Advantages of PML as an iSeries Web Development Language

What is PML
PML is a highly productive language created specifically to help iSeries RPG programmers make the transition to web programming and to provide all the power needed to create simple or sophisticated web applications. PML is extensively supported by WebSmart's IDE, making it easy to learn and easy to work with. Nearly 95% of WebSmart users are RPG programmers. They have proven it is easy to be successful learning web development with PML.

PML – The Best of Web Languages and RPG
PML was specifically designed to take concepts familiar to RPG and iSeries programmers and use them in a true web development world. For example, PML lets you write typical record-level access code, just as RPG does. It also supports a consistent syntax for writing embedded SQL, so you can prepare your applications for wider platform deployment. In addition, PML generates RPG code behind the scenes. You can compare this to the RPG compiler producing MI code – you really wouldn’t want to learn to read it, but you probably realize that MI code is native to the iSeries and therefore gives you the best execution performance. That's true of PML too – because it generates compiled RPG programs, you get the best possible native execution speeds, without having to actually code messy RPG in an unproductive environment.

While many of the implemented programming concepts in PML will be familiar to RPG programmers, the language syntax is similar to free-format RPG. It is also similar to Java, Javascript, PHP, ASP or ColdFusion. These are all languages that have their primary use in web development. For example, millions of web sites, including Yahoo! are written using PHP.

In the Web World, RPG is a Proprietary Language
Contrary to what you might be used to, RPG is in fact a proprietary language. Currently it is used almost exclusively on iSeries systems, and most often for 5250 green-screen programming. While various web development APIs have been built (such as CGIDEV2), these require learning a set of specialized function names and parameters in order to use them effectively. In addition, the most common coding environments for RPG such as SEU are typically most effective for coding green-screen applications, due to constraints in the lengths of source code lines. For example, if you need to embed HTML, it is difficult to do in a traditional green-screen coding environment, compared to green-screen display files. No other platform uses RPG as a language in which to write web applications. It is highly specialized and excellent for iSeries batch and interactive programming, but web development on all other platforms (Windows, Unix, Linux, iSeries PASE environment, etc.) uses more widely available and commonly known languages.

PML Helps Move You To True Web Programming
PML is embedded in WebSmart’s powerful development environment. This environment provides you with real-time color-coding and formatting of your code. It also provides real-time syntax-checking to eliminate compile errors and minimize logic errors. For example, if you type a PML function to write a record to a file, the editor will prompt you to allow you to choose a valid available file name as a parameter- eliminating guesswork and possible mistakes. Also, because PML uses a syntax that is similar to other web languages such as Javascript and PHP you minimize your web development learning curve. If you get familiar in coding in PML, you’ll more easily get comfortable with these languages, too. For example, most commercial web
applications require some level of client-side coding in Javascript to handle routine functionality such as communications between browser windows (such as pop-up calendars and date selections). Learning PML is very similar to learning Javascript, so you can leverage your educational efforts as you make the transition. WebSmart’s IDE also frees you from the constraints of fixed-length line coding that RPG confines you to. This allows you to have code that is indented for easy readability, making it easier to debug than RPG. It also allows you to have long statements where necessary. For example, embedded SQL is often very awkward to code in RPG because it must span several relatively short lines. You are freed from that constraint with PML and WebSmart’s IDE.

**PML Is Optimized For Web Applications**

Under the covers, PML encapsulates many hundreds of lines of ILE code (either RPG, C or CL, for example) into consistently defined one-line functions. While it hides the complexity of this code from you, it embeds optimization routines that make web applications built with PML perform as fast as possible. For example, PML has a function to write HTML to a page (called wrthtml). Behind the scenes, this function uses an algorithm to cache content to be written to the page to minimize calls to native iSeries web APIs that are costly in CPU cycles. Again, this is all transparent to you as a developer – you simply type the function name followed by any parameters, and PML does the rest.

