XIA INFORMATION ARCHITECTS CORPORATION

XML Processing and Application Integration Using Perl

Amsterdam – March 26, 2001

Tomas Hansson, XIA Information Architects

What is Perl?

- Practical Extraction and Report Language
- Larry Wall
- Begun 1993 – Currently at version 5
- “There's more than one way to do it.”

Why Perl?

- Shallow learning curve
- Unlimited expandability
- Object-oriented
- Great library of tools
- Good industry support
- Huge programmer pool

Why Perl for XML?

- Many free modules
- Great pattern recognition
- Quick results
- but
- No easily available validating parser

Tools

- ActiveState's Active Perl Version 5.6 Build 623
- PPM (Perl Package Management) Packages:
  - XML-Parser (Event driven)
  - XML-DOM
  - XML-XPath
  - Tk: XMLViewer

Perl Installation

- Install ActiveState Perl from:
  - Web site: www.activestate.com
  - or
  - From CD

Basic Syntax

- Hello World – Example

Basic Syntax – Variables

- Scalars
- Arrays
- Associative Arrays
- References (pointers)

Basic Syntax – Statements

- if, unless, while, until, for, foreach, continue
- goto, last, next, redo
- do while
- do until

Basic Syntax – Functions (Built-in)

- Arithmetic, conversion, string, array and hash
- Regular Expressions, search and replace
- File, file tests, input/output
- Formats, directory reading, system interaction
- Networking, system V IPC

Basic Syntax – Functions (User-def.)

- sub definitions

Basic Syntax – Packages/Modules

- Database access
- File utilities (basename, dirname, etc.)
- HTML/HTTP
- Math
- Text manipulation
- XML!

Perl vs. Omnimark – The Good

- Perl is free! – Omnimark isn't (anymore)
- Perl is a programming language
- Choice of in-memory processing (DOM)
- Perl is open standard
- Perl is not proprietary
- A lot easier to find Perl programmers

Perl vs. Omnimark – The Bad

- Omnimark doesn't require programming
- Omnimark is validating
- Omnimark has very advanced constructs
- Omnimark is process driven
- Omnimark can parse input or output streams
- Omnimark will process SGML

Perl vs. Python – The Good

- Perl has greater industry recognition
- Python is still considered “academical” and not production quality
- Perl has much better pattern matching and text processing functionality

Perl vs. Python – The Bad

- Python has more XML modules
- Python has validating parsers
- Python has a great user community

Perl vs. XSLT – The Good

- Perl is a programming language
- XSLT has a hard time with complex expressions
- XSLT is resource hungry
- A lot easier to find Perl programmers

Perl vs. XSLT – The Bad

- XSLT is easy to learn
- XSLT has built-in support in existing software
- XSLT is a recognized W3C standard

Literature

- According to Amazon.com:
  - XSLT – 9 matches
  - Omnimark – 3 matches
  - Python – about 25 (+ >100 about Monty Python)
  - Perl – **402!** (Not sure how many on knitting)

When to use Perl?

- Processing too complex for XSLT
- Small budget, quick turn-around time
- Perl expertise available

Perl XML Modules

- XML::Parser (Event-driven)
  - Based on James Clark's Expat (XML Parser Toolkit) parser
  - Expat is a non-validating parser
- XML::DOM
- XML::XPath
- Validating XML parsers are available (Apache's Xerces)

Perl and module installation

- Use PPM to install additional modules
- or
- Extract XML-modules.zip into top-level Perl directory

Example XML -> HTML Conversion

- Canadian Hansard
  - Using Perl
  - Using Omnimark
  - Using XSLT

Example XML Data Extraction

- Retrieve all speakers
  - Using Perl – XML::Parser
  - Using Perl – XML::DOM
  - Using Omnimark
  - Using XSLT

PerlScript

- ActiveX scripting engine
- Installed automatically with ActiveState Perl
- Full language features

XMetaL programming

- Access via:
  - DOM-based interfaces
  - VBA-based interfaces
- Supports Microsoft Scripting Language Interface

Installing XMetaL

Testing XMetaL

Simple DTD

```
<!ELEMENT minutes (title, para+) >

<!ELEMENT title (#PCDATA) >

<!ELEMENT para (name, title?, text) >

<!ELEMENT name (#PCDATA) >
<!ATTLIST name id ID #IMPLIED >

<!ELEMENT title (#PCDATA) >

<!ELEMENT text (#PCDATA) >
```

XMetaL manipulations

- Change “FirstName LastName” to “LastName, FirstName”
- Insert empty title tag
- Populate title tag from “database” file