1 Data Upload (HTTP Multiform)

This data upload provides an HTTP Multiform Post interface to upload data directly into the Meat Messaging system. The content type of this form is: `multipart/form-data.`

The user credentials, the file name and the actual data is provided by the client.
The system will respond with a resulting message.

1.1 Data Upload Input.

The **Test** URL to upload the data would be: https://meatmessaging.info/api/mpprocessfile.aspx

The **Production** URL to upload the data would be: https://meatmessaging.com/api/mpprocessfile.aspx

The Upload page will accept the following fields.

Field	Type	Description
tx_user	Text	Unique mmuid. This is the User Code for a specific Meat Messaging user. The format is 8N. The mmuid is also used to identify the mmcid (eg the specific establishment.). It is recommended that for web services a system user is registered with Meat Messaging for each establishment.
tx_pssw	Text	The password of the user matching the unique mmuid.
tx_function	Text	Control function 0,1,2,3
tx_rffaak	Text	Message reference.
tx_filename	Text	The file name. This file name should comply with the File naming conventions as described.
tx_file	Text	The local file path of the file to be uploaded. This should be a fully qualified path name. The file should contain data that comply with the spec for meat messaging

1.2 File naming

The filename must follow 2-part convention:

- MMCID. 8-digit company identification value unique to each establishment e.g. 10000016.
- RFFAAK. Company specific reference number e.g. 093123450000017555

Example file name: 10000016-093123450000017555.XML

File name length must not exceed 64 characters.

1.3 Control Function

The Function codes are defined as:

Function Code	Function Description	File Required	Notes
1	Create new message.	Y	Create a new message based on the file being uploaded.
2	Update existing message.	Y	Perform a partial update on an existing message using values provided in the uploaded file.
3	Replace existing message.	Y	Remove an existing message and replace it with values provided in the uploaded file.
6	Receipt existing message with all received correctly.	N	Receipt message referenced by parameter tx_rffaak acknowledging all of the consignment has been received correctly.
8	Cancel message.	N	Cancel an existing message that has not yet been receipted.
9	Return data.	N	This will return an XML representation of the message referenced by the parameter tx_rffaak.

1.4 Data Upload Output

The page will respond with a resulting message. The format of the response is as follows (in string format):

<response><result>RESULT</result><rffaak>RFFAAK</rffaak><errors>ERRORS</errors><data>DATA</data></response>

The above fields represented in CAPITALS are:

RESULT	SUCCESS or FAIL
RFFAAK	The unique RFFAAK reference id for the message.
ERRORS	List of errors returned. If empty, no errors occurred. If multiple errors occurred, then will be separated by CRLF.
DATA	XML representation of a message if performing a "Return data" function.

1.5 Sample and Test

A sample html page is available at the following URL:

https://meatmessaging.info/api/uploadtest.html

You can view the source for an example of implementing the API, and also use the page to perform a test upload of a payload to confirm a successful upload of your file format prior to implementing the API in your own system.

1.6 XML Message Definition

The schema used for uploading message data is available under "Docs" section of the Meat Messaging website. The article is "XML Message Definition".