Apache Overview HOWTO

Table of Contents

Table of Contents

Table of Contents

Apache Overview HOWTO

Daniel Lopez Ridruejo, `ridruejo@apache.org`

v0.9, 2002−10−10

This document gives you an overview of the different Apache projects, such as the Apache HTTP server and the Tomcat Servlet and JSP engine. It provides pointers for further information and implementation details.

---

5. Contacting the Author

6. Open Content Open Publication License

---

1. Introduction

This document gives you an overview of the Apache world, including Apache Software Foundation projects such as the Apache web server and commercial and open source third party software. Apache is the most popular server on the Internet. New Apache users, especially those coming from a Windows background, are often unaware of the possibilities of Apache, its useful addons and, more in general, how everything works together. This document aims to show a general picture of such possibilities with a brief description of each one and pointers for further information. The information has been gathered from many sources, including projects' web pages, conference talks, mailing lists, Apache websites and my own hands−on experience. Full credit is given to these authors. Without them and their work, this document would not have been possible or necessary.

1.1 Apache Software Foundation

The Apache Software Foundation provides support for the Apache community of open−source software projects. The Apache projects are characterized by a collaborative, consensus based development process, an open and pragmatic software license, and a desire to create high quality software that leads the way in its field. We consider ourselves not simply a group of projects sharing a server, but rather a community of developers and users.

The ASF is home to many successful Open Source projects, such as the Tomcat Servlet/JSP engine and the ANT build tool.

You can learn more about the foundation here.

1.2 Structure of this document

The first part of this document deals with the Apache Web Server and related modules. It covers the history, architecture and capabilities of the server and describes ways in which you can extend and customize it.

The second part of this document covers projects of the Apache Software Foundation, such as those form the Jakarta and Java XML communities. Rather than organizing the projects around a certain programming language or technology, they are organized based on functionality provided.

---

2. Apache

Apache is the leading internet web server, with over 60% market share, according to the Netcraft survey. Several key factors have contributed to Apache's success:

- The Apache license. It is an open source, BSD−like license that allows for both commercial and non−commercial uses of Apache.
- Talented community of developers with a variety of backgrounds and an open development process based on technical merits.
- Modular architecture. Apache users can easily add functionality or tailor Apache to their specific enviroment.
- Portable: Apache runs on nearly all flavors of Unix (and Linux), Windows, BeOs, mainframes...
- Robustness and security.

Many commercial vendors have adopted Apache−based solutions for their products, including Oracle, Red Hat and IBM. In addition, Covalent provides add−on modules and 24x7 support for Apache.

The following websites use Apache or derivatives. Chances are that if Apache is good enough for them, it is also good enough for you :)

- Amazon.com
- Yahoo!
- W3 Consortium
- Financial Times
- Apple
- MP3.com
- Stanford

>From the Apache website:

The Apache HTTP Server Project is an effort to develop and maintain an open−source HTTP server for modern operating systems including UNIX and Windows NT. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

Apache started its life as modifications to the NCSA Web server, one of the first HTTP servers. You can learn more about Apache's history here:

The Apache project has grown beyond building just a web server into developing other critical server side technologies. The Apache Software Foundation, described in a later section, serves as an umbrella for these projects.

2.1 Architecture

There are two main versions of Apache, the 1.3 series and the 2.0 series. Although both versions are considered production quality, they differ in architecture and capabilities.

2.1.1 Apache 1.3

Apache 1.3 has been ported to a great variety of Unix platforms and is the most widely deployed Web server on the Internet.

Process−based Web server

Apache 1.3 on Unix is a process−based Web server. The Apache program forks several children at startup. Forking means that a parent process makes identical copies of itself, called *children*. Each one of the children can serve a request independent of the others. This approach has the advantage of improved stability: If one of the children misbehaves (runs out of control or has memory leaks) it can be terminated without affecting the others. The stability comes with a performance penalty. In most Unix operating systems, creating processes and context switching (assigning processor time to each process) are expensive operations. Since processes are isolated from each other, they cannot easily share code and data, consuming system resources.

