The Moto Programming Language

A White Paper

Kenneth Kocienda
David Hakim

Webcodex, Inc.
840 C Street #621
San Rafael, CA 94901
http://www.webcodex.com

April 4, 2000

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE DOCUMENT. WEBCODEX, INC. MAY MAKE IMPROVEMENTS AND / OR CHANGES IN THE PRODUCT(S) AND / OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

Introduction *1*

This paper introduces *The Moto Programming Language* (hereinafter referred to as “Moto”), an open-source programming language for developing flexible, robust, high-performance, HTTP server-side Internet applications.

This first section includes a brief discussion of the features of Moto and describes why Moto is a significant advance over comparable technologies that are now available.

Why Moto?

The technology for developing HTTP server-side Internet applications has evolved significantly since the days when applications were typically constructed on top of Common Gateway Interface (CGI) scripts. Today, a wide array of technologies are available for developing feature-rich HTTP server-side Internet applications, technologies, making this type of development easier than ever before.

The most popular offerings fall into several broad categories. There are proprietary solutions like Microsoft Active Server Pages (ASP) and Allaire ColdFusion (ACF). There are “standards-based” technologies like Java Server Pages (JSP) that are offered by several vendors. There are server plug-in architectures like Netscape NSAPI, Microsoft ISAPI, and Apache modules. Finally, there are open source alternatives like PHP.

Having such a range of choices seems wonderful at first glance. Yet, choosing any one of these technologies involves making significant trade-offs. Some run as interpreters and experience serious performance degradation when placed under high load. Some offer only simplistic session management systems that hamper the development of sophisticated applications. Some involve using complicated APIs and are targeted at experienced application developers and software engineers.

The proprietary systems raise a whole range of additional complications. All are expensive. All ship with restrictive licensing agreements. All rely solely on the technology vendor for bug fixes and feature enhancements. Some of the proprietary systems involve making a commitment to a single vendor.

Indeed, none of these technologies combine ease of use, high performance, and robust session management in an open-source package that gives developers the maximum level of flexibility throughout the development and deployment process. But now there is a solution that does: Moto. Moto has been designed from the ground up to deliver features that provide HTTP server-side Internet application developers with a unique combination of power and flexibility:

- Moto is a simple yet full-featured programming language.
- Moto is free of costly and complex third-party dependencies.
- Moto offers dual-mode execution: interpreted mode for quick iterative development; compiled mode for native-code performance.
- Moto has robust session management.
- Moto developers can hide their application code.
- Moto can be extended easily.
- Moto is open source.

Simply put, Moto is the next step in the evolution of HTTP server-side Internet application development technology.

Moto and Comparable Technologies

As the following matrix shows, Moto offers a feature set that is not available from comparable technologies. It is this feature set that makes Moto a uniquely powerful solution for developing HTTP server-side Internet applications.

Feature Comparison Matrix

Feature	Moto	PHP	ASP	ACF	JSP	SPI
No vendor lock-in	X	X				
Simple language	X	X	X	X	X	
Quick code-test-debug cycle	X	X	X	X		
Native-speed execution	X					X
No third-party dependencies	X	X				
Can hide source code	X				X	X
One-file application deployment	X					X
Robust built-in session mgmt.	X		X	X	X	
Open source	X	X				

Legend
ASP: Microsoft Active Server Pages
ACF: Allaire ColdFusion
JSP: Java Server Pages
SPI: Server Plug-in APIs (e.g. Netscape NSAPI and Microsoft ISAPI)

Key Features of Moto 2

This chapter discusses each of the key features of Moto in greater detail. It provides a more in-depth introduction to Moto and the unique features that distinguish this technology from comparable offerings.

Moto is a simple yet full-featured programming language.

Moto is a strongly-typed language that offers flexible control constructs, C-like expressions, and a preprocessor with enhanced features for building complex macro libraries.

Programming in Moto will seem familiar to experienced C, C++, Java, Perl and PHP programmers. The syntax and semantics of Moto owe much to these languages, and a conscious decision was made by the Moto developers to model the language after well-established conventions.

For developers who have not yet picked up any of these "more traditional" programming languages, but are comfortable with writing HTML, Moto provides a shallow learning curve, allowing developers to begin with simple features and advance to more complex functionality at a self-directed, comfortable pace.

