Deleted: Related Standards

TECHNICAL IMPLEMENTATION - INTERACTIVE FF/EDM

Introduction

Industry Goals/ Purpose

NAESB WGQ defined two ways in which flat files could be used to send transactions and transaction responses: interactive and batch. The batch process is described in the Internet ET. This section covers implementation considerations for the use of interactive flat files.

In general, interactive flat file communication is similar to EBB/EDM. For example, both involve human interaction and both use a Web browser and Web server to accomplish their purpose. Interactive flat files differ from EBB/EDM in how the transaction data is prepared. EBB/EDM allows for direct Web page entry of the data elements of the transaction, while flat files are prepared as part of a separate process "off-line".

Deleted: has similarity with

A variety of tools could be used to prepare flat files. However, the intent of the NAESB WGQ standards is to facilitate the preparation of Flat Files by creating standards that are consistent with how spreadsheets, or other tools, save files. Further, the standards were devised to avoid the need for programming (e.g., using spreadsheet macros) in order to create the file. The sender can choose the order of the data elements in the flat file. To maintain flexibility, the recipient must interpret the response in the order of the data elements provided by the Sender. This implies programming to interpret the received file on the part of the recipient.

Deleted: what

Deleted: had in mind was

Deleted:

Deleted: can

Deleted: flexibility for the sender to

Deleted: does

A receiver of a flat file may choose different mechanisms to respond to an interactively uploaded file. While NAESB WGQ has set no standards as to how this should be accomplished, an example is the response may be an HTML screen which highlights any errors found or it may be a file response. As another example, the response could be part of the same Web connection (HTTP round trip) or via an asynchronous mechanism (e.g. notification via email or as results on a Web page).

Deleted: y

Deleted: An

Deleted: interactive

Deleted: process

Deleted: the

Deleted: the user is either notified when the result is available or can go look for the

This portion of the guide assumes an HTTP multipart form file upload. Other implementations (e.g., custom JAVA JSP applet) are not described; however, some of the same considerations described below are applicable.

Applicable Standards

HTTP Post with mult-part forms (RFC 1867)

Secure Sockets Layer (SSL) – HTTPS (RFC 2246) OPEN ISSUE – ENSURE THAT THIS INDEED IS WHAT WE ENDORSE

Minimum Technical Characteristics of the Client Workstation - see Appendix C

Deleted: ¶
see Appendix C

Deleted: Related Standards

Flow Diagram

This paragraph and the following diagram depict a possible flat file upload process. The Sender portion of the flat file process is on the left and the Receiver portion is on right. The Web server will prompt the Sender for a logon id and password. The Web browser and Web server cooperate to ensure encryption of the upload file and the response. The Web server may perform a certain amount of pre-validation before sending the file to the Receiver's backend system for further processing. When the backend system completes the processing, the results are then formatted, possibly as a file, email or an HTML response, and are sent back to the browser or the Sender. If errors are reported in these results, the user would correct them in the source document, resave the flat file and repeat the process. This process would continue until no errors are returned.

Deleted: with the user doing the upload on the left side. A spreadsheet can be used for file creation.

Deleted: The Web server will also cause the browser to prompt for a logon id and password.

Deleted: TSP's

Deleted: its

Deleted: the Web server program gathers the results which may be kept in a database table. It then

Deleted: format

Deleted: s those results

Deleted: sends them

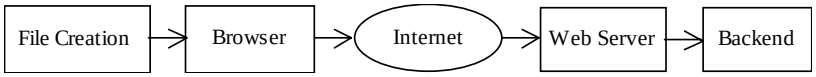
Deleted: The browser then offers the file save dialogue or displays the results as appropriate.

Deleted: spreadsheet

Deleted: input to a

Deleted: and again upload the file

Sender's process to submit a file to the Receiver



Deleted: File Creation Bro

Specification

The Parts of a Page

General

While NAESB WGQ did not either suggest the use of a Web page or determine the design of a Web page for flat file uploads, this section makes suggestions as to how a flat file could be transmitted.

Header Area`

Left side

The top left side of the Web page can provide navigation to the Customer Activities home page and/ or directly to some of its major menu items. That is, it can look exactly like the Header section for EBB/EDM.

