, $v2, $v3) = ($1, $2, $3);
    } else {
        $v1 = $v2 = $v3 = '';
    }

    qq|
        <tr>
            <td class="updateFormRight"> $tag: </td>
            <td class="updateFormLeft">
                ( | . _input_field($n1, $v1, 'TEXT', 3) . qq| )
                | . _input_field($n2, $v2, 'TEXT', 3) . qq| -
                | . _input_field($n3, $v3, 'TEXT', 4) . qq|
            </td>
        </tr>
    |;
}
#
#-----
#
sub update_user_profile ($$)
{
    # Get data
    my ($Person, $params) = @_ ;
    my %person;
    foreach my $key qw(
        first_name last_name
        address city state zip1 zip2
        phone1 phone2 phone3
        fax1 fax2 fax3
        mobile1 mobile2 mobile3
        email password password2 )
    {
        $person{$key} = $params->{$key};
    }

    # Validate data
    unless ( ( OAS::Person::is_valid_first_name $person{ 'first_name' } ) and
        ( OAS::Person::is_valid_last_name $person{ 'last_name' } ) )
    {
        my $html = "<table><tbody><tr><td>Required information is missing:<ul>\n";
        $html .= "<li>First Name</li>\n";
        unless OAS::Person::is_valid_first_name $person{'first_name'};
    }
}

```



```

        $html .= "<li>Last Name</li>\n"
        unless OAS::Person::is_valid_last_name $person{'last_name'};
        $html .= "</ul></td></tr></tbody></table>\n";
display_user_profile($Person, $html, %person);
return;
}
unless ( (defined $person{'password'}) and (defined $person{'password2'}) and
($person{'password'} eq $person{'password2'}) )
{
    my $html = "<table><tbody>"
        . "<tr><td>Passwords do not match</td></tr>"
        . "</tbody></table>\n";
display_user_profile($Person, $html, %person);
return;
}

# Update Simple Values
foreach my $key qw(first_name last_name address city state email ) {
    eval qq|\$Person->$key(\$person{\$key})|;
}

# Update Zip
$Person->zip($person{'zip1'} . $person{'zip2'});

# Update Phones
foreach my $key qw(phone fax mobile) {
    eval "\$Person->$key(\$person{". $key . '1'}. $person{'. $key . '2'}. $person{'. $key . '3'})";
}

# Update Password
$Person->password($person{'password'}) if
(DEFAULT_PASSWORD ne $person{'password'});

# Save
$Person->Save or error_message($Person->error_message);

display_user_profile($Person);
}
#
#-----
#
sub error_message ($)
{
    my $message = (defined $_[0]) ? $_[0] : '<unknown>';

    print qq|
        <h1>Error:</h1>
        <ul>
            <li>$message</li>
        </ul>
    |. OAS::CGI::html_footer;
    exit;
}
#
#-----
1;

```

## Appendix D – Converter Source Code

This appendix contains the source code for all the converters used to do document to graphic rendering. More information on this process is found in Section 4.2.2.

### D.1. copy

This converter is used for documents that only need to be copied into the system (i.e., graphics).

```
#!/usr/bin/perl -w

use warnings;
use strict;
use OAS::Document qw/is_valid_document_id/;
use OAS::Utils qw/:files/;
use OAS::Converter qw/:errors/;
use Image::Size;

my $in_file = shift;
my $out_path = shift;
my $id = shift;
my $doc = new OAS::Document;

my $in_filename = file_name $in_file;
my $in_path = file_path $in_file;

conversion_fail('Invalid In Filename', CONVERSION_ERROR_FILENAME)
    unless is_valid_filename $in_filename;
conversion_fail('Invalid In Path', CONVERSION_ERROR_FILENAME)
    unless is_valid_path $in_path;
conversion_fail('Invalid Out Path', CONVERSION_ERROR_PATH)
    unless is_valid_path $out_path;
conversion_fail('Invalid Document ID', CONVERSION_ERROR_ID)
    unless is_valid_document_id $id;
conversion_fail('Unknown Document ID', CONVERSION_ERROR_DATABASE)
    unless $doc->Load($id);

my $extension = file_extension $in_file;
my $filename = $out_path.'/'.$id.'-0000.'.$extension;
$filename =~ s|//|/|g;

conversion_fail("Copy Failed: $!", CONVERSION_ERROR_EXTERNAL)
    if system 'cp', $in_file, $filename;
```

```

my ($width, $height, $stream) = imgsize($filename);
$width = (defined $width) ? $width : 0;
$height = (defined $height) ? $height : 0;

my $page_number = $doc->add_page;
$doc->page_image( $page_number, $filename );
$doc->page_width( $page_number, $width );
$doc->page_height( $page_number, $height );
conversion_fail('Save Failed: '.$doc->error_message, CONVERSION_ERROR_DATABASE)
    unless $doc->Save;
exit CONVERSION_ERROR_NONE;

```

## D.2. gs\_wrap

This shell script performs the actual *Ghostscript* conversion used to convert *Postscript* or *PDF* files to a graphic set.

```

#!/bin/bash
#
# Command-line Arguments:
#
# 1: Input filename      [fully-qualified]
# 2: Output filename stem [fully-qualified]
# 3: Resolution          [e.g. 72]
#
if [ $# -ne 3 ]
then
    echo ERROR: Missing Parameters
    exit 1
fi
gs -dSAFER -dBATCH -dNOPAUSE -q -dQUIET -dJPEGQ=100 -sDEVICE=jpeg -r$3 -dTextAlphaBits=4
-sOutputFile=$2-%04d.jpg $1 >/dev/null
exit 0;

```

## D.3. image

This is another copy of the script found in Appendix D.1, using a different name.

The functionality is the same.

```

#!/usr/bin/perl -w

use warnings;
use strict;
use OAS::Document qw/is_valid_document_id/;
use OAS::Utils qw/:files/;
use OAS::Converter qw/:errors/;
use Image::Size;
use Image::Magick;

my $in_file = shift;
my $out_path = shift;
my $id = shift;
my $doc = new OAS::Document;

```

```

my $in_filename = file_name $in_file;
my $in_path     = file_path $in_file;

conversion_fail('Invalid In Filename', CONVERSION_ERROR_FILENAME)
    unless is_valid_filename $in_filename;
conversion_fail('Invalid In Path', CONVERSION_ERROR_FILENAME)
    unless is_valid_path $in_path;
conversion_fail('Invalid Out Path', CONVERSION_ERROR_PATH)
    unless is_valid_path $out_path;
conversion_fail('Invalid Document ID', CONVERSION_ERROR_ID)
    unless is_valid_document_id $id;
conversion_fail('Unknown Document ID', CONVERSION_ERROR_DATABASE)
    unless $doc->Load($id);

my $filename = $out_path.'/'.$id.'-0000.jpg';
$filename =~ s|//|/|g;

conversion_fail("Copy Failed: $!", CONVERSION_ERROR_EXTERNAL) if system 'convert',
    $in_file, $filename;

my ($width, $height, $stream) = imgsize($filename);
$width = (defined $width) ? $width : 0;
$height = (defined $height) ? $height : 0;

my $page_number = $doc->add_page;
$doc->page_image( $page_number, $filename );
$doc->page_width( $page_number, $width );
$doc->page_height( $page_number, $height );
conversion_fail('Save Failed: '.$doc->error_message, CONVERSION_ERROR_DATABASE)
    unless $doc->Save;
exit CONVERSION_ERROR_NONE;

```

#### D.4. oas\_conv\_server.pl

This is the source code for the *Win32* server that handled the creation of *Postscript* files. This is described in Section 4.2.2.3. The client is in Appendix D.9.

```

#!/usr/bin/perl

use strict;
use warnings;
use IO::Socket;
use Win32::OLE;

#=====
# CONSTANTS & GLOBALS
#
use constant wdPrintAllDocument    => 0;
use constant wdPrintDocumentContent => 0;
use constant wdPrintAllPages      => 0;

my $PORT      = 9000;
my $TEMP_DIR  = "C:\\test\\t3mp"; # NO TRAILING \s
my $LOG_CONSOLE = 1;
my $LOG_FILE   = 0;
#
#=====

```

```

#=====
# MAIN
#
    &log('Server Starting');
    &listen_for_connections(&setup_port);
    exit;
#
#=====

#-----
#
sub setup_port ()
{
    &log("Attempting to bind to port $PORT");
    my $sock = IO::Socket::INET->new(
        PeerAddr => 'oas.et.byu.edu',
        Listen   => 20,
        LocalPort => $PORT,
        Proto    => 'tcp',
        Type     => SOCK_STREAM
    ) or die "Cannot Bind to Port $PORT";
    &log("Bound to port $PORT");
    return $sock;
}
#
#-----
#
sub listen_for_connections ($)
{
    my $sock = $_[0];
    die &error('Invalid socket') unless (defined($sock) and $sock);
    &log('Waiting for connections');
    while (my $con = $sock->accept) {
        my ($port, $iaddr) = sockaddr_in($con->peername);
        my $peer = inet_ntoa($iaddr);
        $peer = (defined $peer) ? $peer : '<unknown>';
        &log("New Connection from $peer:$port");
        &process_file($con);
        $con->close;
        &log('Connection Closed');
    }
}
#
#-----
#
sub process_file ($)
{
    my $con = $_[0];
    return &error('Invalid connection') unless (defined($con) and $con);

    # Receive info from client
    my ($name, $ext, $size, $file);
    {
        $name = <$con>;
        return &error("Filename: $!") unless defined $name;
        $name =~ s/\s//g;
        return &error('Invalid filename') unless $name =~ /^[w\.\-]+$/;
        print $con 'OK'.v13.10;
    }
    {
        $ext = <$con>;
        return &error("Extension: $!") unless defined $ext;
        $ext =~ s/\s//g;
        return &error('Invalid extension') unless $ext =~ /^[a-z\d]+$/i;
        print $con 'OK'.v13.10;
    }
    {
        $size = <$con>;
        return &error("Size: $!") unless defined $size;
    }
}

```

```

    $size =~ s/\s//g;
    return &error('Invalid size') unless $size =~ /\d{1,10}$/;
    print $con 'OK'.v13.10;
}

my $l_file = 0;
# TODO Fix infinite loop here
while ($l_file < $size) {
    $file .= <$con>;
    $l_file = length $file;
}
chop $file; chop $file;
$l_file = length $file;
return &error('Actual size does not match expected size')
    unless $size == $l_file;
print $con 'OK'.v13.10;
}

&log("Processing [$name|$ext|$size]");

# Build temporary file
$name =~ s/\W/~//g;
$name =~ /([\w~]+)/;
$name = "$l.$ext";
open(OUT, ">$TEMP_DIR\\$name")
    or return &error("Tempfile [$TEMP_DIR\\$name]: $!");
binmode OUT;
print OUT $file;
close OUT;

# Process file
if ('csv' eq $ext) { &process_Excel ($name) }
elsif ('doc' eq $ext) { &process_Word ($name) }
elsif ('ppt' eq $ext) { &process_PowerPoint ($name) }
elsif ('rtf' eq $ext) { &process_Word ($name) }
elsif ('wpd' eq $ext) { &process_Word ($name) }
elsif ('xls' eq $ext) { &process_Excel ($name) }
else {
    return &error("Unknown file extension: $ext");
}

# Send converted .ps file
if (open(PS, "$TEMP_DIR\\$name.ps")) {
    binmode PS;
    local $/ = undef;
    my $ps = <PS>;
    close PS;
    my $l_ps = length $ps;
    &log("Sending [ps|$l_ps]");
    print $con $l_ps, v13.10;
    print $con $ps, v13.10;
    &log('Processing complete');
} else {
    &error('Processing Failed');
}

# Delete temporary files
unlink "$TEMP_DIR\\$name" or &error("Unlink File: $!");
unlink "$TEMP_DIR\\$name.ps" or &error("Unlink PS: $!");
}
#
#-----
#
sub process_Excel ($)
{
    my $name = $_[0];
    die ('Missing filename') unless defined ($name) and $name;
    &log('Converting Excel format to .ps');
    my $excel = Win32::OLE->new('Excel.Application', 'Quit');

```

```

$excel->{'Visible'} = 0;
if ($excel->Workbooks->Open("$TEMP_DIR\\$name")) {
    $excel->ActiveWorkbook->PrintOut({
        PrintToFile => 1,
        PrToFileName => "$TEMP_DIR\\$name.ps",
    });
    $excel->ActiveWorkbook->Save;
    $excel->ActiveWorkbook->Saved(1);
} else {
    return &error('Unable to open spreadsheet: ' . Win32::OLE->LastError());
}
}
#
#-----
#
sub process_PowerPoint ($)
{
#TODO FIX THIS FUNCTION
my $name = $_[0];
die ('Missing filename') unless defined($name) and $name;
&log('Converting PowerPoint format to .ps');
my $ppt = Win32::OLE->new('PowerPoint.Application', 'Quit');
$ppt->{'Visible'} = 1;
if ($ppt->Presentations->Open("$TEMP_DIR\\$name")) {
    $ppt->ActivePresentation->PrintOut({
        PrintToFile => "$TEMP_DIR\\$name.ps",
    }) or die Win32::OLE->LastError();
    $ppt->ActivePresentation->Saved(1);
} else {
    return &error('Unable to open document: ' . Win32::OLE->LastError());
}
}
#
#-----
#
sub process_Word ($)
{
my $name = $_[0];
die ('Missing filename') unless defined($name) and $name;
&log('Converting Word format to .ps');
my $word = Win32::OLE->new('Word.Application', 'Quit');
$word->{'Visible'} = 0;
if ($word->Documents->Open("$TEMP_DIR\\$name")) {
    $word->ActiveDocument->PrintOut({
        Background => 0,
        Append => 0,
        Range => wdPrintAllDocument,
        Item => wdPrintDocumentContent,
        Copies => 1,
        PageType => wdPrintAllPages,
        PrintToFile => 1,
        OutputFilename => "$TEMP_DIR\\$name.ps",
    });
    $word->ActiveDocument->Save;
    $word->ActiveDocument->Saved(1);
} else {
    return &error('Unable to open document: ' . Win32::OLE->LastError());
}
}
#
#-----
#
sub error ($)
{
my $message = defined($_[0]) ? $_[0] : '<unknown>';
&log("ERROR: $message");
}
#

```

```

#-----
#
sub log ($)
{
    my $message = defined($_[0]) ? $_[0] : '<no entry>';
    my $entry   = &timestamp . " $message\n";
    print $entry if $LOG_CONSOLE;
}
#-----
#
sub timestamp ()
{
    my ($sec, $min, $hour, $day, $mon, $year, undef) = localtime;
    $mon++;
    $year += 1900;
    $sec = "0$sec" if length $sec < 2;
    $min = "0$min" if length $min < 2;
    $hour = "0$hour" if length $hour < 2;
    $day = "0$day" if length $day < 2;
    $mon = "0$mon" if length $mon < 2;
    return "[$year.$mon.$day $hour:$min:$sec]";
}
#-----

```

## D.5. OOo

This script is used to convert *OpenOffice.org* file formats. A description of the process of configuring the *vncserver* is included. The need for *vncserver* is discussed in Section 4.2.2.2. The macro is based off of the DocConverter v.2.0 macros available at [http://sourceforge.net/project/showfiles.php?group\\_id=87718&package\\_id=95532](http://sourceforge.net/project/showfiles.php?group_id=87718&package_id=95532). This section is licensed under the LGPL.

