EmailLabs API
Rev 4.07.00
1.0 Basic Setup & Frequently Asked Questions

1.1 What is the API?
1.2 What can the API do?
1.3 Why would a customer want to use the API?
1.4 How is the API commonly used?
1.5 What are some other uses of the API?
1.6 What does API stand for?
1.7 What syntax and protocol does the API use?
1.8 What programming languages can use the API?
1.9 What is an example of an API command?

2.0 Data Handling

2.1 Add - Add a new record
2.2 Update - Update an existing record
2.3 Upload - Upload a file
2.4 Query-Data - Retrieve demographic data
2.5 Query-Stats - Retrieve mailing-related data
2.6 Query-Listdata - Retrieve all records within a mailing list
2.7 Download - Download all records from a list

3.0 Demographic Handling

3.1 Add - Add a new demographic
3.2 Update - Update an existing demographic
3.3 Query - Retrieve information about a specific demographic
3.4 Query-Enable - Retrieve enabled demographics
3.5 Query-All - Retrieve all demographics
3.6 Enable - Enable demographics
3.7 Disable - Disable demographics

4.0 Message Handling

4.1 Add - Add a new message
4.2 Update - Update an existing message
4.3 Schedule - Schedule, pause, cancel, or resume a message
4.4 Proof - Create a proof message
4.5 Query-Data - Retrieve message data
4.6 Query-Stats - Retrieve statistical data
4.7 Query-Listdata - Retrieve all messages in list
4.8 Copy - Copy a message
4.9 Query-DeliveryDetails - Retrieve delivery data of specific message

5.0 Filter Handling

5.1 Add - Add a new filter
5.2 Update - Update an existing filter
5.3 Delete - Delete an existing filter
5.4 Generate - Generate filter results
5.5 Query-Data - Grab current filter results

6.0 List Handling

6.1 Query-Listdata - Query all existing mailing lists
6.2 Add - Create a new mailing list
6.3 Delete - Delete a mailing list
1.0 Basic Setup & Frequently Asked Questions

This API is designed to give EmailLabs customers direct programming level access to the key functions of managing and sending email to large data sets. For accounts that are API enabled, the EmailLabs systems are accessible via the protocols described below via HTTP posts on ports 80 or 443.

The general format of submitting requests to the API is as follows:
- Requests are submitted via HTTP POST or GET. We recommend HTTP POST for best results.
- Depending on the server farm you login to (see your login URL) requests sent to the following locations:
  - http://www.uptilt.com/API/mailing_list.html
  - http://www.up0.net/API/mailing_list.html
- Three additional fields must be included in each request. They are named (and are case sensitive):
  - type
  - activity
  - input (must be URL encoded)
Values for each of these fields vary according to the type of request.
- All responses are preceded by the character set encoding (<?xml version="1.0" encoding="iso-8859-1"?>)
- Values in your DATASET must be HTML special character encoded. For example:
  <DATASET>
  <SITE_ID></SITE_ID>
  <MLID></MLID>
  <DATA type="message-html">This is my first &amp; only HTML message</DATA>
  <DATASET>

1.1 What is the API?
A series of programming commands that allows an EmailLabs customer to perform many actions that they can do via their online interface.

1.2 What can the API do?
Many things that can be done by an end-user via the online EmailLabs login can also be done programmatically via the EmailLabs API. The functionality of the user console and the API are very close.

1.3 Why would a customer want to use the API?
The best use of the API is to automate user activities that are better handled by a computer system. Repetitive tasks such as uploading, downloading, or daily message sending can often be sped up through the use of the API.

1.4 How is the API commonly used?
The most common use of the EmailLabs API is as a means of passing customer data from an in-house CRM system or database into the EmailLabs application.

1.5 What are some other uses of the API?
The API is also used by EmailLabs partners to re-implement the EmailLabs user interface as well as by customers who want to use a portion of the EmailLabs functionality as a part of their application.

1.6 What does API stand for?
Application Programming Interface

1.7 What syntax and protocol does the API use?
The API uses XML syntax over HTTP (or HTTPS) protocols

1.8 What programming languages can use the API?
Any programming language can be used to access the API, just so long as it can make TCP/IP requests. JAVA, PHP, ASP and many other languages include XML libraries that make handling the syntax of the EmailLabs API simple.
1.9 What is an example of an API command?

OVERVIEW OF COMMAND

CASE 1

Activity: Query-Data
Use this activity to retrieve demographic data related to a particular record. The response data will return demographic values, unique ID, and the state of the record (if it is trashed). The only required field is the email address.

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

type: record
activity: query-data

INPUT FROM CUSTOMER
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="email">testemail@uptilt.com</DATA>
</DATASET>

RESPONSE FROM EMAILLABS
<DATASET><TYPE>success</TYPE>
  <RECORD>
    <DATA type="demographic" id="1">Joe</DATA>
    <DATA type="demographic" id="2">Boo</DATA>
    <DATA type="demographic" id="3">Mr.</DATA>
    <DATA type="extra" id="state">unsubscribed</DATA>
    <DATA type="extra" id="statedate">2002-01-23</DATA>
    <DATA type="extra" id="uid">ab468d02ef</DATA>
  </RECORD>
</DATASET>

CASE 2

The following is an example the "Query-Data" API call that will pull up the stats on a hypothetical member record of "esales@yourdomain.com"

EXAMPLE A (Correct):
   &input=%3CDATASET%3E%0D%0A%3CSITE_ID%3E1111%3C%2FSITE_ID%3E%0D%0A%3CMLID%3E0000%3C%2FMLID%3E%0D%0A%3CDATA%0D%0A%3Demail%22%3Eesales%40yourdomain.com%3C%2FDATA%3E%0D%0A%3C%2FDATASET%3E

instead of:
EXAMPLE B (Incorrect):
   https://www.uptilt.com/API/mailing_list.html?siteid=1111&mlid=00000&UPTA
   DMN_ACTIVITY=query-data&type=record&email=esales@yourdomain.com

There are three parameters you MUST pass through at all times:
1) Record
2) Activity
3) Input

Only in the Input parameter do you specify the mailing list ID and Site ID. These were missing in Example B (above).

The Input parameter for the activity you're doing must include the following:
This is encoded the INPUT parameter as seen in Example A.

**PHP Example (not fully functional):**

```php
<?php
include('xml_class.inc');

## contains site_id for account  mlid for mailing list
$siteid=xxxxx;
$mlid=xxxxx;

function call_API($type,$activity,$input)
{
    $form_url="http://www.uptilt.com/API/mailing_list.html";
    $form_method="POST";
    $form_data['type']=$type;
    $form_data['activity']=$activity;
    $form_data['input']=$input;

    // Build the request data string
    $request_data = "";
    while(list($name,$value)=each($form_data))
    {
        if (strlen($request_data) > 0)
            $request_data .= '&';
        $request_data .= $name.'='.urlencode($value);
    }

    // Build the request
    $ur = parse_url($form_url);
    $form_method = strtoupper($form_method);
    $request  = $form_method." " . $form_method;
    switch ($form_method)
    {
        case 'GET':
            $request = $url['path']."?".$request_data." HTTP/1.0\r\n".
                        "Host: ".$url['host'].'\r\n";
            break;
        case 'POST':
            $request = $url['path']." HTTP/1.0\r\n".
                        "Host: ".$url['host'].'\r\n".
                        "Content-type: application/x-www-form-urlencoded\r\n".
                        "Content-length: ".strlen($request_data)."\n\r\n".
                        $request_data;
            break;
        default:
            die("Invalid method: ".$form_method);
    }

    // Open the connection
    $fp = fsockopen($url['host'], 80, $err_num, $err_msg, 30);
    $response="";
    if ($fp)
    {
        // Submit form
        fputs($fp, $request);

        // Get the response
        while (!feof($fp))
        {
```
{  
    $tmp=fgets($fp, 1024);
    $response.= $tmp;
}
fclose($fp);
?>
2.0 Data Handling

Adding/Modifying/Querying a record

This provides all activities associated with an individual record.

API Type value: record
Required Fields: <SITE_ID> Site ID
<MLID> Mailing List ID

Valid Activities:
Add       Add a new record
Update    Update an existing record
Upload    Upload a file
Query-Data Retrieve demographic data
Query-Stats Retrieve mailing-related data
Query-Listdata Retrieve all records within a mailing list
Download  Download all records from a mailing list
Encoding  Defaults to iso-8859-1, allows you to choose what you want

2.1 Activity: Add

Use this activity to add a new record to the mailing list. The only required field is the email address. Each field has an attribute named "type" and "id". If the type is “demographic”, then the value of “id” must be the Demographic ID number of that demographic (you can determine the demographic ID on the Member Records page within the respective mailing list in EmailLabs system). If the type is “extra”, then “id” must be “proof” or “trigger”. The return code will include the 10-digit unique ID of the new record.

Functional Description

Adding Email Address [required]
To add an email address, use the email type and specify the email address.

Adding with Demographics [optional]
You can also insert demographic values along with the email address. Use the demographic input. You can determine the demographic ID on the Member Records page within the respective mailing list in EmailLabs system. To add demographics of type “Multi-Select List” or “Multi-Checkbox”, please be sure to use double pipe characters “||” in between each element of that field.