Moto is free of costly and complex third-party dependencies.

Most technologies for developing HTTP server-side Internet applications are proprietary, requiring the purchase of a license. These licenses can be expensive and, of course, only cover the current shipping version.

In addition, many of these same technologies have complex system requirements, and call for the purchase specific operating systems, web servers, or database drivers—all of which have costs, licenses, and dependencies of their own. Clearly, this complex web of licensing and dependencies can become daunting.

Moto sidesteps this license and dependency trap. Since Moto is open source, it is free for all commercial and non-commercial use. In addition, Moto has been designed to work with other open source software, such as the Linux operating system and the Apache HTTP Server, so the environment and supporting code needed to run Moto applications will always be available free of cost.

Moto offers "dual-mode" execution: interpreted mode for quick iterative development; compiled mode for "native-code" performance.

The Moto language ships with both a language interpreter and compiler, and this dual-mode execution capability is one of the Moto language's more compelling features. The reason for dual mode execution is simple: it enables the quick code/test/debug cycles that are possible when developing with interpreted languages, without sacrificing the speed of native code execution.

What's more, using dual mode execution could not be simpler. While creating Moto applications, developers execute them with the interpreter, and they get the quick "make the change, see the change" feedback that makes iterative development proceed quickly. Once the application is finished and is ready to move into production, the Moto compiler is used to build the application into a machine-native binary—and the application runs anywhere from 100x to 1000x faster than it did with the interpreter.

Moto offers robust session management.

Session management has long been one of the more complicated issues in HTTP server-side Internet application development. Typically, session management is implemented with client-side cookies or some ad-hoc

URL passing mechanism. While this can be made to work, a common result is that developing complex session management schemes is a task that consumes a large amount of application development time. It is also very often a one-off effort that needs to be modified for each new application page or feature.

Moto offers a way out. Moto includes robust, HTTP server-side session management, providing semantics for managing application-wide, session-scoped, and page-specific data. This system makes a straightforward process out of developing applications that involve multi-page interactions. And yet, Moto makes all this powerful HTTP server-side functionality available through a simple-to-use API built around a super-slim URL passing mechanism.

Moto developers can hide their application code.

For contractors who developed HTTP server-side Internet applications, shipping source code out to customers is part of the normal process. While this may be acceptable to some, many developers may want to hide code in which they have invested a great deal of time and effort. Unfortunately, with interpreted languages like PHP and Microsoft ASP, there is no choice: when the application ships, the source code ships with it.

Moto offers a choice. Since Moto applications can be compiled down to a machine-native binary, the application source code is not required to run the application, as it is in an intrepreted-language environment. With Moto, developers can control the degree to which customers see source code.

Moto can be extended easily.

With Moto, new extensions integrate smoothly. Moto includes a fully-documented extension API for extending the Moto language and runtime system. Moto also offers an Interface Definition Language (IDL) for binding to pre-existing C language libraries. And, since Moto is open source, there are no secrets, and no guess work is needed to see how the Moto internals have been designed and implemented.

Moto is open source.

Moto developers will never wonder about what will happen to their technology investment, as they would if the vendor of their proprietary technology gets acquired, retargets its business objectives away from Internet application technology, or worst of all, goes out of business.

The Moto source code is open, which guarantees it will always be possible to support and extend the technology.

Moreover, the fact that Moto is open-source technology is the result of a purposeful choice. In recent years, projects like GNU, the Linux operating system, and the Apache HTTP Server have demonstrated the clear benefits of open source development. Moto hopes to build on the momentum generated by these projects and the open source ethic to deliver the best, most efficient and, most powerful HTTP server-side Internet application technology yet seen.

The Moto Programming Language—A New Beginning

Moto is the next step in HTTP server-side Internet application development technology. No longer must developers compromise when choosing a technology for creating dynamic web applications. No longer must developers settle for the slowness of interpreted languages. No longer must developers struggle with complicated application deployments involving hundreds of pages. No longer must developers give away application code. No longer must developers work around inadequate support for complex multi-page interactions. No longer must developers lock in to a single technology vendor.

With Moto, high-performance, robust session management, open source, and the greatest degree of design, development, and deployment flexibility is available today.