```

#!/usr/bin/perl -w

use strict;
use warnings;
use OAS::Utils qw(:files);
use OAS::Document qw(is_valid_document_id);
use OAS::Converter qw(:errors :resolution);

#|||||
# NOTES
#
# This script's macro is licensed under the LGPL. It is based off of the DocConverter
# script available at:
#
#     http://sourceforge.net/project/showfiles.php?group_id=87718&package_id=95532
#     via http://www.oocomacros.org/
#
# The use of this script requires that the web server user (e.g. apache) not only has
# access to the X display (xhost), but also has been registered as an actual user for
# OpenOffice.org. This involves the following steps:
#

```



```

# 1. Create a home directory for the user
# 2. Give the user shell access
# 3. Login or su as the user
# 4. Execute the OpenOffice.org executable and complete the new user portion
# 5. Create the necessary macro
# 6. Logoff as the user
# 7. Remove shell access
#
# The macro is a OpenOffice.org Basic macro as follows:
#
#
# Sub toPDF(strFile as string)
#   Dim oDoc as Object
#   Dim strFilterSubName as String
#
#   strUrl = ConvertToUrl( strFile )
#   oDoc = StarDesktop.loadComponentFromURL( strUrl, "_blank", 0,
#       array(MakePropertyValue("Hidden",false)) )
#
#   If not IsNull(oDoc) Then
#       strFilterSubName = ""
#       ' select appropriate filter
#       If oDoc.SupportsService("com.sun.star.presentation.PresentationDocument") Then
#           strFilterSubName = "impress_pdf_Export"
#       ElseIf oDoc.SupportsService("com.sun.star.sheet.SpreadsheetDocument") Then
#           strFilterSubName = "calc_pdf_Export"
#       ElseIf oDoc.SupportsService("com.sun.star.text.WebDocument") Then
#           strFilterSubName = "writer_web_pdf_Export"
#       ElseIf oDoc.SupportsService("com.sun.star.text.GlobalDocument") Then
#           strFilterSubName = "writer_globaldocument_pdf_Export"
#       ElseIf oDoc.SupportsService("com.sun.star.text.TextDocument") Then
#           strFilterSubName = "writer_pdf_Export"
#       ElseIf oDoc.SupportsService("com.sun.star.drawing.DrawingDocument") Then
#           strFilterSubName = "draw_pdf_Export"
#       ElseIf oDoc.SupportsService("com.sun.star.formula.FormulaProperties") Then
#           strFilterSubName = "math_pdf_Export"
#       ElseIf oDoc.SupportsService("com.sun.star.chart.ChartDocument") Then
#           strFilterSubName = "chart_pdf_Export"
#       Else
#           '
#       EndIf
#   EndIf
#
#   If Len(strFilterSubName) > 0 Then
#       oDoc.storeToUrl( ConvertToUrl( strFile & ".pdf" ),
#           array(MakePropertyValue("FilterName", strFilterSubName),
#               MakePropertyValue("CompressMode", "1" )))
#   EndIf
#
#   oDoc.close(True)
# End Sub
#
# Function MakePropertyValue( Optional cName As String, Optional uValue ) As
#   com.sun.star.beans.PropertyValue
#   oPropertyValue = createUnoStruct( "com.sun.star.beans.PropertyValue" )
#   If Not IsMissing( cName ) Then
#       oPropertyValue.Name = cName
#   EndIf
#   If Not IsMissing( uValue ) Then
#       oPropertyValue.Value = uValue
#   EndIf
#   MakePropertyValue() = oPropertyValue
# End Function
#
#
# It is also necessary to setup a vncserver that the system can log into. The vncserver
# service can be started with chkconfig. However, it needs to be updated to give a
# vncserver user name and display number. vncserver then needs to be run from the home

```

```

# directory of the user while logged in as that user, to create the necessary
# configuration files. The user's ~/.vnc/xstart file should be modified so as not to run
# any unnecessary programs and to grant access to local users via xhost +local:
#
#|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

#=====
# CONSTANTS & GLOBALS
#
my $DEBUG      = 0;
my $Doc        = new OAS::Document;
my $Converter  = new OAS::Converter;
my $Filename   = shift;
my $Path       = shift;
my $ID         = shift;
my $Resolution = shift;
#
#=====

#=====
# MAIN
#
&_init;
&print_file;
#
#=====

#-----
#
sub _init ()
{
    my $in_filename = file_name $Filename;
    my $in_path      = file_path $Filename;

    conversion_fail( 'Invalid In Filename', CONVERSION_ERROR_FILENAME )
        unless is_valid_filename $in_filename;

    conversion_fail( 'Invalid In Path',      CONVERSION_ERROR_FILENAME )
        unless is_valid_path $in_path;

    conversion_fail( 'Invalid Out Path',     CONVERSION_ERROR_PATH      )
        unless is_valid_path $Path;

    conversion_fail( 'Invalid Document ID',  CONVERSION_ERROR_ID          )
        unless is_valid_document_id $ID;

    conversion_fail( 'Unknown Document ID', CONVERSION_ERROR_DATABASE )
        unless $Doc->Load($ID);

    $Resolution = DEFAULT_RESOLUTION unless is_valid_resolution $Resolution;
}
#
#-----
#
sub print_file () {

    # Convert print job
    $ENV{DISPLAY} = ':1.0';
    system 'openoffice.org-2.0',
        '-headless',
        '-nologo',

```

```

        qq|macro:///Standard.OAS.toPDF($Filename)|;
$Converter->Load('pdf');
system $Converter->command, "$Filename.pdf", $Path, $ID, $Resolution;
unlink "$Filename.pdf";
}
#
#-----

```

## D.6. plain\_text

This converter uses *enscript* to convert plain text formats. This is similar to the process used in Appendix D.8, but the command-line arguments passed to *enscript* and the resulting layout are different.

```

#!/usr/bin/perl -w

use strict;
use warnings;
use OAS::Utils      qw(:files);
use OAS::Document  qw(is_valid_document_id);
use OAS::Converter qw(:errors :resolution);

#=====
# CONSTANTS & GLOBALS
#
my $DEBUG      = 0;
my $Doc        = new OAS::Document;
my $Converter  = new OAS::Converter;
my $Filename   = shift;
my $Path       = shift;
my $ID         = shift;
my $Resolution = shift;
#
#=====

#=====
# MAIN
#
&_init;
&print_file;
#
#=====

#-----
#
sub _init ()
{
    my $in_filename = file_name $Filename;
    my $in_path     = file_path $Filename;

    conversion_fail( 'Invalid In Filename', CONVERSION_ERROR_FILENAME )
        unless is_valid_filename $in_filename;
}

```

```

conversion_fail( 'Invalid In Path',      CONVERSION_ERROR_FILENAME )
                unless is_valid_path $in_path;

conversion_fail( 'Invalid Out Path',     CONVERSION_ERROR_PATH      )
                unless is_valid_path $Path;

conversion_fail( 'Invalid Document ID',  CONVERSION_ERROR_ID       )
                unless is_valid_document_id $ID;

conversion_fail( 'Unknown Document ID',  CONVERSION_ERROR_DATABASE )
                unless $Doc->Load($ID);

$Resolution = DEFAULT_RESOLUTION unless is_valid_resolution $Resolution;
}
#
#-----
#
sub print_file () {

    # Convert print job
    system 'enscript',
           '-l',
           '--no-header',
           '--highlight',
           '--print-anyway',
           '--no-job-header',
           '--quiet',
           '--color',
           $Filename,
           "--output=$Filename.ps";
    $Converter->Load('ps');
    system $Converter->command, "$Filename.ps", $Path, $ID, $Resolution;
    unlink "$Filename.ps";
}
#
#-----

```

## D.7. ps\_pdf

This is used to convert *Postscript* and *PDF* files. It handles the database calls and such before passing control to *gs\_wrap*, as described in Appendix D.2.

```

#!/usr/bin/perl -w

use strict;
use warnings;
use IO::Socket;
use OAS::System ();
use OAS::Utils qw(:files);
use OAS::Document qw(is_valid_document_id);
use OAS::Converter qw(:errors :resolution);
use Image::Size;

#-----
# CONSTANTS & GLOBALS
#
my $DEBUG      = 0;
my $Doc        = new OAS::Document;
my $Filename   = shift;
my $Path       = shift;

```

```

        my $ID          = shift;
        my $Resolution = shift;
#
#=====
#
#=====
# MAIN
#
        &init;
        &convert_to_graphics;
        &save_pages;
        exit CONVERSION_ERROR_NONE;
#
#=====
#-----
#
sub init ()
{
    my $in_filename = file_name $Filename;
    my $in_path      = file_path $Filename;

    conversion_fail( 'Invalid In Filename', CONVERSION_ERROR_FILENAME )
        unless is_valid_filename $in_filename;

    conversion_fail( 'Invalid In Path',      CONVERSION_ERROR_FILENAME )
        unless is_valid_path $in_path;

    conversion_fail( 'Invalid Out Path',     CONVERSION_ERROR_PATH      )
        unless is_valid_path $Path;

    conversion_fail( 'Invalid Document ID',  CONVERSION_ERROR_ID        )
        unless is_valid_document_id $ID;

    conversion_fail( 'Unknown Document ID', CONVERSION_ERROR_DATABASE )
        unless $Doc->Load($ID);

    $Resolution = DEFAULT_RESOLUTION unless is_valid_resolution $Resolution;
}
#
#-----
#
sub convert_to_graphics ()
{
    print "Saving Converted File\n"           if $DEBUG;
    print "Converting to Graphics\n"         if $DEBUG;
    my $command = $OAS::System::Conf->home_dir . 'bin/converters/gs_wrap';
    system $command, $Filename, $Path.$ID, $Resolution;
    print "Conversion Complete\n"           if $DEBUG;
}
#
#-----
#
sub save_pages ()
{
    print "Getting List of Files\n"           if $DEBUG;
    opendir(IMAGES, $Path)
        or conversion_fail "Cannot Open Directory: $!",
            CONVERSION_ERROR_FILESYSTEM;
    my @files = grep { not /^\.{1,2}\z/ } readdir IMAGES;
    closedir IMAGES;
    print 'Total Pages: ' . scalar(@files) . "\n" if $DEBUG;

    foreach my $file (sort @files) {
        my ($width, $height, $stream) = imgsize($Path.$file);
        $width = (defined $width) ? $width : 0;
    }
}

```

```

    $height = (defined $height) ? $height : 0;
    print "File $file is $width x $height\n"           if $DEBUG;

    my $page_number = $Doc->add_page;
    print "Adding Page # $page_number\n"             if $DEBUG;
    $Doc->page_image( $page_number, $Path.$file );
    $Doc->page_width( $page_number, $width );
    $Doc->page_height( $page_number, $height );
}

conversion_fail('Save Failed: '.$Doc->error_message, CONVERSION_ERROR_DATABASE)
    unless $Doc->Save;
}
#
#-----

```

## D.8. source\_code

Similar to Appendix D.6, this converts source code utilizing *enscript*. However, this converter outputs in A4 landscape mode, which tends to be better for source code.

Syntax highlighting is also enabled for source code types recognized by *enscript*.

