

Printed Forms Control using Variables and JavaScript

You may have designed forms in myEvolv that are intended to be printed from the system. For example, we have a couple of face sheets that we routinely print from myEvolv and keep in paper form in case Evolv is not accessible. One of the face sheets contains critical health information that we print and share with emergency medical staff or hospitals if a client goes to the emergency room. The face sheets are great in myEvolv because every time they are accessed, they are updated to pull in the most up-to-date information. Once the information is printed, it can become outdated and it can be difficult to tell how old the data may be.

We use two variable fields on the face sheets that record which user printed the face sheet and on what date so that as the paper version are floating around in the facilities, we are able to determine how stale the information is and destroy any copies that are deemed too old.

DOB	Age	Gender
3/1/2002	17	Female

Current Location
Facility: Raise the Age Facility (RTA101) - 1550 Champlin Ave., UTICA, NY 13502
Foster Home Placement

Why variables?

Variables do not save data to the database. They only display data on the forms, so they are ideal for this application.

Other than that, variables work more or less the same as a form field so you can apply use the same JavaScript as you would on a form field without needing to make adjustments. If you find that you want to store the data in the database, you can. Just use a field instead of a variable.

Form Setup

Add two variables in the top Group on the form: **Generated On** & **Generated By**

Preview Form: **Client Information Sheet**

Personal Information		
0	Generated On	Variable
1	Generated By	Variable
2	Service Track	Variable
3		Report

Generated On

Properties

Variable

Variable

Caption	Generated On
Order In Group	0
Variable Name	generated_dt
Data Type	DateTime[datetime] <input type="button" value="..."/>
Instructions	<input type="button" value="..."/>
Display Size	
Max Size Allowed	
Is Add Only	<input type="checkbox"/>
Is Visible	<input checked="" type="checkbox"/>
Is Visible on Subform	<input type="checkbox"/>
Is Listed	<input type="checkbox"/>
Is Modifiable	<input type="checkbox"/>
Required	<input type="checkbox"/>

Generated By

Properties

Variable

Variable

Caption	Generated By
Order In Group	1
Variable Name	generated_by
Data Type	Regular String[varchar(50)] <input type="button" value="..."/>
Instructions	<input type="button" value="..."/>
Display Size	25
Max Size Allowed	
Is Add Only	<input type="checkbox"/>
Is Visible	<input checked="" type="checkbox"/>
Is Visible on Subform	<input type="checkbox"/>
Is Listed	<input type="checkbox"/>
Is Modifiable	<input type="checkbox"/>
Required	<input type="checkbox"/>

Generated On is a DateTime variable and **Generated By** is a Regular String variable.

I make them not-modifiable so that users cannot change their value before printing to fake the data.

I also found that a display size of 25 for the **Generated By** field just about matches the display width of a DateTime field so it looks nicer.

JavaScript

Generated On – Default Value

This one is easy and probably one that you have already used or seen used elsewhere.

```
todayDateTime();
```

This is a JavaScript function written by Netsmart that default's the current Date/Time into a field already properly formatted for the field.

If your field is a Date Only type or displayed as Date Only, you can use this instead:

```
todayDate();
```

Generated By – Default Value

Another single line of code for this variable but it is a very useful Netsmart function that I use all the time.

```
getDataValue('staff_view', 'staff_id', workerID, 'staff_name');
```

1 2 3 4

The `getDataValue()` function looks up a record in a table or view (1) where the table or view's column (2) value is equal to the provided value (3) and returns the value from the other column (4) from that record.

In other words, this function is looking in the `staff_view`, which is a system view with data about staff and gets the row from that view where `staff_view.staff_id = workerID` and returns the value of `staff_view.staff_name` from that record.

`workerID` is a JavaScript session variable that is already set by myEvolv for you to use and it contains the GUID of the logged in user. So when the code is executed while you are logged in, it should get your name from the `staff_view`. If someone else is logged in and the code executes, it should pull in that user's name.