Initializing Forms with Data

On this page:

Default field values

A form designed with all the form fields blank (empty with no values) in the Form Designer, will load with no default field values. There are several ways to populate your form with default values to assist the user in filling in your form.

In order of precedence, as the form loads (test/use) form fields are potentially populated with default values:

1. Default values entered when the form was created in the Form Designer
2. Values from an XML document(s) sent in the Post request for the form or OnInit.
3. URL parameters appended to the form URL via the Live Forms_data URL parameter
4. Values retrieved as the form loads from any Document URIs
5. And finally default values set via Business Rules as the form loads and during use. See Setting default values in rules and using REST Services in Rules for further details.

_data

Click here for a quick refresher about adding URL parameters

If you are appending parameters to a URL, use the **?** for separating the form/flow URL from the parameters and use the **&** to separate parameters from each other. For example, let's say you want to append the Pacific timezone parameter to the share URL of your form/flow.

Here is an example using the Link(Email/Web page) URL for a workflow:

Add ?_formTz=America/Los_Angeles to the URL. Use the ? since there are no existing parameters appended.

```
https://app.frevvo.com:443/frevvo/web/tn/mycompany/user/max/app/_N2Z-EN4ZEeSMHtcWMmy
rrQ/flowtype/_Wdk4AecIEeSyBMGi6J8M3Q/popupform?_formTz=America/Los_Angeles
```

Here is an example using the Raw Flow URL for the same workflow:

Add &_formTz=America/Los_Angeles after the last existing paremeter. Use the & since you are separating parameters from each other.

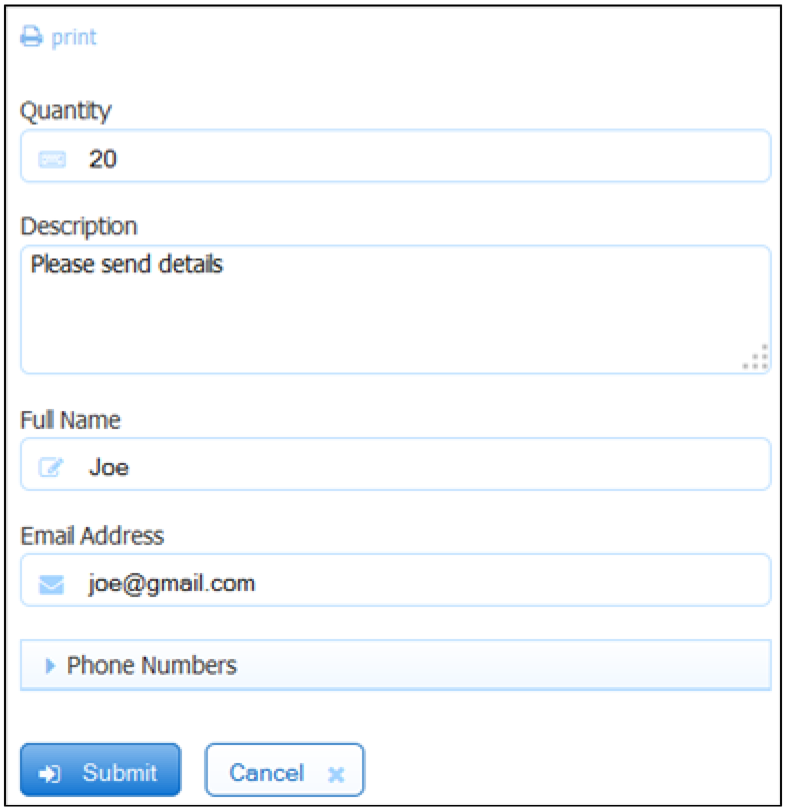
```
https://app.frevvo.com:443/frevvo/web/tn/mycompany/user/max/app/_N2Z-EN4ZEeSMHtcWMmy
rrQ/flowtype/_Wdk4AecIEeSyBMGi6J8M3Q?_method=post&embed=true&_formTz=America/Los_Ang
eles
```

Form fields can be initialized directly from the form's HTTP URL. You can do this by using the special (reserverd) URL parameter called _data. Below is an example of a form initialized via _data. The syntax for _data is based on Rison. This form was initialized by appending the following to the URL from the form's Share Email/Web Page form link..

```
?_data=(EMail2787:'joe@gmail.com',Name:'Joe',Desc:'Please send details',Q:'20')
```

While not mandatory in all cases it is highly recommended that you enclose the values in single quotes. Thus Name:Joe should be added to the URL instead as Name:'Joe'. Also certain characters may need to be URL encoded. See quoting values below for details.