**PML Has Features For Native Use of Existing OPM or ILE Code (RPG, CL, COBOL, etc.)**

PML does not constrain you from using existing code written in any native iSeries language, including RPG code. For example, you can directly call RPG programs, modules or service programs. In addition, WebSmart’s IDE eliminates the headaches of having to worry about how to bind in any modules at compile-time- it takes care of that housekeeping for you, behind the scenes. This makes it easy for you leverage your existing code base, so you can take extensive advantage of your code assets.

**PML Has Many Powerful Functions To Build Complex Web Applications**

PML includes all the functions for conventional business database programming along with an extensive set of functions useful for web programming. For example, PML includes:

- Functions for standard Web interaction, including redirection to other pages, handling passing values between pages and the server, constructing URLs to redisplay pages, functions to make pages safe by stripping out unwanted HTML prior to sending data, functions for creating web components such as data-aware dropdown boxes, etc.
- Functions to support direct access to IFS (stream) files, making it easy to create or read unstructured data such as text files or email attachments.
- Functions for handling iSeries user profile security (login, password checking, etc.) to make creating custom HTML-based login pages easy.
- Functions for iSeries-specific tasks such as working with library lists and data areas, running iSeries commands, calling programs and modules, etc.
- Functions for SOA web services – allowing secure or unsecured server-to-server communications over the HTTP protocol possible. This is useful for collating information from many partner sites such as FedEx for shipping status or mapping sites for maps and integrating them in your own pages.
- Functions for AES (128-bit) encryption – allowing secure storage of sensitive data such as login info or credit card numbers in server-side databases or client-side cookies.
- Functions for web session management and cookie management – making it possible to write sophisticated e-commerce applications that manage logins with potential timeouts, etc. and ‘remember me’ features to facilitate return visits to your site.
- Date manipulation and arithmetic functions – making it easy to calculate date ranges, future dates, etc. These functions help eliminate many lines of RPG coding by handling common business needs for date data easily.
- Multi-language support functions – making it easy to present a website in different languages for different users.

**PML Is Highly Productive when compared to RPG**

PML is specifically engineered for web applications, making it highly productive to code in. For example, one line of PML can produce a drop-down box with a filtered list of values, populated using a database file. Here’s a commented example:

```pml
// Display a dropdownbox named "contacts" of all the contact
tnames for a Customer from the Customer (CUSTMASTF) file.
// Subset the list using the input customer number (custno) as
// a key.
// Show the Contact name (cmcontname) in the dropdown and
return the contact number (cmcontno) when the user selects it.

crtklist(custkey, custno); // Klist of customer number

dropdownbox("contacts", CUSTMASTF, cmcontno, cmcontname,
contactno, custkey );
```

Compare this to the *minimum* amount of RPG code required to do the same thing, assuming you already have a function written (like outweb_ in this example) that lets you write server content to a browser:
As you can see, PML shields you from the complex and cumbersome coding requirements of RPG to output even commonly used HTML components to web pages.

PML Shields You From Future Technology Changes

Because PML encapsulates the underlying code and hides it from you, it makes it easy for us to prepare for future technologies. Already, PML can generate RPG or Java servlets. You simply write in one language, PML, then target your applications for either RPG or Java servlets, or both. This is a powerful feature, because even though RPG and Java are coded completely differently, you only need to know how to write in one language- PML. In the future, we can readily target PML to other languages or common web technologies. These could all run on other platforms besides the iSeries. In contrast, RPG is highly specialized for the iSeries server and for iSeries business programming, and is proprietary in syntax and form.

