



Whitepaper

Last Updated: Unknown?!?

*How LISTSERV® Maestro Can
Help You Keep Your
Subscribers' Data Current*

August 2, 2010

Copyright © 2010 L-Soft international, Inc.

Information in this document is subject to change without notice. Companies, names, and data used for example herein are fictitious unless otherwise noted. Some screen captures have been cropped and/or edited for emphasis or descriptive purposes.

Permission is granted to copy this document, at no charge and in its entirety, if the copies are not used for commercial advantage, the source is cited, and the present copyright notice is included in all copies. Recipients of such copies are equally bound to abide by the present conditions. Prior written permission is required for any commercial use of this document, in whole or in part, and for any partial reproduction of the contents of this document exceeding 50 lines of up to 80 characters, or equivalent.

L-Soft invites comments on its documentation. Please feel free to send your comments by email to: manuals@lsoft.com

Copyright © 2010, L-Soft international, Inc.

All Rights Reserved Worldwide.

LISTSERV is a registered trademark licensed to L-Soft Sweden and L-Soft international, Inc.

All other trademarks, both marked and not marked, are the property of their respective owners.

Introduction

The purpose of this white paper is to describe how you can use LISTSERV Maestro to create and manage accurate “last updated” fields for your subscribers’ data. With this field, you’ll be able to see how current your subscribers’ data is. In addition, when you want to remind your subscribers to update their data, this white paper will also show you how to send and customize messages that will allow your subscribers to easily update their data without actually logging into the Membership Area.

Creating a “Timestamp” Profile Field

For many data administrators, keeping track of their subscribers’ data and its accuracy is a major concern. Simply having a way to keep track of the last date and time that a specific record was updated would be beneficiary. Because of this need, LISTSERV Maestro 4.0-5 and later has made it possible for you to create a “timestamp” that helps you keep your subscribers information up-to-date. This “timestamp” is created by adding a special derivation rule formula to a derived profile field.

To define a "timestamp" profile field for a specific dataset, go to the Recipient Dataset Details screen for that dataset, and then select the **Dataset** menu from the Toolbar and select **Edit Dataset Settings**. The Recipient Dataset Definition wizard opens. On the Profile Fields screen, click the **Add Field** link. Enter a suitable **Name** and **Display Name**, for example “CHANGESTAMP” and “Last Updated”. Click the **Data Type** drop-down arrow and select **Derived**, and then click the **Input Type** drop-down arrow and select **Hidden**. (This input type ensures that subscribers won't see this change timestamp on their pages). Click **[Next]** to define the derivation rule.

Recipient Dataset Definition

General Profile Fields Profile Field Details Summary

Cancel Save & Exit <- Back Next ->

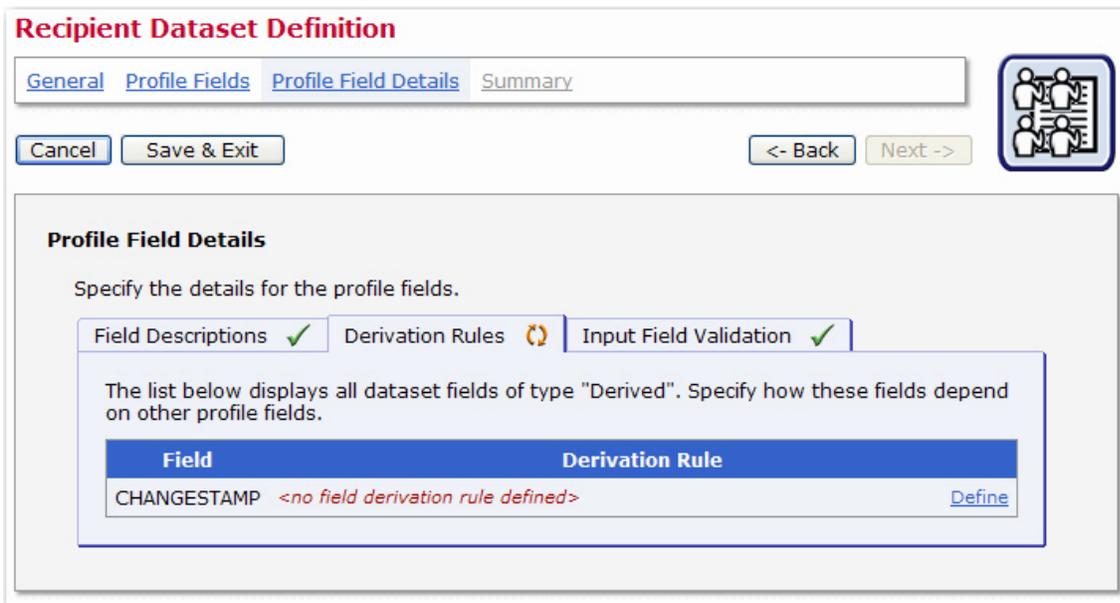
Shared Dataset Fields

Specify the shared dataset fields. These fields are shared among all mailing lists in this dataset. Click "Add Field" to add a new field to the list. Click on the links next to a field to select it for editing or to remove the field from the list.