Windows support

Apache 1.3 is the first version of Apache to support Windows, although the port is not considered to be as stable as its Unix counterparts. This is due to the fact that the server had been designed with Unix in mind and the Windows port was a later addition that did not integrate very well.

Modular

Apache 1.3 has a modular architecture. You can enable or disable modules to add and remove Web server functionality. You can customize Apache to improve performance and security. In addition to modules bundled with the server, there is a great number of third party modules, providing extended functionality.

2.1.2 Apache 2.0

Apache 2.0 is the latest and greatest version of the Apache server. The architecture contains significant improvements over the 1.3 series. The following are some of them.

Multi Processing Modules

Apache 2.0 abstracts the request processing architecture in special server modules, called Multi Processing modules (MPMs). This means that Apache can be configured to be a pure process−based server, a purely threaded server or a mixture of those models. Threads are contained inside processes and run simultaneously. Unlike processes, threads can share data and code. Threads are thus more "lighweight" than processes, and in most cases threaded servers scale better than process based servers. The disadvantage is that the server is less reliable, since if a thread misbehaves it can corrupt data or code belonging to other threads.

Protocol Modules

The protocol handling has been encapsulated in its own layer in Apache 2.0. That means it is possible to write modules to serve protocols other than HTTP, such as POP3 for mail or FTP for file transfer. These protocol modules can take advantage of a solid server framework and module functionality, such as authentication and dynamic content generation. This means that, for example, you can authenticate your POP3 users against the same user database Apache uses for web requests and that FTP content can be generated dynamically using PHP, CGI or any other technologies explained later in this document.

Module and filter architecture.

Apache 2.0 maintains the 1.3 modular architecture and adds an additional extension mechanism: filters. Filters allow modules to modify the content generated by other modules. They can encrypt, scan for viruses or compress not only static files but dynamically generated content.

Compatibility issues

Unfortunately, though the module API is similar between versions, they are not identical and Apache 1.3 modules need to be ported to the new architecture. Most mainstream modules such as PHP and mod_perl already have Apache 2.0 versions and others, such as mod_dav and mod_ssl, are now part of the server distribution. Running modules on a threaded architecture requires specific changes to modules. Modules distributed with Apache have undergone those changes and are considered `thread−safe', but third−party modules or libraries may not. If you need one of those, you will be limited to running Apache as a pure process−based server.

Portable

Apache runs equally well now on Windows and Unix platforms thanks to the Apache Portable Runtime (APR) library. It abstracts the differences among operating systems, such as file or network access APIs. Porting Apache to a new platform is often as simple as porting the Apache Portable Runtime. This abstraction layer also provides for platform−specific tuning and optimization.

2.2 Security

Apache provides several security−related modules for securing and restricting access to the server.

Authentication

Authentication modules allow you to determine the identity of a client, usually by verifying an username and password against a backend database. Apache includes modules to authenticate against plain text and database files. Additional authentication modules exist that connect Apache to existing security frameworks or databases, including: NT Domain controller, Oracle, mySQL, PostgresSQL and so on.

The LDAP modules are specially interesting, as they allow integration with company and enterprise wide existing directory services. You can find these modules at http://modules.apache.org. An Apache 2.0 LDAP module can be found at the Apache website.

Access Control

Apache provides the mod_access module that can restrict access to resources based on parameters of the client request, such as the presence of a specific header or the IP address or hostname of the client. Third party modules allow you to restrict access to clients that misbehave, as explained in later sections on performance and bandwidth control.

SSL/TLS

The Secure Sockets Layer/Transport Layer Security protocols allow data between the Web server and client to be encrypted. In Apache 1.3, the protocols are implemented by mod_ssl, which is distributed separately from the mod_ssl website and requires applying patches to the server. This was necessary because of export regulations on encryption. Most of those restrictions have since then being lifted and starting with Apache 2.0, mod_ssl is now included as a base module with Apache.