Also note that the only supported format for dates is the XSD schema date format; you must use YYYY-MM-DD.



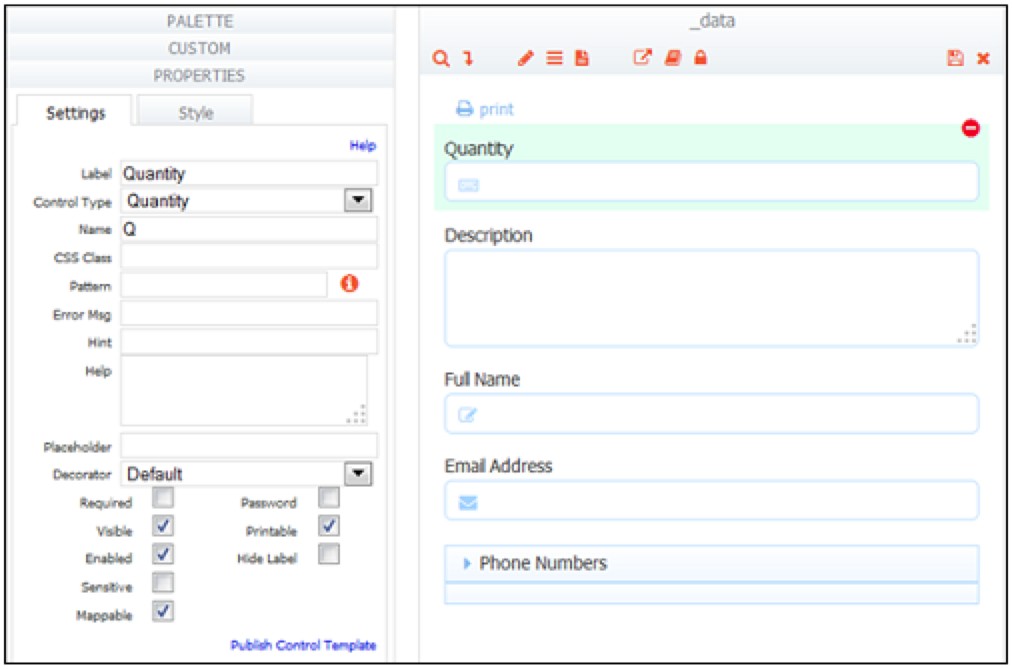
Quantity
20

Description
Please send details

Full Name
Joe

Email Address
joe@gmail.com

The form fields are addressed in the _data parameter by their Name property if the control was added from the palette and by the XSD element name if the control was added from data sources. In the example above the form has 4 controls. The control labeled Email Address has the name EMmail2787. The controls labeled Full Name has the name 'Name'. Description is named Desc and Quantity is named Q. See the Form Designer below with the Quantity control selected and you can see that it's name in the Properties Settings is 'Q'.



Label	Quantity
Control Type	Quantity
Name	Q
CSS Class	
Pattern	
Error Msg	
Hint	
Help	
Placeholder	
Decorator	Default
Required	☐
Visible	✓
Enabled	✓
Sensitive	☐
Mappable	✓
Password	☐
Printable	✓
Hide Label	☐

Here are some examples:

- ?_data=(EMail2787:'joe@gmail.com') sets the value of the control named EMail2787 to joe@gmail.com
- ?_data=(A:'good',B:'bad') sets the controls named A and B to good and bad respectively

If you append the _data parameters to the form/flow raw link or if you are using the Embedded form (script) to share your form the "?" does not work. Use the "&" character,instead of the "?" to append the _data parameters.