Right side

The top right side of the Web page can provide for invocation of page functions as it does for EBB/EDM. Since uploading a flat file does not have need for most of the EBB/EDM functions, this portion of the page may be limited to such things as the "Submit" function.

Forms Area

The Forms Area will be uncomplicated for Interactive Flat File uploads. Its exact look

will depend on how interactivity is implemented and whether optional response types are made available. At minimum, it needs to have a text box to specify the file to be uploaded. This text box will be accompanied by a "Browse" button to allow a graphical selection of the file versus having to type its full path and name. This button is provided automatically by the browser. It is also necessary to include a "Submit" button near (e.g., immediately below) the text box for the file name. This button is necessary as part of a multipart form. The "Submit" function mentioned above in the right side of the Functions Area could be made to programmatically (e.g., using Javascript) "click" this "Submit" button. If alternative response types (see Intro above) are provided, such choices could be made available with a drop-down list box. It may make sense to provide this ahead of (e.g., above) the text box which provides entry of the file name. Two other possible controls include a dropdown from which to choose the TSP being nominated and a text box to indicate the DUNS number of the nominator. These would simulate the "to" and "from" fields in the batch EDM process. An example of what this may look like is provided in a subsequent section. As it is unlikely that this collection of user interface controls will require much screen real estate, it may make sense to allow a larger portion of the screen for response information if it is an HTML screen response.

Matrix Area

The matrix area could be used for an HTML response if that alternative is made available. If so, it is also desirable that it be as consistent as possible with the look and feel of the response resulting from EBB/EDM (assuming it is implemented on the site along with Interactive Flat file capability).

Page Functions

As was stated above, there might not be many functions besides the "Submit" function. The Submit function will have the effect of uploading the flat file for processing by the back end system. Depending upon the specific implementation, it may generate an acknowledgement of the receipt of the uploaded file, errors encountered in the prevalidation (if any) and/or the actual results of the backend processing (e.g., Quick Response info).

Page Format

To accomplish a file upload, the Forms Area must include a multi-part form which requires a special HTML values for the Form tag which are ENCTYPE="multipart/form-data", ACTION="scriptname" and METHOD="POST" where scriptname is the script or program which processes the upload file on the Web server. The form will also contain a tag specifying a file as a type of input such as the following: <input type="file" size="30" name="input-data">. It is this tag which causes the browser to create a text box and a button for browsing to a specific file. The NAESB WGQ-specified browser release (i.e., version 4 or better) ensures that mulitpart forms are supported.

File Creation

As was mentioned in the Industry Goals section, it is envisioned that the creation of the required flat file format be possible without programming. Specifically, what the designers had in mind was the use of a spreadsheet to accomplish this. The user would first type a "heading" row which contains the names of the data elements being uploaded (see NAESB WGQ Standard 4.3.81). Then the user would type appropriate data values in subsequent rows of the spreadsheet (note NAESB WGQ Standard 4.3.82). When all data is entered, the user would

choose a file save menu and choose a file type of "comma separated values". The user must carefully note where this file is saved so that it can be chosen in the browser Forms area as described above.

To facilitate the repeated use of this spreadsheet, it would make sense to save a spreadsheet in its native format including the heading information, thus allowing reuse of this as a template for subsequent nominations. If this is done, the user must be careful not to choose this native format file (e.g., for Excel this would be the .xls file) as the file to be uploaded, as it will not be of the proper file type (it is a binary file and not the one with the necessary text layout). Other spreadsheet features may be employed to avoid having to repeatedly enter data (e.g., the contract identifier) which does not change from row to row.

While the vision includes no programming, it does not preclude the use of macros or other "front ends" to make it easier for the user to create the proper file format. For example, a special program with a customized form for data entry could be written which facilitates easier data entry or integration with an existing system. This program would have the responsibility of taking the form data and arranging into a format compliant with the standard (see NAESB WGQ Standard 4.3.80).

Uploading Mechanism

If both EBB/EDM and Interactive FF/EDM are available, it may be useful to have submenus for each under the appropriate NAESB WGQ standard menu. Once this menu is chosen, the user can be presented a Web page as described above under the Parts of a Page and Page Format sections.

