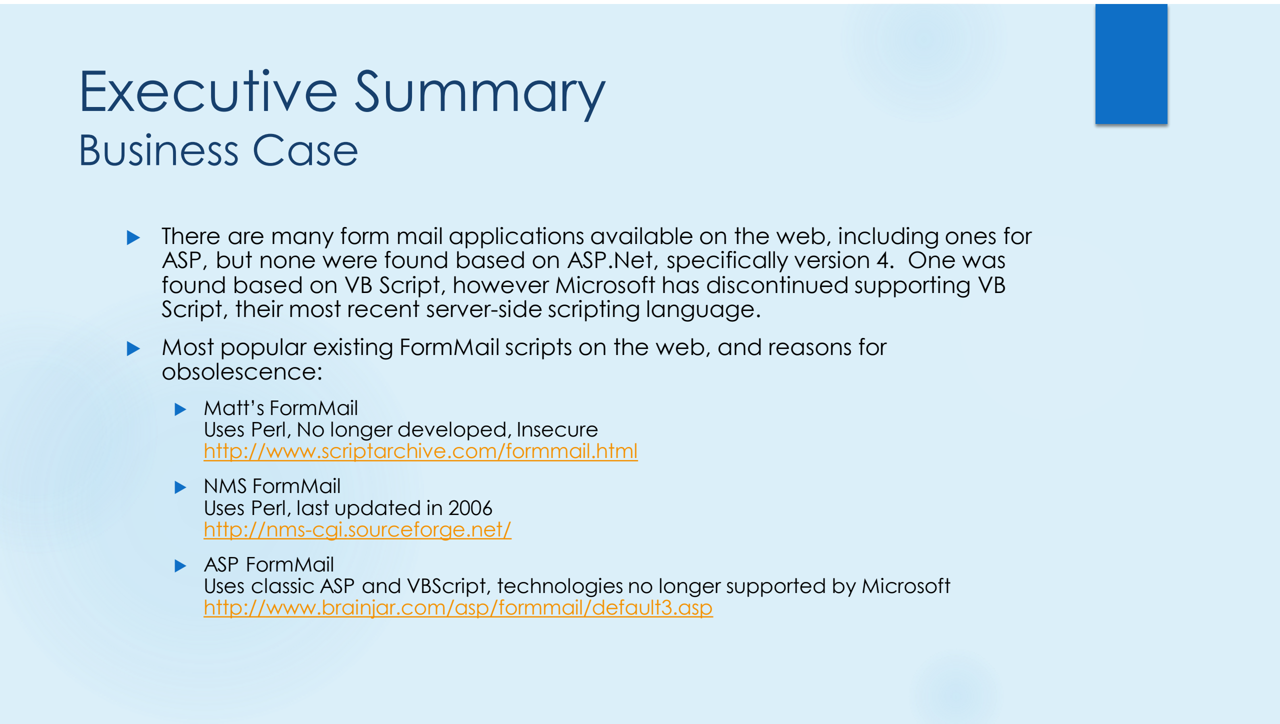
FormMail for ASP.NET

Alex Gust
CPET 499 Web Systems

October 31, 2014

Executive Summary
Background

- A form mail application can be used alongside static web sites or sites with client-side logic only to allow forms to be processed and sent via email to a specified recipient.
- Forms are typically processed using server-side logic in the form of PHP, ASP.Net, or similar technologies. Many simple web sites, however, also need the ability to collect information via web forms.
- A form mail application is a self-contained script or program that resides in an executable directory of the web server, and can accept POST data from an HTML page. Once the POST data is received, the form mail application will compile the information in a readable format and then email it to a specified recipient



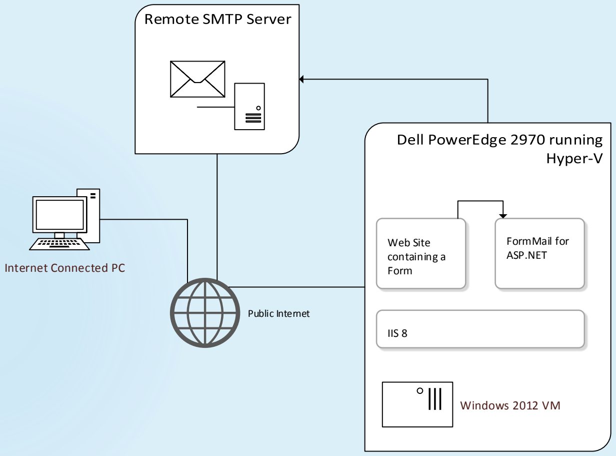
Executive Summary
Business Case

- There are many form mail applications available on the web, including ones for ASP, but none were found based on ASP.Net, specifically version 4. One was found based on VB Script, however Microsoft has discontinued supporting VB Script, their most recent server-side scripting language.
- Most popular existing FormMail scripts on the web, and reasons for obsolescence:
  - Matt's FormMail
Uses Perl, No longer developed, Insecure
http://www.scriptarchive.com/formmail.html
  - NMS FormMail
Uses Perl, last updated in 2006
http://nms-cgi.sourceforge.net/
  - ASP FormMail
Uses classic ASP and VBScript, technologies no longer supported by Microsoft
http://www.brainjar.com/asp/formmail/default3.asp