Flagging as a Proof List Member [optional]
When creating a message, before you send it to everyone on your list, you can send a test copy of your message to everyone who is flagged a Proof List Member. This allows you to send proofs for review before doing a regular sending out of your campaign. Use the extra input below and set its ID to “proof” with a value of “yes”.

Apply Triggers to the Email Added [optional]
You can have the system check the email address being added against any enabled triggers that are active within the mailing list. Use the extra input with an ID of “trigger” and a value of “yes”. For example, if you had a trigger message setup to send out a newsletter when someone joins for the first time, using this input will allow the individual to receive that trigger message as soon as you add them in via the API.

Assigning a User a Specific State
You can add a user to a specific state in the mailing list. Use the extra input below with the ID of “state” and the value of active, bounced, unsubscribed, or trashed.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding Emails</td>
<td>email</td>
<td>-</td>
<td>(email address)</td>
</tr>
<tr>
<td>Adding Demographic Values</td>
<td>demographic</td>
<td>(demographic id*)</td>
<td>(value of field)</td>
</tr>
<tr>
<td>Flaggng as Proof Member</td>
<td>extra</td>
<td>proof</td>
<td>yes</td>
</tr>
<tr>
<td>Check against Triggers</td>
<td>extra</td>
<td>trigger</td>
<td>yes</td>
</tr>
<tr>
<td>Select stated of the user</td>
<td>extra</td>
<td>state</td>
<td>active, bounced, unsubscribed, trashed</td>
</tr>
<tr>
<td>Encoding</td>
<td>extra</td>
<td>encoding</td>
<td>(encoding of your choice - blank will default to iso-8859-1)</td>
</tr>
</tbody>
</table>

Note: parameters within parenthesis “( )” indicate respective values and not the name of the field.
Format

<DATA type="choose a type below" id="enter respective id value here">value</DATA>

EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):

- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

type: record
activity: add
Input:

<Data>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="email">testemail@uptilt.com</DATA>
  <DATA type="demographic" id="1">Joe</DATA>
  <DATA type="demographic" id="2">Boo</DATA>
  <DATA type="demographic" id="3">Mr.</DATA>
  <DATA type="demographic" id="4">Newsletters||Promotions||Alerts</DATA>
  <DATA type="extra" id="proof">yes</DATA>
  <DATA type="extra" id="trigger">yes</DATA>
</DATA>

The above example created a new record with the email address: testemail@uptilt.com. This record will also be placed in the “Proof List” and checked against any existing triggers.

Response:

<Data> <TYPE>success</TYPE> <DATA>a196ee799a</DATA> </DATA>

2.2 Activity: Update

Use this activity to update an existing record in the mailing list. The only required field is the email address. The fields used for this activity are the same as the fields used in “Add”.

Functional Description

Updating Email Addresses [optional]
To update an existing record’s email address, use extra with the ID of “new_email” and then specify the new address.

Adding with Demographics [optional]
You can also insert demographic values along with the email address. Use the demographic input. You can determine the demographic ID on the Member Records page within the respective mailing list in EmailLabs system. To add demographics of type “Multi-Select List” or “Multi-Checkbox”, please be sure to use double pipe characters “||” in between each element of that field.

Flagging as a Proof List Member [optional]
When creating a message, before you send it to everyone on your list, you can send a test copy of your message to everyone who is flagged a Proof List Member. This allows you to send proofs for review before doing a regular sending out of your campaign. Use the extra input below and set its ID to “proof” with a value of “yes”.

Apply Triggers to the Email Added [optional]
You can have the system check the email address being added against any enabled triggers that are active within the mailing list. Use the extra input with an ID of “trigger” and a value of “yes”. For example, if you had a trigger message setup to send out a newsletter when someone joins for the first time, using this input will allow the individual to receive that trigger message as soon as you add them in via the API.

Assigning a User a Specific State [optional]
You can add a user to a specific state in the mailing list. Use the extra input below with the ID of “state” and the value of active, bounced, unsubscribed, or trashed.
Look Up User by Unique ID [optional]
You have the ability to look up a user based on the EmailLabs unique id number assigned to that user in the system. Use the extra input below with the ID of "id" and the value set as the unique ID of the user from the EmailLabs system.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updating Emails</td>
<td>extra</td>
<td>new_email</td>
<td>(email address)</td>
</tr>
<tr>
<td>Adding Demographic Values</td>
<td>demographic</td>
<td>id*</td>
<td>(value of field)</td>
</tr>
<tr>
<td>Flagging as Proof Member</td>
<td>extra</td>
<td>proof</td>
<td>yes</td>
</tr>
<tr>
<td>Check against Triggers</td>
<td>extra</td>
<td>trigger</td>
<td>yes</td>
</tr>
<tr>
<td>Remove from MultiSelect</td>
<td>extra</td>
<td>ms_option</td>
<td>delete</td>
</tr>
<tr>
<td>Add to MultiSelect</td>
<td>extra</td>
<td>ms_option</td>
<td>add</td>
</tr>
<tr>
<td>Assign user state</td>
<td>extra</td>
<td>state</td>
<td>active, bounced, unsubscribed, trashed</td>
</tr>
<tr>
<td>Look up by UID</td>
<td>extra</td>
<td>ID</td>
<td>(emaillabs unique id)</td>
</tr>
<tr>
<td>Encoding</td>
<td>extra</td>
<td>encoding</td>
<td>(encoding of your choice - blank will default to iso-8859-1)</td>
</tr>
</tbody>
</table>

Note: parameters within parenthesis "( )" indicate respective values and not the name of the field.

Format

<DATA type="choose a type below" id="enter respective id value here">value</DATA>

EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

type: record
activity: update
Input:

<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="email">testemail@uptilt.com</DATA>
  <DATA type="demographic" id="1">Joe</DATA>
  <DATA type="demographic" id="2">Boo</DATA>
  <DATA type="demographic" id="3">Mr.</DATA>
  <DATA type="demographic" id="4">Newsletters||Promotions||Alerts</DATA>
  <DATA type="extra" id="proof">yes</DATA>
  <DATA type="extra" id="state">unsubscribed</DATA>
  <DATA type="extra" id="new_email">updatedaddress@uptilt.com</DATA>
</DATASET>

Response:

<DATASET><TYPE>success</TYPE><DATA>a196ee799a</DATA></DATASET>
2.3 Activity: Upload

Use this activity to upload a file to the mailing list. The location of file should be accessible via the web. There are different options you may pass in addition to the filename:

**Functional Description**

**File**
Specifies the URL or FTP location of the file to be uploaded.

**Validate**
Determines whether or not each email address is validated for correct format (something@something.something).

**Type**
Emails uploaded will be one of these types: proof, unsubscribed, bounced, trashed

**Update**
Update addresses that already exist. If “only” is selected, don’t insert new entries, only update existing.

**Email**
Specify an email address to receive notification when upload is complete. If none is specified, system will return a code when complete. You can only specify one email address in the call.

**Delimiter**
Field delimiter for your file

**Apply Triggers to the Email Added [optional]**
You can have the system check the email address being added against any enabled triggers that are active within the mailing list. Use the extra input with an ID of “trigger” and a value of “yes”. For example, if you had a trigger message setup to send out a newsletter when someone joins for the first time, using this input will allow the individual to receive that trigger message as soon as you add them in via the API.

**Append not replace [optional]**
When using multiple select lists or multiple check boxes to append the new value to the records demographic rather than replace the value. Use the extra input below and set its ID to “append” with a value of “on”.

**Delete blanks**
Used to blank out the value of a demographic. To use this, the file must contain a blank column that has that demographic as a header. Use the extra input below and set its id to “delete_blank” with a value of “on”.

**Untrash**
If an entry is in the trashed state (unsubscribed, bounced, admin trashed), this flag will set them back to the active state. Otherwise the record will remain trashed. Use the extra input below and set its ID to “untrash” with a value of “on”.

**Clear Trigger History**
Allows you to clear the trigger history of all or a select number of triggers in a particular mailing list. This can be use to clear out any scheduled triggers for all members in the upload. Use the extra input below and set its id to “clear_trigger_history” with the value either “all” for all triggers in that list or a comma delimited list of trigger id’s.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>extra</td>
<td>file</td>
<td>(<a href="http://web.com/file.txt">http://web.com/file.txt</a>) or (ftp://username:<a href="mailto:password@ftp.com">password@ftp.com</a>/file.txt)</td>
</tr>
<tr>
<td>Validate</td>
<td>extra</td>
<td>validate</td>
<td>on</td>
</tr>
<tr>
<td>Type</td>
<td>extra</td>
<td>type</td>
<td>proof, unsubscribed, bounced, trashed</td>
</tr>
<tr>
<td>Update</td>
<td>extra</td>
<td>update</td>
<td>on/off/only</td>
</tr>
<tr>
<td>Email</td>
<td>email</td>
<td>--</td>
<td>(email address to receive notice)</td>
</tr>
<tr>
<td>Delimiter</td>
<td>extra</td>
<td>delimiter</td>
<td>,</td>
</tr>
<tr>
<td>Triggers</td>
<td>extra</td>
<td>trigger</td>
<td>yes</td>
</tr>
</tbody>
</table>
Append extra append on
Delete Blanks extra delete_blanks on
Untrash extra untrash on
Clear trigger history extra clear_trigger_history all or comma delimited list of trigger id's
Encoding extra encoding (encoding of your choice - blank will default to iso-8859-1)

EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

type: record
activity: upload

Input:
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="email">myemail@address.com</DATA>
  <DATA type="extra" id="file">http://www.example.com/myfile.csv</DATA>
  <DATA type="extra" id="validate">on</DATA>
  <DATA type="extra" id="type">active</DATA>
  <DATA type="extra" id="update">on</DATA>
</DATASET>

Response:
<DATASET><TYPE>success</TYPE><DATA>a196ee799a</DATA></DATASET>

2.4 Activity: Query-Data
Use this activity to retrieve demographic data related to a particular record. The response data will return demographic values, unique ID, and the state of the record (if it is trashed). The only required field is the email address.

EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

type: record
activity: query-data

Input:
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="email">testemail@uptilt.com</DATA>
</DATASET>
2.5 Activity: Query-Stats

Use this activity to retrieve statistical data (sent, clickthru and open history) related to a particular record. The response data will return the statistic type, the subject and date of the message, timestamp and frequency of the statistic. The only required field is the email address.

EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

<table>
<thead>
<tr>
<th>type:</th>
<th>record</th>
</tr>
</thead>
<tbody>
<tr>
<td>activity:</td>
<td>query-stats</td>
</tr>
</tbody>
</table>

Input:

```xml
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="email">testemail@uptilt.com</DATA>
</DATASET>
```

Response:

```xml
<DATASET><TYPE>success</TYPE>
  <RECORD>
    <DATA type="type" id="open">142</DATA>
    <DATA type="subject">Your Weekly Newsletter</DATA>
    <DATA type="delivery">3600</DATA>
    <DATA type="timestamp">1024253</DATA>
    <DATA type="date">2002-01-14</DATA>
    <DATA type="frequency">5</DATA>
  </RECORD>
  <RECORD>
    <DATA type="type" id="click">142</DATA>
    <DATA type="subject">Your Weekly Newsletter</DATA>
    <DATA type="delivery">3600</DATA>
    <DATA type="timestamp">1025242</DATA>
    <DATA type="date">2002-01-14</DATA>
    <DATA type="frequency">2</DATA>
    <DATA type="location">http://www.uptilt.com</DATA>
  </RECORD>
  <RECORD>
    <DATA type="type" id="sent">142</DATA>
    <DATA type="subject">Your Weekly Newsletter</DATA>
    <DATA type="timestamp">1025242</DATA>
  </RECORD>
</DATASET>
```
The above response lists 5 opens to message Your Weekly Newsletter, delivered at 1:00 am. The message has 2 clicks to www.uptilt.com. The message was sent to only one person. The “delivery” value is the time the message was sent that day (in seconds) since midnight. For example, 3600 corresponds to 1 am, and 39600 corresponds to 11 am (3600x11 hours). The “date” value is the day that the message was sent. The “timestamp” value is the UNIX timestamp at which the statistic was recorded, and “frequency” is the number of times the statistic was recorded.

### 2.6 Activity: Query-Listdata

Use this activity to retrieve all records from this list. The response data will return demographic values, unique ID, and the state of each record (if it is trashed). If “page” is supplied, page X will be returned, where each page has “pagelimit” records per page. If “pagelimit” is not supplied, the “pagelimit” defaults to 25. If “pagelimit” is set to 0, then all the record will be returned.

**Available Extra Parameters:**
- **pagelimit**: # records displayed per page (setting this to 0 = unlimited)
- **page**: page # based on pagelimit value
- **type**: Valid values are: active, trashed, unsubscribed, bounced, trashedbyadmin
  - If “type” is supplied, only those whose type occurred during this supplied date are returned.
- **Date**: Provide date format in textual format (December 5, 2004)
- **Encoding**: Defaults to iso-8859-1, allows you to choose what you

**Functional Description**

- **Show by Date** [optional]
- **# of Records to Show per Page** [optional]
  - This will determine how many records to show per page. It will also determine how many pages you will have in total (the more records per page displayed, the fewer pages you’ll have).
- **Go to Page Number** [optional]
  - To display a specific page number (e.g. 45 of 200).
- **Show by Type** [optional]
  - Selectively display records in your mailing list of a certain record status type: active, trashed, unsubscribed, bounced, trashedbyadmin. “Active” means they will receive messages as normal. “Trashed” means they will not receive future messages. “Trashed” is further categorized into three subgroups (to clarify reasons why they were trashed): “bounced”, “unsubscribed”, or “trashedbyadmin”.

**Function** | **Type** | **ID** | **Value**
--- | --- | --- | ---
Show by Date | extra | date | (Month DD, YYYY)
Go to Page Number | extra | page | (page number)
# of Records to Show per Page | extra | pagelimit | (number or de)
Show by Type | extra | type | active,trashed,unsubscribed, bounced, trashedbyadmin
Encoding | extra | encoding | (encoding of your choice - blank will default to iso-8859-1)

**Format**

```
<Data type="choose a type below" id="enter respective id value here">value</Data>
```
EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

type: record
activity: query-listdata

Input:
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="extra" id="page">2</DATA>
  <DATA type="extra" id="pagelimit">30</DATA>
  <DATA type="extra" id="type">active</DATA>
</DATASET>

Response:
<DATASET><TYPE>success</TYPE>
  <RECORD>
    <DATA type="email">jboo@emaillabs.com</DATA>
    <DATA type="demographic" id="1">Joe</DATA>
    <DATA type="demographic" id="2">Boo</DATA>
    <DATA type="demographic" id="3">Mr.</DATA>
    <DATA type="extra" id="state">active</DATA>
    <DATA type="extra" id="statedate">2002-01-23</DATA>
    <DATA type="extra" id="uid">ab468d02ef</DATA>
  </RECORD>
  <RECORD>
    <DATA type="email">jane@emaillabs.com</DATA>
    <DATA type="demographic" id="1">Jane</DATA>
    <DATA type="demographic" id="2">Doe</DATA>
    <DATA type="demographic" id="3">Ms.</DATA>
    <DATA type="extra" id="uid">ab468d0888</DATA>
  </RECORD>
  <RECORD>
    <DATA type="email">john@emaillabs.com</DATA>
    <DATA type="demographic" id="1">John</DATA>
    <DATA type="demographic" id="2">Doe</DATA>
    <DATA type="demographic" id="3">Mr.</DATA>
    <DATA type="extra" id="state">active</DATA>
    <DATA type="extra" id="statedate">2002-01-31</DATA>
    <DATA type="extra" id="uid">cc468d02ef</DATA>
  </RECORD>
</DATASET>

2.7 Activity: Download

Use this activity to download all records from this list. The response data will return a successful message and notification that the download has started. Once finished the download will email a link to the 'email_notify' address.

Available Extra Parameters:

- **Email_notify**: email address that will be notified with the download is finished. Email will contain a link to the file.
- **delimiter**: delimiter user in the file. Can be anything.
- **type**: Valid values are: active, trashed, unsubscribed, bounced, trashedbyadmin
- **join date**: Valid value is an 8 digit unix time stamp (IE: 20041201 = December 1, 2004)
- **trash date**: Valid value is an 8 digit unix time stamp (IE: 20041201 = December 1, 2004)
- **Encoding**: Default is iso-8859-1, you can choose your own however
Functional Description

Email Notify [optional]
The email address the download notification will be sent to. The address that will receive the link to the file downloaded. Only one email address can be specified.

Type of records [optional]
Allows you to download a specific state for the user. Valid values are active, trashed, bounced, unsubscribed

Delimiter [optional]
Allows you to specify the delimiter to use in the file. Most common is a comma (,) or a pipe (|)

Join Date [optional]
Will download only the users with that specific joindate. In UNIX timestamp format.

Trash Date [optional]
Will download only the users with that specific trashdate. In UNIX timestamp format.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email notify</td>
<td>extra</td>
<td>EMAIL_NOTIFY</td>
<td>(your email address)</td>
</tr>
<tr>
<td>Type of records</td>
<td>extra</td>
<td>type</td>
<td>active, trashed, bounced, unsubscribed</td>
</tr>
<tr>
<td>Delimiter in the file</td>
<td>extra</td>
<td>delimiter</td>
<td>(specify delimiter)</td>
</tr>
<tr>
<td>JoinDate</td>
<td>extra</td>
<td>JOIN_DATE</td>
<td>YYYYMMDD</td>
</tr>
<tr>
<td>TrashDate</td>
<td>extra</td>
<td>TRASH_DATE</td>
<td>YYYYMMDD</td>
</tr>
<tr>
<td>Encoding</td>
<td>extra</td>
<td>encoding</td>
<td>(encoding of your choice - blank will default to iso-8859-1)</td>
</tr>
</tbody>
</table>

Format

<DATA type="choose a type below" id="enter respective id value here">value</DATA>

EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

| type: record  |
| activity: download |

Input:

<DATASET>
    <SITE_ID>123</SITE_ID>
    <MLID>345</MLID>
    <DATA type="extra" id="email_notify">test@uptilt.com</DATA>
    <DATA type="extra" id="type">active</DATA>
</DATASET>
Response:

```xml
<DATASET><TYPE>success</TYPE>
</DATASET>
```
3.0 Demographic Handling

This provides all activities associated with demographics.