2.3 Proxy

A proxy is a program that performs requests on behalf of another. There are different kind of Web proxies. A traditional HTTP proxy, also called a *forward proxy*, accepts requests from clients (usually Web browsers), contacts the remote server, and returns the responses.

A reverse proxy is a Web server that is placed in front of other servers, providing a unified front end and offloading certain tasks, such as SSL processing, from the backend Web servers.

Apache supports both types of proxy, caching of proxied content and differente proxy backends such as FTP.

2.4 Performance and scalability

Raw performance is only one of the factors to consider in a web server (flexibility and stability come usually first).

Having said that, there are solutions to improve performance on heavy loaded webservers serving static content. If you are in the hosting business Apache also provides ways in which you can measure and control bandwidth usage. Throttling in this context usually means slowing down the delivery of content based on the file requested, a specific client IP address and so on. This is done to prevent abuse.

- **mod_mmap**: Included in current Apache 1.3 releases, it maps to memory a statically configured list of files that are frequently requested but infrequently changed. This functionality is included in mod_file_cache in Apache 2.
- **Mod_bandwidth**: This Apache 1.3 module enables the setting of server−wide or per connection bandwidth limits, based on the specific directory, size of files and remote IP/domain.
- **Bandwidth share module**: provides bandwidth throttling and balancing by client IP address. It supports Apache 1.3 and earlier versions of Apache 2.
- **Mod_throttle**:Throttle bandwidth per virtual host or user. For Apache 1.3

Load Balancing

Using the Apache reverse proxy and mod_rewrite you can have an Apache process distributing requests among a variety of backend web servers. You can find more information at http://www.apache.org/docs/misc/rewriteguide.html

Additionally, mod_backhand is an Apache 1.3 module that allows seamless redirection of HTTP requests from one web server to another. This redirection can be used to target machines with under−utilized resources, thus providing fine−grained, per−request load balancing of web requests. You can find more information at http://www.backhand.org/.

Compression

Apache 2.0 includes mod_deflate, a filtering module that compresses content before delivering it to clients. This saves bandwidth but can have a performance impact. The mod_gzip module provides this functionality for Apache 1.3

2.5 CGI scripts

CGI stands for Common Gateway Interface. CGI programs are external programs that are called when a user requests a certain page. The CGI program receives information from the web server (form variable values, type of browser, IP address of the client and so on) and uses that information to output a web page to the client.

Apache has support for CGIs and there is a third−party Apache 1.3 module that provides support for the FastCGI protocol. It avoids the performance penalties associated with starting and stopping a CGI program with every request. You can find it at http://fastcgi.com/

2.6 Development Platform Integration

Web applications are written in high−level languages such as Java, Perl, C# and so on and Apache has several modules that integrate them with the server. In many cases the modules expose the Apache API so entire Apache modules can be written in those languages.

Perl

mod_perl is one of the most veteran and successful Apache projects. It embeds a Perl interpreter in Apache and allows access to the web server internals from Perl. This allows for entire modules to be written in Perl or a mixture of Perl and C code. In the 1.3 Apache versions, one interpreter has to be embedded in each child, since the server is multiprocess based. In heavy traffic dynamic sites, the increased size could make a difference. In threaded versions of Apache 2.0 mod_perl allows for sharing of code, data and session state among interpreters. This results in a faster, leaner solution.

mod_perl is in itself another platform, with a great variety of modules available such as Mason and Embperl for embedding Perl in HTML pages and AxKit for XML−driven templates.