```

#!/usr/bin/perl -w

use strict;
use warnings;
use Image::Magick;
use OAS::Utils    qw(:files);
use OAS::Document qw(is_valid_document_id);
use OAS::Converter qw(:errors :resolution);

#=====
# CONSTANTS & GLOBALS
#
my $DEBUG      = 0;
my $Doc        = new OAS::Document;
my $Converter  = new OAS::Converter;
my $Filename   = shift;
my $Path       = shift;
my $ID         = shift;
my $Resolution = shift;
#
#=====

#=====
# MAIN
#
&_init;
&print_file;
#
#=====

```

```

#-----
#
sub _init ()
{
    my $in_filename = file_name $Filename;
    my $in_path      = file_path $Filename;

    conversion_fail( 'Invalid In Filename', CONVERSION_ERROR_FILENAME )
        unless is_valid_filename $in_filename;

    conversion_fail( 'Invalid In Path',      CONVERSION_ERROR_FILENAME )
        unless is_valid_path $in_path;
    conversion_fail( 'Invalid Out Path',     CONVERSION_ERROR_PATH      )
        unless is_valid_path $Path;

    conversion_fail( 'Invalid Document ID', CONVERSION_ERROR_ID        )
        unless is_valid_document_id $ID;

    conversion_fail( 'Unknown Document ID', CONVERSION_ERROR_DATABASE )
        unless $Doc->Load($ID);

    $Resolution = DEFAULT_RESOLUTION unless is_valid_resolution $Resolution;
}
#
#-----
#
sub print_file () {

    # Convert print job
    system 'enscript',
        '-1',
        '--no-header',
        '--highlight',
        '--print-anyway',
        '--no-job-header',
        '--quiet',
        '--line-numbers',
        '--media=A4',
        '--landscape',
        '--margins=0:5:10:10',
        $Filename,
        "--output=$Filename.ps";
    $Converter->Load('ps');
    system $Converter->command, "$Filename.ps", $Path, $ID, $Resolution;
    unlink "$Filename.ps";

    # Rectify all images
    $Doc->Load($Doc->id);
    foreach my $page (1..$Doc->total_pages) {
        my $ifile = $Doc->page_image($page);
        my $image = new Image::Magick;
        $image->Read($ifile);
        $image->Rotate(degrees=>90);
        $image->Write($ifile);
    }
}
#
#-----

```

## D.9. win32

This is the client for the server found in Appendix D.4. This handles the communication with the *Win32* server and then converts the received *Postscript* file.

```

#!/usr/bin/perl -w

use strict;
use warnings;
use IO::Socket;
use OAS::Utils qw(:files);
use OAS::Document qw(is_valid_document_id);
use OAS::Converter qw(:errors :resolution);
use Image::Size;

=====
# CONSTANTS & GLOBALS
#
my $HOST      = 'oas2.et.byu.edu';
my $PORT      = 9000;
my $DEBUG     = 0;
my $TIMEOUT   = 180;

my $Server    = undef;
my $Orig_File = undef;
my $PS_File   = undef;
my $Doc       = new OAS::Document;
my $Filename  = shift;
my $Path      = shift;
my $ID       = shift;
my $Resolution = shift;
#
=====

#-----
# MAIN
#
$| = 1;
&init;
&read_file;
&connect_to_server;
&send_file;
&receive_conversion;
my $converter = new OAS::Converter;
    $converter->Load('ps');
system $converter->command, "$Filename.ps", $Path, $ID, $Resolution;
unlink "$Filename.ps";
#
=====

#-----
#
sub init ()
{
    my $in_filename = file_name $Filename;
    my $in_path      = file_path $Filename;

    conversion_fail( 'Invalid In Filename', CONVERSION_ERROR_FILENAME )
        unless is_valid_filename $in_filename;

    conversion_fail( 'Invalid In Path',      CONVERSION_ERROR_FILENAME )
        unless is_valid_path $in_path;

    conversion_fail( 'Invalid Out Path',     CONVERSION_ERROR_PATH      )
        unless is_valid_path $Path;

    conversion_fail( 'Invalid Document ID',  CONVERSION_ERROR_ID        )
        unless is_valid_document_id $ID;

    $Resolution = DEFAULT_RESOLUTION unless is_valid_resolution $Resolution;
}

```



```

#
#-----
#
sub read_file ()
{
    local $/ = undef;
    open(IN, $Filename)
        or conversion_fail "Cannot open $Filename: $!", CONVERSION_ERROR_FILESYSTEM;
    binmode IN;

    $Orig_File = <IN>;
    close IN;
}
#
#-----
#
sub connect_to_server ()
{
    print "Attempting to connect to $HOST:$PORT\n" if $DEBUG;
    $Server = IO::Socket::INET->new(
        PeerAddr => $HOST,
        PeerPort => $PORT,
        Proto    => 'tcp',
        Type     => SOCK_STREAM
    ) or conversion_fail "Could not connect to server $HOST:$PORT: $!",
        CONVERSION_ERROR_NETWORK;
    print "Connected to $HOST:$PORT\n" if $DEBUG;
}
#
#-----
#
sub send_file ()
{
    my $response = undef;
    my $file_name = file_name $Filename;
    my $extension = file_extension $file_name;
    my $file_size = length $Orig_File;

    print "Sending Filename: $file_name\n"           if $DEBUG;
    print $Server $file_name, v13.10;
    $response = <$Server>;
    conversion_fail 'Bad Response', CONVERSION_ERROR_NETWORK
        unless $response eq ('OK'.v13.10);

    print "Sending Extension: $extension\n"         if $DEBUG;
    print $Server $extension, v13.10;
    $response = <$Server>;
    conversion_fail 'Bad Response', CONVERSION_ERROR_NETWORK
        unless $response eq ('OK'.v13.10);

    print "Sending File Size: $file_size\n"        if $DEBUG;
    print $Server $file_size, v13.10;
    $response = <$Server>;
    conversion_fail 'Bad Response', CONVERSION_ERROR_NETWORK
        unless $response eq ('OK'.v13.10);

    print "Sending File\n"                         if $DEBUG;
    print $Server $Orig_File, v13.10;
    $response = <$Server>;
    conversion_fail 'Bad Response', CONVERSION_ERROR_NETWORK
        unless $response eq ('OK'.v13.10);
}
#
#-----
#
sub receive_conversion ()
{

```

```

print "Waiting for Converted File\n"                if $DEBUG;

my ($actual_size, $expected_size);
{
    $expected_size = <$Server>;
    $expected_size =~ s/\s//g;
    conversion_fail 'Invalid size', CONVERSION_ERROR_EXTERNAL
        unless $expected_size =~ /\d{1,10}$/;
    print "Expected File Size: $expected_size\n"    if $DEBUG
}

print "Receiving Converted File\n"                if $DEBUG;
$actual_size = 0;
$PS_File     = '';
alarm($TIMEOUT);
while ($actual_size < $expected_size) {
    $PS_File .= <$Server>;
    $actual_size = length $PS_File;
}
alarm(0);
chop $PS_File; chop $PS_File;
$actual_size = length $PS_File;
print "Actual File Size: $actual_size\n"          if $DEBUG;
conversion_fail 'Actual size does not match expected size',
    CONVERSION_ERROR_EXTERNAL
    unless $actual_size == $expected_size;

print "Saving Converted File\n"                    if $DEBUG;
open(PS, ">$Filename.ps")
    or conversion_fail "Cannot Open .ps File: $!", CONVERSION_ERROR_FILESYSTEM;
binmode PS;
print PS $PS_File;
close PS;
}
#
#-----

```



## Appendix E – JavaScript Source Code

This appendix contains the source code for the different external *JavaScript* files used by *OAS*.

### E.1. annotation.js

```
/////////////////////////////////////////////////////////////////
// Name: annotation.js
// Ver.: 1.00
// Date: 02 Aug 2006
/////////////////////////////////////////////////////////////////

//=====
// GLOBALS & CONSTANTS
//
//-----
// Initialization Variables
//
var Document_Width      = 0;
var Document_Height    = 0;
var DOCUMENT_OFFSET_X  = 160;
var DOCUMENT_OFFSET_Y  = 60;
//
//-----
// Settings Variables
//
var Blocking            = false;
var DRAW_MODE          = 0;
var TEXT_MODE          = 1;
var DEFAULT_MODE       = DRAW_MODE;
var Current_Mode       = DEFAULT_MODE;
var Pen = {
    pen_size:           null,
    pen_color:          null,
    font_name:          null,
    font_size:          null,
    font_color:         null,
    font_background:    null
};
var Tracer = {
    div:                'tracerWidget',
    size_x:             null,
    size_y:             null,
    offset_x:           null,
    offset_y:           null,

```

```

        home_x:    null,
        home_y:    null
};
var Mouse = {
    x: 0,
    y: 0
};
//
//-----
// Drawing Variables
//
var Array_X          = null;
var Array_Y          = null;
var Record           = false;
var Record_Start     = 0
var Coordinate_Count = 0;
var MAX_COORDINATE_VALUE = 8192;
var MIN_COORDINATE_VALUE = 0;
var Min_X = MAX_COORDINATE_VALUE;
var Min_Y = MAX_COORDINATE_VALUE;
var Max_X = MIN_COORDINATE_VALUE;
var Max_Y = MIN_COORDINATE_VALUE;
//
//-----
// Trace Variables
//
var TRACE_MIN_STEPS = 20;
var TRACE_MAX_STEPS = 200;
var Trace_Data = {
    timer_id:    null,
    delay:       25, //ms
    index:       0,
    x:           0,
    y:           0,
    steps:       0,
    side:        0,
    step_length: 0,
    x_steps:     0,
    y_steps:     0
};
//
//-----
// Cookie Variables
//
var Cookie_Data = {
    mode:         null,
    pen_size:     null,
    pen_color:    null,
    font_name:    null,
    font_pitch:   null,
    font_color:   null,
    font_background: null,
    use_tracer:   null
}
//
//=====

//-----
// INITIALIZATION FUNCTIONS
//
function document_init(w,h)
{
    // Document Size
    Document_Width  = w;
    Document_Height = h;

```

```

// Pen Setup
set_pointer(Tracer, 10, 10, '/images/ball.gif');

// Layer Setup
div_show('menuSidebar');

// Event Setup
document.onmousemove = mouseMove;
document.onmousedown = mouseDown;
document.onmouseup = mouseUp;
window.onscroll = documentScroll;

// Trace Setup
init_arrays();
Trace_Data.timer_id = setInterval('trace_path()', Trace_Data.delay);

}
//
//-----
//
function init_arrays()
{
    // Build empty arrays
    if (null != Array_X)
    {
        delete Array_X;
        delete Array_Y;
    };
    Array_X = new Array();
    Array_Y = new Array();

    // Load default pen information from HTML
    Array_X[0] = 'PEN_WEIGHT';
    Array_Y[0] = document.settings.pen_size.value;
    Array_X[1] = 'PEN_COLOR';
    Array_Y[1] = document.settings.pen_color.value;
};
//
//-----
//
function set_pointer(this_pointer, size_x, size_y, src)
{
    var offset_x = Math.floor(size_x / 2);
    var offset_y = Math.floor(size_y / 2);
    var div_info = '';

    // Set dimensional data
    this_pointer.size_x = size_x;
    this_pointer.size_y = size_y;
    this_pointer.offset_x = offset_x;
    this_pointer.offset_y = offset_y;
    this_pointer.home_x = Math.floor(DOCUMENT_OFFSET_X / 2) - offset_x;
    this_pointer.home_y = DOCUMENT_OFFSET_Y + offset_y;

    // Update <DIV> information
    div_info = '';
    div_content(this_pointer.div, div_info);
}
//
//-----

```

```

//
function set_pen()
{
    var next_idx = Array_X.length;

    // Get attributes from HTML form and add to list if necessary
    if (Pen.pen_size != document.settings.pen_size.value)
    {
        Pen.pen_size      = document.settings.pen_size.value;
        Array_X[next_idx] = 'PEN_WEIGHT';
        Array_Y[next_idx] = Pen.pen_size;
        next_idx++;
    }
    if (Pen.pen_color != document.settings.pen_color.value)
    {
        Pen.pen_color = document.settings.pen_color.value;
        Array_X[next_idx] = 'PEN_COLOR';
        Array_Y[next_idx] = Pen.pen_color;
        next_idx++;
    }
    Pen.font_name      = document.annotation_text_control.font.value;
    Pen.font_size      = document.annotation_text_control.pitch.value;
    Pen.font_color     = document.annotation_text_control.color.value;
    Pen.font_background = document.annotation_text_control.background.value;
    set_cookies();
}
//
//-----

//-----
// COOKIE FUNCTIONS
//
function set_cookies ()
{
    var cookie = Pen.font_name + '|'
                + Pen.font_size + '|'
                + Pen.font_color + '|'
                + Pen.font_background ;
    createCookie('text',    cookie,                                180);
    createCookie('pen',    Pen.pen_size + '|' + Pen.pen_color,    180);
    createCookie('mode',   mode_to_text(Current_Mode),            180);
    createCookie('tracer', document.settings.use_tracer.checked, 180);
}
//
//-----
//
function read_cookies ()
{
    var cookie_text  = readCookie('text');
    var cookie_pen   = readCookie('pen');
    var cookie_mode  = readCookie('mode');
    var cookie_tracer = readCookie('tracer');
    if (cookie_mode) {
        if (cookie_mode && ('null' != cookie_mode)) {
            Cookie_Data.mode = cookie_mode;
        }
    }
    if (cookie_pen) {
        var pen = cookie_pen.split('|');
        if (pen[0] && ('null' != pen[0])) { Cookie_Data.pen_size = pen[0]; }
        if (pen[1] && ('null' != pen[1])) { Cookie_Data.pen_color = pen[1]; }
    }
    if (cookie_text) {
        var text = cookie_text.split('|');

```

```

        if (text[0] && ('null' != text[0])) {
            Cookie_Data.font_name = text[0];
        }
        if (text[1] && ('null' != text[1])) {
            Cookie_Data.font_pitch = text[1];
        }
        if (text[2] && ('null' != text[2])) {
            Cookie_Data.font_color = text[2];
        }
        if (text[3] && ('null' != text[3])) {
            Cookie_Data.font_background = text[3];
        }
    }
    if (cookie_tracer != null) {
        Cookie_Data.use_tracer = cookie_tracer;
    }
}
//
//-----
//
function update_from_cookies ()
{
    if (Cookie_Data.mode) { set_mode(text_to_mode(Cookie_Data.mode))
}
    if (Cookie_Data.pen_size) {
        document.settings.pen_size.value = Cookie_Data.pen_size;
    }
    if (Cookie_Data.pen_color) {
        document.settings.pen_color.value = Cookie_Data.pen_color;
    }
    if (Cookie_Data.font_name) {
        document.annotation_text_control.font.value = Cookie_Data.font_name;
    }
    if (Cookie_Data.font_pitch) {
        document.annotation_text_control.pitch.value = Cookie_Data.font_pitch;
    }
    if (Cookie_Data.font_color) {
        document.annotation_text_control.color.value = Cookie_Data.font_color;
    }
    if (Cookie_Data.font_background) {
        document.annotation_text_control.background.value = Cookie_Data.font_background;
    }
    if (Cookie_Data.use_tracer != null) {
        document.settings.use_tracer.checked =
            ('true' == Cookie_Data.use_tracer) ? true : false;
    }
    Pen.pen_size      = document.settings.pen_size.value;
    Pen.pen_color     = document.settings.pen_color.value;
    Pen.font_name     = document.annotation_text_control.font.value;
    Pen.font_size     = document.annotation_text_control.pitch.value;
    Pen.font_color    = document.annotation_text_control.color.value;
    Pen.font_backgroud = document.annotation_text_control.background.value;
    set_cookies();
}
//
//-----