<table>
<thead>
<tr>
<th>API Type value:</th>
<th>demographic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required Fields:</td>
<td>&lt;SITE ID&gt; Site ID</td>
</tr>
<tr>
<td></td>
<td>&lt;MLID&gt; Mailing List ID</td>
</tr>
</tbody>
</table>

Valid Activities:
- Add: Add a new demographic
- Update: Update an existing demographic
- Query: Queries the values of the specified demographic
- Query-Enabled: Retrieve enabled demographics
- Query-All: Retrieve all demographics
- Enable: Enable demographics
- Disable: Disable demographics

3.1 Activity: Add

Use this activity to add a new demographic. The fields available are as follows (required fields in orange):

**Functional Description**

Add [text/area/select list/radio button/multiple select list/multiple checkbox]

These functions allow you to create a specific demographic of the respective type.

Select list display length

This parameter allows you to set the number of options to display in a multiple select list box.

Enable demographic

This flags the demographic as “enabled”; allowing you to use it within the system.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add text box</td>
<td>text</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Add text area</td>
<td>text area</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Add select list</td>
<td>select list</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Add radio button</td>
<td>radio button</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Add multi-select list</td>
<td>multiple select list</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Add multi-checkbox</td>
<td>multiple checkbox</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Add option for select</td>
<td>option</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Select list display length</td>
<td>size</td>
<td>n/a</td>
<td>(number of lines to show)</td>
</tr>
<tr>
<td>Enable demographic</td>
<td>state</td>
<td>n/a</td>
<td>enabled</td>
</tr>
</tbody>
</table>

**Format**

```xml
<Data type="choose a type below" id="enter respective id value here">value</Data>
```

**EXAMPLE**

Post data to: Depending on the server farm you login to (see your login URL):
- [http://www.uptilt.com/API/mailing_list.html](http://www.uptilt.com/API/mailing_list.html)
- [http://www.up0.net/API/mailing_list.html](http://www.up0.net/API/mailing_list.html)
- [http://www.elabs3.com/API/mailing_list.html](http://www.elabs3.com/API/mailing_list.html)
- [http://www.elabs7.com/API/mailing_list.html](http://www.elabs7.com/API/mailing_list.html)

**Input:**

```xml
<Data type="demographic" activity="add">
```

www.emallabs.com  •  866-632-4522 Sales  •  877-787-8458 SUPPORT
4400 Bohannon Dr, Suite 200  •  Menlo Park CA 94025
The above example created a new demographic called “Where do you live?” which is a dropdown selection list made up of 7 choices, and enabled. The response generated includes the unique id for this demographic.

### 3.2 Activity: Update

Use this activity to update an existing demographic. The required fields are site id, mailing list id, and demographic id. **The existing demographic will be deleted first, and then updated with the data contained in this update request.**

#### Functional Description

**id** *(required)*

Specifies the demographic to be changed by its ID number.

**state**

Toggles demographics as “enabled” or “disabled”.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>--</td>
<td>--</td>
<td>(number of id)</td>
</tr>
<tr>
<td>state</td>
<td>--</td>
<td>--</td>
<td>disabled, enabled</td>
</tr>
<tr>
<td>Change to text box</td>
<td>text</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Change to text area</td>
<td>textarea</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Change to select list</td>
<td>select list</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Change to radio button</td>
<td>radio button</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Change to multi-select list</td>
<td>multiple select list</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Change to multi-checkbox</td>
<td>multiple checkbox</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
<tr>
<td>Change option for select</td>
<td>option</td>
<td>n/a</td>
<td>(name of demographic)</td>
</tr>
</tbody>
</table>

#### Format

```html
<Data type="choose a type below" id="enter respective id value here">value</Data>
```

#### EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

**type:** demographic  
**activity:** update

Input:
```
<Data>
<SITE_ID>123</SITE_ID>
<MLID>345</MLID>
<Data type="id">1425</Data>
</DATASET>
```
3.3 Activity: Query
Use this activity to retrieve data about a specific demographic. The required fields are site id, mailing list id, and demographic id.

**EXAMPLE**

Post data to: Depending on the server farm you login to (see your login URL):
- [http://www.uptilt.com/API/mailing_list.html](http://www.uptilt.com/API/mailing_list.html)
- [http://www.up0.net/API/mailing_list.html](http://www.up0.net/API/mailing_list.html)
- [http://www.elabs3.com/API/mailing_list.html](http://www.elabs3.com/API/mailing_list.html)
- [http://www.elabs7.com/API/mailing_list.html](http://www.elabs7.com/API/mailing_list.html)

**Input:**
```xml
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="id">1425</DATA>
</DATASET>
```

**Response:**
```xml
<DATASET><TYPE>success</TYPE>
  <RECORD>
    <DATA type="select list">Where do you live?</DATA>
    <DATA type="option">North America</DATA>
    <DATA type="option">South America</DATA>
    <DATA type="option">Europe</DATA>
    <DATA type="option">Africa</DATA>
    <DATA type="option">Asia</DATA>
    <DATA type="option">Australia</DATA>
    <DATA type="option">Antarctica</DATA>
    <DATA type="state">enabled</DATA>
  </RECORD>
</DATASET>
```

3.4 Activity: Query-Enabled
Use this activity to retrieve all enabled demographics.

**EXAMPLE**

Post data to: Depending on the server farm you login to (see your login URL):
- [http://www.uptilt.com/API/mailing_list.html](http://www.uptilt.com/API/mailing_list.html)
- [http://www.up0.net/API/mailing_list.html](http://www.up0.net/API/mailing_list.html)
- [http://www.elabs3.com/API/mailing_list.html](http://www.elabs3.com/API/mailing_list.html)
- [http://www.elabs7.com/API/mailing_list.html](http://www.elabs7.com/API/mailing_list.html)

**Input:**
```xml
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
</DATASET>
```
Response:

```xml
<DATASET><TYPE>success</TYPE>
  <RECORD>
    <DATA type="name">Where do you live?</DATA>
    <DATA type="id">1425</DATA>
    <DATA type="type">select list</DATA>
  </RECORD>
  <RECORD>
    <DATA type="name">First Name</DATA>
    <DATA type="id">1426</DATA>
    <DATA type="type">text</DATA>
  </RECORD>
  <RECORD>
    <DATA type="name">Last Name</DATA>
    <DATA type="id">1427</DATA>
    <DATA type="type">text</DATA>
  </RECORD>
</DATASET>
```

### 3.5 Activity: Query-All

Use this activity to retrieve all demographics.

**EXAMPLE**

Post data to: Depending on the server farm you login to (see your login URL):

- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

<table>
<thead>
<tr>
<th>type</th>
<th>demographic</th>
</tr>
</thead>
<tbody>
<tr>
<td>activity</td>
<td>query-all</td>
</tr>
</tbody>
</table>

**Input:**

```xml
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
</DATASET>
```

**Response:**

```xml
<DATASET><TYPE>success</TYPE>
  <RECORD>
    <DATA type="name">Where do you live?</DATA>
    <DATA type="id">1425</DATA>
    <DATA type="type">select list</DATA>
    <DATA type="state">enabled</DATA>
  </RECORD>
  <RECORD>
    <DATA type="name">First Name</DATA>
    <DATA type="id">1426</DATA>
    <DATA type="type">text</DATA>
    <DATA type="state">enabled</DATA>
  </RECORD>
  <RECORD>
    <DATA type="name">Last Name</DATA>
    <DATA type="id">1427</DATA>
    <DATA type="type">text</DATA>
    <DATA type="state">enabled</DATA>
  </RECORD>
  <RECORD>
    <DATA type="name">Middle Name</DATA>
    <DATA type="id">1428</DATA>
    <DATA type="type">text</DATA>
  </RECORD>
</DATASET>
```
3.6 **Activity: Enable**

Use this activity to enable one or more demographics in the list. The response data will contain comma separated list of all currently enabled demographics.

**Functional Description**

This works similar to other demographic update calls, but allows you to enable multiple demographics at once.

**EXAMPLE**

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

<table>
<thead>
<tr>
<th>type</th>
<th>demographic</th>
<th>activity</th>
<th>enable</th>
</tr>
</thead>
</table>

**Input:**
```
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="id">2</DATA>
  <DATA type="id">3</DATA>
</DATASET>
```

**Response:**
```
<DATASET>
  <TYPE>success</TYPE>
  <DATA>2,3,1425,3300</DATA>
</DATASET>
```

3.7 **Activity: Disable**

Use this activity to disable one or more demographics in the list. The response data will contain comma separated list of all currently enabled demographics.

**Functional Description**

This works similar to other demographic update calls, but allows you to disable multiple demographics at once.

**EXAMPLE**

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

<table>
<thead>
<tr>
<th>type</th>
<th>demographic</th>
<th>activity</th>
<th>disable</th>
</tr>
</thead>
</table>

**Input:**
```
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="id">2</DATA>
  <DATA type="id">3</DATA>
</DATASET>
```

**Response:**
```
<DATASET>
  <TYPE>success</TYPE>
  <DATA>2,3,1425,3300</DATA>
</DATASET>
```
Input:

```xml
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="id">2</DATA>
  <DATA type="id">3</DATA>
</DATASET>
```

Response:

```xml
<DATASET>
  <TYPE>success</TYPE>
  <DATA>1425,3300</DATA>
</DATASET>
```
4.0 Message Handling

This provides all activities associated with an individual message.