PHP

From the PHP website: *PHP is a server−side, cross−platform, HTML embedded scripting language*. It is the most popular module for Apache and this is due to a variety of reasons:

- Learning curve is quite low
- Great documentation
- Extensive database support
- Modularity

PHP has a modular design. Among many others, there are modules that provide support for:

- Database connetivity for popular databases such as Oracle, MS−SQL server, ODBC interface, MySQL, mSQL, PostgreSQL and so on.
- XML support
- File transfer: FTP
- HTTP
- Directory support: LDAP

- Mail support: IMAP, POP3, NNTP
- PDF document generation
- CORBA
- SNMP

You only need to compile/use the modules you need. PHP can be used with Apache, as an external CGI or with other webservers. It is crossplatform and it runs on most flavors of Unix and Windows. If you come from a Windows background, you probably have used Internet Information Server with Active Server Pages and MS−SQL Server. A common replacement in the Unix world for this trio is Apache with PHP and MySQL. Since PHP works:

- with Apache and with Microsoft IIS
- with MySQL and with MS−SQL server
- on Unix and on Windows

you have a nice, gradual migration path from a Microsoft−centric solution to Unix based solutions.

Python

Python is a popular object oriented scripting language. Mod_Python, which is now an official Apache project, allows you to integrate Python with the Apache web server. You can develop complex web applications or accelerate existing Python CGI scripts. Recent versions run on Apache 2.0.

Tcl

The Tcl Apache project integrates Tcl with the Apache webserver. Tcl is a lightweight, extensible scripting language. You can learn more about Tcl here. There are several modules currently under the Apache Tcl umbrella:

- Both Mod_dtcl and Neowebscript allow embedding Tcl on HTML pages. Rivet combines the best of both modules.
- Mod_tcl takes an approach similar to mod_perl, exposing the Apache API.
- WebSH provides a Tcl Web application environment

Microsoft technologies

Several modules allow integration with Microsoft languages and technologies such as the .Net framework or Active Server Pages.

.Net

mod_haydn integrates Mono with Apache and exposes the Apache API to the .Net framework, allowing you to write modules in C#, for example. Covalent provides mod_asp.net, an commercial Windows module that allows Apache to run ASP.Net applications, allowing you to replace Microsoft IIS.

ASP

ASP stands for Active Server Pages and is a Microsoft technology that allows you to embed code, usually Visual Basic, in HTML pages. Several companies such as ChilliSoft and Stryon provide products that can run

ASP applications on Unix environments.

ISAPI

ISAPI is an API that you can use to extend Microsoft IIS, similarly to how you would use the Apache API. Apache includes a module mod_isapi that mirrors this functionality and allows you to run ISAPI modules.

Java

Most applications servers, such as those from Oracle, IBM and BEA provide modules to integrate with the Apache web server. Additionally, several modules such as mod_jk and mod_webapp allow you to connect to Tomcat, a Servlet and JavaServer Pages container that is also part of the Apache Software Foundation.

Modules for other languages

This document has described modules for popular server side languages such as Perl, Python and PHP. You can find additional language modules (JavaScript, Haskell, Ruby and others) at the Apache modules directory.

2.7 Management

An important part of Web server administration includes building, configuring and monitoring different servers.

Build tools

Apache can be extended and customized in many different ways. Integration of different modules with the server can sometimes be a difficult task. Tools such as the Apache Toolbox can make this task easier, by providing a menu driven build framework.

User Interfaces for Apache

Apache is configured thru text configuration files, and that sometimes can be hard, specially for people coming from a Windows background. There are open source graphical tools that make this task easier:

- Comanche, by yours truly, is crossplatform and runs on Unix/Linux, Windows and Mac.
- Webmin: A nice web based interface.
- gui.apache.org: GUI interfaces for Apache project. Programs are in various degrees of development.

SNMP

SNMP stands for Simple Network Management Protocol. It allows monitoring and management of network servers, equipment and so on. SNMP modules for Apache help manage large deployments of web servers, measure the quality of service offered and integration of Apache with existing management frameworks.

- Open source Mod SNMP for Apache 1.3.
- Covalent SNMP provides a commercial SNMP module, support for the latest SNMPv3 standard, integration with HP–Openview, Tivoli and so on.

2.8 Publishing

Authors of Web content require a means of managing that content and uploading it to the server. One of the protocols used for this purpose is DAV (Distributed Authoring and Versioning). DAV is an extension to the HTTP protocol that enables users and applications to publish and modify Web content. DAV technology is widely implemented, Microsoft supports it at the operating system level (WebFolders) and in its Office suite. Same goes for Apple OS−X and a variety of third party products from Adobe, Oracle and so on. You can get the mod_dav module for Apache 1.3 at http://www.webdav.org/mod_dav/. In Apache 2.0, mod_dav is included with the base distribution.

