Clarion, the Web, OOP and You

Building a Web Site with Clarion for Windows

Tom Hebenstreit
Consulting Services

Introduction

- Leveraging your current skills to create powerful web sites
- Gathering and displaying information interactively
- Using Clarion to write WWW scripts
- Using Templates to simplify your Web programs
- Using OOP to simplify using web objects

Topics to Cover

- World Wide Web Basics
- Communicating with a Web Server
- Processing Input
- Creating Web Pages dynamically
- Transferring Files to/from the web

Web Basics

- HTTP - the language of the Web
- Hyper Text Markup Language (HTML) - the language of browsers
- What is a web page?
  - Static Pages
  - Dynamic Pages
- Cookies

How the web works

- Browser request via URL or link (HTTP)
- The Web server figures out what to do
  - Static Pages
    - » Server finds page (file) and sends it back
  - Dynamic Pages/Links
    - » Server runs your program, which creates a pages and passes it back to the server
    - » Server returns page created by your program
- Browser parses/displays the page (HTML)

Linking your programs with the Web Server

- IDC (Internet Database Connector)
- Standard CGI (Common Gateway Interface)
- Win-CGI (CGI adapted to Windows)
- ISAPI (Internet Server Application Program Interface)
- Other methods (ADO, ActiveX, JDBC, etc.)

Internet Database Connector (IDC)

- Advantages
  - No programming required
  - Uses HTX files
- Disadvantages
  - Limited control
  - Requires ODBC connection
  - Basic knowledge of SQL needed
  - Developed for IIS

WIN-CGI

- Adapted from CGI to overcome limited Windows support for Unix methods
- Advantages
  - Simple interface, no API calls
  - Programs can do whatever you want
- Disadvantages
  - Limited support under IIS
  - Programs load/unload each time they are run
  - Extensive use of temp files to pass information

Standard CGI

- Developed for use on Unix Web Servers
- Advantages
  - Supported by virtually all servers
  - Use environment to pass information
  - Programs can do whatever you want
- Disadvantages
  - Programs load/unload each time they are run
  - Must use API calls to get Form data

Internet Server Application Program Interface (ISAPI)

- Proprietary - developed by Microsoft for IIS
- Advantages
  - Runs in Server program space
  - Fast response
- Disadvantages
  - Very complicated to program to natively
  - Clarion is not 'thread safe'

Using Tornado/ISAPI

- Tornado OCX mediates between Clarion and ISAPI
- Advantages
  - Fast
  - Supports HTX (no writing HTML)
  - Run multiple instances of program
- Disadvantages
  - Must purchase Tornado OCX
  - Requires ISP to install OCX on server

Gathering Information: Web Queries and Forms

- Creating the Form
- Linking to or invoking your program
- 'GET' versus 'POST'
- Field Types
  - Entry
  - Checkboxes
  - Options
  - Drop lists
  - Multi-line Text area

Validating Web Form Input

- Very limited control over input
- MaxLength for Entry fields
- No browser limits on TextArea fields
- Checking can only be done after Submit
  - Checking before your program using JavaScript, VBscript, Java, etc.
    - » Not supported by all browsers
  - Checking within your program

Basic Tools for Web Work

- Clarion for Windows 32-bit
- Some knowledge of HTML
- A Web Page authoring program
- A Web Browser (preferably Netscape, IE and an older browser)
- A Web Server for Testing

Writing a Clarion Script

- Be aware of target Web Server
- Choose the method to use
  - Speed, standardization, simplicity
- Create your Application
  - Process Requests and Form Data
  - Creating Pages to Return

Processing a Web Request: GET

- All information is passed in the 'Query String' field as one long string
- Query String is 'URL Encoded'
  - Basic format is Field=Value
  - Spaces converted to '+'
  - Non-standard characters are passed as 'Escape Sequences'

Example: URL Encoding

- Raw Form Data
  - Name=Bob Smith!
  - Phone=(310) 111-2222
- URL Encoded
  - Name=Bob+Smith%21&Email=%28310%29+111-2222
- Templates, etc., provide methods to automatically decode (be sure to use them!)

Processing a Web Form: POST

- Information is passed in Field=Value pairs
- Must retrieve each field individually
- Field data may need to be 'cleaned up' (remove escape sequences, etc.)
- How options and checkboxes are passed
- Creating a 'Cancel' option

Creating Pages to Send Back

- Process the input request
- Decide what to do with it
- Return something
  - Forms
  - Tables of records
  - Handling errors or missing data
    - » Using 'Back' versus a URL/Link

Creating Pages: Writing HTML line by line

- Works with all methods (Win-CGI, CGI, Tornado/ISAPI)
- Must understand basics of HTML
- Your script creates the entire page via HTML hard-coded into the program
- Changing the page means changing the program

Creating Pages: Using an HTX file

- HTX file exists totally separate from your program
- No HTML knowledge required
- Create string of '<%field%>=data' pairs to pass to server
- Web Server combines HTX file and your variables into an HTML page to send back

Creating Pages: Using Psuedo HTX

- Pre-designed Web Page is stored in Memo or ASCII File
- Perform your own 'merge' of data and HTML
- Read, process and write each line
- Program does not need recompile for simple changes to page

Real Life

Examples of working applications

FTP - File Transfer Protocol

- Used to transfer files between computers linked via the Internet
- Bi-directional - both upload and download
- Operates in a client/server fashion
- Must have User Name/Password to login
- Anonymous FTP allows anyone to access a site (or a portion of one)
- Transfers always initiated by the client

Transferring Files via FTP

- Manual/Interactive methods
  - Via your online service (AOL or CSi)
  - Using a dedicated program (CuteFTP, etc.)
  - Win 95 FTP command line
- Automated
  - Win 95 FTP program via batch file
  - Adding FTP to your programs using WinInet.DLL

Using WININET FTP Services

- Establishing the connection
- Retrieving directory Listings
- Working with directories
- Working with files
  - Upload/Download
  - Rename, Delete, Etc.
- Ending the session
- Other services (HTTP, Cookie, Gopher)

Examining the Code, Classes and Templates

- Win-CGI Examples
  - Classes
- CGI Examples
  - Classes
- Tornado/ISAPI Examples
  - Output classes

Learning More…

- Study the examples
- Books
  - *Up to date list provided at DevCon*
- Web Sites
  - *Up to date list provided at DevCon*
- Other resources

Summary

- You can create scripts as simple or complicated as you want using Clarion.
- You can write your own HTML or avoid it completely
- You can write, compile, test and debug all on one machine
- Using the provided Templates and examples -- IT'S EASY!!

Questions

- Ask away!

*** Thank you for attending! ***