

# The PSP Language

## Introduction

The PSP (Pre-compiled Server Pages) programming language is designed to provide developers of dynamic web content with the debugging output common to compiled languages within the structure of scripting languages popular for web programming. A PSP compiler will give debugging output against PSP code until the code fits its simple, yet strict syntax. Properly written PSP code will be compiled into guaranteed syntax-perfect JSP pages.

The strengths of the PSP language will include:

- Verbose debugging output
- Separate dynamic rendering code from static html from.
- Java functionality
- Algorithmic things, like control structures!
- Simplified Java syntax

## Background

JavaServer Pages (JSP) allows you to create dynamic web pages. In its basic form JSP pages are HTML pages with special embedded tags that execute application logic to generate dynamic content. JSP are an extension to the Java Servlet API. A JSP file is compiled into a Java Servlet, which executes in a Servlet engine on the application server. Therefore, JSP pages have full Java functionality and have all the benefits of a typical Java program.

One of the pitfalls new JSP developers run across is that they don't spend much time on design and they soon end up with a "fat" JSP that contains both presentation logic and business logic. They soon realize that their JSP files, because of the mixture of HTML syntax and JSP tags, have become unreadable. By separating out the application logic for rendering dynamic content into a separate file we can maintain the purity of the HTML source and avoid a point of confusion for application logic developers.

## **The Advantage of Pre-compilation**

With any current web programming language, such as Perl, PHP, or JSP, no code is parsed until a page is viewed. If there is a syntax error in that code, nothing will be displayed - because the program didn't parse correctly, there is ZERO OUTPUT! With no debugging output, it is the programmer's responsibility to read through his or her entire program to find a subtle missing semi-colon or closing parenthesis. Instead of wasting time and energy like this, a programmer could write his code in PSP, and use the PSP compiler to generate syntax-perfect JSP pages.

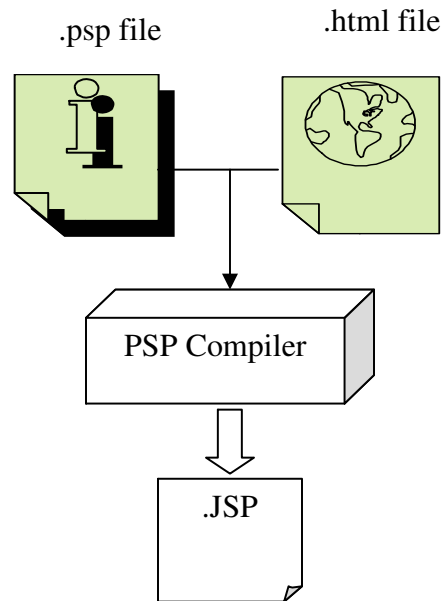
The PSP compiler will first check for proper PSP syntax. If there are mistakes, debugging output will specify what the mistakes are and which line they occur on. If syntax is correct, the PSP compiler will generate a JSP code file with perfect JSP syntax, from bits as simple as semi-colons and braces to structures as complex as for-loops and if-else statements - and of course the evasive `<%` and `%>`.

## **Java Functionality**

Because PSP compiles into JSP pages, PSP offers much of the functionality of the Java language - dynamic content from 'request' and 'session', complex control structures, HTML user interfaces, and platform independence. With PSP, the powerful features of JSP are wrapped in an extremely convenient programming environment.

But PSP will not offer FULL Java functionality. PSP will force good programming habits by leaving the most powerful processing jobs to be handled by back-end server code. The PSP language is focused on providing a user interface with dynamic information gleaned from more powerful back-end software.

## Design



**Fig1:** Design overview of the PSP compiler

The PSP compiler is designed with the intention of separating the code for rendering dynamic content from the static content that is used as the baseline. It does so by reconciling a dynamic .psp and a static .html file, and combining the two into a .jsp file. As such, the compiler recognizes both PSP source files and HTML.

The compiler first parses the PSP code and compiles it then uses the PSP code to add dynamic content to the HTML file. After which, the resulting file is a JSP. If there is an error at any of these steps, the compiler exits and reports the errors in a context-specific, user-friendly manner. Error reporting will include information as to whether it was the .psp file or the linking step that generated the error, as well as line number(s) of erroneous code, and as much detail about what caused the error as possible.

PSP syntax is designed to be as convenient and familiar as possible. It is readily understandable and readily programmable by a Java developer.

## Basic Syntax

Below is an overview of the basic syntax for the PSP language. The syntax is intended to be very similar to Java syntax as that is typically a required skill set for your average JSP programmer. The following syntax gives you an idea of how the language is used to create dynamic content.

- Forms and Input Fields

HTML	<code>&lt;form name="form1" method="post" action=""&gt;</code>
PSP syntax	<code>\$form1.action = "cnn.com"</code>
JSP	<code>&lt;form name="form1" method="post" action="&lt;% out.print("cnn.com") %&gt;"&gt;</code>

- Conditional Expressions

HTML	<code>&lt;input type="text" value=""&gt;</code>
PSP syntax	<code>if (a == 2) then \$form.fieldname = "HELLO"</code>
JSP	<code>&lt;input type="text" value="&lt;% if (a == 2) out.print(O'Neil);%&gt;"&gt;</code>

## Limitations

The target audience for the PSP language is novice JSP developers. Currently, the language only provides basic functionality for generating dynamic web content. HTML and JSP are both extensive languages therefore we will only focus on a narrow area, forms, input fields and tables.

## Conclusion

The goal of the PSP language is to remedy the failings of other dynamic web scripting languages. Existing languages primarily fail due to difficulties in error detection. As a result a considerable amount of time is spent debugging. We have outlined a design that should create a valuable compiler that addresses some of the shortcoming on current technologies used to render dynamic content on the web. We feel that the separation of the code for rendering dynamic and the static portions of web documents will make PSP an attractive alternative to web developers. The power of PSP lies in its ability to simplify the overall development effort by generating JSP that are guaranteed to compile. While its power is only in its infancy PSP shows a great deal of promise in terms of the solutions it may one day solve.