Previous to DAV, Microsoft had its own publishing protocol, integrated with the Microsoft FrontPage tool. You can add server−side support for Frontpage using the modules at http://www.rtr.com/Ready−to−Run_Software/, though due to the way they integrate with Apache they are not considered secure.

2.9 Protocol modules

Apache 2.0 introduced the concept of protocol modules. That means that developers can reuse the Apache server framework to implement new protocols such as those dealing with mail and file transfer. mod_ftp is a commercial Apache−based FTP module from Covalent. mod_pop3 is an open source module that implements the POP3 protocol, commonly used by mail readers to retrieve messages from mail servers.

2.10 Virtual Hosting

Apache provides extensive virtual hosting support which means that you can serve multiple websites from a single server. In Apache 2.0, with the per−child MPM you can have multiple children, each one serving a different domain under different Unix user ids. This is very important for security in shared hosting scenarios, as it allows you to isolate customers from each other. The following are additional, alternative, virtual hosting modules.

- mod_dynvhost
- mod_pweb
- mod_v2h

2.11 Commercial support

Apache is the web server of choice for many commercial entities, including big enterprises. These companies have certain requirements when adopting a technology, specially one that is at the core of their Internet strategy, such as Web servers. Those requirements include performance, stability, management capabilities, support, professional services and integration with legacy systems. A number of commercial companies, such as IBM, Red Hat and Covalent, provide the products and services necessary to make Apache meet the needs of Enterprise customers.

In addition, many other companies and OEMs ship Apache as a bundled web server with their products.

3. ASF Projects

Although Apache is probably the most popular, the Apache Software Foundation is home to many other projects. This section provides an overview of the most relevant ones, organized logically. Most of them belong either to the Jakarta project and the XML project. The Jakarta project hosts Java−based projects and the XML project hosts, surprise, XML−related projects.

3.1 Applications and Frameworks

The following are application and development frameworks that are part of the ASF.

3.1.1 Servers

The following are some ASF server projects.

Tomcat

Tomcat is the flagship product of the Jakarta project. It is the official reference implementation for the Java Servlet and JavaServer Pages technologies.

You can learn more in the Tomcat homepage.

JAMES (Java Apache Mail Enterprise Server)

Complementary to the other Apache server side technologies, JAMES provides *a 100% pure Java server designed to be a complete and portable enterprise mail engine solution based on currently available open protocols (SMTP, POP3, IMAP, HTTP)*

More information can be found here.

Lucene

Jakarta Lucene is a high−performance, full−featured text search engine written in Java and part of the Jakarta project. You can find more information at http://jakarta.apache.org/lucene/

Jetspeed

Jetspeed is a web based portal written in Java. It has a modular API that allows aggregation of different data sources (XML, SMTP, iCalendar)

3.1.2 Content management

The following are projects related to content management

Slide

Slide is a high−level content management framework. Conceptually, it provides a hierarchical organization of binary content which can be stored into arbitrary, heterogenous, distributed data stores. In addition, Slide integrates security, locking and versioning services. It also provides a WebDAV server and client implementation. You can learn more at the Slide home page.

Alexandria

Alexandria is an integrated documentation management system. It brings together technologies common to many open source projects like CVS and JavaDoc. The goal is to integrate source code and documentation to encourage code documentation and sharing. More information at http://jakarta.apache.org/alexandria/index.html

3.1.3 Frameworks

The following are application development frameworks.

Turbine

Turbine is a servlet based framework that allows experienced Java developers to quickly build secure web applications. Turbine brings together a platform for running Java code and reusable components. Some of its features include: Integration with template systems, MVC style development, Access Control Lists, localization support and so on. You can find more information at the Turbine web site.