Receipt Programming

Interpreting a multipart form upload

A multipart form is sent to the Web server using a layout described in the applicable Internet Request For Comment (RFC), currently RFC 1867. This RFC describes how a multipart form allows the uploading of a variety of MIME types from a single form, one of which is a File type. As part of the upload, an HTTP header is sent indicating the string of characters which acts as a delimiter for each part of the upload form. If the form is processed by a traditional Common Gateway Interface (CGI) program (e.g., using C/C++ or Perl or others), it will have to parse the data using the RFC as a specification of data format.

Using a commercial component to assist

For some Web servers it may be possible to obtain a commercially available component which reduces the task of receiving an uploaded file to simple object method and property syntax.

Assigning data element values (parsing the uploaded file)

Once the file has been successfully received by the Web server, it may be useful to pre-validate it as much as possible. For this to be done, the individual elements of the file need to be parsed and, presumably, saved to an array or data base table. Assigning the data elements to the proper storage area is facilitated by the first row which provides standardized abbreviations (see NAESB WGQ Standard 4.3.81) for each position in the

delimited file's records (or rows).

Pre-validations

At this stage it may be possible to reject the uploaded file for various reasons, thus avoiding sending "garbage data" to the backend system. This could be the result of an unrecognized header row data element name. It may also be due to the discovery that the file is binary, indicating a probable mistake by the sending party (e.g., upload of the spreadsheet's native format or another unexpected format). In any case, the goal here is to avoid unnecessarily burdening the backend and providing the quickest possible response to the user.

Synchronous Vs Asynchronous

As was mentioned in the Industry Goals section, a variety of implementations are possible for Interactive Flat Files. One type of implementation could be characterized as "synchronous" where the user waits for the reply from the backend validations as part of the same HTTP round trip. In other words, after pressing the Submit button, the system returns a response confirming the receipt of the uploaded file followed by the completely validated response to the browser which is waiting for that response.

A different implementation may only acknowledge receipt of the uploaded file and will make the results of the backend validation available some time later. The user may or may not be notified of the availability of the full validation response. If not, they may periodically check a particular Web link for a list of available responses. NAESB WGQ was intentionally silent as regards how the EBB/EDM or Interactive FF/EDM accomplish showing validation results.

Yet other implementations may be possible.

Interface to backend system

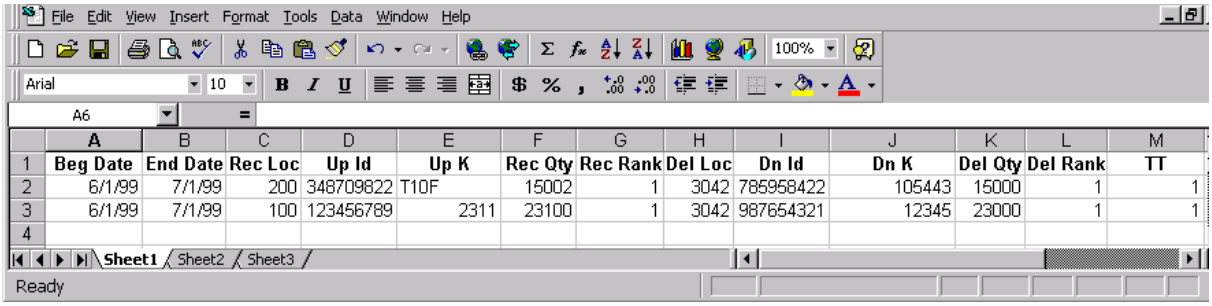
NAESB WGQ standards make no attempt to specify backend mechanisms, so this is completely up to the individual providers. Typical implementations may include two-tier (traditional client/server applications), two-tier with data base stored procedures or three-tier. Again, other implementations are possible, and this guide makes no attempt to be complete.

Formatting the response

As mentioned above, the response can be presented in an HTML screen or in a flat file. This may be based on an option provided to the sender on the upload form. If it is a flat file response, it must conform to the NAESB WGQ standards which include flexibility in the order of data elements within a record (or row). It may be more "user friendly" to have a well-defined (presumably published on the provider's Web site) sequence so as to avoid making the user incur programming time and expense otherwise necessary to handle a variable sequence.

