

# DemocracyInAction Imports Tipsheet

## GENERAL BEHAVIORS

\*Imports are a way to take records stored in an external source and move them into DemocracyInAction's database *en masse*. You might do this when setting up your account with records you've used in a previous e-mail application, when synchronizing between our database and an outside one, or to append updated information from your ongoing activities – donations you've received through the mail, names on a sign-up sheet from a rally, or anything at all.

\*You can also use the system to do quick large-scale updates to existing data – assigning many people to a common group, or adding donation history.

## ESSENTIAL DUPLICATION LOGIC

\*With the importer, as with inputs arriving from action forms, the system looks for duplicates in a set sequence:

1. Supporter\_KEY (the unique supporter ID number assigned by our database)
2. uid (an arbitrary field that you can populate if you've used an external database that has assigned its own ID numbers)
3. e-mail address

\*If the system's default behavior matches any of these fields with an existing record in your database, it will treat the import as an *update* to that record and either append or overwrite data, as appropriate (although it will not overwrite filled fields with empty fields)

## BASIC IMPORTS

\*The importer works in a two-step process. Most imports are straightforward additions of people to an organization's supporter list, or perhaps updates of existing supporters.

\*On the first page, under Your List -> Import Supporter List, you'll do three things:

1. Browse for a file to import (or, for smaller files no larger than a few hundred, just copy and paste directly into the big white box). Note that the file must be a delimited data file; if you're working in Excel or another spreadsheet application, you'll need to Save As. See the instructions at the top of the import page.
2. Tell the system what character delimits the file (that is, designates the break between fields). For instance, a .csv comma-separated file is delimited by commas.
  - 2a. Optionally, you can also tell it that this file resembles a previous import. This is not necessary, but might save you time in the next step if you import similar files frequently.
3. Hit Submit

\*On the second page, you'll get a vertical list of all the fields in the top line of your file, next to pull-down menus of fields in the DemocracyInAction system to which you can import (including your custom fields). The system will try to guess which of your fields go with which of ours, but you'll almost certainly need to go through the menus to instruct it yourself. (Hint: watch out for the system guessing “Last Created” on a field called “Last Name”) You can also tell the system to Ignore any fields you don't need to import.

\*To import to Boolean (checkbox) fields, you'll need to use 1 for checked/true, and 0 for unchecked/false. To import to multiple-choice custom fields, you'll need to use the number 0,1,2,3... instead of the names to which you've assigned these choices. To import to DIA pull-down menu fields such as Title, you'll need to use the *exact* names that DemocracyInAction uses.

\*New records will by default have their Receive Email box checked (value=1), which means the new record will be an eligible e-mail recipient. However, you can import to this field like any other, so you can instruct the system otherwise. An import that updates a record whose Receive Email box is already unchecked will *not* re-check that box unless you're explicitly importing to that field and the import file has a value of 1 in the field you're matching with Receive Email.

\*If page two of the import process is garbled, the system either didn't understand your import file, or you used the wrong delimiter. Go back to step one and start again.

### **SOME ADDITIONAL OPTIONS**

\*Optionally, you can also have everyone in this import assigned to a Group by using the menu immediately below the field-matching section – a handy way to flag people for future reference. You might also do this in cases where creating a group is the primary purpose of the import: importing a file consisting of nothing but e-mail addresses where all of those addresses are already in your database, for instance, while using the groups feature would automatically assign all those people to a readily queryable group.

\*The importer allows you to control the default overwriting behavior using the advanced options further down the page

1. Disable deduplication altogether
2. Designate a different field to deduplicate on (not recommended unless you're sure you know what you're doing)
3. Tell the system not to treat new imports as newer data
4. Instruct the system to overwrite filled fields with empty fields in cases where an imported file has blank values.

\*You may also want to tell the system to skip 1 line, if the top line of your file contains generic field names, if you want to avoid creating a record whose e-mail address is E-mail and first name is Fname.

\*To the right of the field-matching area you can designate “Replacement Options” which allows you to substitute values in your import file on the fly. We recommend using find-and-replace in a local application instead, so you can be certain what you're getting. However, if you have a file with some people who are marked as “Subscribed” and others marked “Unsubscribed”, you could use Replacement Options to translate those values into 1 and 0 respectively for import to the Receive Email field using the syntax outlined on the web page.