//-----
// EVENT HANDLERS
//
function mouseMove(e)
{
    set_mouse_current_coordinates(e);
}

```



```

// Record coordinates if mouse button is pressed and within document area
if (Record && in_document_area(Mouse.x, Mouse.y))
{
    var next_idx = Array_X.length;
    var new_x     = Mouse.x - DOCUMENT_OFFSET_X;
    var new_y     = Mouse.y - DOCUMENT_OFFSET_Y;

    // Sanity check
    with (Math) {
        new_x = min(new_x, MAX_COORDINATE_VALUE);
        new_x = max(new_x, MIN_COORDINATE_VALUE);
        new_y = min(new_y, MAX_COORDINATE_VALUE);
        new_y = max(new_y, MIN_COORDINATE_VALUE);
    }

    // Make assignments
    Array_X[next_idx] = new_x;
    Array_Y[next_idx] = new_y;
    Coordinate_Count++;

    // Check for new min and/or max values
    with (Math) {
        Min_X = min(new_x, Min_X);
        Max_X = max(new_x, Max_X);
        Min_Y = min(new_y, Min_Y);
        Max_Y = max(new_y, Max_Y);
    }
}
//
//-----
//
function mouseDown(e)
{
    if (in_document_area(Mouse.x, Mouse.y)) {
        Record = true;
        Record_Start = Array_X.length;

        // Update Pen Status
        var next_idx = Array_X.length;
        Array_X[next_idx] = 'PEN_DOWN';
        Array_Y[next_idx] = 'PEN_DOWN';
        Coordinate_Count++;

        mouseMove(e);
    }
}
//
//-----
//
function mouseUp(e)
{
    if (Record) {

        // Update Pen Status
        var next_idx = Array_X.length;
        Array_X[next_idx] = 'PEN_UP';
        Array_Y[next_idx] = 'PEN_UP';
        Coordinate_Count++;
    }
    Record = false;
}
//
//-----
//
function documentScroll(e)
{
    if (Array_X.length > Record_Start) {

```

```

        Array_X = Array_X.slice(0, Record_Start);
        Array_Y = Array_Y.slice(0, Record_Start);
    }
    mouseUp(e);
}
//
//-----
//
function set_mouse_current_coordinates(e)
{
    // Check to see if the event can be referenced
    if ( !e ) {
        if ( window.event ) {
            // DOM
            e = window.event;
        } else {
            // CRITICAL ERROR: Cannot reference event
            notify_failure("set_mouse_current_coordinates (event)");
            return;
        }
    }

    // Determine mouse's current coordinates
    if ( typeof( e.pageX ) == 'number' ) {
        Mouse.x = e.pageX;
        Mouse.y = e.pageY;
    } else {
        if ( typeof( e.clientX ) == 'number' ) {
            Mouse.x = e.clientX;
            Mouse.y = e.clientY;
            if ( !( ( window.navigator.userAgent.indexOf( 'Opera' ) + 1 ) ||
                ( window.ScriptEngine && ScriptEngine().indexOf( 'InScript' ) + 1 ) ||
                window.navigator.vendor == 'KDE' ) )
            {
                if ( document.body &&
                    ( document.body.scrollLeft || document.body.scrollTop ) )
                {
                    // IE 4, 5 & 6 (in non-standards compliant mode)
                    Mouse.x += document.body.scrollLeft;
                    Mouse.y += document.body.scrollTop;
                } else if( document.documentElement &&
                    ( document.documentElement.scrollLeft ||
                    document.documentElement.scrollTop ) )
                {
                    // IE 6 (in standards compliant mode)
                    Mouse.x += document.documentElement.scrollLeft;
                    Mouse.y += document.documentElement.scrollTop;
                }
            }
        }
    } else {
        // CRITICAL ERROR: Cannot get mouse coordinates
        notify_failure("set_mouse_current_coordinates (coordinates)");
        return;
    }
}
//
//-----
//
function add_annotation_form_submit ()
{
    var submit = true;
    var c_mode = mode_to_text(Current_Mode);
    Blocking = true;
    div_hide(Tracer.div);

    // Populate <FORM> with correct data before submission
    document.add_annotation.mode.value = c_mode;
    document.add_annotation.text.value = '1';
}

```

```

document.add_annotation.x_min.value      = Min_X;
document.add_annotation.y_min.value      = Min_Y;
document.add_annotation.x_max.value      = Max_X;
document.add_annotation.y_max.value      = Max_Y;
document.add_annotation.x_data.value     = Array_X.join(',');
document.add_annotation.y_data.value     = Array_Y.join(',');
document.add_annotation.x_offset.value   = DOCUMENT_OFFSET_X;
document.add_annotation.y_offset.value   = DOCUMENT_OFFSET_Y;
document.add_annotation.background.value = 'yellow';
document.add_annotation.referrer.value   = unescape(document.URL);

if ('TEXT' == c_mode) {
    submit = false;
    div_show('annotationText');
    document.annotation_text_control.text.focus();
} else {
    if (0 == document.add_annotation.x_max.value) {
        submit = false;
    }
}

return submit;
}
//
//-----
//
function add_annotation_text_submit ()
{
    // Update <FORM> data
    document.add_annotation.text.value     =
    document.annotation_text_control.text.value;
    document.add_annotation.font.value     =
    document.annotation_text_control.font.value;
    document.add_annotation.font_pitch.value =
    document.annotation_text_control.pitch.value;
    document.add_annotation.font_color.value =
    document.annotation_text_control.color.value;
    document.add_annotation.background.value =
    document.annotation_text_control.background.value;

    // Check annotation size
    if (0 == document.add_annotation.x_max.value) {
        document.add_annotation.x_min.value = 0;
        document.add_annotation.y_min.value = 0;
        document.add_annotation.x_max.value = 200;
        document.add_annotation.y_max.value = 100;
    }

    // Submit annotationform
    document.add_annotation.submit();
}
//
//-----

//-----
// SETTINGS FUNCTIONS
//
function set_mode(new_mode)
{
    switch (new_mode) {
    case DRAW_MODE:
        Current_Mode = DRAW_MODE;
        document.settings.mode[DRAW_MODE].checked = true;
        document.settings.mode.value = mode_to_text(DRAW_MODE);
        break;

```

```

    case TEXT_MODE:
        Current_Mode = TEXT_MODE;
        document.settings.mode[TEXT_MODE].checked = true;
        document.settings.mode.value = mode_to_text(TEXT_MODE);
        break;
    default:
        notify_failure("set_mode (Invalid Mode)");
        set_mode(DEFAULT_MODE);
    }
    set_cookies();
}
//
//-----
//
function mode_to_text(mode_type)
{
    var mode_as_text = '';

    switch (mode_type) {
    case DRAW_MODE:
        mode_as_text = 'DRAW';
        break;
    case TEXT_MODE:
        mode_as_text = 'TEXT';
        break;
    default:
        notify_failure('mode_to_text (Invalid Mode)');
    }

    return mode_as_text;
}
//
//-----
//
function text_to_mode(mode_type)
{
    var text_as_mode = null;

    if ('DRAW' == mode_type) { text_as_mode = DRAW_MODE }
    else if ('TEXT' == mode_type) { text_as_mode = TEXT_MODE }
    else {
        notify_failure('text_to_mode (Invalid Mode Text)');
    }

    return text_as_mode;
}
//
//-----

//-----
// TRACE FUNCTIONS
//
function trace_path()
{
    // Check trace delay
    if (Trace_Data.delay != parseInt(document.settings.trace_speed.value)) {
        Trace_Data.delay = parseInt(document.settings.trace_speed.value);
        clearInterval(Trace_Data.timer_id);
        Trace_Data.timer_id = setInterval('trace_path()', Trace_Data.delay);
    }

    // Only do trace if necessary
    if (Blocking || !document.settings.use_tracer.checked) {
        div_hide(Tracer.div);
        return;
    }
}

```

```

if (0 < Coordinate_Count) {
    var new_x = null;
    var new_y = null;

    // Determine Mode and perform trace
    switch (Current_Mode) {
    case DRAW_MODE:
        Trace_Data.index = Trace_Data.index % Array_X.length;
        if (!isNaN(Array_X[Trace_Data.index]))
        {
            new_x = Array_X[Trace_Data.index];
            new_y = Array_Y[Trace_Data.index];
        }
        break;
    case TEXT_MODE:
        if (1 == Coordinate_Count) {
            // Only one coordinate recorded
            Trace_Data.index = Array_X.length - 1;
            if (isNaN(Array_X[Trace_Data.index])) {
                // Something has been selected since coordinate was
                // choosen
                // Reset and use old values
                new_x = Trace_Data.x + Tracer.offset_x -
                    DOCUMENT_OFFSET_X;
                new_y = Trace_Data.y + Tracer.offset_y -
                    DOCUMENT_OFFSET_Y;
            } else {
                // Coordinate the last in array
                new_x = Array_X[Trace_Data.index];
                new_y = Array_Y[Trace_Data.index];
            }
        } else {
            calculate_steps();
            var current_step = Trace_Data.index;

            if (current_step < Trace_Data.x_steps) {
                // First Side
                Trace_Data.side = 1;
                new_x = Min_X + (current_step *
                    Trace_Data.step_length);
                new_y = Min_Y;
            } else { current_step -= Trace_Data.x_steps;
            if (current_step < Trace_Data.y_steps) {
                // Second Side
                Trace_Data.side = 2;
                new_x = Max_X;
                new_y = Min_Y + (current_step *
                    Trace_Data.step_length);
            } else { current_step -= Trace_Data.y_steps;
            if (current_step < Trace_Data.x_steps) {
                // Third Side
                Trace_Data.side = 3;
                new_x = Max_X - (current_step *
                    Trace_Data.step_length);
                new_y = Max_Y;
            } else { current_step -= Trace_Data.x_steps;
            // Fourth Side
                Trace_Data.side = 4;
                new_x = Min_X;
                new_y = Max_Y - (current_step *
                    Trace_Data.step_length);
            }}}
        }
        break;

```

```

default:
    notify_failure('trace_path (Invalid Mode)');
    set_mode(DEFAULT_MODE);
    return;
}

// Trace to next position
if (null != new_x) {
    Trace_Data.x = new_x - Tracer.offset_x + DOCUMENT_OFFSET_X;
    Trace_Data.y = new_y - Tracer.offset_y + DOCUMENT_OFFSET_Y;
    div_move(Tracer.div, Trace_Data.x, Trace_Data.y);
    div_show(Tracer.div);
}
Trace_Data.index++;
}
}
//-----
//
//
function calculate_steps()
{
    // Determine steps needed for each side
    var x_length, y_length, total_length, slope;

    // Determine lengths of sides
    x_length = Max_X - Min_X;
    y_length = Max_Y - Min_Y;
    total_length = 2 * (x_length + y_length);

    // Determine slope of regression line for steps per length
    slope = (TRACE_MAX_STEPS - TRACE_MIN_STEPS) / MAX_COORDINATE_VALUE;

    with (Math) {
        // Calculate total steps using regression line
        Trace_Data.steps = TRACE_MIN_STEPS + (total_length * slope);
        // Make sure total steps is within TRACE_MAX_STEPS
        Trace_Data.steps = min(TRACE_MAX_STEPS, floor(Trace_Data.steps));
        // Make sure total steps is a multiple of 4
        Trace_Data.steps = Trace_Data.steps + 4 - (Trace_Data.steps % 4);
        // Make sure total steps is within TRACE_MAX_STEPS once more
        if (Trace_Data.steps > TRACE_MAX_STEPS) {
            Trace_Data.steps -= 4;
        }

        // Calculate step_length
        Trace_Data.step_length = floor(2 * (x_length + y_length) /
            Trace_Data.steps);

        // Determine number of steps in each dimension
        Trace_Data.x_steps = floor((x_length * Trace_Data.steps) / total_length);
        Trace_Data.y_steps = floor((Trace_Data.steps - (2 * Trace_Data.x_steps)) /
            2);

        // Set indexes
        Trace_Data.index = Trace_Data.index % Trace_Data.steps;
    }
}
//-----
//
//
function clear_path()
{
    init_arrays();
    Coordinate_Count = 0;
    Trace_Data.index = 0;
    Trace_Data.side = 0;
    Trace_Data.x = 0;
    Trace_Data.y = 0;
}

```

```

Trace_Data.steps = 0;
Min_X = Min_Y = MAX_COORDINATE_VALUE;
Max_X = Max_Y = MIN_COORDINATE_VALUE;
div_hide(Tracer.div);
div_move(Tracer.div, Tracer.home_x, Tracer.home_y);
}
//
//-----

//-----
// UTILITY FUNCTIONS
//
function in_document_area(x, y)
{
    return (x >= DOCUMENT_OFFSET_X) && (y >= DOCUMENT_OFFSET_Y);
}
//
//-----
//
function set_element_text(element_name, new_text)
{
    var element_id = document.getElementById(element_name);
    element_id.firstChild.nodeValue = new_text;
}
//
//-----

```

## E.2. cookie.js