**API Type value:** message

**Required Fields:**
- `<SITE_ID>` Site ID
- `<MLID>` Mailing List ID

**Valid Activities:**
- **Add** Add a new message
- **Update** Update an existing message
- **Schedule** Schedule, pause, cancel, or resume a message
- **Query-Data** Retrieve message data
- **Query-Stats** Retrieve statistical data
- **Query-Listdata** Retrieve all messages in list
- **Copy** Copy a message

4.1 Activity: Add

Use this activity to add a new message to the mailing list. The fields available are as follows:

**Functional Description**

**Subject** *(required)*
Subject line of your message. You can use MailMerge functions within your subject, but you must properly escape the bracket characters with the appropriate XML protocol. You must also represent any html special characters with the corresponding special char. (IE: & = &amp;)

**From-Email** *(required)*
The email address displayed in the “From” field. This is the “Reply-To” email address. You can use MailMerge functions, but you must properly escape the bracket characters with the appropriate XML protocol, and it must be an email address other than the recipient's address.

**From-Name** *(required)*
The name displayed in the “From” field. This labels the Reply-To email address with a phrase. Names like “XYZ Company Newsletter” are common. You can use MailMerge functions, but you must properly escape the bracket characters with the appropriate XML protocol.

**Message-Format** *(required)*
This specifies if your email message is formatted for HTML (multi-part) or TEXT (plain text only). Multi-part messages bundle the HTML, Plain Text, and AOL versions of your message into one email.

**Message-HTML**
Specify the HTML content of your message. You must also represent any html special characters with the corresponding special char. (IE: & = &amp;)

**Message-TEXT**
Specify the Text content of your message. You must also represent any html special characters with the corresponding special char. (IE: & = &amp;)

**Message-AOL**
Specify the AOL content of your message. Current versions of AOL (v6.0+) will typically receive the HTML version of the message and not this version. You must also represent any html special characters with the corresponding special char. (IE: & = &amp;)

**Clickthru**
“on” or “off” to specify if tracking is enabled. If enabled, your hyperlinks in your message will be converted into trackable links and listed on your message reports. This affects only the HTML version of your message.

**Clickthru-TEXT**
“on” or “off” to specify if tracking for web addresses in TEXT version of your message is converted into trackable links.

**Attachment**
Specifies the URL or FTP location of the file to be attached. NOTE: The file size must NOT exceed maximum allowable attachment size.

**MM_Attachment**
Specifies whether the attachment should be mail merged. A value of “on” means the attachment will be mail merged. A value of blank or not specifying this parameter means it will not be mail merged. Optional.
Charset
Specify which character set to use for this email. By default, us-ascii is used.
big5 / gb2312 / euc-jp / euc-kr / koi8-r / windows-1251 / windows-1250
(default: us-ascii)

HTMLEncoding
If your readers are having a difficult time viewing your message because of certain uncommon characters appearing, you can try using this parameter: “qp” = Quoted Printable (default: 7-bit) Do not change this if you are not having any problems.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>subject</td>
<td>n/a</td>
<td>(Subject of the message)</td>
</tr>
<tr>
<td>From-Email</td>
<td>from-email</td>
<td>n/a</td>
<td>(The email address displayed in the “From” field)</td>
</tr>
<tr>
<td>From-Name</td>
<td>from-name</td>
<td>n/a</td>
<td>(The name displayed in the “From” field)</td>
</tr>
<tr>
<td>Message-Format</td>
<td>message-format</td>
<td>n/a</td>
<td>HTML, TEXT</td>
</tr>
<tr>
<td>Message-HTML</td>
<td>message-html</td>
<td>n/a</td>
<td>(HTML message)</td>
</tr>
<tr>
<td>Message-TEXT</td>
<td>message-text</td>
<td>n/a</td>
<td>(Text message)</td>
</tr>
<tr>
<td>Message-AOL</td>
<td>message-AOL</td>
<td>n/a</td>
<td>(AOL message)</td>
</tr>
<tr>
<td>Clickthru</td>
<td>clickthru</td>
<td>n/a</td>
<td>on, off</td>
</tr>
<tr>
<td>Clickthru-TEXT</td>
<td>clickthru-text</td>
<td>n/a</td>
<td>on, off</td>
</tr>
<tr>
<td>Attachment</td>
<td>attachment</td>
<td>n/a</td>
<td>(<a href="http://web.com/file.txt">http://web.com/file.txt</a>) or (ftp://username:<a href="mailto:password@ftp.com">password@ftp.com</a>/file.txt)</td>
</tr>
<tr>
<td>MM_Attachment</td>
<td>mm_attachment</td>
<td>n/a</td>
<td>on</td>
</tr>
<tr>
<td>Charset</td>
<td>charset</td>
<td>n/a</td>
<td>us-ascii (default), big5, gb2312, euc-jp, euc-kr, koi8-r, windows-125, windows-1250 (ADD THE OTHERS)</td>
</tr>
<tr>
<td>HTMLEncoding</td>
<td>HTMLEncoding</td>
<td>n/a</td>
<td>qp = Quoted Printable (default: 7-bit)</td>
</tr>
</tbody>
</table>

Format

<DATA type="choose a type below" id="enter respective id value here">value</DATA>

EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):

- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

type: message
activity: add
Input:

```xml
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="subject">A Test HTML Message</DATA>
  <DATA type="from-email">newsletters@emaillabs.com</DATA>
  <DATA type="from-name">Updates from Emaillabs</DATA>
  <DATA type="message-html">I'm the HTML message!</DATA>
  <DATA type="message-text">I'm the Text message!</DATA>
  <DATA type="message-AOL">I'm the AOL message!</DATA>
  <DATA type="clickthru">on</DATA>
  <DATA type="clickthru-text">on</DATA>
</DATASET>
```

The above example created a new HTML message with clickthrus enabled for both the HTML and Text versions of the message. The response from this post will be a unique identifier for this message.

4.2 Activity: Update

Use this activity to update an existing message in the mailing list. The possible fields are the same as in “Add”. Any fields which are not included in this post will not change the original values of the message. The only required fields are site id, mailing list id, and message id.

**Functional Description**

**Subject [required]**

Subject line of your message. You can use MailMerge functions within your subject, but you must properly escape the bracket characters with the appropriate XML protocol.

**From-Email [required]**

The email address displayed in the “From” field. This is the “Reply-To” email address. You can use MailMerge functions, but you must properly escape the bracket characters with the appropriate XML protocol, and it must be an email address other than the recipient's address.

**From-Name [required]**

The name displayed in the “From” field. This labels the Reply-To email address with a phrase. Names like “XYZ Company Newsletter” are common. You can use MailMerge functions, but you must properly escape the bracket characters with the appropriate XML protocol.

**Message-Format [required]**

This specifies if your email message is formatted for HTML (multi-part) or TEXT (plain text only). Multi-part messages bundle the HTML, Plain Text, and AOL versions of your message into one email.

**Message-HTML**

Specify the HTML content of your message. You must also represent any html special characters with the corresponding special char. (IE: & = &amp;)

**Message-TEXT**

Specify the Text content of your message. You must also represent any html special characters with the corresponding special char. (IE: & = &amp;)

**Message-AOL**

Specify the AOL content of your message. Current versions of AOL (v6.0+) will typically receive the HTML version of the message and not this version. You must also represent any html special characters with the corresponding special char. (IE: & = &amp;)

**Clickthru**

“on” or “off” to specify if tracking is enabled. If enabled, your hyperlinks in your message will be converted into trackable links and listed on your message reports. This affects only the HTML version of your message.

**Clickthru-TEXT**

“on” or “off” to specify if tracking for web addresses in TEXT version of your message is converted into trackable links.

**Remove_Attachment**

A value of “on” will remove the existing message attachment. Optional.
Charset
Specify which character set to use for this email. By default, us-ascii is used.
big5 / gb2312 / euc-jp / euc-kr / koi8-r / windows-1251 / windows-1250
(default: us-ascii)

HTMLencoding
If your readers are having a difficult time viewing your message because of certain uncommon characters appearing, you can try using this parameter: “qp” = Quoted Printable (default: 7-bit) Do not change this if you are not having any

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>subject</td>
<td>n/a</td>
<td>(Subject of the message)</td>
</tr>
<tr>
<td>From-Email</td>
<td>from-email</td>
<td>n/a</td>
<td>(The email address displayed in the “From” field)</td>
</tr>
<tr>
<td>From-Name</td>
<td>from-name</td>
<td>n/a</td>
<td>(The name displayed in the “From” field)</td>
</tr>
<tr>
<td>Message-Format</td>
<td>message-format</td>
<td>n/a</td>
<td>HTML, TEXT</td>
</tr>
<tr>
<td>Message-HTML</td>
<td>message-html</td>
<td>n/a</td>
<td>(HTML message)</td>
</tr>
<tr>
<td>Message-TEXT</td>
<td>message-text</td>
<td>n/a</td>
<td>(Text message)</td>
</tr>
<tr>
<td>Message-AOL</td>
<td>message-AOL</td>
<td>n/a</td>
<td>(AOL message)</td>
</tr>
<tr>
<td>Clickthru</td>
<td>clickthru</td>
<td>n/a</td>
<td>on, off</td>
</tr>
<tr>
<td>Clickthru-TEXT</td>
<td>clickthru-text</td>
<td>n/a</td>
<td>on, off</td>
</tr>
<tr>
<td>Remove_Attachment</td>
<td>remove_attachment</td>
<td>n/a</td>
<td>on</td>
</tr>
<tr>
<td>Charset</td>
<td>charset</td>
<td>n/a</td>
<td>us-ascii (default), big5, gb2312, euc-jp,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>euc-kr, koi8-r, windows-125, windows-1250</td>
</tr>
<tr>
<td>HTMLencoding</td>
<td>HTMLencoding</td>
<td>n/a</td>
<td>qp = Quoted Printable (default: 7-bit)</td>
</tr>
</tbody>
</table>

Format
</DATA type="choose a type below" id="enter respective id value here">value</DATA>

EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

<DATASET>
<SITE_ID>123</SITE_ID>
<MID>345</MID>
<MID>24251</MID>
</DATASET>

<DATA type="subject">A test html message</DATA>
<DATA type="from-email">test@emaillabs.com</DATA>
<DATA type="clickthru-text">off</DATA>
### 4.3 Activity: Schedule

Use this activity to schedule the delivery of a message. You may use this function to schedule, pause, resume, or cancel a message. The only required fields are site id, mailing list id, message id, and action.

