Gateway 486 f:\pagevill\designer\sform.wri
2/4/98
2/12/98
2/20/98 - Contains DRAFT material for John
3/14/98 - Updated DRAFT material for John

Arlington Web Services, Inc.

AWS Secure Electronic Forms

Preliminary Release February 20, 1998

Overview

Secure Electonic Forms are HTML forms that have information entered using a secure browser link. The information is automatically encrypted and is retrieved using a secure browser link with the proper access code. Some of the information (like order number) can be sent un-secured to an e-mail address and/or a fax machine. This system makes it easy for people who know only HTML to create electronic forms like www.pageville.com/website/eform/ccard.htm.

The entire secure electronics form system is created using HTML and pattern files. Pattern files are bacially HTML files. You must have a Pageville Account which can be a demo account. You must have your own administration HTML file (admin.htm). The admin.htm page is used to download this and other files. Since you have this file, you should already have a Pageville Account with an admin.htm file in it. During the download process, we strongly recommended that you print out the page that has the values for your Pageville Account for the Pageville codes you will need to update. The Pageville codes are as follows:

MetroCode
CustomerCode
SiteDomain
SiteDirectory

In this documentation **tag** means an HTML tag and **command** means a Pageville processing command like <!!CustomerCode>.

Special Pageville Commands

Some Pageville commands (e.g. special tags) have special purposes for secure forms as follows:

<!!ToEmailFax>
All data after this command until the <!!EndEmailFax> command is sent to e-mail and/or fax.

<!!EndEmailFax>
Ends output to e-mail/fax.

EmailTo: address
This must be after a <!!ToEmailFax> command and before the <!!EndEmailFax>

DRAFT
Multiple "EmailTo" commands may be included.

<!!ObjectIdGet>
This gets the next Object (message) Sequence Number which can used when saving to a file or showing ID number.

<!!ObjectIdShow,m????>
This show the object name but the questions marks are replaced the the Object Sequence Number. If the sequence number is 20, the above object name will be m0020. The question marks must be togeher.

<!!ToBuffer>
All data after this command is saved to a buffer until a <!!SaveToFile . . > command is reached.

<!!SaveToFile,m????.msg,Secure>
SaveToFile is followed with filename, template and optional Secure. The Template can include a Pageville command. For example: <!!SaveToFile,<!!ObjectId>.mod,Secure>.
The Secure option means data can only be retreive using an access code with a secure browser link.

<!!RenameFile,from,to)
Renames the from name to the to name.

<!!DeleteFile,filename>

<!!Pause,seconds>   **NOT YET PROGRAMMED**

DRAFT
<!!SaveToFile,m????.log,Append>
Save contents to an existing file by appending to end. This file will not be encrypted.
This will allow appending a single line at the end of a file to log submissions. The command for the single line can be something like:
"<!!ShortDateStamp>","<!!FirstName>", . . .
If the data is a single line a carriage return and line feed will also be appended.

John NOTE: I will make sure we get quote quote when there is no data.

<!!DateStamp>
Outputs the date and time the information was submitted.

<!!DateStampInfo>
Outputs the date and time the information was submitted if ReleaseInfo was also submitted.

<!!DateStampHash>
This will output of three characters which are created from the date stamp.

DRAFT
<!!ShortDateStamp>
<!!DateStampShort>

Output will be: yymmdd

DRAFT
<!!ShortTimeStamp>
<!!DateStampTime>
Output will be: hhmmss
This can be used to provide a unique hidden FTP directory for downloading
files.

<!!MemberHandle>
The MemberHandle can be used to allow members that have been setup to access the message.

<!!ResposeAccess>
Also if ResponseAccess commands is used, its value will also be used as an additional key.

Outputs - Basic Forms

A basic secure form pattern file normally has two or three outputs.

(1) The first is an HTML file that will be saved as a secure document in Pageville. It is suggested that this file be named m????.msn. The question marks will automatically replaced by a 4 digit sequence number. The number of question marks can be more than or fewer than four. When the owner first views at file with a ".msn" extension, the extension is changed to ".msg". When the delete command is used for a file with a ".msn" or ".msg" extension, the extension is changed to ".msd" and the file is really deleted automatically seven days later.

(2) This displays on submitter's screen to notify message was sent.

(3) This is an e-mail/fax that is sent to the site owner. It just informs that a message is available. This output is optional.

Files to Create - Basic Forms

There is an HTML file (ccard1.htm) and a pattern file (ccard2.pat) included in the downloaded material for electronic forms. The two files required for the basic secured forms can be created by renaming and editting each of these using the check lists below.

Check List - Basic Forms

Check list for creating HTML files like "ccard1.htm"

1. Create a HTML POST FORM to your own specifications. The FORM METHOD tag must be the same as in the ccard1.htm file which is as follows:

```
<FORM METHOD="POST"  ACTION="https://www.pageville.com/cgi-bin/dispatch.pl"
ENCTYPE="x-www-form-encoded">
```

2. Copy all the INPUT TYPE = HIDDEN tags from the ccard1.htm file and insert them after the FORM METHOD tag. Replace the following Pageville commands with the proper value for your site:

MetroCode
CustomerCode
ReleaseGroup
SiteName
SiteDomain
SiteDirectory

3. Change the INPUT HIDDEN tag Action by replacing the filename after the pound sign (ccard2.pat) with the name of the pattern file you are creating.