URL Template Variables

One common use for _data is in the Form Action Post and the Go to Url, and in the Doc Action Post. You can use Templatized Strings to set the Url parameter values dynamically from the form field values. One example: imagine your form contains a field named 'custName'. &_data=(fname:'{custName}') sets the value of fname to the value in the form field named custName. The value in the form field custName is replace when this _data is interpreted. If the post is to a form Url and that next form has a field named 'fname' that field will automatically be initialized to the value of 'custName'.

Quoting values

Values such as 02 are converted to numbers and would actually become a 2 rather than 02. If your form does require a 02 rather than a 2 you can quote the value to prevent this conversion. Thus &_data=(customerId:'02') would set the value of the control named customerId to 02. Whereas if you used &_data=(customerId:02) then the value in customerId would be set to 2.

We highly recommend quoting all values, and it is even more important when using templates because you may not have control over the values users enter into your fields. Thus &_data=(customerId:'{Id}') is preferred over &_data=(customerId:{Id}).

If you are using _data to pass values entered into form 1 to initialize fields in form 2 AND the fields in form 1 are not required (meaning when the form is submitted the fields may not contain values), you **must** quote the templates. For example if Id is **not** a required field then this &_data=(customerId:{Id}) will fail when the field has no value. To prevent this failure always use quotes.

Certain characters such as % need to be URL encoded. The w3schools.com website describes in detail the reason certain characters must be encoded when used in Urls. And it also contains a tool that lets you enter your string and converts it to a properly encoded string. For example if your value is '% complete' this must be written as '%25 complete'.

The single and double quote characters themselves must be escaped with the Live Forms escape character '!'. For example the string 'Nancy's Horse' must be written as 'Nancy!'s Horse'. If the '!' is missing from the string the apostrophe will be mistaken as the end of string character.

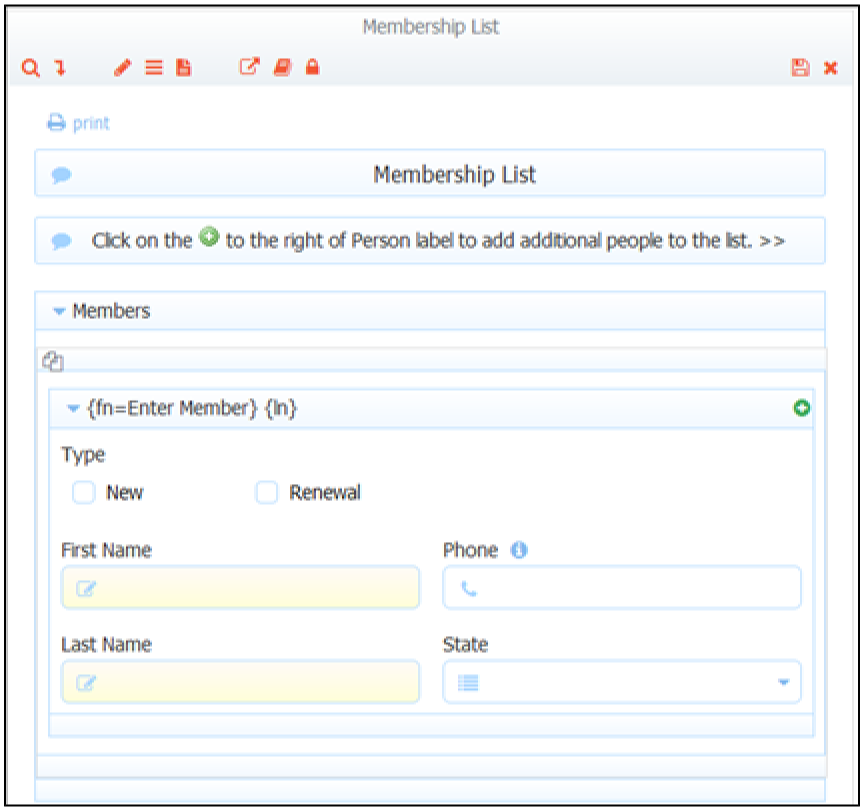
Repeat Controls

Initializing repeats requires a special syntax. :!(<value1>,<value2>,etc..!). In the current release Url Templates variables are not supported for repeats.

- &_data=(A:!(one,two!)) expects to find two controls named A (a repeat) and sets them to one and two.

If _data contains more instance of the repeating item then on the form by default, then it will create new items as the form is loading, as if the user had clicked the "+" button.