Avalon

If you are familiar with Perl or BSD systems, Avalon is roughly the equivalent of CPAN or the Ports collection for Java Apache technologies. It does not only provide guidelines for a common repository of code, it goes one step further: *is an effort to create, design, develop and maintain a common framework for server applications written using the Java language.* It provides the means so server side Java projects can be easily integrated and build on each other. You can find more information at the Avalon web site.

3.2 Presentation

The following template systems, transformation engines and other presentation related projects.

Cocoon

Cocoon leverages other Apache XML technologies like Xerces, Xalan and FOP to provide a comprehensive XML publishing framework. The framework can talk to many different data sources and can transform the content into several different delivery formats such as PDF, HTML, XML and RTF. It can run as a servlet or as a command line program. You can learn more about Cocoon at the project homepage

Velocity

Velocity is a Java based template engine. It can be used as a stand−alone utility for generating source code, HTML, reports, or it can be combined with other systems to provide template services. Velocity has a Model View Controller paradigm that enforces separation of Java code and the HTML template. You can learn more about Velocity here.

AxKit

AxKit is a popular XML−based Application Server for mod_perl and Apache. It allows separation of content and presentation and provides on−the−fly conversion from XML to any format.

Xalan

Xalan is an XSLT processor available for Java and C++. XSL is a style sheet language for XML. The T is for Transformation. XML is good at storing structured data (information). You sometimes need to display this data to the user or apply some other transformation. Xalan takes the original XML document, reads transformation configuration (stylesheet) and outputs HTML, plain text or another XML document. You can learn more about Xalan at the Xalan Java and Xalan C++ project homepages.

FOP

From the website: *FOP is a Java application that reads a formatting object tree and then turns it into a PDF document*. So FOP takes an XML document and outputs PDF, in a similar way that Xalan does with HTML or text. You can learn more about FOP here.

3.3 Parsers and Document Access libraries

The following are different libraries that can be used to parse and manipulate a variety of document formats.

Xerces

The Xerces project provides XML parsers for a variety of languages, including Java, C++ and Perl. The Perl bindings are based on the C++ sources. An XML parser is a tool used for programatic access to XML documents. This is a description of the standards supported by Xerces:

- DOM: DOM stands for Document Object Model. XML documents are hierarchical by nature (nested tags). XML documents can be accessed thru a tree like interface. The process is as follows:
    - Parse document
    - Build tree
    - add/delete/modify nodes
    - Serialize tree
- SAX:Simple API for XML. This is a stream based API. This means that we will receive callbacks as elements are encountered. These callbacks can be used to construct a DOM tree for example.
- XML Namespaces
- XML Schema: The XML standard provides the syntax for writing documents. XML Schema provides the tools for defining the *contents* of the XML document (semantics). It allows to define that a certain element in the document must be an integer between 10 and 20 or contain an IP address.

The Xerces XML project initial code base was donated by IBM. You can find more information in the Xerces Java, Xerces C++ and Xerces Perl homepages.

Batik

Batik is a Java based toolkit for applications that want to use images in the Scalable Vector Graphics (SVG) format for various purposes, such as viewing, generation or manipulation.

It is XML centric and compliant with the W3C specification. It is a bit atypical from other Apache projects, in that it provides a graphical component. Batik provides hooks to extend the framework thru custom tags and it allows conversion from SVG to other formats like JPEG or PNG. You can learn more at the Batik homepage

POI

The POI project consists of APIs for manipulating various file formats based upon Microsoft's OLE 2 Compound Document format using pure Java. This includes Word and Excel documents. You can find more information at http://jakarta.apache.org/poi/

3.4 Interoperability

The following are libraries for remote communication and interoperability between servers.

SOAP

Apache SOAP ("Simple Object Access Protocol") and Axis are implementations of the SOAP protocol

SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts:

- *An envelope that defines a framework for describing what is in a message and how to process it*,
- *a set of encoding rules for expressing instances of application−defined datatypes*, and
- *a convention for representing remote procedure calls and responses*.