Check list for creating the pattern file which is like "ccard2.pat":

1. Start by using a copy of "ccard2.pat"
2. Replace material that is before the <!!ObjectIdGet> Command.
3. Replace the material after the Print Output Area marker and before the End Print Output Area marker. The material should be the same itmes as the form posting tags in the HTML file created above. The INPUT tags need to be replaced with the value of each by tag using the appropriate Pageville <!! . . . > command.

Make other changes as desired. All material after the <!!ToBuffer> command and before the <!! Save . . . > command becomes the file that is used to print the submitted material and the responses

Outputs - Advanced Forms

An advanced secure form pattern file normally has three or four outputs.

1. The first is an HTML file that will be saved as a secure document in Pageville. It is suggested that is file be named as m????.mod. This file will be almost the same as the HTML file created above. This file allows the site owner and the submitter to edit changes to the information.
2. The second is also an HTML file that will be saved as a secure document in Pageville. It is suggested that this file be named as m????.fin. This file is formatted to allow the information to be printed.
3. This displays on submitter's screen to notify message was sent.
4. This is an e-mail/fax that is sent to the site owner. It just informs that a message is available. This output is optional.

Check List - Advanced Forms

Check list for creating HTML files like "ccard1ad.htm"

This is the same as the above "Check list for creating HTML files like ccard1.htm" except there is a "Response Access Code".

Check list for creating the pattern file which is like "ccard2ad.pat":

1. Start by using a copy of "ccard2ad.pat"
2. Replace material that is before the <!!ObjectIdGet> Command.
3. Replace the pattern file name in the input tag with name Action. The pattern filename is after the pound sign.
4. Replace the material after the Update Work Area marker and before the End Update Work Area marker. The material should be much like the form posting tags in the HTML file created above. A VALUE should be added to each text tag using the appropriate Pageville <!! . . . > command. You can make one or more of the items fixed as follows:
   a. Remove the tag for updating the value.
   b. Insert a HIDDEN tag that uses the variable name for both the tag name and the tag value. Then show the tag value. For example to keep the same value for InputCity:

      ```
      <INPUT TYPE="HIDDEN" NAME="InputCity" VALUE="<!!InputCity>">
      City: <!!InputCity>
      ```

5. Replace the material after the Print Output Area marker and before the End Print Output Area marker. The material should be the same itmes as the form posting tags in the HTML file created above. The INPUT tags need to be replaced with the value of each by tag using the appropriate Pageville <!! . . . > command.

Make other changes as desired. All material after the first <!!ToBuffer> command and before the first <!!Save . . . > command becomes the file that is used to update and/or respond to the submission.

All material after the second <!!ToBuffer> command and before the second <!!Save . . . > command becomes the file that is used to print the submitted material and the responses.

Check For Submitted Messages

A HTML file that is functionally like ckmsg.htm file is t be used to check for submitted messages. We have included the file that is used in the DEMO as an example. The example has six selections. Three for basic messages and three for advanced messages. It is unlikely all six would be used except for a DEMO. The purpose of this file is to allow the site owner to see the available messages and delete messages.

Each "FileExtend" command serves two purposes. File extensions are entered outside of parentheses and separated by commas. The material within the parentheses is display when the list of available messges is shown.

To use this file for a non-demo site requires a little editing. Edit the HIDDEN tages. Replace the following Pageville commands with the proper value for your site:

    MetroCode
    CustomerCode
    ReleaseGroup
    SiteName
    SiteDomain
    SiteDirectory

Edit the other material.

View A Submitted Message

A HTML file that is functionally like viewmsg.htm file is t be used to view, edit and/or print a specific message. We have included the file that is used in the DEMO as an example. The example has four selections. Two for basic messages and two for advanced messages. It is unlikely all four would be used except for a DEMO. The purpose of this file is to allow the person who submitted a message to see the response. It can also be used by site owner to see a specific message.

To use this file for a non-demo site requires a little editing. Edit the HIDDEN tages. Replace the following Pageville commands with the proper value for your site:

MetroCode
CustomerCode
ReleaseGroup
SiteName
SiteDomain
SiteDirectory

Edit the other material.

DRAFT

Creating Submission Log Files

A log file can be created that accumulates submitted information. Use the SaveToFile command with Append to do this. Example:

```
<!!SaveToFile,my.log,Append>
```

This will save contents to an existing file by appending to end. If the file does not exist, it will be created. This file will not be encrypted. This will allow appending a single line at the end of a file to log submissions. The command for the single line can be something like:
"<!!ShortDateStamp>","<!!FirstName>", . . .
If the data is a single line, a carriage return and line feed will also be appended.

DRAFT

Viewing Submitted Log Data

Log data can be viewed without seperating existing and future data. Just use an ~include(filename) command is a pattern file. The filename must be the log filename.

DRAFT

FTP Downloading of Log Data Data

The following commands can be used to download log data:

```
<!!RenameFile,my.log,mylog.tmp)
<!!Pause,2>
```

```
<!!FtpFiles,message/mylog.tmp,order>
```

The file my,log is renamed to mylog.tmp. New data will be written to a new my.log. The 2 second pause is to allow completion of any data currently being submitted. The FTP command will place a copy of mylog.tmp into the directory:

```
/ftp/pageville/customer/order
```

where customer is customer's site directory. For example with order equal to m????, sequence is 23 and customer nextech, the copy will be:

```
/ftp/pageville/nextech/m0023/mylog.tmp
```

You can specify multiply files as follows:

```
<!!FtpFiles,message/my*.tmp,order>
```

Also you can have multiple FtpFiles commands.