```

////////////////////////////////////
// Name: cookie.js
// Ver.: 1.00
// Date: 01 Aug 2006
// Note: Script provided by: http://www.quirksmode.org/js/cookies.html
//       Used with permission: http://www.quirksmode.org/about/copyright.html
////////////////////////////////////

//-----
// COOKIE FUNCTIONS
//
function createCookie(name,value,days)
{
    if (days)
    {
        var date = new Date();
        date.setTime(date.getTime()+(days*24*60*60*1000));
        var expires = "; expires="+date.toGMTString();
    }
    else var expires = "";
    document.cookie = name+"="+value+expires+"; path=/";
}
//
//-----
//
function readCookie(name)
{
    var nameEQ = name + "=";
    var ca = document.cookie.split(';');

    for(var i=0;i < ca.length;i++)
    {

```

```

        var c = ca[i];
        while (c.charAt(0)==' ') c = c.substring(1,c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length,c.length);
    }
    return null;
}
//
//-----
//
function eraseCookie(name)
{
    createCookie(name,"",-1);
}
//
//-----

```

### E.3. div\_manager.js

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Name: div_manager.js
// Ver.: 1.00
// Date: 17 Jul 2006
// Note: Based on script examples at:
//       - http://www.howtocreate.co.uk/tutorials/index.php?tut=0&part=14
//       - http://www.howtocreate.co.uk/jslibs/htmlhigh/showhidelevels.html
//       These examples are Copyright 2001, Mark Wilton-Jones, used by permission.
//       See http://www.howtocreate.co.uk/jslibs/termsOfUse.html for licensing and
//       distribution terms.
//
//       The script shown here is heavily modified to fit within OAS. Variable and
//       function names have been changed throughout. In addition, the script has been
//       expanded with a large number of additional functions (2006).
//
//       Please visit the listed websites for the original script. Original comments
//       are included per licensing agreement.
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

/*****
          DHTML Show/Hide multi level script written by Mark Wilton-Jones - 2001
*****/

```

Please see <http://www.howtocreate.co.uk/jslibs/> for details and a demo of this script  
Please see <http://www.howtocreate.co.uk/jslibs/termsOfUse.html> for terms of use

To use:

Inbetween the <head> tags, put:

```

<script type="text/javascript" language="javascript1.2">
<!--
//this is optional and therefore does not need to be defined if unwanted
var myalternative = "LOCATION OF ALTERNATIVE PAGE, JUST IN CASE THEIR BROWSER
DOESN'T GET IT RIGHT";
//-->
</script>
<script src="PATH TO SCRIPT/showhidelevels.js" type="text/javascript"
language="javascript1.2"></script>

```

if you haven't got an alternative (You should have one for all those people out there who can't use JavaScript anyway . . .), simply omit the line defining var myalternative;

Now create the bits that you want to show and hide (only one will be shown at a time):



```
<div id="ID OF DIV"
style="position:absolute;left:20px;top:25px;visibility:hidden;">
    contents go in here
</div>
```

You can change the numbers after the 'left:' and 'top:' bits to position it on the page  
The name of the div must contain only letters and numbers (1st character must be a letter)  
or Mozilla gets it wrong

---

To show a div, put:

```
<a href="javascript:showdiv('ID OF DIV',LEVEL);" onmouseover="showdiv('ID OF
DIV',LEVEL)">Link words</a>
```

DO NOT leave out the " and ' bits.  
For the first level produced by the original links, set LEVEL to 1  
For the level produced by the first level links, set LEVEL to 2 etc.

---

To hide the divs that are showing at level LEVEL+, put:

```
<a href="javascript:hideDivsToLevel(LEVEL);"
onmouseover="hideDivsToLevel(LEVEL)">Link Words</a>
```

If a new div is being displayed, the old one will be hidden anyway.

---

To hide a div that is showing but has not been set as showing using the script, put:

```
<a href="javascript:hideDivSpecial('ID OF DIV');" onmouseover="hideDivSpecial('ID
OF DIV'">Link Words</a>
```

DO NOT leave out the " and ' bits.

\*\*\*\*\*  
And here's the actual code  
\*\*\*\*\*/

```
//You will notice that I never actually detect the browser type, I just
//detect its capabilities and then use them. This means that if someone
//uses a browser that I don't know, but is able to use either method, it
//will still work.

//=====
// SETUP & CONSTANTS
//
window.onerror = null;

// initialise all variables

var USE_PX = document.childNodes ? 'px' : '';
//
//=====

//-----
// DIV CONTROL FUNCTIONS
//
function div_reference(div_id, document_id) // getRefToDivNest
{
    if (!document_id)
    {
        document_id = document;
    }
    if (document.layers)
    {
```



```

        if (document_id.layers[div_id])
        {
            return document_id.layers[div_id];
        } else {
            var x, y;
            for (x = 0, y; !y && x < document_id.layers.length; x++)
            {
                y = div_reference(div_id, document_id.layers[x].document);
            }
            return y;
        }
    }

    if (document.getElementById)
    {
        return document.getElementById(div_id);
    }
    if (document.all)
    {
        return document.all[div_id];
    }
    return document[div_id];
}
//
//-----
//
function div_show(this_div) // showdiv
{
    // this function shows the div
    var div_ref = div_reference(this_div);
    if (div_ref.style)
    {
        // DOM compliant
        div_ref.style.visibility = 'visible';
    } else {
        if (div_ref.visibility)
        {
            // Netscape and old versions of Mozilla compliant
            div_ref.visibility = 'show';
        } else {
            // Nothing found, no known way of changing the style
            // CRITICAL ERROR: Cannot show <DIV>
            notify_failure("div_show");
            return;
        }
    }
}
//
//-----
//
function div_hide(this_div) // hideDivSpecial
{
    // hide a div
    var div_ref = div_reference(this_div);
    if (div_ref.style)
    {
        // DOM compliant
        div_ref.style.visibility = 'hidden';
    } else {
        if (div_ref.visibility)
        {
            // Netscape and old versions of Mozilla compliant
            div_ref.visibility = 'hide';
        } else {
            // Nothing found, no known way of changing the style
            // CRITICAL ERROR: Cannot hide <DIV>
            notify_failure("div_hide");
            return;
        }
    }
}

```

```

    }
}
//
//-----
//
function div_move(this_div, new_x, new_y)
{
    var div_ref = div_reference(this_div);
    var div_style = (div_ref.style) ? div_ref.style : div_ref;

    // Move <DIV> to new location
    div_style.left = new_x + USE_PX;
    if (!isNaN(new_y)) {
        div_style.top = new_y + USE_PX;
    }
}
//
//-----
//
function div_move_offset(this_div, offset_x, offset_y)
{
    var div_ref = div_reference(this_div);
    var div_style = (div_ref.style) ? div_ref.style : div_ref;

    // Calculate new location
    var new_x = parseInt(div_style.left) + parseInt(offset_x);
    var new_y = parseInt(div_style.top) + parseInt(offset_y);

    // Move <DIV> to new location
    div_style.left = new_x + USE_PX;
    div_style.top = new_y + USE_PX;
}
//
//-----
//
function div_level(this_div, new_z)
{
    var div_ref = div_reference(this_div);
    var div_style = (div_ref.style) ? div_ref.style : div_ref;

    // Move <DIV> to new Z-index
    div_style.zIndex = parseInt(new_z);
}
//
//-----
//
function div_width(this_div, new_w)
{
    var div_ref = div_reference(this_div);
    var div_style = (div_ref.style) ? div_ref.style : div_ref;

    div_style.width = new_w;
}
//
//-----
//
function div_height(this_div, new_h)
{
    var div_ref = div_reference(this_div);
    var div_style = (div_ref.style) ? div_ref.style : div_ref;

    div_style.height = new_h;
}
//
//-----
//

```

```

function div_content(this_div, new_content)
{
    div_ref = div_reference(this_div);

    if ('undefined' != typeof(div_ref.innerHTML))
    {
        div_ref.innerHTML = new_content;
    } else if( div_ref.document && div_ref.document != window.document ) {
        div_ref.document.open();
        div_ref.document.write(new_content);
        div_ref.document.close();
    } else if( window.frames && window.frames.length &&
        window.frames['nameOfIframe'] )
    {
        div_ref = window.frames['nameOfIframe'].window;
        div_ref.document.open();
        div_ref.document.write(new_content);
        div_ref.document.close();
    }
}
//
//-----
//
function notify_failure(errorMessage)
{
    window.alert( "ERROR: "+ errorMessage );
}
//
//-----

```

## E.4. menu.js

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Name: menu.js
// Ver.: 1.00
// Date: 17 Jul 2006
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//=====
// GLOBALS
//
var Menu_JS      = null;
var Total_Menus = 0;
var NameIdx      = 0;
var ShowIdx      = 1;
//
//=====

//-----
// MENU FUNCTIONS
//
function create_menu (menu_name_array)
{
    var i;
    for (i in menu_name_array) {
        Total_Menus++;
    }
    Menu_JS = new Array(Total_Menus)
    for (i in menu_name_array) {
        Menu_JS[i] = new Array(2);
    }
}

```

```

        Menu_JS[i][NameIdx] = menu_name_array[i]
        Menu_JS[i][ShowIdx] = false;
    }
}
//
//-----
//
function show_menu(idx)
{
    Menu_JS[idx][ShowIdx] = true;
    div_show(Menu_JS[idx][NameIdx]);
}
//
//-----
//

function hide_menu(idx)
{
    Menu_JS[idx][ShowIdx] = false;
    div_hide(Menu_JS[idx][NameIdx]);
}
//
//-----
//

function toggle_menu(idx)
{
    // Toggle selected submenu
    if (Menu_JS[idx][ShowIdx]) { // Currently visible
        hide_menu(idx)
    } else { // Currently hidden
        show_menu(idx)
    }
    // Hide other submenus
    var i
    for (i in Menu_JS) {
        if (i != idx) {
            hide_menu(i)
        }
    }
    // Disable loading of link
    return false;
}
//
//-----
//

function helper_app(url, window_width, window_height, menuIdx)
{
    var top = (screen.availHeight - window_height) / 2;
    var left = (screen.availWidth - window_width) / 2;

    var options =
        'alwaysRaised' +
        'toolbar=no,' +
        'menubar=no,' +
        'status=no,' +
        'width=' + window_width + ',' +
        'height=' + window_height + ',' +
        'top=' + top + ',' +
        'left=' + left

    open(url, 'helperApp', options)

    hide_menu(menuIdx)

    return false;
}
//
//-----

```

```

//-----
// UTILITY MENU FUNCTIONS
//
function window_width ()
{
    var w;

    // Determine window size
    if ( typeof( window.innerWidth ) == 'number' )
    {
        // Non-IE browser
        w = window.innerWidth;

    } else if ( ( document.documentElement ) &&
        ( document.documentElement.clientWidth ||
        document.documentElement.clientHeight ) )
    {
        // IE 6+ in 'standards compliant mode'
        w = document.documentElement.clientWidth;
    } else if ( document.body && ( document.body.clientWidth ||
        document.body.clientHeight ) )
    {
        // IE 4 compatible
        w = document.body.clientWidth;
    }

    return w;
}
//
//-----
//
function window_height ()
{
    var h;

    // Determine window size
    if ( typeof( window.innerWidth ) == 'number' )
    {
        // Non-IE browser
        h = window.innerHeight;
    } else if ( ( document.documentElement ) &&
        ( document.documentElement.clientWidth ||
        document.documentElement.clientHeight ) )
    {
        // IE 6+ in 'standards compliant mode'
        h = document.documentElement.clientHeight;
    } else if ( document.body && ( document.body.clientWidth ||
        document.body.clientHeight ) )
    {
        // IE 4 compatible
        h = document.body.clientHeight;
    }

    return h;
}
//
//-----

```



## Appendix F – CSS Source Code

This appendix includes the source code for the different CSSv2 style sheets used by *OAS*.

### F.1. oas.css

This is the overall style sheet for the system. A monolithic style sheet for the majority of the system was used as it could simply be downloaded by the client once and then cached. It also provides a standardized look and feel for the entire system.