Basically you can think of SOAP as an remote procedure call system, based on HTTP and XML. On the one hand this means it is verbose and slow compared to other systems. On the other hand it eases interoperatibility, debugging and development of clients and servers for a variety of languages since most modern languages have HTTP and XML modules. You can learn more at the Apache SOAP homepage

XML−RPC

The XML−RPC project is a Java implementation of the XML−RPC protocol, a light−weight protocol similar and predecessor to SOAP.

XML security

The XML security project provides XML document signature verification for secure exchange of documents.

3.5 Development

Apache Portable Runtime

The APR project provides a portability layer that abstracts a number of APIs for file manipulation, network access and so on. It is written in C and works on most Unix flavors, Windows and a variety of other systems. It is the basis for Apache 2.0

Ant

Ant is a Java based build tool. It has a modular API and can be extended by creating new tasks. It is driven by XML configuration files.

Byte Code Library

The Byte Code Engineering Library (BCEL) is a library to analyze, create, and manipulate binary Java class files.

Log4j

This package provides a logging framework that Java applications can use. It can be enabled at runtime without modifying the binary and has been designed with performance in mind. It can be found at http://jakarta.apache.org/log4j/

ORO and Regexp

ORO is a complete package that provides regular expression support for Java. It includes Perl5 regular expression support, glob expressions and so on. All under the Apache license. You can learn more about ORO at http://jakarta.apache.org/oro/index.html. There is another ASF lightweight regular expression package, Regexp.

Struts

Struts is an Apache project that tries to bring the Model−View−Controller (MVC) design paradigm to web development. It builds on Servlet and JavaServer Pages technologies. The model part is made up of Java server objects, which represent the internal state of the application. The view part is constructed via JavaServer Pages (JSP), which is a combination of static HTML/XML and Java. JSPs also allow the developer to define new tags. The controller part consists of servlets, which take requests (GET/POST) from the client, perform actions on the model and update the view by providing the appropriate JSP. You can learn more at the Struts project pages.

Taglibs

The JavaServer pages technology allows developers to provide functionality by adding custom tags. The Taglibs project intends to be a common repository for these extensions. It includes tags for common utilities (i.e. date), SQL database access and so on.

You can learn about TagLibs at http://jakarta.apache.org/taglibs/. More documentation is included in the package.

Database

OJB is a database mapping tool that allows persistance and storage of Java objects in relational databases. Xindice is a native XML database for storing and querying XML documents.

Commons

The Commons project provides a great variety of reusable Java components with minimal dependencies.

3.6 Testing

The following ASF projects cover testing and performance analisys.

httpd−test

The httpd−test project provides a testing framework for the Apache web server and tools such as flood for HTTP load testing.

Cactus

Cactus is a testing framework for testing server side Java code such as Servlets and EJBs.

JMeter

This is a testing tool written in Java with a GUI frontend. It can be obtained at http://jakarta.apache.org/jmeter/.

Lakta

Lakta is an end−to−end HTTP testing tool

Watchdog

The Watchdog project is a suite of validation sets for the Servlet and JavaServer Pages specification.

---

4. Where to find more information

Additional Apache related resources

4.1 Websites

The following are some useful websites

- Apache Website
- Apache Week
- Apache modules directory
- Apache today
- Apache World
- Slashdot Apache section

4.2 Books

I maintain a list of books related to this document. It is not a comprehensive list, but rather I include only those books that I have personally found well−written and useful.

4.3 Support forums

You can find the Apache users mailing list at http://httpd.apache.org/lists.html. Similar lists exist for the rest of projects mentioned there. Make sure you read the Frequently Asked Questions document before posting . You can also get support in the newsgroup comp.infosystems.www.servers.unix at http://groups.google.com.

If you want commercial support, consider contacting Covalent, which provides expert support for Apache (at a fee, of course). If you are using Apache on Linux, your Linux vendor may have support plans that include Apache.