#### Functional Description

<table>
<thead>
<tr>
<th>Action</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>schedule / pause / resume / cancel / proof</td>
<td>n/a</td>
<td>schedule, pause, resume, cancel, proof</td>
<td></td>
</tr>
<tr>
<td>Delivery-Year</td>
<td>delivery-year</td>
<td>n/a</td>
<td>(YYYY), defaults to current</td>
</tr>
<tr>
<td>Delivery-Month</td>
<td>delivery-month</td>
<td>n/a</td>
<td>1..12, (Defaults to the current date)</td>
</tr>
<tr>
<td>Delivery-Day</td>
<td>delivery-day</td>
<td>n/a</td>
<td>0..31 (defaults to the current date)</td>
</tr>
<tr>
<td>Delivery-Hour</td>
<td>delivery-hour</td>
<td>n/a</td>
<td>0 to 23, value representing hour</td>
</tr>
<tr>
<td>Randomgroup-Max</td>
<td>randomgroup-max</td>
<td>n/a</td>
<td>(n)</td>
</tr>
<tr>
<td>Randomgroup-Select</td>
<td>randomgroup-select</td>
<td>n/a</td>
<td>1,2,3... n</td>
</tr>
<tr>
<td>Rule</td>
<td>rule</td>
<td>n/a</td>
<td>(filter ID)</td>
</tr>
</tbody>
</table>

#### Format

```
<DATA type="choose a type below" id="enter respective id value here">value</DATA>
```

#### Example

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

---
type: message
activity: schedule

Input:

```xml
<DataSet>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <MID>24251</MID>
  <Data type="action">schedule</Data>
  <Data type="delivery-year">2002</Data>
  <Data type="delivery-month">12</Data>
  <Data type="delivery-day">25</Data>
  <Data type="delivery-hour">8</Data>
  <Data type="randomgroup-max">4</Data>
  <Data type="randomgroup-select">1,2</Data>
  <Data type="rule">1425</Data>
</DataSet>
```

### 4.4 Activity: Proof

Use this activity to create a proof message. The only required fields are site id, mailing list id, and message id. The message id corresponds to the ID of the message to be proofed. This message sends to those who have been flagged as a proof list member.

**EXAMPLE**

Post data to: Depending on the server farm you login to (see your login URL):

- [http://www.uptilt.com/API/mailing_list.html](http://www.uptilt.com/API/mailing_list.html)
- [http://www.up0.net/API/mailing_list.html](http://www.up0.net/API/mailing_list.html)
- [http://www.elabs3.com/API/mailing_list.html](http://www.elabs3.com/API/mailing_list.html)
- [http://www.elabs7.com/API/mailing_list.html](http://www.elabs7.com/API/mailing_list.html)

Input:

```xml
<DataSet>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <MID>24251</MID>
</DataSet>
```

### 4.5 Activity: Query-Data

Use this activity to retrieve data related to a particular message. The only required fields are site id, mailing list id, and message id. Fields returned:

**Functional Description (Input)**

- **SITE_ID**
  Enter your Site ID or Customer ID in this field

- **MLID**
  Enter your Mailing List ID into this field

- **MID**
  Enter your unique Message ID into this field

**Response Description (Output)**

- **Subject**
  Subject line of your message. You can use MailMerge functions within your subject, but you must properly escape the bracket characters with the appropriate XML protocol.
From-Email
The email address displayed in the “From” field. This is the “Reply-To” email address. You can use MailMerge functions, but you must properly escape the bracket characters with the appropriate XML protocol, and it must be an email address other than the recipient’s address.

From-Name
The name displayed in the “From” field. This labels the Reply-To email address with a phrase. Names like “XYZ Company Newsletter” are common. You can use MailMerge functions, but you must properly escape the bracket characters with the appropriate XML protocol.

Message-Format
This specifies if your email message is formatted for HTML (multi-part) or TEXT (plain text only). Multi-part messages bundle the HTML, Plain Text, and AOL versions of your message into one email.

Message-HTML
Specify the HTML content of your message.

Message-TEXT
Specify the Text content of your message

Message-AOL
Specify the AOL content of your message. Current versions of AOL (v6.0+) will typically receive the HTML version of the message and not this version.

Clickthru
“on” or “off” to specify if tracking is enabled. If enabled, your hyperlinks in your message will be converted into trackable links and listed on your message reports. This affects only the HTML version of your message.

Clickthru-TEXT
“on” or “off” to specify if tracking for web addresses in TEXT version of your message is converted into trackable links.

Charset
Specify which character set to use for this email. By default, us-ascii is used.
big5 / gb2312 / euc-jp / euc-kr / koi8-r / windows-1251 / windows-1250
(default: us-ascii)

HTMLencoding
If your readers are having a difficult time viewing your message because of certain uncommon characters appearing, you can try using this parameter: “qp” = Quoted Printable (default: 7-bit)

Rule
The ID of the rule to be applied to this message.
Sent
Status of the message: yes / no / waiting to be scheduled / paused

Delivery
Time of message delivery (seconds after midnight of the day in question)

Randomgroup-Max
Random groups created for this message

Randomgroup-Select
Random groups selected for delivery

EXAMPLE
Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptl.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html
type: message
activity: query-data

Input:

```xml
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <MID>24251</MID>
</DATASET>
```

Response:

```xml
<DATASET><TYPE>success</TYPE>
  <RECORD>
    <DATA type="subject">A Test HTML Message</DATA>
    <DATA type="from-email">newsletters@emaillabs.com</DATA>
    <DATA type="from-name">Updates from Emaillabs</DATA>
    <DATA type="message-format">HTML</DATA>
    <DATA type="message-html">I 'm the HTML message!</DATA>
    <DATA type="message-text">I'm the Text message!</DATA>
    <DATA type="message-aol">I'm the AOL message!</DATA>
    <DATA type="rule">1425</DATA>
    <DATA type="sent">yes</DATA>
    <DATA type="delivery">1402342</DATA>
    <DATA type="charset"></DATA>
    <DATA type="htmlencoding"></DATA>
    <DATA type="randomgroup-max">4</DATA>
    <DATA type="randomgroup-select">1,2</DATA>
    <DATA type="clickthru">on</DATA>
    <DATA type="clickthru-text">on</DATA>
  </RECORD>
</DATASET>
```

### 4.6 Activity: Query-Stats

Use this activity to retrieve statistical data (sent, clickthru, action-tracking, and open history) related to a particular message. The only required fields are site id, mailing list id, and message id. Fields returned:

**Functional Description (Input)**

**SITE_ID**
Enter your Site ID or Customer ID in this field

**MLID**
Enter your Mailing List ID into this field

**MID**
Enter your unique Message ID into this field

**XXX**
Specify the statistic you want to pull

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SITE_ID</td>
<td>n/a</td>
<td>n/a</td>
<td>(SITE ID number)</td>
</tr>
<tr>
<td>MLID</td>
<td>n/a</td>
<td>n/a</td>
<td>(MLID number)</td>
</tr>
<tr>
<td>MID</td>
<td>n/a</td>
<td>n/a</td>
<td>(MID number)</td>
</tr>
<tr>
<td>(XXX)</td>
<td>n/a</td>
<td>n/a</td>
<td>(OPENED, SENT, CLICKED-DETAILS, ACTION-TRACKING)</td>
</tr>
</tbody>
</table>

**Response Description (Output)**
Stats-Sent  
Total # of members who were sent this message

Stats-Opens  
Total # of times this message was opened

Stats-Opens-Unique  
Total # of unique members who opened this message

Stats-Clicks-Unique  
Total # of unique members who clicked at least one link in the message

Stats-Bounces  
Total # of bounces

Stats-Unsubscribes  
Total # of unsubscribes

Clicks-Total  
Total # of clicks for a particular link

Clicks-Unique  
Total # of unique clicks for a particular link

Clicks-ID  
Unique ID of a particular link

Action-Tracking  
Total # of action tracking hits

Clicked-Details  
Shows everyone who clicked a link (and what link) for the given message. In the output, a “type” field is returned ("H" or "T") to stand for an HTML click or a Text click respectively.