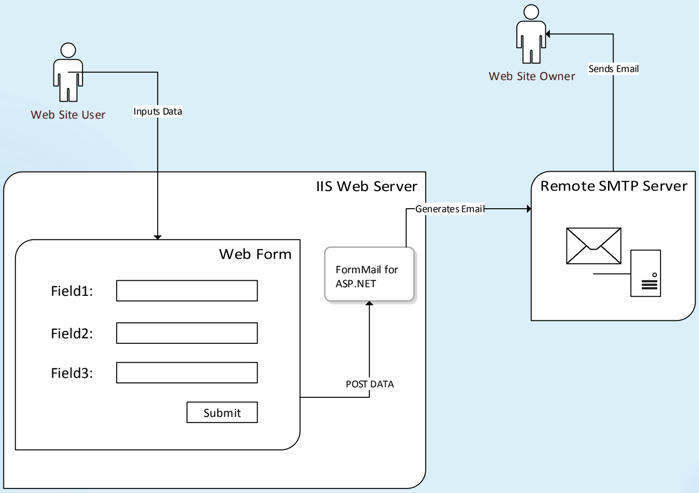
Executive Summary
Proposed Solution

- Build a portable, reusable form mail application based on C# and the windows hosting platform running .Net 4. Newer .Net libraries are to be utilized (in the System.Net.Mail namespace). The completed FormMail application shall be published to the Internet and made available for free download.

Infrastructure Diagram



Interaction Diagram



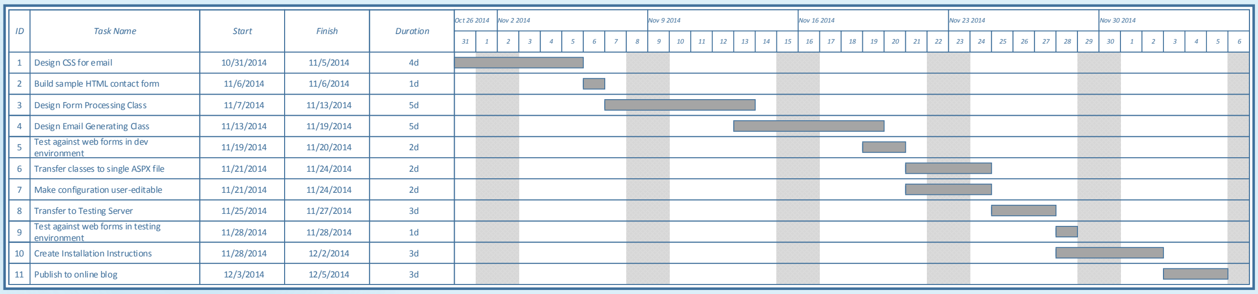
Requirements

- The FormMail Application shall be built as a .Net application, either with a compiled DLL file, or, preferably as a single uncompiled ASPX file containing all code.
- The application shall be user configurable based on either POST data, or a text configuration file.
  - User may specify form fields to include (or exclude) in the generated email.
  - User may specify form fields that are required for form to be submitted.
  - User may specify HTML pages to which the user is redirected after successful or failed form submission.
- The application shall support external SMTP servers with authentication, or via the local server (assuming it too supports SMTP with authentication
- Installation procedure should be simple – i.e. copying files to a web server and setting appropriate execute permissions.
- Security measures shall be implemented to prevent the FormMail application from being abused by spammers or anyone external to the web server on which it runs.
- The form mail application shall not have a user-facing component, but shall make its data available in the form of a further POST or GET request, or by means of redirection to a user specified page.
- The application shall be accommodating to Text Boxes, Radio Buttons, and Check Boxes on the user form. Small to medium sized forms are supported (Approximately 5 – 20 fields).
- Emailed output shall accommodate short text strings, long text strings (Greater than 254 characters, but less than 10,000 characters), and multiple entry controls (Such as Radio Buttons).
- Emailed output shall be in HTML format and viewable on both desktop and mobile email clients. Mobile design will implement a reflexive CSS design to display without horizontal scrolling.

Task List

1. Design CSS for email
   1. Build design suitable for desktop viewing
   2. Build design suitable for mobile viewing
2. Build Sample HTML Contact Form
   1. Build form with textboxes only
   2. Add radio buttons and check boxes
3. Design Form Processing Class
4. Design Email Generating Class
5. Test functionality against various web forms
   1. Test against text boxes, radio buttons and check boxes
6. Transfer Web Project Classes to single ASPX page
7. Move User Customizable Fields to User Editable Config Area
8. Transfer from Development (Visual Studio machine) to Testing Server (IIS Only)
9. Test functionality against various web forms
   1. Test against text boxes, radio buttons and check boxes
10. Create Installation Instructions
11. Publish to online blog

Schedule



ID	Task Name	Start	Finish	Duration
1	Design CSS for email	10/31/2014	11/5/2014	4d
2	Build sample HTML contact form	11/6/2014	11/6/2014	1d
3	Design Form Processing Class	11/7/2014	11/13/2014	5d
4	Design Email Generating Class	11/13/2014	11/19/2014	5d
5	Test against web forms in dev environment	11/19/2014	11/20/2014	2d
6	Transfer classes to single ASPX file	11/21/2014	11/24/2014	2d
7	Make configuration user-editable	11/21/2014	11/24/2014	2d
8	Transfer to Testing Server	11/25/2014	11/27/2014	3d
9	Test against web forms in testing environment	11/28/2014	11/28/2014	1d
10	Create Installation Instructions	11/28/2014	12/2/2014	3d
11	Publish to online blog	12/3/2014	12/5/2014	3d

Cost

- Assuming 2 hours per working day, no work on weekends
- Tasks total: 31 hours