Name	Display Name	Data Type	Input Type
EMAIL	E-Mail Address	Text	Mandatory
FNAME	First Name	Text	Optional Edit Remove
LNAME	Last Name	Text	Optional Edit Remove
CHANGESTAMP	Last Updated	Derived	Hidden Reset Up Down Remove

[Add Field](#)

On the Profile Field Details screen, click the Derivation Rules tab. For the newly added CHANGESTAMP field, click the **Define** link.



In the Derivation Rule window, set the derivation rule of this new field to the following formula for the derivation rule:

```
If(true, CurrentMillis, Length(FIELD_LIST))
```

where you replace `FIELD_LIST` with a "+" concatenation of all other profile fields in your dataset (or hosted list), like this:

```
&FIELD1; + &FIELD2; + &FIELD3; + [etc]...
```

However, you need to observe a few rules depending on each field's type so that this concatenation list results in a correct formula syntax and is accepted by the system:

- **Text and Mandatory field types** – There is nothing special to observe, just add the field to the list, i.e. write `&FIELD;`
- **Text and Optional field types** – Use the optional-field syntax to add the field, i.e. write `[&FIELD; ""]`
- **Number and Mandatory field types** – There is nothing special to observe, just add the field to the list, i.e. write `&FIELD;`
- **Number and Optional field types** – Use the optional-field syntax to add the field, i.e. write `[&FIELD; 0]`
- **Boolean field types** – Wrap the field in the `ToNum()` function, i.e. write `ToNum(&FIELD;)`
- **Tracking Permission field types** – See “Boolean field types”
- **Single Select (Mandatory or Optional) field types** – There is nothing special to observe, just add the field to the list, i.e. write `&FIELD;`
- **Multiple Select (Mandatory or Optional) field types** – Wrap the field in the `Count()` function, i.e. write `Count(&FIELD;)`
- **Derived field types** – This field does not need to be mentioned in the formula at all, so simply leave it out.

Example

Assume that you have the following profile fields:

TEXTOPT	(optional text)
TEXTMAND	(mandatory text)
NUMOPT	(optional number)
NUMMAND	(mandatory number)
BOOL	(boolean)
SINGLE	(single select)
MULTI	(multiple select)

then, the derivation formula would look like this:

```
If(true, CurrentMillis, Length([&TEXTOPT; "" ] + &TEXTMAND; + [&NUMOPT; 0] +  
&NUMMAND; + ToNum(&BOOL;) + &SINGLE; + Count(&MULTI;)))
```

or, the same with linebreaks for better readability:

```
If(true,  
CurrentMillis,  
Length([&TEXTOPT; "" ] +  
    &TEXTMAND; +  
    [&NUMOPT; 0] +  
    &NUMMAND; +  
    ToNum(&BOOL;) +  
    &SINGLE; +  
    Count(&MULTI;)  
    )  
)
```

Restriction: It is not possible to have a derived field in a hosted list that references (in its formula) profile fields that are defined in the dataset. Similarly, it is not possible to have a derived field in a dataset that references (in its formula) profile fields that are defined in any of the hosted lists of the dataset.

Therefore, if you have both profile fields in the dataset (i.e. shared fields for all lists) and individual profile fields in a hosted list, then you can **not** have a change-timestamp for all of these profile fields (from dataset and list) at once.

Your only option is to have two change-timestamps: One for the dataset profile fields (defined as a derived field in the dataset, referencing all other dataset profile fields, but no list profile fields), and one for the list profile fields (defined as a derived field in the list, referencing all other list profile fields, but no dataset profile fields).

Important Notes about the Derivation Rule Formula

You need to know that a derived field's value is recalculated whenever the value of any of the other profile fields that are mentioned in the derivation rule formula changes. Therefore, what we need to come up with is a formula, whose result is the current time (so that it can be used as a timestamp), but which is also recalculated whenever any of the (other) profile fields are changed. Because of this, the formula must not only result in the current time, but it also must mention all the other profile fields somewhere.

This is what the above formula does, using the "If" function:

1. The first argument of the function is the condition. In this case, the condition is set to a constant "true".
2. The second argument of the function is the result that is to be returned if the condition is true. In this case, since the condition is always "true" (see above), the value of the

second argument is always the value that is returned by the function. Therefore, we simply set "CurrentMillis" as the second argument, which will have the effect that the second argument is equal to the current time; and, since the second argument will always be returned as the result of the If-function itself (because of the constant "true" condition), the result of the whole If-function (and thus the whole derivation rule formula) will be the current time, just as we need it for our timestamp.

3. The third argument of the function is the result that is to be returned if the condition is false. In this case, since the condition is always "true" (see above), this means that the value of this second argument will **never** be returned. So we are free to put into this third argument whatever we want (as long as it is syntactically valid and results in a Number). We take advantage of this (i.e. that the third argument is always ignored). As said above, to make the derivation rule recalculate its value (to the current time) each time any of the other profile fields are changed, we need to mention all of these other profile fields somewhere in the formula, but we don't want them to actually influence the return value of the formula (which is supposed to be the current time). And, since the third argument is ignored, we can use it for exactly this purpose, i.e. to mention all the other profile fields without influencing the result of the formula, which is the purpose of the "FIELD_LIST".