PML Is Extensible

You can encapsulate your own routines into simple one-line functions with PML. For example, if you have a complex pricing routine written in RPG, with PML it is possible to place a simple PML ‘wrapper’ around that code, where the insides are either a code snippet such as a subroutine, or a

```plaintext
C    CUSTKEY       KLIST
C    KFLD          custno
* Write HTML to start creating the dropdown box:
C    CALLP  outweb_('<select size="1" name="')
C    CALLP  outweb_('contacts' )
C    CALLP  outweb_('>')
* Position at the customer number:
C    custkey     SETLL    RCUSTMST
* Loop to read all contacts of this customer
C    custkey     READE(N) RCUSTMST
C    DOW         %eof = *OFF
C    IF          cmcontno = contactno
C    CALLP       outweb_('<option selected value="')
C    ELSE
C    CALLP       outweb_('value="')
C    ENDIF
C    CALLP       outweb_(%editc(cmcontno : 'X'))
C    CALLP       outweb_('>')
C    CALLP       outweb_(cmcontname)
C    CALLP       outweb_(</option>')</n
* Next record:
C    custkey     READE(N) RCUSTMST
C    ENDDO
* Write the closing tag for the dropdown:
C    CALLP  outweb_(</select>')
```
self-contained compiled program or module. This gives you the added advantage of correct prompting and syntax checking of your PML code, so that like shipped PML functions you are protected from syntax and compile errors. What's more, you are not confined to wrapping PML around RPG - the underlying code could be written in any ILE language, including CL or C. For example, if you've written code to use some of the system API's, you may find it helpful to define PML functions to simplify calling those API's.

**Tools For Learning PML**

While learning any new language may seem daunting at first, we have provided many supporting features to help you learn PML as easily as possible. These include:

- Intelligent templates and wizards in WebSmart's IDE – these write all the code for you (both PML and HTML) to create fully functioning programs with no programming required. The PML code is completely documented and explained, so you can use the programs you write with the templates as a starting point for learning PML.
- A comprehensive User Guide (in print or online) that steps you through building a complete working and practical web application. At various steps along the way, the PML used to perform certain tasks is explained in detail.
- A comprehensive Reference Guide that describes the PML language syntax in detail and explains every PML function completely. Most functions also have detailed working example code snippets you can cut and paste directly into your applications.
- PML snippets in the IDE. You can drag and drop these directly into your code so that you have pre-built code directly available for you.
- WebSmart's IDE lets you drag and drop function names directly into the code, so you don't even have to remember how to spell them - each function has a brief description of its purpose next to it.
- WebSmart's IDE has intelligent prompting for function parameters, including context-sensitive lists of valid values. It also has context-sensitive color-coding and code indentation, and real-time error checking.
- WebSmart's IDE has context-sensitive Help – hit F1 on any function name and the online Reference Guide will be positioned at that function. From here you can see example code or related functions that might be helpful.
- Comprehensive online knowledge base of articles related to both PML and WebSmart development in general. Hundreds of articles are available to give you help in solving common web programming problems.
- Online access to hundreds of example WebSmart programs. We are continually updating our library of WebSmart examples so you can exploit our expertise and expand your PML skills – all at no additional cost.
- A PML to RPG Coding 'cheat-sheet'. This shows the most common coding constructs in RPG, and explains their equivalent in PML.

**WebSmart and PML Ease the Challenges of Web Development**

Doing web development will require some change in your thinking, and some re-education. It is quite different than green-screen programming. Web programming requires what is commonly referred to as a "stateless programming" approach that requires a different way of designing and thinking about development. It also requires you to learn some HTML programming instead of SDA or display file DDS coding. You want to have graphical pages that are easy to navigate and run efficiently on your iSeries and that look and behave like true web applications. No matter what tool or language you use, these issues are the same.
All development tools incorporate a unique language or set of functions, whether they claim to be RPG-based or not; otherwise they would not allow you to do something for an environment that's different than iSeries RPG green screen programming. Even if a product purports to use conventional RPG, it will still require you to use its own proprietary functions or programming constructs that are different than native RPG operation codes. While there's a learning curve for any tool, WebSmart insulates you from the long web programming learning curve and allows you to quickly become an expert. This is partly because PML is especially designed for web programming no matter what the underlying technology. And, WebSmart’s IDE is specifically built with hundreds of features to make coding in PML as simple, productive and foolproof as possible.

WebSmart and PML are the superior choice to help you take on the challenges of Web Development today while protecting your investment for the future.