

JangoMail Tutorial

Advanced Filtering of List Members

Summary

This tutorial teaches you how to filter a List to send to only members that match a certain criteria. Perhaps you only want to send to the Hotmail addresses in a List. Perhaps you only want to send to those List members whose "Age" is over 30. Perhaps you only want to send to those members of the List that have joined the List in the last 60 days. Perhaps you only want to send to those members of the List who have clicked at least one link in a prior email campaign. You can accomplish all of these by filtering your List.

You can filter your List based on two kinds of criteria:

1. The information you have stored for each List member in the List's fields, such as FirstName, LastName, Age, DOB, City
2. Information stored in your JangoMail account about the List member's past behavior, such as clicks and opens

We also have a new way to filter list members from the **Reports** section. Read more on our blog on how to [Use a Report to Create and Send to an Email List](#).

Getting Started

When you wish to send to a filtered portion of your List rather than the entire List, the steps you take are a bit different. Instead of starting on the "Messages" tab, like you would when sending to an entire List, you start with the "Lists" tab. From this tab, you filter the List, and then go to the "Messages" section.

After navigating to the **Lists** tab, click the Edit Icon next to the List that you wish to filter. Navigate to the **List Tools** tab. Then click the **Filter & Send** option.

List Details ▶ Cookie Customers

Members Fields List Settings List Tools

Members & Fields
Add or edit the members or fields in this list.

Filter & Send
Filter the list and send a message to it.

Auto-Responders
Set up auto-responders and view activity.

Unsubscribe all members
Adds all members of this list to your account's unsubscribe list.

Delete all members
Remove all members from the list.

Delete members by ad-hoc entry
Remove list members by entering their addresses in a text box.

Create an Opt In Form
Design and publish the form your visitors can use to **add** themselves to this list.

There are two modes to help you compose your filter criteria, **Simple** and **Advanced**. The Simple mode provides an easy-to-use interface to build simple queries based only on List field criteria. The Advanced mode is more sophisticated and allows you to compose more complex filter criteria by typing in freehand SQL.

This tutorial is focused more on the Advanced mode, and will give examples in freehand SQL.

Note: This tutorial is for advanced users. While programming knowledge is not necessary, an understanding of some SQL will help.

Some Examples

If you have a List called "My Contacts" and you want to send an email to only the Hotmail.com addresses in this List, your WHERE clause would look like:

```
emailaddress like '%hotmail.com%'
```

The "emailaddress" field is present in all Lists, so it can always be used in your SQL statements. Let's assume that additionally you have these fields in the List:

```
FirstName  
LastName  
Age
```

You want to send to those members of the List whose age is greater than or equal to 30.

The WHERE clause would look like:

```
CONVERT(int, Age) >= 30
```

It is important to note the use of the CONVERT statement here. JangoMail treats all List field values as strings. In order to compare the field "Age" to the number 30, it first has to be treated like a number itself, in this case an integer. The SQL CONVERT function allows us to cast "Age" to an integer.

Finally, let's assume you want to send to only those members of the List that have joined in the last 30 days. Another built-in field in all Lists is the "submission_date" field that stores the date and time that an email address joined the List. You can use this field along with the SQL DATEDIFF function to create the appropriate WHERE clause:

```
DATEDIFF(Day, submission_date, GetDate()) <= 30
```

In this case, DATEDIFF calculates the number of days in between submission_date and GetDate(). GetDate() always returns the current system date/time in the EST/New York time zone.

Filters Based on List Member Behavior

So far, we have covered filters based on a List member's individual profile – that is, the data that is stored in the List member fields. You also have the ability to filter Lists based off the List member's past behavior, such as clicks and opens of a particular campaign in the past.

For example, if you want to send to only members of the "My Contacts" List that have clicked on any link in your account's history, your WHERE clause would look like:

```
emailaddress in (##clicked##)
```

In this case, ##clicked## is a virtual JangoMail table that contains the email addresses of all those that have clicked any URL in your account's history. Similarly, if you wanted to

send to only members of the "My Contacts" List that have opened any email in your account's history, your WHERE clause would look like:

emailaddress in (##opened##)

The full list of virtual tables, including format variations based on additional filter criteria can be found below:

##clicked##	email addresses that have clicked any URL in any emailing in your account's history
##clicked-http://www.mybigstore.com##	email addresses that have clicked a specific URL in your account's history
##MassEmailID.clicked## example: ##47874874.clicked##	email addresses that have clicked any URL in a previously sent mass email campaign
##MassEmailID.clicked-http://www.mybigstore.com## example: ##47874874.clicked-http://www.mybigstore.com##	email addresses that have clicked a specific URL in a previously sent mass email campaign
##opened##	email addresses that have opened email in your account's history
##MassEmailID.opened## example: ##47874874.opened##	email addresses that have opened a specific previously sent mass email campaign
##forwarded##	email addresses that have forwarded any email in your account's history
##MassEmailID.forwarded## example: ##47874874.forwarded##	email addresses that have forwarded a specific previously sent mass email campaign
##replied##	email addresses that have replied to any mass email in your account's history <i>Note: The "replied" filters will only work if JangoMail is receiving and processing replies rather than you receiving them directly.</i>
##MassEmailID.replied## example: ##47874874.replied##	email addresses that have replied to a specific previously sent mass email campaign
##received##	email addresses that have received any mass email in your account's history
##MassEmailID.received## example: ##47874874.received##	email addresses that have received a specific previously sent mass email campaign
##action-URL## example: ##action-http://www.mybigstore.com/purchase.asp##	email addresses that have landed on a specific page within your web site