```
/* General */
body
{
background: white;
}
a
{
color: black;
text-decoration: none;
}

/* Main Menu Bar */
#menuSystem
{
position: absolute;
left: 0px;
top: 0px;
z-index: 1001;
height: 25px;
background-color: #D0D0D0;
border-top-style: solid;
border-bottom-style: solid;
border-top-width: thin;
border-bottom-width: thin;
}
a.menuHead
{
font-size: 15px;
height: 25px;
padding-left: 10px;
```



```

padding-right: 10px;
font-family: sans-serif;
}
a.menuHead:hover
{
background-color: #F0F0F0;
}
span.userLoggedIn
{
color: #707070;
font-size: 9px;
font-family: sans-serif;
padding-left: 350px;
padding-right: 5px;
white-space: nowrap;
}

/* Submenu Display */
.submenuSystem
{
position: absolute;
top: 23px;
background-color: #F0F0F0;
width: 100px;
z-index: 1003;
visibility: hidden;
border-style: outset;
overflow: hidden;
}
a.menuItem
{
font-size: 12px;
font-style: normal;
font-family: arial,sans-serif;
padding-left: 2px;
padding-right: 100px;
list-style-type: none;
border-style: none;
width: 50px;
}
a.menuDisabled
{
color: #D0D0D0;
font-size: 12px;
font-style: normal;
font-family: arial,sans-serif;
padding-left: 2px;
padding-right: 100px;
list-style-type: none;
border-style: none;
width: 50px;
cursor: default;
}
a.menuItem:hover
{
border-top-style: dotted;
border-bottom-style: dotted;
border-width: thin;
}
#fileMenu
{
left: 0px;
}
#editMenu
{
left: 42px;
}

```

```

#viewMenu
{
left: 87px;
}
#helpMenu
{
left: 139px;
}

/* Information Bar */
#menuInfobar
{
position: absolute;
top: 24px;
left: 0px;
height: 25px;
z-index: 1002;
background-color: #F0F0F0;
border-bottom-style: solid;
border-bottom-width: thin;
text-align: left;
}
input.pageNavToolbar
{
color: black;
background-color: #F0F0F0;
height: 20px;
font-size: 12px;
font-family: sans-serif;
cursor: pointer;
}
input.pageNavDisabled
{
color: #D0D0D0;
background-color: #F0F0F0;
height: 20px;
font-size: 12px;
font-family: sans-serif;
cursor: default;
}
td.pageNavToolbar
{
font-size: 12px;
font-family: sans-serif;
text-align: center;
white-space: nowrap;
}
span.infoTitle
{
font-size: 14px;
font-family: sans-serif;
padding-left: 1em;
padding-right: 1em;
margin-left: 1em;
margin-right: 1em;
background: white;
border-style: solid;
border-width: thin;
}
span.infoAuthor
{
font-size: 10px;
font-family: sans-serif;
color: #A0A0A0;
font-weight: bolder;
}

```

```

margin-left: .5em;
margin-right: .5em;
}

/* Side Bar */
#menuSidebar
{
position: absolute;
top: 49px;
left: 0px;
width: 160px;
z-index: 1001;
background-color: #F0F0F0;
border-top-style: solid;
border-top-width: thin;
text-align: left;
visibility: visible;
}
td.sideControlLeft
{
font-size: 12px;
font-family: sans-serif;
text-align: left;
}
td.sideControlCenter
{
font-size: 12px;
font-family: sans-serif;
text-align: center;
}
td.sideControlRight
{
font-size: 12px;
font-family: sans-serif;
text-align: right;
}
td.sideControlAnnotation
{
font-size: 10px;
font-family: sans-serif;
text-align: left;
}
a.sideControlAnnotation
{
color: blue;
font-size: 10px;
font-family: sans-serif;
text-align: left;
text-decoration: underline;
}
td.sideControlHeading
{
background: #E4E4E4;
font-size: 12px;
font-family: sans-serif;
font-weight: bold;
text-align: center;
}
td.sideControlWing
{
width: 25px;
text-align: center;
}
td.sideControlTitle
{
background: #D0D0D0;
font-size: 12px;
font-family: sans-serif;
}

```

```

font-weight: bolder;
text-align: center;
font-style: italic;
}

/* Annotation Text Control */
td.annotationTextControl
{
font-size: 12px;
font-family: sans-serif;
font-weight: bolder;
padding-left: 1em;
}

/* Widgets */
#tracerWidget
{
position: absolute;
top: 20px;
left: 20px;
width: 10px;
height: 10px;
visibility: hidden;
z-index: 1;
}

/* Welcome Screen */
#welcomeScreen
{
position: absolute;
background: white;
left: 0px;
top: 30px;
width: 100%;
text-align: center;
float: none;
z-index: 1;
}
.welcomeText
{
font-style: normal;
font-family: arial,sans-serif;
}

/* Update Forms */
#updateForm
{
position: relative;
top: 45px;
width: 100%;
}
th.updateForm
{
color: white;
background: black;
font-size: 18px;
font-family: sans-serif;
text-align: center;
padding-right: 3em;
padding-left: 3em;
}
td.updateFormRight
{
font-size: 14px;
font-family: sans-serif;
text-align: right;
}

```

```

td.updateFormLeft
{
font-size: 14px;
font-family: sans-serif;
text-align: left;
}
tr.editListOdd
{
color: black;
background: #F0F0F0;
font-size: 12px;
font-family: sans-serif;
text-align: left;
}
tr.editListEven
{
color: black;
background: #D0D0D0;
font-size: 12px;
font-family: sans-serif;
text-align: left;
}

```

## F.2. open\_document.css

This style sheet is used for the Open Document dialog described in Section 4.3.

```

/* General */
body
{
background-attachment: fixed;
}

th
{
background: black;
color: white;
font-weight: bold;
font-size: 10px;
font-family: sans-serif;
text-align: center;
}

a.heading
{
color: white;
font-weight: bold;
font-size: 10px;
font-family: sans-serif;
text-align: center;
}

a.title
{
color: blue;
text-decoration: underline;
font-size: 10px;
font-weight: bolder;
}

```

```
td.timestamp
{
font-size: 8px;
font-weight: bold;
text-align: center;
}

td.title
{
background: #F0F0FF;
font-size: 10px;
font-weight: bolder;
text-align: left;
}

td.author
{
font-size: 10px;
font-weight: bolder;
text-align: center;
}

td.acl
{
font-size: 8px;
font-weight: bolder;
text-align: center;
}
```



## Appendix G – SQL Database Schema

The following *SQL* schema was optimized for *MySQL 5.x* servers. It may not be directly compatible with other *DBMS*. It assumes the database has already been created.

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

DROP TABLE IF EXISTS `acl_group`;
CREATE TABLE `acl_group` (
  `group_name` char(32) NOT NULL,
  `document_id` char(16) NOT NULL,
  `acl` enum('R','D','A','Ra','Da','M') NOT NULL,
  KEY `group_name` (`group_name`),
  KEY `document_id` (`document_id`),
  CONSTRAINT `acl_group_ibfk_1`
    FOREIGN KEY (`group_name`) REFERENCES `groups` (`name`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `acl_group_ibfk_2`
    FOREIGN KEY (`document_id`) REFERENCES `document` (`id`)
    ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `acl_user`;
CREATE TABLE `acl_user` (
  `person_id` char(10) NOT NULL,
  `document_id` char(16) NOT NULL,
  `acl` enum('R','D','A','Ra','Da','M') NOT NULL,
  KEY `person_id` (`person_id`),
  KEY `document_id` (`document_id`),
  CONSTRAINT `acl_user_ibfk_1`
    FOREIGN KEY (`person_id`) REFERENCES `person` (`id`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `acl_user_ibfk_2`
    FOREIGN KEY (`document_id`) REFERENCES `document` (`id`)
    ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `annotation`;
CREATE TABLE `annotation` (
  `id` char(16) NOT NULL,
  `author` char(10) NOT NULL,
  `owner` char(10) NOT NULL,
  `document_id` char(16) NOT NULL,
```



```

`page` int(11) NOT NULL,
`type` enum('Image','Text') NOT NULL,
`width` int(11) NOT NULL,
`height` int(11) NOT NULL,
`x_pos` int(11) NOT NULL,
`y_pos` int(11) NOT NULL,
`timestamp` bigint(10) NOT NULL,
`font` char(20) default NULL,
`pitch` int(2) default NULL,
`color` char(20) default NULL,
`background` char(16) default NULL,
`image` varchar(256) default NULL,
`text` varchar(256) default NULL,
PRIMARY KEY (`id`),
KEY `author` (`author`),
KEY `owner` (`owner`),
KEY `document_id` (`document_id`),
KEY `page` (`page`),
KEY `owner_2` (`owner`),
KEY `document_id_2` (`document_id`,`page`),
CONSTRAINT `annotation_ibfk_1`
  FOREIGN KEY (`author`) REFERENCES `person` (`id`)
  ON UPDATE CASCADE,
CONSTRAINT `annotation_ibfk_2`
  FOREIGN KEY (`owner`) REFERENCES `person` (`id`)
  ON UPDATE CASCADE,
CONSTRAINT `annotation_ibfk_3`
  FOREIGN KEY (`document_id`,`page`) REFERENCES `page` (`id`,`page`)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `converter`;
CREATE TABLE `converter` (
  `in_ext` char(8) NOT NULL,
  `out_ext` char(8) NOT NULL,
  `command` varchar(256) NOT NULL,
  `description` varchar(256) NOT NULL,
  PRIMARY KEY (`in_ext`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*140000 ALTER TABLE `converter` DISABLE KEYS */;
LOCK TABLES `converter` WRITE;
INSERT INTO `converter` VALUES
('c','jpg','/usr/local/oas/bin/converters/source_code','C Source Code'),
('cpp','jpg','/usr/local/oas/bin/converters/source_code','C++ Source Code'),
('css','jpg','/usr/local/oas/bin/converters/plain_text','CSS Source Code'),
('csv','jpg','/usr/local/oas/bin/converters/win32','Comma-delimited Spreadsheet'),
('doc','jpg','/usr/local/oas/bin/converters/win32','Microsoft Word Document'),
('gif','gif','/usr/local/oas/bin/converters/copy','GIF Image'),
('h','jpg','/usr/local/oas/bin/converters/source_code','C/C++ Header File'),
('htm','jpg','/usr/local/oas/bin/converters/source_code','HTML Source Code'),
('html','jpg','/usr/local/oas/bin/converters/source_code','HTML Source Code'),
('jpeg','jpg','/usr/local/oas/bin/converters/image','JPEG Image'),
('jpg','jpg','/usr/local/oas/bin/converters/copy','JPEG Image'),
('js','jpg','/usr/local/oas/bin/converters/source_code','JavaScript Source Code'),
('odg','jpg','/usr/local/oas/bin/converters/OOo','OpenOffice.org Draw Drawing'),
('odp','jpg','/usr/local/oas/bin/converters/OOo','OpenOffice.org Impress Presentation'),
('ods','jpg','/usr/local/oas/bin/converters/OOo','OpenOffice.org Calc Spreadsheet'),
('odt','jpg','/usr/local/oas/bin/converters/OOo','OpenOffice.org Writer Document'),
('pdf','jpg','/usr/local/oas/bin/converters/ps_pdf','Portable Document Format'),
('pl','jpg','/usr/local/oas/bin/converters/source_code','Perl Script Source Code'),
('pm','jpg','/usr/local/oas/bin/converters/source_code','Perl Module Source Code'),
('png','png','/usr/local/oas/bin/converters/copy','PNG Image'),
('ppt','jpg','/usr/local/oas/bin/converters/OOo','Microsoft PowerPoint Presentation'),
('ps','jpg','/usr/local/oas/bin/converters/ps_pdf','Postscript File'),
('rtf','jpg','/usr/local/oas/bin/converters/win32','Rich-Text Format Document'),
('sh','jpg','/usr/local/oas/bin/converters/source_code','Shell Programming Source Code'),
('txt','jpg','/usr/local/oas/bin/converters/plain_text','Plain Text'),
('wpd','jpg','/usr/local/oas/bin/converters/win32','Word Perfect Document'),

```

```

('wrl','jpg','/usr/local/oas/bin/converters/source_code','VRML Source Code'),
('xls','jpg','/usr/local/oas/bin/converters/win32','Microsoft Excel Spreadsheet');
UNLOCK TABLES;
/*!40000 ALTER TABLE `converter` ENABLE KEYS */;

DROP TABLE IF EXISTS `document`;
CREATE TABLE `document` (
  `id` char(16) NOT NULL,
  `author` char(10) NOT NULL,
  `owner` char(10) NOT NULL,
  `title` varchar(128) NOT NULL,
  `timestamp` bigint(10) NOT NULL,
  `original` varchar(256) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `author` (`author`),
  KEY `owner` (`owner`),
  CONSTRAINT `document_ibfk_1`
  FOREIGN KEY (`author`) REFERENCES `person` (`id`)
  ON UPDATE CASCADE,
  CONSTRAINT `document_ibfk_2`
  FOREIGN KEY (`owner`) REFERENCES `person` (`id`)
  ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `group_membership`;
CREATE TABLE `group_membership` (
  `person_id` char(10) NOT NULL,
  `group_name` char(32) NOT NULL,
  PRIMARY KEY (`person_id`,`group_name`),
  KEY `group_name` (`group_name`),
  CONSTRAINT `group_membership_ibfk_1`
  FOREIGN KEY (`person_id`) REFERENCES `person` (`id`)
  ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `group_membership_ibfk_2`
  FOREIGN KEY (`group_name`) REFERENCES `groups` (`name`)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `groups`;
CREATE TABLE `groups` (
  `name` char(32) NOT NULL,
  `owner` char(10) default NULL,
  PRIMARY KEY (`name`),
  KEY `owner` (`owner`),
  CONSTRAINT `groups_ibfk_1`
  FOREIGN KEY (`owner`) REFERENCES `person` (`id`)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40000 ALTER TABLE `groups` DISABLE KEYS */;
LOCK TABLES `groups` WRITE;
INSERT INTO `groups` VALUES ('ADMIN',NULL);
UNLOCK TABLES;
/*!40000 ALTER TABLE `groups` ENABLE KEYS */;

DROP TABLE IF EXISTS `page`;
CREATE TABLE `page` (
  `id` char(16) NOT NULL,
  `page` int(11) NOT NULL,
  `image` varchar(256) NOT NULL,
  `width` int(11) NOT NULL,
  `height` int(11) NOT NULL,
  PRIMARY KEY (`id`,`page`),
  CONSTRAINT `page_ibfk_1`
  FOREIGN KEY (`id`) REFERENCES `document` (`id`)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

DROP TABLE IF EXISTS `person`;
CREATE TABLE `person` (
  `id` char(10) NOT NULL,
  `first_name` varchar(24) NOT NULL,
  `last_name` varchar(48) NOT NULL,
  `address` varchar(48) default NULL,
  `city` varchar(24) default NULL,
  `state` char(2) default NULL,
  `zip` bigint(9) unsigned default NULL,
  `phone` bigint(10) unsigned default NULL,
  `fax` bigint(10) unsigned default NULL,
  `mobile` bigint(10) unsigned default NULL,
  `email` varchar(64) default NULL,
  `password` char(20) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `last_name` (`last_name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `service`;
CREATE TABLE `service` (
  `name` char(36) NOT NULL,
  `url` varchar(128) NOT NULL,
  `description` varchar(256) default NULL,
  PRIMARY KEY (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*!40000 ALTER TABLE `service` DISABLE KEYS */;
LOCK TABLES `service` WRITE;
INSERT INTO `service` VALUES
('AddAnnotation','/mp/AddAnnotation','Add an Annotation to a document'),
('AdminConverter','/mp/admin/Converter','Manage Document Converters'),
('AdminDeleteConverter','/mp/admin/DeleteConverter','Delete Document Converter'),
('AdminDeleteService','/mp/admin/DeleteService','Delete Web Service'),
('AdminDeleteUser','/mp/admin/DeleteUser','Delete System User'),
('AdminService','/mp/admin/Service','Manage Web Services'),
('AdminUser','/mp/admin/User','Manage System Users'),
('AnnotateDocument','/mp/AnnotateDocument','Display Document in Annotation Mode'),
('ChangeDocumentOwner','/cgi-bin/oas/ChangeDocumentOwner','Change Document Owner'),
('CreateDocument','/cgi-bin/oas/CreateDocument','Create New Document'),
('DeleteAnnotation','/mp/DeleteAnnotation','Deletes Annotation'),
('DeleteDocument','/cgi-bin/oas/DeleteDocument','Deletes Document'),
('DocumentPermissions','/mp/DocumentPermissions','Manage Document Permissions'),
('EditAnnotation','/mp/EditAnnotation','Edit Text Annotation'),
('FileFormats','/mp/FileFormats','Lists File Formats that can be Converted'),
('GroupManager','/mp/GroupManager','Manage Groups'),
('IndexMenu','/mp/IndexMenu','Creates Menu for SSI Pages'),
('Logout','/Logout','Logout Current User from System'),
('OpenDocument','/mp/OpenDocument','Open Document Dialog'),
('ReadDocument','/mp/ReadDocument','Display Document with Annotations'),
('ShowDocument','/mp/ShowDocument','Display Document without Annotations'),
('UserProfile','/mp/UserProfile','Manage User Profile');
UNLOCK TABLES;
/*!40000 ALTER TABLE `service` ENABLE KEYS */;

DROP TABLE IF EXISTS `system`;
CREATE TABLE `system` (
  `configuration` char(36) NOT NULL,
  `home_dir` varchar(256) NOT NULL,
  `www_dir` varchar(256) NOT NULL,
  `www_url` varchar(256) NOT NULL,
  `perl_dir` varchar(256) NOT NULL,
  PRIMARY KEY (`configuration`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40000 ALTER TABLE `system` DISABLE KEYS */;
LOCK TABLES `system` WRITE;