**EXAMPLE**

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

type: message  
activity: query-stats

Input:

```
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <MID>24251</MID>
  <DATA type="ACTION-TRACKING"></DATA>
</DATASET>
```

Response:

```
<DATASET><TYPE>success</TYPE><RECORD>
  <DATA type="stats-sent">10500</DATA>
  <DATA type="stats-opens">5142</DATA>
  <DATA type="stats-opens-unique">3024</DATA>
  <DATA type="stats-clicks-unique">2112</DATA>
  <DATA type="stats-bounces">2</DATA>
  <DATA type="stats-unsubscribes">11</DATA>
</RECORD>
```
<DATA type="clicks-total" id="http://www.emaillabs.com">1024</DATA>
<DATA type="clicks-unique" id="http://www.emaillabs.com">823</DATA>
<DATA type="clicks-id" id="http://www.emaillabs.com">ab409128e</DATA>
</RECORD>
<RECORD>
<DATA type="clicks-total" id="http://www.yahoo.com">1024</DATA>
<DATA type="clicks-unique" id="http://www.yahoo.com">823</DATA>
<DATA type="clicks-id" id="http://www.yahoo.com">0b409120e</DATA>
</RECORD>
<RECORD>
<DATA type="clicks-total" id="http://www.emaillabs.com/quote.html">1024</DATA>
<DATA type="clicks-unique" id="http://www.emaillabs.com/quote.html">823</DATA>
<DATA type="clicks-id" id="http://www.emaillabs.com/quote.html">c24091287</DATA>
</RECORD>
</DATASET>

The above response lists 3 clicks. You can see that the click stats can be grouped by their “id” identifier.

4.7 Activity: Query-Listdata
Use this activity to retrieve all messages in the list. The only required fields are site id and mailing list id. To limit messages in a certain month, provide DATA types of month and year.

Functional Description (Input)

SITE_ID
Enter your Site ID or Customer ID in this field

MLID
Enter your Mailing List ID into this field

Month
Enter the month you wish to view.

Year
Enter the year you wish to view.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SITE_ID</td>
<td>n/a</td>
<td>n/a</td>
<td>(SITE ID number)</td>
</tr>
<tr>
<td>MLID</td>
<td>n/a</td>
<td>n/a</td>
<td>(MLID number)</td>
</tr>
<tr>
<td>MID</td>
<td>n/a</td>
<td>n/a</td>
<td>(MID number)</td>
</tr>
</tbody>
</table>

Response Description (Output)

Subject
Subject of the message

Message-Format
HTML or TEXT

Sent
Status of the message: yes / no / waiting to be scheduled / paused

Date
Date of message delivery (YYYY-MM-DD)
Delivery
Time of message delivery (seconds from midnight of given day)

Stats-Sent
# of messages sent

MID
Message ID

**EXAMPLE**
Post data to: Depending on the server farm you login to (see your login URL):
- [http://www.uptilt.com/API/mailing_list.html](http://www.uptilt.com/API/mailing_list.html)
- [http://www.up0.net/API/mailing_list.html](http://www.up0.net/API/mailing_list.html)
- [http://www.elabs3.com/API/mailing_list.html](http://www.elabs3.com/API/mailing_list.html)
- [http://www.elabs7.com/API/mailing_list.html](http://www.elabs7.com/API/mailing_list.html)

**Input:**
```
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="month">10</DATA>
  <DATA type="year">2002</DATA>
</DATASET>
```

**Response:**
```
<DATASET><TYPE>success</TYPE>
  <RECORD>
    <DATA type="subject">A Test HTML Message</DATA>
    <DATA type="message-format">HTML</DATA>
    <DATA type="sent">yes</DATA>
    <DATA type="delivery">1402342</DATA>
    <DATA type="date">2002-05-22</DATA>
    <DATA type="stats-sent">100423</DATA>
    <DATA type="mid">25132</DATA>
  </RECORD>
  <RECORD>
    <DATA type="subject">Another Test HTML Message</DATA>
    <DATA type="message-format">TEXT</DATA>
    <DATA type="sent">Sending</DATA>
    <DATA type="delivery">1402342</DATA>
    <DATA type="date">2002-06-22</DATA>
    <DATA type="stats-sent">15</DATA>
    <DATA type="mid">25137</DATA>
  </RECORD>
  <RECORD>
    <DATA type="subject">Last Test HTML Message</DATA>
    <DATA type="message-format">HTML</DATA>
    <DATA type="sent">waiting to be scheduled</DATA>
    <DATA type="delivery">3600</DATA>
    <DATA type="date">2002-08-12</DATA>
    <DATA type="stats-sent">0</DATA>
    <DATA type="mid">25141</DATA>
  </RECORD>
</DATASET>
```
4.8 Activity: Copy

Use this activity to copy an existing message in the mailing list. The only required fields are site id, mailing list id, and message id. The message id of the new, copied message is returned.

EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

```xml
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <MID>24251</MID>
</DATASET>
```

4.9 Activity: Query-Deliverydetails

Use this activity to retrieve the delivery details of a particular message. The required fields are site id, mailing list id, and message id. Returned will be an RSS feed of delivery information: Which server(s) sent the message, how many and at what time.

Functional Description (Input)

SITE_ID
Enter your Site ID or Customer ID in this field

MLID
Enter your Mailing List ID into this field

MID
Message ID of the message.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SITE_ID</td>
<td>n/a</td>
<td>n/a</td>
<td>(SITE ID number)</td>
</tr>
<tr>
<td>MLID</td>
<td>n/a</td>
<td>n/a</td>
<td>(MLID number)</td>
</tr>
<tr>
<td>MID</td>
<td>n/a</td>
<td>n/a</td>
<td>(MID number)</td>
</tr>
</tbody>
</table>

Response Description (Output)

```xml
<?xml version="1.0" encoding="iso-8859-1" ?>
<rss version="2.0">
  <channel>
    <title>Delivery Details for 1-1-14</title>
    <pubDate>07 09 05 14:04:29 PST</pubDate>
    <item>
      <title>702704-Mail10-1-1126199890</title>
      <link></link>
      <description>1000</description>
    </item>
  </channel>
</rss>
```
EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

Input:

```xml
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <MID>11111</MID>
</DATASET>
```