### **IMPORTING DONATIONS**

\*You can use the importer to populate a supporter's history of donations. In this case, you'll use a single file with one row per donation, each row containing one of the three variables the system uses to match identity (Supporter\_KEY, uid or e-mail) and then the appropriate transaction information. For instance:

a@b.com	2006-02-14	20.00
a@b.com	2006-08-22	35.00
c@d.com	2005-12-29	100.00

... would add two donations to a@b.com's donor history dated February 14, 2006 and August 22, 2006, and one to c@d.com's record dated December 29, 2005.

\*Your fields in such an import must *exactly* match the naming conventions and formats that DIA fields use. Notably, dollar amounts should be XXXX.XX (no currency symbols or commas) and dates should be YYYY-MM-DD. You should be able to convert to these formats in a spreadsheet if your data does not arrive that way by default.

## **CREATING MULTIPLE GROUPS**

\*If you have a large number of flags or other variables that you'll need to translate into DemocracyInAction, you can handle that two different ways.

\*The slower, systematic way is to populate each group into its own column using boolean (1 or 0) values to designate membership – and create each group under Your List -> Manage Groups. You can then select the specific group to match each column.

\*You can also create these groups dynamically *through the importer itself*. In this case, you'll want a tab-delimited file (or any non-comma-delimited format) with one field that's a comma-delimited sequence of Group identities such as:

Board, Major Donor, Banquet Committee

The system will automatically create groups to match these names (excluding the commas) as it imports and populate people into these groups. As long as the names are exactly consistent (non-case sensitive), everyone will be added to the same group, so these records:

a@b.com	Board, Major Donor, Banquet Committee
c@d.com	Board, Banquet Committee, Volunteer

Will create four groups: Board (2 members), Banquet Committee (2 members), Major Donor (1 member) and Volunteer (1 member)

## **BEYOND THE SUPPORTER TABLE**

\*You can actually import entries to **any** DemocracyInAction table, not just the supporter table. “Recipients of Actions” is provided by default in the first step of the process; this menu allows you to import potential custom recipients that you can use in your campaigns.

\*However, to import to another table not listed here, simply enter it in the “write-in table” box. If you properly name the table (and you can see the tables you're in throughout the system as you navigate by watching your url), the second step of the importer will show you a pull-down menu of fields in that custom table. *You'll get the best use of this if you understand how relational databases work in general, and DemocracyInAction's in particular. This option is only recommended for advanced users.*

*Exemplar: Importing Petition Signers or Event Attendees*

\*One way to use this write-in feature is to import **petition signers** or **event attendees**. The step-by-step process below outlines the procedure for petitions, but it can be tweaked for events as well.

Overview. We have a table called supporter\_petition that establishes this connection. The key fields there are the supporter\_key number and the petition\_key number. So you need to get your Excel file of petition signers into a format like this. Here's how:

1. Import your file just into the supporter list, putting them all in a group as you do it. Plain-boring import here.
2. Turn around and export this group by querying the group and exporting it. The purpose of these first two steps is to get a supporter\_key number in your file, so you'll need to make sure you choose to include it. (you can, actually, export nothing but the supporter\_key).
3. Get the petition\_key number from the Petitions section (or from the user-facing page) -- you can see it in the url.
4. Open up your export file in Excel (or whatever), create a new column, and mass-paste the petition key number into it for all the records you've exported. So you should now have a file something like this:

supporter\_key,petition\_key

12345,999  
24680,999  
36912,999

(etc.) ... a different supporter key for each line, the same petition key for each line.

5. Save this file as a comma- or tab-delimited file.
6. Your List -> Import Supporter List. Browse for the file you've just saved and designate the delimiting character, just like normal. Only for this, use the "write-in table" (on the right side of the screen, below the pull-down menu for the delimiter) supporter\_petition.
7. Submit, and on to step two.
8. In step two, you should have many new and exotic choices. You'll obviously want to match the supporter\_key to supporter\_key and petition\_key to petition\_key. That's really all that's needed, but you can use as many fields as you like -- and if you're displaying signatures in a user-facing version on your petitions, you'll probably want to include names and (where possible) comments at the very least. You can put those into the data file as well, of course (that probably means you'll be importing them and then re-exporting them along with the rest - comments may require a custom blob field). (Be careful with comments, in case the comment bloc itself includes commas, line breaks or tabs that might confuse the importer.)