```

```
INSERT INTO `system` VALUES (
  'default',
  '/usr/local/oas/',
  '/usr/local/oas/htdocs/',
  '/',
  '/usr/lib/perl5/site_perl/OAS/'
);
UNLOCK TABLES;

/*!40000 ALTER TABLE `system` ENABLE KEYS */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```



## Appendix H – Apache Configuration File: httpd.conf

```
#####
# This is the main Apache server configuration file. #
#####

### Section 1: Global Environment
#####
#
ServerTokens OS
# Do NOT add a slash at the end of the directory path.
ServerRoot "/etc/httpd"
PidFile run/httpd.pid
Timeout 180
KeepAlive Off
MaxKeepAliveRequests 100
KeepAliveTimeout 15
<IfModule prefork.c>
    StartServers      8
    MinSpareServers   5
    MaxSpareServers   20
    ServerLimit       256
    MaxClients        256
    MaxRequestsPerChild 4000
</IfModule>
<IfModule worker.c>
    StartServers      2
    MaxClients        150
    MinSpareThreads   25
    MaxSpareThreads   75
    ThreadsPerChild   25
    MaxRequestsPerChild 0
</IfModule>
#Listen 12.34.56.78:80
Listen 80

# Dynamic Shared Object (DSO) Support
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule auth_digest_module modules/mod_auth_digest.so
LoadModule authn_file_module modules/mod_authn_file.so
LoadModule authn_alias_module modules/mod_authn_alias.so
LoadModule authn_anon_module modules/mod_authn_anon.so
LoadModule authn_dbm_module modules/mod_authn_dbm.so
LoadModule authn_default_module modules/mod_authn_default.so
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule authz_owner_module modules/mod_authz_owner.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_dbm_module modules/mod_authz_dbm.so
LoadModule authz_default_module modules/mod_authz_default.so
#LoadModule ldap_module modules/mod_ldap.so
#LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
LoadModule include_module modules/mod_include.so
LoadModule log_config_module modules/mod_log_config.so
```

```

LoadModule logio_module modules/mod_logio.so
LoadModule env_module modules/mod_env.so
LoadModule ext_filter_module modules/mod_ext_filter.so
LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule expires_module modules/mod_expires.so
LoadModule deflate_module modules/mod_deflate.so
LoadModule headers_module modules/mod_headers.so
LoadModule usertrack_module modules/mod_usertrack.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule mime_module modules/mod_mime.so
#LoadModule dav_module modules/mod_dav.so
LoadModule status_module modules/mod_status.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule info_module modules/mod_info.so
#LoadModule dav_fs_module modules/mod_dav_fs.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule dir_module modules/mod_dir.so
LoadModule actions_module modules/mod_actions.so
#LoadModule speling_module modules/mod_speling.so
#LoadModule userdir_module modules/mod_userdir.so
LoadModule alias_module modules/mod_alias.so
LoadModule rewrite_module modules/mod_rewrite.so
#LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
#LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule cache_module modules/mod_cache.so
LoadModule suexec_module modules/mod_suexec.so
LoadModule disk_cache_module modules/mod_disk_cache.so
LoadModule file_cache_module modules/mod_file_cache.so
LoadModule mem_cache_module modules/mod_mem_cache.so
LoadModule cgi_module modules/mod_cgi.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
#LoadModule asis_module modules/mod_asis.so
LoadModule mysql_auth_module modules/mod_auth_mysql.so
LoadModule perl_module modules/mod_perl.so

#ExtendedStatus On
User apache
Group apache

### Section 2: 'Main' server configuration
#####
#
ServerAdmin trev@byu.edu
ServerName oas.et.byu.edu
UseCanonicalName On

DocumentRoot "/usr/local/oas/htdocs"
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
<Directory "/usr/local/oas/htdocs">
    Options Includes
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
<Directory "/usr/local/oas/htdocs/d/">
    Options Includes
    AllowOverride AuthConfig
    Order allow,deny

```

```

    Allow from all
    AuthName "OAS - Online Annotation System"
    AuthType Basic
    AuthMySQLHost localhost
    AuthMySQLDB oas
    AuthMySQLUser oas
    AuthMySQLPassword oas
    AuthMySQLUserTable person
    AuthMySQLNameField id
    AuthMySQLPasswordField password
    AuthMySQLEnable On
    AuthMySQLPwEncryption crypt
    Require valid-user
</Directory>
<Directory "/usr/local/oas/htdocs/logout/">
    Options None
    AllowOverride None
    Order allow,deny
    Allow from all
    AuthName "OAS - Online Annotation System"
    AuthType Basic
    AuthMySQLHost localhost
    AuthMySQLDB oas
    AuthMySQLUser oas
    AuthMySQLPassword oas
    AuthMySQLUserTable person
    AuthMySQLNameField id
    AuthMySQLPasswordField password
    AuthMySQLEnable On
    AuthMySQLPwEncryption crypt
    Require user
</Directory>
<IfModule mod_userdir.c>
    UserDir disable
</IfModule>
DirectoryIndex index.html index.html.var index.shtml

AccessFileName .htaccess
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>

TypesConfig /etc/mime.types
DefaultType text/plain
<IfModule mod_mime_magic.c>
# MIMEMagicFile /usr/share/magic.mime
    MIMEMagicFile conf/magic
</IfModule>

HostnameLookups Off

ErrorLog logs/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
#CustomLog logs/referer_log referer
CustomLog logs/agent_log agent
CustomLog logs/access_log combined

ServerSignature On

Alias /icons/ "/var/www/icons/"
<Directory "/var/www/icons">
    Options Indexes MultiViews
    AllowOverride None

```



```

        Order allow,deny
        Allow from all
</Directory>

ScriptAlias /cgi-bin/ "/usr/local/oas/cgi-bin/"
<Directory "/usr/local/oas/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
<Directory "/usr/local/oas/cgi-bin/oas">
    AuthName "OAS - Online Annotation System"
    AuthType Basic
    AuthMySQLHost localhost
    AuthMySQLDB oas
    AuthMySQLUser oas
    AuthMySQLPassword oas
    AuthMySQLUserTable person
    AuthMySQLNameField id
    AuthMySQLPasswordField password
    AuthMySQLEnable On
    AuthMySQLPwEncryption crypt
    Require valid-user
</Directory>

# Redirect permanent /foo http://www.example.com/bar

AddLanguage ca .ca
AddLanguage cs .cz .cs
AddLanguage da .dk
AddLanguage de .de
AddLanguage el .el
AddLanguage en .en
AddLanguage eo .eo
AddLanguage es .es
AddLanguage et .et
AddLanguage fr .fr
AddLanguage he .he
AddLanguage hr .hr
AddLanguage it .it
AddLanguage ja .ja
AddLanguage ko .ko
AddLanguage ltz .ltz
AddLanguage nl .nl
AddLanguage nn .nn
AddLanguage no .no
AddLanguage pl .po
AddLanguage pt .pt
AddLanguage pt-BR .pt-br
AddLanguage ru .ru
AddLanguage sv .sv
AddLanguage zh-CN .zh-cn
AddLanguage zh-TW .zh-tw
LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt pt-BR ru
sv zh-CN zh-TW
ForceLanguagePriority Prefer Fallback
AddDefaultCharset UTF-8

AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddHandler type-map var
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml

Alias /error/ "/var/www/error/"
<IfModule mod_negotiation.c>
<IfModule mod_include.c>

```

```

    <Directory "/var/www/error">
        AllowOverride None
        Options IncludesNoExec
        AddOutputFilter Includes html
        AddHandler type-map var
        Order allow,deny
        Allow from all
        LanguagePriority en es de fr
        ForceLanguagePriority Prefer Fallback
    </Directory>
</IfModule>
</IfModule>

BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0
BrowserMatch "Microsoft Data Access Internet Publishing Provider" redirect-carefully
BrowserMatch "MS FrontPage" redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^WebDAVFS/1.[0123]" redirect-carefully
BrowserMatch "^gnome-vfs/1.0" redirect-carefully
BrowserMatch "^XML Spy" redirect-carefully
BrowserMatch "^Dreamweaver-WebDAV-SCM1" redirect-carefully

### Section 3: Perl_Mod Handlers
#####
#
PerlSwitches -T

<Location "/mp">
    Options None
    Order allow,deny
    Allow from all
    AuthName "OAS - Online Annotation System"
    AuthType Basic
    AuthMySQLHost localhost
    AuthMySQLDB oas
    AuthMySQLUser oas
    AuthMySQLPassword oas
    AuthMySQLUserTable person
    AuthMySQLNameField id
    AuthMySQLPasswordField password
    AuthMySQLEnable On
    AuthMySQLPwEncryption crypt
    Require valid-user
</Location>
<Location "/mp/admin">
    AuthName "OAS - Online Annotation System"
    AuthType Basic
    AuthMySQLHost localhost
    AuthMySQLDB oas
    AuthMySQLUser oas
    AuthMySQLPassword oas
    AuthMySQLUserTable person
    AuthMySQLNameField id
    AuthMySQLPasswordField password
    AuthMySQLGroupTable "person, group_membership"
    AuthMySQLGroupCondition "person.id = group_membership.person_id"
    AuthMySQLGroupField group_name
    AuthMySQLEnable On
    AuthMySQLPwEncryption crypt
    Require group ADMIN
</Location>

```

```

PerlModule OAS::Apache2::AddAnnotation
<Location /mp/AddAnnotation>
    SetHandler perl-script
    PerlHandler OAS::Apache2::AddAnnotation
</Location>

PerlModule OAS::Apache2::AdminConverter
<Location /mp/admin/Converter>
    SetHandler perl-script
    PerlHandler OAS::Apache2::AdminConverter
</Location>

PerlModule OAS::Apache2::AdminDeleteConverter
<Location /mp/admin/DeleteConverter>
    SetHandler perl-script
    PerlHandler OAS::Apache2::AdminDeleteConverter
</Location>

PerlModule OAS::Apache2::AdminDeleteService
<Location /mp/admin/DeleteService>
    SetHandler perl-script
    PerlHandler OAS::Apache2::AdminDeleteService
</Location>

PerlModule OAS::Apache2::AdminDeleteUser
<Location /mp/admin/DeleteUser>
    SetHandler perl-script
    PerlHandler OAS::Apache2::AdminDeleteUser
</Location>

PerlModule OAS::Apache2::AdminService
<Location /mp/admin/Service>
    SetHandler perl-script
    PerlHandler OAS::Apache2::AdminService
</Location>

PerlModule OAS::Apache2::AdminUser
<Location /mp/admin/User>
    SetHandler perl-script
    PerlHandler OAS::Apache2::AdminUser
</Location>

PerlModule OAS::Apache2::AnnotateDocument
<Location /mp/AnnotateDocument>
    SetHandler perl-script
    PerlHandler OAS::Apache2::AnnotateDocument
</Location>

PerlModule OAS::Apache2::DeleteAnnotation
<Location /mp/DeleteAnnotation>
    SetHandler perl-script
    PerlHandler OAS::Apache2::DeleteAnnotation
</Location>

PerlModule OAS::Apache2::DocumentPermissions
<Location /mp/DocumentPermissions>
    SetHandler perl-script
    PerlHandler OAS::Apache2::DocumentPermissions
</Location>

PerlModule OAS::Apache2::EditAnnotation
<Location /mp/EditAnnotation>
    SetHandler perl-script
    PerlHandler OAS::Apache2::EditAnnotation
</Location>