---

5. Contacting the Author

You can contact me at daniel @ rawbyte.com . I welcome suggestions and corrections, but please, please, do not send me messages asking me to troubleshoot your Apache installation. I just do not have the time to answer people individually. If you need support, please refer to the resources mentioned above.

5.1 Translations

If you want to contribute a translation of this document you should use the SGML source. Check http://www.tldp.org for info. Please drop me a note so I can make sure you get the most recent version.

---

6. Open Content Open Publication License

Open Publication License Draft v1.0, 8 June 1999 (text version)

6.1 REQUIREMENTS ON BOTH UNMODIFIED AND MODIFIED VERSIONS

The Open Publication works may be reproduced and distributed in whole or in part, in any medium physical or electronic, provided that the terms of this license are adhered to, and that this license or an incorporation of it by reference (with any options elected by the author(s) and/or publisher) is displayed in the reproduction.

Proper form for an incorporation by reference is as follows:

Copyright (c) <year> by <author's name or designee>. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, vX.Y or later (the latest version is presently available at http://www.opencontent.org/openpub/). The reference must be immediately followed with any options elected by the author(s) and/or publisher of the document (see section VI).

Commercial redistribution of Open Publication−licensed material is permitted.

Any publication in standard (paper) book form shall require the citation of the original publisher and author. The publisher and author's names shall appear on all outer surfaces of the book. On all outer surfaces of the book the original publisher's name shall be as large as the title of the work and cited as possessive with respect to the title.

6.2 COPYRIGHT

The copyright to each Open Publication is owned by its author(s) or designee.

6.3 SCOPE OF LICENSE

The following license terms apply to all Open Publication works, unless otherwise explicitly stated in the document.

Mere aggregation of Open Publication works or a portion of an Open Publication work with other works or programs on the same media shall not cause this license to apply to those other works. The aggregate work shall contain a notice specifying the inclusion of the Open Publication material and appropriate copyright notice.

SEVERABILITY. If any part of this license is found to be unenforceable in any jurisdiction, the remaining portions of the license remain in force.

NO WARRANTY. Open Publication works are licensed and provided "as is" without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non−infringement.

6.4 REQUIREMENTS ON MODIFIED WORKS

All modified versions of documents covered by this license, including translations, anthologies, compilations and partial documents, must meet the following requirements:

- 1. The modified version must be labeled as such.
- 2. The person making the modifications must be identified and the modifications dated.
- 3. Acknowledgement of the original author and publisher if applicable must be retained according to normal academic citation practices.
- 4. The location of the original unmodified document must be identified.
- 5. The original author's (or authors') name(s) may not be used to assert or imply endorsement of the resulting document without the original author's (or authors') permission.

6.5 GOOD−PRACTICE RECOMMENDATIONS

In addition to the requirements of this license, it is requested from and strongly recommended of redistributors that:

- 1. If you are distributing Open Publication works on hardcopy or CD−ROM, you provide email notification to the authors of your intent to redistribute at least thirty days before your manuscript or media freeze, to give the authors time to provide updated documents. This notification should describe modifications, if any, made to the document.

- 2. All substantive modifications (including deletions) be either clearly marked up in the document or else described in an attachment to the document.
- 3. Finally, while it is not mandatory under this license, it is considered good form to offer a free copy of any hardcopy and CD−ROM expression of an Open Publication−licensed work to its author(s).

6.6 LICENSE OPTIONS

The author(s) and/or publisher of an Open Publication−licensed document may elect certain options by appending language to the reference to or copy of the license. These options are considered part of the license instance and must be included with the license (or its incorporation by reference) in derived works.

A. To prohibit distribution of substantively modified versions without the explicit permission of the author(s). "Substantive modification" is defined as a change to the semantic content of the document, and excludes mere changes in format or typographical corrections.

To accomplish this, add the phrase `Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.' to the license reference or copy.

B. To prohibit any publication of this work or derivative works in whole or in part in standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the copyright holder.

To accomplish this, add the phrase 'Distribution of the work or derivative of the work in any standard (paper) book form is prohibited unless prior permission is obtained from the copyright holder.' to the license reference or copy.