Note: Notice how we put an additional "Length()" function around the "FIELD_LIST". This ensures that the data type of the value from "FIELD_LIST" is of the Number type to match the data type of "CurrentMillis", which is also Number. This is because the "FIELD_LIST" will be a concatenation of all other profile fields (to "mention" them in the formula); and, the data type of such a concatenation is usually Text, which is why we put the "Length()" function around it to convert the Text type back into a Number type.

Requesting Profile Field Updates

When you want to remind your subscribers to update their data, LISTSERV Maestro 4.0 and later will let you create a message with a special system drop-in that will direct your subscribers to the External Profile Edit Page. This page will let your subscribers easily update specific profile fields without logging into the Membership Area.

Using the ProfileEditPageURL System Drop-In

This drop-in is called `ProfileEditPageURL` and, if included in the mail body, it is replaced with a URL to the External Profile Edit Page.

The drop-in must be written in a special form because when you write this drop-in into your message you will need to include information about which profile fields will be queried from the subscriber. The syntax is as follows:

```
{ { *ProfileEditPageURL FIELDNAME_LIST } }
```

where `FIELDNAME_LIST` must be replaced with a comma-separated list of the names of all profile fields that the External Profile Edit Page will query from the subscriber.

Examples:

- `{ { *ProfileEditPageURL HOBBY } }`

will be replaced with an URL that points to an External Profile Edit Page that asks the subscriber to submit a value for the HOBBY profile field.

- `{ { *ProfileEditPageURL STREET, ZIP_CODE, CITY } }`

will be replaced with an URL that points to an External Profile Edit Page that will ask the subscriber to submit values for the STREET, ZIP_CODE, and CITY profile fields.

Customizing the External Profile Edit Page

The External Profile Edit Page is a dynamic page that queries certain profile fields from the subscriber. The fields that are queried by the page during a visit depends on the parameters in the URL by which the page was accessed. And, the parameters in the URL in turn are created by the ProfileEditPageURL system drop-in depending on the field names that you include in the FIELDNAME_LIST of that system drop-in.

Since the External Profile Edit Page is such a dynamic page, there are several features specifically designed for it in the Subscriber Interface:

1. If you select this page for customization, then at the bottom of the screen you see several checkboxes – one checkbox for each available profile field. If you check/uncheck the checkboxes, and then click [Apply], you will then see a preview of this page with exactly these profile fields visible. This lets you simulate the look and feel of this page for each desired profile field combination.

With the examples of above: If you are planning to use `{{*ProfileEditPageURL STREET, ZIP_CODE, CITY}}` to query the address data of the subscriber, and you first want to check that the page will look correctly if these three profile fields are displayed, then go to the External Profile Edit Page, check the boxes for STREET, ZIP_CODE, and CITY, uncheck all other boxes, and then click [Apply]. The preview will then show the page just as it will look to subscribers if you use the system drop-in with STREET, ZIP_CODE, and CITY.

The screenshot shows a web-based configuration interface for the External Profile Edit Page. At the top, there are two tabs: "Draft Version" and "Production Version". Below the tabs, it says "Header and Footer: Inherited." with a small icon. The main content area is a preview of the page, showing a blue header bar, a title "Edit Profile For '[List name goes here]'", and a prompt "Please supply your information below." followed by three input fields: "Street Address:", "Zip Code:", and "City:". Below these fields are "OK" and "Cancel" buttons. At the bottom of the preview area is a scrollbar. Below the preview, there is a dropdown menu labeled "Preview page with profile fields of:" with "The dataset" selected. Below this are several checkboxes for profile fields: TRACK, FNAME, LNAME, AGE, COUNTY, STREET, ZIP_CODE, and CITY. The checkboxes for STREET, ZIP_CODE, and CITY are checked. An "Apply" button is located to the right of the checkboxes.

2. If the `{{*ProfileEditPageURL FIELDNAME_LIST}}` system drop-in is used in a mailing to a dataset (see Section 1.10 Sending Messages to an Entire Dataset), then `FIELDNAME_LIST` must only contain profile fields that are fields of the dataset. It can not contain any fields that belong to any of the HRLs or HLLs in the dataset.

However, if the mailing goes to a HRL or HLL, then `FIELDNAME_LIST` may contain all profile fields that are either in the HRL/HLL itself or in the dataset that contains the HRL/HLL.

The Subscriber Interface for the External Profile Edit Page reflects this: just above the previously mentioned checkboxes, you have a Preview page with profile fields of drop-down menu that allows you to select which profile fields you want to see. If you select The Dataset, then you will only see checkboxes for the dataset profile fields. This simulates the situation when you use the system drop-in in a mailing to the dataset. If you select The "LIST", then you will see all checkboxes for the list XYZ plus all of the dataset. This simulates the situation when you use the system drop-in in a mailing to list XYZ where you can use all combined profile fields from the list and dataset in `FIELDNAME_LIST`.

References

LISTSERV Maestro User's Manual

<http://www.lsoft.com/resources/manuals.asp>