PerlModule OAS::Apache2::FileFormats
<Location /mp/FileFormats>

```

```

        SetHandler perl-script
        PerlHandler OAS::Apache2::FileFormats
    </Location>

    PerlModule OAS::Apache2::GroupManager
    <Location /mp/GroupManager>
        SetHandler perl-script
        PerlHandler OAS::Apache2::GroupManager
    </Location>

    PerlModule OAS::Apache2::IndexMenu
    <Location /mp/IndexMenu>
        SetHandler perl-script
        PerlHandler OAS::Apache2::IndexMenu
    </Location>

    PerlModule OAS::Apache2::Logout
    <Location /Logout>
        SetHandler perl-script
        PerlHandler OAS::Apache2::Logout
    </Location>

    PerlModule OAS::Apache2::OpenDocument
    <Location /mp/OpenDocument>
        SetHandler perl-script
        PerlHandler OAS::Apache2::OpenDocument
    </Location>

    PerlModule OAS::Apache2::ReadDocument
    <Location /mp/ReadDocument>
        SetHandler perl-script
        PerlHandler OAS::Apache2::ReadDocument
    </Location>

    PerlModule OAS::Apache2::ShowDocument
    <Location /mp/ShowDocument>
        SetHandler perl-script
        PerlHandler OAS::Apache2::ShowDocument
    </Location>

    PerlModule OAS::Apache2::UserProfile
    <Location /mp/UserProfile>
        SetHandler perl-script
        PerlHandler OAS::Apache2::UserProfile
    </Location>

### Section 4: Startup Script
#####
#
<Perl>
</Perl>

```



## Appendix I – Supported File Formats

Table I1 lists all of the tested file formats for *OAS*. It should be noted that while source code formats are essentially plain-text, special formatting, as described in Section 4.2.2 is applied to these formats. Therefore, they have their own entry in the table.

Table I1: Supported File Formats

File Extension	File Format
c	C Source Code
c++	C++ Source Code
css	CSS Source Code
csv	Comma-delimited Spreadsheet
doc	Microsoft Word Document
gif	GIF Image
h	C/C++ Header File
htm/html	HTML Source Code
jpg/jpeg	JPEG Image
js	JavaScript Source Code
odg	OpenOffice.org Draw Drawing
odp	OpenOffice.org Impress Presentation
ods	OpenOffice.org Calc Spreadsheet
odt	OpenOffice.org Writer Document
pdf	Portable Document Format
pl	Perl Script Source Code
pm	Perl Module Source Code
png	PNG Image
ppt	Microsoft PowerPoint Presentation
ps	Postscript File

Table I1 – *Continued*

File Extension	File Format
rtf	Rich-Text Format Document
sh	Shell Programming Source Code
txt	Plain Text
wpd	Word Perfect Document
wrl	VRML Source Code
xls	Microsoft Excel Spreadsheet

## Appendix J – Supported Web Browsers

Table J1: Supported Web Browsers

Operating System	Web Browser	Supported?
Windows XP Home	Internet Explorer 6.0	Yes
	Firefox 1.0	Yes
	Opera 9.00	Yes
Windows XP Tablet PC Edition 2005	Internet Explorer 7.0 RC1	Yes
	Firefox 1.5	Yes
	Mozilla 1.7.3	Yes
Mac OS X Jaguar	Safari 1.3	Yes (no tracer)
Linux Fedora Core 5	Firefox 1.5	Yes
	Opera 9.00	Yes
	Mozilla 1.7.13	Yes
	Konquerer 3.5.3	Yes
	“Epiphany” GNOME Web Browser 2.14.2.1	Yes





## Appendix K – Annotation in Literature

This appendix lists a number of annotation systems, models, and frameworks found in literature, along with references. This is not meant to be an exhaustive list. Instead it provides insight into the breadth of research that has been done. Many of these systems approach annotation from a wide variety of paradigms and process domains.

### 3Book

- Hong, Chi, and Card 2005

### Adobe Acrobat

- Brush et al. 2001
- Hong, Chi, and Card 2005
- Mock 2004
- Olsen, Taufer, and Fails 2004
- Plimmer and Mason 2006
- Popyack and Herrmann 2003
- Timmins 2004
- Wang and Chen 2004

### Alias Sketchbook Pro

- Olsen, Taufer, and Fails 2004

### ALT

- Gabrielli and Law 2003

### Amaya

- Kahan and Koivunen 2001
- Koivunen 2005
- Vatton 2006

### AnnotateImage

- Bottoni et al. 2004

#### Annotation Engine

- Takeda and Suthers 2002

#### Annotator

- Brush et al. 2002

#### Annotea

- Bottoni et al. 2004
- Handschuh and Staab 2002
- Hansen 2006
- Kahan and Koivunen 2001
- Koivunen 2005
- Segawa 2006
- Swick et al. 2002
- Vatton 2006

#### Aquanet

- Olsen, Taufer, and Fails 2004

#### Arakne

- Hansen 2006

#### Augment

- Hansen 2006

#### BSCW

- Brush et al. 2002

#### Chimera

- Hansen 2006

#### ComMentor

- Brush et al. 2002
- Hansen 2006
- Takeda and Suthers 2002

#### Comments

- Cadiz, Gupta, and Grudin 2000

#### CommonSpace

- Cadiz, Gupta, and Grudin 2000

#### ComMotion

- Hansen 2006

#### CoNotes

- Cadiz, Gupta, and Grudin 2000
- Takeda and Suthers 2002

#### CREAM

- Cimiano, Handschuh, and Staab 2004
- Handschuh and Staab 2002
- Handschuh, Staab, and Maedche 2001
- Wu, Zhang, and Yu 2006

#### Crit.org

- Brush et al. 2002

#### Critlink

- Handschuh and Staab 2002
- Takeda and Suthers 2002

#### Cyberguide

- Hansen 2006

#### Devise Hypermedia

- Hansen 2006

#### Dexter Hypertext Reference Model

- Hansen 2006

#### Diary Composer (DC)

- Sevasti and Christos 2000

#### Digital Graffiti

- Hansen 2006
- Carter et al. 2004

#### Dynamite

- Olsen, Taufer, and Fails 2004
- Schilit, Golovchinsky, and Price 1998
- Wolfe 2000

#### EPost

- Brush 2002

#### E-Quill

- Olsen, Taufer, and Fails 2004

#### ESP Game

- Volkmer, Smith, and Natsev 2005

#### EVA

- Volkmer, Smith, and Natsev 2005

#### Freestyle

- Olsen, Taufer, and Fails 2004

#### Geonotes

- Hansen 2006

#### gIBIS

- Olsen, Taufer, and Fails 2004

#### HATS

- Kim, Slater, and Whitehead 2004

#### HP Websigns

- Hansen 2006

#### HyCon

- Hansen 2006

#### HyConExplorer

- Hansen 2006

#### I2Cnet

- Bottoni et al. 2004

#### Imark plug-in (IE)

- Hansen 2006
- Kahan et al. 2001

#### INote

- Bottoni et al. 2004

#### Intraspect

- Brush et al. 2002

#### Kukakuka

- Takeda and Suthers 2002

#### Journal of Interactive Media in Education

- Takeda and Suthers 2002

#### Livelink

- Brush et al. 2002

#### Lotus Notes

- Cadiz, Gupta, and Grudin 2000

#### M2Screen

- Mock 2004

#### MADCOW

- Bottoni et al. 2004
- Bottoni et al. 2006

#### Madeus

- Bottoni et al. 2004

#### Microcosm

- Hansen 2006

#### Microsoft eBook Reader

- Brush et al. 2001

#### Microsoft Journal

- Harmon, Helps, and Bailey 2005
- Mock 2004
- Timmins 2004
- Willis and Miertschin 2004

#### Microsoft Word

- Cadiz, Gupta, and Grudin 2000
- Harmon, Helps, and Bailey 2005
- Hong, Chi, and Card 2005
- Mock 2004
- Plimmer and Mason 2006
- Timmins 2004
- Wang and Chen 2004
- Willis and Miertschin 2004

#### MRAS

- Bottoni et al. 2004
- Cadiz, Gupta, and Grudin 2000

#### Notecards

- Olsen, Taufer, and Fails 2004

#### OHP-NAV model

- Hansen 2006

#### OneNote

- Mock 2004
- Plimmer and Mason 2006
- Timmins 2004
- Wang and Chen 2004
- Willis and Miertschin 2004

#### Ont-O-Mat (Onto-Mat)

- Cimiano, Handschuh, and Staab 2004
- Handschuh and Staab 2002
- Handschuh, Staab, and Maedche 2001

#### PAIS

- Bottoni et al. 2004

#### PANKOW

- Cimiano, Handschuh, and Staab 2004

#### Pink

- Takeda and Suthers 2002

#### PREP

- Cadiz, Gupta, and Grudin 2000
- Zheng, Booth, and McGrenere 2006

#### Protege

- Xu, Zhang, and Yu 2006

#### QuickPlace

- Brush et al. 2002

#### Quilt

- Brush et al. 2002
- Cadiz, Gupta, and Grudin 2000
- Zheng, Booth, and McGrenere 2006

#### Re:mark

- Wang and Chen 2004

Ricoh MovieTool

- Volkmer, Smith, and Natsev 2005

SASSE

- Zheng, Booth, and McGrenere 2006

ScreenCrayons

- Olsen, Taufer, and Fails 2004

SharePoint

- Takeda and Suthers 2002
- Wang and Chen 2004

Stick-e notes

- Hansen 2006

ThirdVoice

- Handschuh and Staab 2002

Trevi

- Dmitriev et al. 2006

Vannotea

- Bottoni et al. 2004

VideoAnnEx

- Bottoni et al. 2004
- Volkmer, Smith, and Natsev 2005

Vidya

- Shevade and Sundaram 2003

VIPER Annotation Tool

- Volkmer, Smith, and Natsev 2005

WebAnn

- Brush 2002
- Gabrielli and Law 2003
- Marshall and Brush 2002

Web Discussions

- Brush et al. 2002



#### WEBTOUR

- Sastry, Lewis, and Pizano 1999

#### Webvise

- Hansen 2006

#### WISPA

- Sannomiya et al. 2001

#### XLibris

- Barger and Moscovich 2003
- Golovchinsky and Denoue 2002
- Hong, Chi, and Card 2005
- Olsen, Taufer, and Fails 2004
- Schilit, Golovchinsky, and Price 1998
- Wolfe 2000

#### XMetal Reviewer

- Zheng, Booth, and McGrenere 2006

#### Xspect

- Hansen 2006

#### Yawas

- Handschuh and Staab 2002

## Appendix L – Open-source Course Management Systems

The list of open-source course management systems, shown in Table L1, was compiled by compiled by Scott Leslie on the EdTechPost Wiki (Leslie 2004).

Table L1: Open-source Course Management Systems

System	Sponsor/Developer	Country of Origin	License Type
.LRN	MIT/Heidelberg	US/Germany	GPL
Adept		Russia	GPL
ATutor 1.2	University of Toronto	Canada	GPL
Bazaar 7	University of Athabasca	Canada	GPL
Boddington	University of Leeds	U.K.	Boddington System Software License
CHEF	University of Michigan	US	
Claroline 1.4	Université catholique de Louvain	France	GPL
ClassWeb 2.0	University of California Los Angeles	US	GPL
Coursework	Stanford University	US	
eClass.Net	Tulane University	US	BSD
eConf	University of Namur	Belgium	GPL
Eduzope/Eduplone	Coalition (Infrae, Plone, others)	Europe	GPL
eLecture Online Lecturing System	Christian and Thomas Lang (Karl-Franzens-Universität Graz)	Austria	
Eledge 1.2	Chuck Wight (University of Utah)	US	GPL
e-Tutor	University of Ottawa	Canada	GPL

Table L1 – *Continued*

System	Sponsor/Developer	Country of Origin	License Type
File3	University of Art and Design Helsinki	Finland	GPL
Freestyle Learning	University of Muenster	Germany	
GANESHA	Anéma Formation	France	GPL
H2O Project	Harvard Law School	US	GPL
Ilias	University of Cologne	Germany	GPL
Interact	Christchurch College of Education	New Zealand	GPL
Interactive Learning Environment (ILE)	Brad Cox	US	BSD/MIT
Internet Course Reader	TeleLearning National Centre of Excellence	Canada	LGPL
Jones e-education V2002	Jones advisory Group	US	
KEWL	University of Western Cape	South Africa	
LearnLoop	ITuniversity in Gothenburg	Sweden	GPL
LogiCampus	Tap Internet & Tarrant County College	US	Custom
LON-CAPA	Michigan State University	US	GPL
OLAT	University of Zuerich	Switzerland	Apache style
Manhattan Virtual Classroom 0.93	Western New England College	US	GPL

Table L1 – *Continued*

System	Sponsor/Developer	Country of Origin	License Type
MimerDesk 1.5.3.1	Ionstream	Finland	GPL
MnITS Internet Teaching System	Daniel Bartholomew	US	GPL
Moodle 1.1	Martin Dougiamas	Australia	GPL
OpenCourse	Paul Jones	US	GPL
Open Learning Manangement System (O-LMS)	University of Utah	US	
OpenUSS	University of Münster	Germany	GPL
Segue	Middlebury College	US	
Shadownet	University of Missouri-Columbia	US	GPL
Spaghettilearning.com		Italy	GPL
sTeam	Univeristy of Paderborn	Germany	
Stellar	Massachusetts Institute of Technology	US	
TextWeaver	San Diego State University	US	LGPL
Uni Open Platform	FernUniversitaet in Hagen	Germany	GPL
WBT-Master	Graz University of Technology	Austria	GPL
WebWork	University of Rochester	US	
Whiteboard	Todd Templeton	US	GPL



## Appendix M – OAS Logo

This is the official logo for the *OAS* project.



Figure M1: *OAS* Logo