Examples

Sample spreadsheet

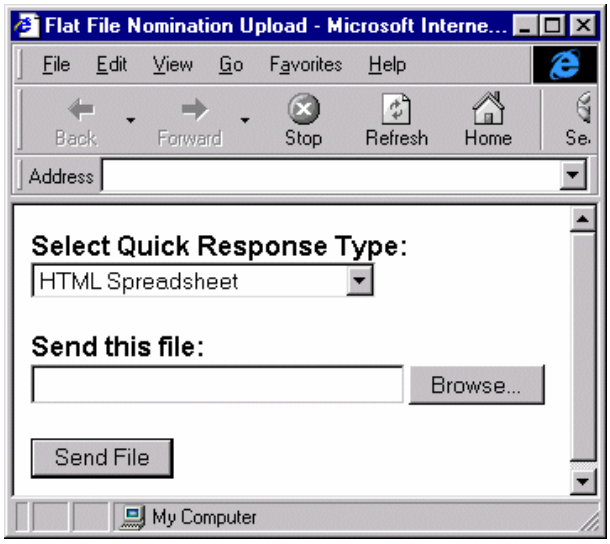


	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Beg Date	End Date	Rec Loc	Up Id	Up K	Rec Qty	Rec Rank	Del Loc	Dn Id	Dn K	Del Qty	Del Rank	TT
2	6/1/99	7/1/99	200	348709822	T10F	15002	1	3042	785958422	105443	15000	1	1
3	6/1/99	7/1/99	100	123456789	2311	23100	1	3042	987654321	12345	23000	1	1
4													

Flat file saved from the spreadsheet

Beg Date,End Date,Rec Loc,Up ID,Up K,Rec Qty,Rec Rank,Del Loc,Dn ID,Dn K,Del Qty,Del Rank,TT
19990601,19990701,28476,420824973,Q10C,1000,1,30948,293841234,W02R,970,1,01
19990601,19990701,34521,009712345,0200,25309,999,6111,087654765,P109,24500,999,01

Sample HTML upload form



HTML for Sample Form

The following is the HTML for the above (note the user of multipart form and the post method):

```
<html>
<head>
<title>Flat File Nomination Upload</title>
</head>
        <form ENCTYPE="multipart/form-data" ACTION="ProcessUpload.asp" METHOD="POST">
                <p><strong>Select Quick Response Type: </strong><br>
                <select name="QRType" size="1">
                        <option value="Spreadsheet">HTML Spreadsheet</option>
                        <option value="Echo">HTML Echo of Input with Errors</option>
                        <option value="Tab">Tab Delimited Flat File</option>
                        <option value="Comma">Comma Delimited Flat File</option>
                        <option value="Fixed">Fixed Format Flat File</option>
                </select></p>
                <p><strong>Send this file:<br></strong>
                <input type="file" size="30" name="input-data"></p>
                <p><strong><input type="submit" value="Send File"></strong></p>
        </form>
</body>
</html>
```

Security

Authentication

NAESB WGQ Standard 4.3.84 calls for use of Basic Authentication. This is a standard part of the HTTP specification. Without use of encryption, this would be a clear text transmission of user id and password. To avoid this, merely protect the page from which the logon is invoked with Secure Sockets Layer encryption as described below. Note that where the user id and password information is maintained, it is different for different Web environments. You may want to consider providing the ability for users to change their password.

Encryption

NAESB WGQ Standard 4.3.83 calls for the use of 128-bit encryption using Secure Socket Layer (SSL) technology. SSL is accomplished by obtaining a certificate from providers and using Web servers capable of using these certificates to accomplish SSL. The browsers specified in the Standard Client Configuration standard are known to be able to handle SSL mechanisms. Any pages to be protected with SSL need to be invoked with the HTTPS protocol by using “https” versus “http” as part of the hyperlink (HREF) name. Note that this means using a Fully Qualified versus Relative link name. This, in turn, causes a new DNS lookup from the browser. When the hostname is provided by more than one machine, this may result in the request being sent to a different machine. This would only cause problems where necessary state information is being maintained in the memory of the Web server’s machine.