	<i>Note: The "action" filters will only work if you are properly using JangoMail's Activity Tracking Feature.</i>
##action-CustomParameter## example: ##action-purchase=y##	email addresses that have landed on a page within your web site that has been tagged by you with a specific parameter
##MassEmailID.action-URL## example: ##47874874.action-http://www.site.com/buy.asp##	email addresses that have landed on specific page within your web site after clicking through from a specific mass email campaign
##MassEmailID.action-CustomParameter## example: ##47874874.action-purchase=y##	email addresses that have landed on a page within your web site that has been tagged by you with a specific parameter, after having clicked through from a specific mass email campaign

*Notes: The mass email ID for a campaign can be retrieved from the "Sending Complete" email notification or from the **Reports --> Open Report Dashboard** section. Your account's history is limited to how long JangoMail stores reporting data - 90 days for most types of reporting data.*

More Examples

- To send to members of your List that have clicked the URL <http://www.mybigstore.com/Product1.asp> in your account's history:

WHERE

emailaddress in (##clicked-http://www.mybigstore.com/Product1.asp##)

- To send to members of your List that have opened but never clicked anything in your account's history:

WHERE

emailaddress in (##opened##) and emailaddress NOT in (##clicked##)

- To send to just the yahoo.com addresses that opened mass email 34930234

WHERE

emailaddress like '%yahoo.com%' and emailaddress in (##34930234.opened##)

- To send to just those members whose birthday is today and have previously made a purchase on your web site (assuming DOB is a field in your List)

WHERE

Month(DOB) = Month(GetDate()) and Day(DOB) = Day(GetDate()) and emailaddress in (##action-http://www.mybigstore.com/purchase_confirm.html##)

How To Schedule a Follow-up Email Right When You Send The First Email

Let's say that you want to first send a mass email to a List of 1000 contacts, and as a follow-up you would like to have an email sent 24 hours later to just those members of the List that opened the first email. Here's what you would do:

1. Send a regular mass email to the entire List, and note the campaign's ID number. This number can be retrieved from the "Sending Complete" email notification or from the **Reports --> Open Report Dashboard** section.
2. Next, go to the **Lists** section, and select the List that you just sent to.
3. Navigate to the **List Tools** tab and click the **Filter & Send** option.
4. Switch to the **Advanced Query Editor** tab.
5. In the WHERE box, type: `emailaddress in (##XXXXX.opened##)`, where XXXXX is the first mass email's ID number.
6. Click the **Filter and Send** button.
7. You will be taken to the **Messages** section, and the green box at the top of the page will likely tell you that 0 members of you List have been selected. *This is okay, since at this point, you probably don't have any opens on your first email campaign. But when the email is actually sent, 24 hours from now, there will be opens on that first email campaign.*
8. Compose your follow-up email. In the **Schedule** tab, select **Later**, enter in a date/time 24 hours from now, and make sure that **Should JangoMail pre-process this message now?** is set to **No**. This will enable JangoMail to re-query your List at the time of email sending.
9. Click the **Send** button.

Using with the API

To use advanced list filters with the API (<https://api.jangomail.com>), you would pass in the filter criteria just as you would any ordinary List filter criteria. In the `SendMassEmail`, `SendMassEmailPrevious`, or `SendMassEmailRaw` methods, you would pass in the filter criteria into the `ToGroupFilter` input parameter in the following format:

```
[Group Name]: [WHERE clause]
```

For example, if the List (Group) was called "My Contacts", the input parameter's value might look like this if you wished to send to all List members that had previously clicked:

```
My Contacts: emailaddress in (##clicked##)
```

For more information, please see the API documentation on the three above referenced methods at <http://api.jangomail.com/help>.

Optimization Tips

To optimize the speed at which your Lists are filtered:

1. Don't keep extraneous fields in your List. For example, if your List has the fields `FirstName` and `LastName`, but you're not using them, delete these fields from the List. It will make a significant difference in the speed at which your List is filtered.

2. Replace the * in the SELECT clause with only the fields you need for this particular emailing. The fields should be separated by commas. At a minimum, you will need the "emailaddress" field. But if you are not personalizing your Subject/Message with other List field data, then you don't need any other fields to be a part of the SELECT statement.
3. In the Advanced Filter mode, click the "**Query is large, so only show sample records**" checkbox if you anticipate that your query will take a long time or will be processing a large number of records. Checking this box will make the filter run approximately 95% faster—the only disadvantage to checking this box is that you won't be able to **see** your filtered data on the following page; you will only be able to see five sample records.