Response:

```xml
<?xml version="1.0" encoding="iso-8859-1" ?>
<rss version="2.0">
  <channel>
    <title>Delivery Details for 1-1-14</title>
    <pubDate>07 09 05 14:04:29 PST</pubDate>
    <item>
      <title>702704-Mail10-1-1126199890</title>
      <link></link>
      <description>1000</description>
    </item>
    <item>
      <title>702704-Mail10-2-1126199900</title>
      <link></link>
      <description>1000</description>
    </item>
    <item>
      <title>702704-Mail5-3-1126199911</title>
      <link></link>
      <description>1000</description>
    </item>
    <item>
      <title>702704-Mail5-0-1126199890</title>
      <link></link>
      <description>1000</description>
    </item>
  </channel>
</rss>

5.0 Filter Handling

This provides all activities associated with filters.

**API Type value:** filter

**Required Fields:**
- `<SITE_ID>` Site ID
- `<MLID>` Mailing List ID

**Optional Fields:**
- `<DATA type="id">` Filter ID

**Valid Activities:**
- **Add** Add a new filter
- **Update** Update an existing filter
- **Delete** Delete an existing filter
- **Generate** Generate an already existing filter
- **Query-data** To query the results of the filter

5.1 Activity: Add

Use this activity to create a new filter. The fields available are as follows (required fields in orange):

- **Name** Name of the filter to be created or updated.
- **Email** Email Restriction
- **Demographic** Demographic Restriction
- **Activity** Activity Restriction

**Functional Description**

**Name** *(required)*

Name of the filter to be created or updated.

**Activity: Sent**

Define an activity restriction based on whether persons were sent a specific message.

**Activity: Opened**

Define an activity restriction based on whether persons opened a specific message. You can also specify a threshold of how many times they opened a specific message.

**Activity: Opened & Clicked**

Define an activity restriction based on whether persons opened and clicked on a specific link in a specific message.

**Activity: Clicked**

Define an activity restriction based on whether persons clicked on a specific link.

**Activity: Inverse**

Modify a previous activity restriction by inverting its conditions.

**Demographic: Value is a Constant**

Define a demographic-based restriction to find people who match a certain value. For example, if “City” equals “San Francisco”.

**Demographic: Value in a Range**

Define a demographic-based restriction to find people who match a certain range of values. For example, if “Zip/Postal Code” equals between “7000” to “99999”.

**Demographic: Value is Less than / Greater than a Constant**

Define a demographic-based restriction to find people who have a specific demographic value that is either greater than or less than a constant you define. For example, if “Products Purchased" is greater than “2”.

**Demographic: Inverse**

Modify a previous demographic restriction by inverting its conditions.
<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name [required]</td>
<td>name</td>
<td>--</td>
<td>(name of filter)</td>
</tr>
<tr>
<td>Activity: Sent</td>
<td>activity</td>
<td>(message id)</td>
<td>sent</td>
</tr>
<tr>
<td>Activity: Opened</td>
<td>activity</td>
<td>(message id)</td>
<td>opened</td>
</tr>
<tr>
<td>Activity: Opened &amp; Clicked</td>
<td>activity</td>
<td>(message id)</td>
<td>openedclicked</td>
</tr>
<tr>
<td>Activity: Clicked</td>
<td>activity</td>
<td>(message id)</td>
<td>clicked</td>
</tr>
<tr>
<td>Activity: Inverse</td>
<td>extra</td>
<td>inverse</td>
<td>A-(message id)</td>
</tr>
<tr>
<td>Demographic: Value is a Constant</td>
<td>demographic</td>
<td>(demographic id)</td>
<td>(demographic value)</td>
</tr>
<tr>
<td>Demographic: Value in a Range</td>
<td>demographic</td>
<td>(demographic id)</td>
<td>(demographic value)</td>
</tr>
<tr>
<td></td>
<td>extra</td>
<td>between</td>
<td>D-(demographic id)</td>
</tr>
<tr>
<td>Demographic: Value is Less than / Greater than a Constant</td>
<td>demographic</td>
<td>(demographic id)</td>
<td>(demographic value)</td>
</tr>
<tr>
<td></td>
<td>extra</td>
<td>lessthan</td>
<td>D-(demographic id)</td>
</tr>
<tr>
<td></td>
<td>extra</td>
<td>greaterthan</td>
<td>D-(demographic id)</td>
</tr>
</tbody>
</table>

**Format**

<DATA type="choose a type below" id="enter respective id value here">value</DATA>

**EXAMPLE**

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

<table>
<thead>
<tr>
<th>type</th>
<th>filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>activity</td>
<td>add</td>
</tr>
</tbody>
</table>
The above example created a new Filter that selects users whose name is not Joe, Joey, or Joseph, and was not sent the message with the message id 1425. The response from this post will be a unique identifier for this filter.

5.2 Activity: Update
Use this activity to update a filter. The fields available are as follows:

**Functional Description**

**Name [required]**
Name of the filter to be created or updated.

**Activity: Sent**
Define an activity restriction based on whether persons were sent a specific message.

**Activity: Opened**
Define an activity restriction based on whether persons opened a specific message. You can also specify a threshold of how many times they opened a specific message.

**Activity: Opened & Clicked**
Define an activity restriction based on whether persons opened and clicked on a specific link in a specific message.

**Activity: Clicked**
Define an activity restriction based on whether persons clicked on a specific link (primary used for Text-only users)

**Activity: Inverse**
Modify a previous activity restriction by inverting its conditions

**Demographic: Value is a Constant**
Define a demographic-based restriction to find people who match a certain value. For example, if “City” equals “San Francisco”.

**Demographic: Value in a Range**
Define a demographic-based restriction to find people who match a certain range of values. For example, if “Zip/Postal Code” equals between “7000” to “99999”.

**Demographic: Value is Less than / Greater than a Constant**
Define a demographic-based restriction to find people who have a specific demographic value that is either greater than or less than a constant you define. For example, if “Products Purchased” is greater than “2”.

**Demographic: Inverse**
Modify a previous demographic restriction by inverting its conditions

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>name</td>
<td>--</td>
<td>(name of filter)</td>
</tr>
<tr>
<td>Activity:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
activity (message id) sent

Activity: Opened
activity (message id) opened

Activity: Opened & Clicked
activity (message id) openedclicked

Activity: Clicked
activity (message id) clicked

Activity: Inverse
extra inverse A-(message id)

Demographic: Value is a Constant
demographic (demographic id) (demographic value)

Demographic: Value in a Range
demographic (demographic id) (demographic value)
extra between D-(demographic value)

Demographic: Value is Less than / Greater than a Constant
demographic (demographic id) (demographic value)
extra lessthan D-(demographic value)
-OR-
extra greaterthan D-(demographic id)

Demographic: Inverse
extra inverse D-(demographic id)

**Format**

<DATA type="choose a type below" id="enter respective id value here">value</DATA>

**EXAMPLE**

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

Type: filter
Activity: update

Input:

<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="id">14325</DATA>
  <DATA type="name">A Test Filter</DATA>
  <DATA type="demographic" id="1">Joe</DATA>
  <DATA type="demographic" id="1">Joe</DATA>
</DATASET>
The above example updated an existing Filter that selects users whose name is not Joe, Joey, or Joseph, and was not sent the message with the message id 1425.

5.3 Activity: Delete
Use this activity to delete a filter. The fields available are as follows.

**Functional Description**

- **Name** *required*
  - Name of the filter

- **ID** *required*
  - Filter ID

**EXAMPLE**

Post data to: Depending on the server farm you login to (see your login URL):
- [http://www.uptilt.com/API/mailing_list.html](http://www.uptilt.com/API/mailing_list.html)
- [http://www.up0.net/API/mailing_list.html](http://www.up0.net/API/mailing_list.html)
- [http://www.elabs3.com/API/mailing_list.html](http://www.elabs3.com/API/mailing_list.html)
- [http://www.elabs7.com/API/mailing_list.html](http://www.elabs7.com/API/mailing_list.html)

- **type:** filter
- **activity:** delete

**Input:**

```xml
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="id">14325</DATA>
</DATASET>
```

5.4 Activity: Generate
Use this activity to generate a filter. The fields available are as follows.

**Functional Description**

- **Name** *required*
  - Name of the filter

- **ID** *required*
  - Filter ID

- **Email** *required*
  - This is the email address that will receive notice once the filter results are generated and ready for download/use.

**EXAMPLE**

Post data to: Depending on the server farm you login to (see your login URL):
- [http://www.uptilt.com/API/mailing_list.html](http://www.uptilt.com/API/mailing_list.html)
- [http://www.up0.net/API/mailing_list.html](http://www.up0.net/API/mailing_list.html)
- [http://www.elabs3.com/API/mailing_list.html](http://www.elabs3.com/API/mailing_list.html)
- [http://www.elabs7.com/API/mailing_list.html](http://www.elabs7.com/API/mailing_list.html)

- **type:** filter
activity: generate

Input:

```xml
<DATASET>
  <MLID>123</MLID>
  <DATA type="extra" id="email">notifyme@here.com</DATA>
  <DATA type="id">55555</DATA>
</DATASET>
```

### 5.5 Activity: Query-Data

Use this activity to query results of a filter. The fields available are as follows.

#### Functional Description

<table>
<thead>
<tr>
<th>Name</th>
<th>[required]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the filter</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>[required]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter ID</td>
<td></td>
</tr>
</tbody>
</table>

#### EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):
- [http://www.uptilt.com/API/mailing_list.html](http://www.uptilt.com/API/mailing_list.html)
- [http://www.up0.net/API/mailing_list.html](http://www.up0.net/API/mailing_list.html)
- [http://www.elabs3.com/API/mailing_list.html](http://www.elabs3.com/API/mailing_list.html)
- [http://www.elabs7.com/API/mailing_list.html](http://www.elabs7.com/API/mailing_list.html)

<table>
<thead>
<tr>
<th>type:</th>
<th>filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>activity:</td>
<td>query-data</td>
</tr>
</tbody>
</table>

Input:

```xml
<DATASET>
  <SITE_ID>123</SITE_ID>
  <MLID>345</MLID>
  <DATA type="id">14325</DATA>
</DATASET>
```
6.0 List Handling

This provides all activities associated with list management.

API Type value: list
Required Fields: <SITE_ID> Site ID

Valid Activities:
- Query-Listdata: Query all existing mailing lists
- Add: Create a new mailing list
- Delete: Delete a mailing list
- Query-Listdata Stats: Query all mailing lists and their stats

6.1 Activity: Query-Listdata

Use this activity to get information about all lists available. The only required field is Site ID.

EXAMPLE

Post data to: Depending on the server farm you login to (see your login URL):

- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

Input:
```xml
<DATASET>
  <SITE_ID>123</SITE_ID>
</DATASET>
```

Response:
```xml
<DATASET><TYPE>success</TYPE>
  <RECORD>
    <DATA type="name" id="142">mailing list #1</DATA>
    <DATA type="messages" id="142">14</DATA>
    <DATA type="last-message" id="142">Aug 1, 2002</DATA>
    <DATA type="members" id="142">1420</DATA>
  </RECORD>
  <RECORD>
    <DATA type="name" id="143">mailing list #2</DATA>
    <DATA type="messages" id="143">1</DATA>
    <DATA type="last-message" id="143">Aug 5, 2002</DATA>
    <DATA type="members" id="143">124420</DATA>
  </RECORD>
</DATASET>
```

The above query returned two mailing lists, each identified with the mailing list id (specified by the “id” property).

6.2 Activity: Add

Use this activity to create a new mailing list. The only required field is Name.

Functional Description

Name [required]
Specify the name of your new mailing list

Format
<DATA type="name" value="EXAMPLE"

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

type: list
activity: add

Input:

<Data type="name">Newest Mailing List</Data>

Response:

<DataSet><Type>success</Type><Data type="mild">1425</Data></DataSet>

6.3 Activity: Delete
Use this activity to delete an existing mailing list.

Functional Description

mlid [required]
Specify the mailing list ID number to delete

Format

<MLID>IDnumber</MLID>

<Example>

Post data to: Depending on the server farm you login to (see your login URL):
- http://www.uptilt.com/API/mailing_list.html
- http://www.up0.net/API/mailing_list.html

type: list
activity: delete

Input:

<Data type="name">Newest Mailing List</Data>

Response:

<DataSet><Type>success</Type><Data type="mild">Mailing List (1234) Deleted</Data></DataSet>