This membership list form contain a single default empty member item. The controls First Name and Last Name are named fn and ln respectively.


Membership List

print

Membership List

Click on the to the right of Person label to add additional people to the list. >>

Members

{fn=Enter Member} {ln}

Type

New Renewal

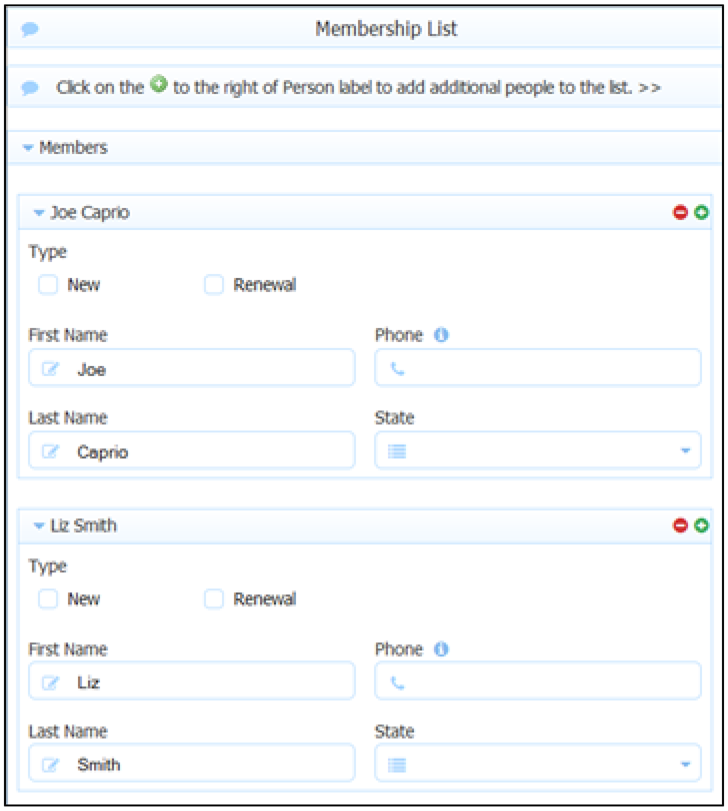
First Name

Phone

Last Name

State

Appending _data=(fn:!(Joe,Liz!),ln:!(Caprio,Smith!) to the form's share URL will cause the form to automatically add a 2nd repeating member and to initialize the first and last names for both members. This form also contains a templatized label in the section control. Notice how the templates are also replaced with the _data values when the form loads.


Membership List

Click on the to the right of Person label to add additional people to the list. >>

Members

Joe Caprio

Type

New Renewal

First Name: Joe

Phone

Last Name: Caprio

State

Liz Smith

Type

New Renewal

First Name: Liz

Phone

Last Name: Smith

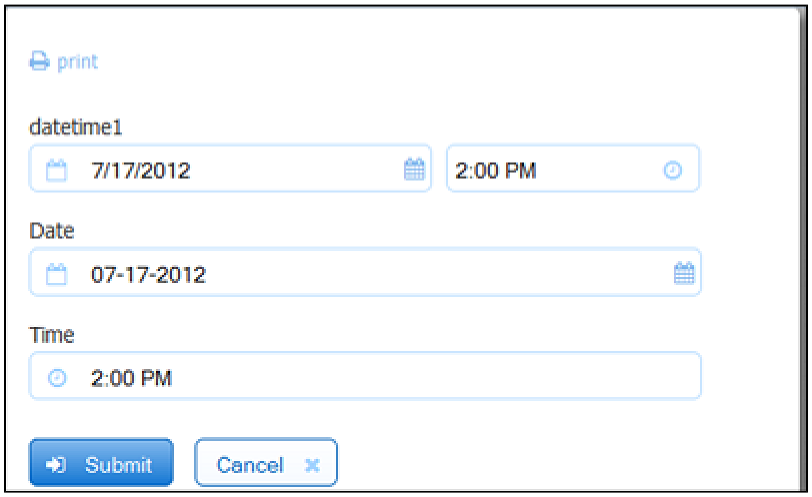
State

Time, Date/Time Controls

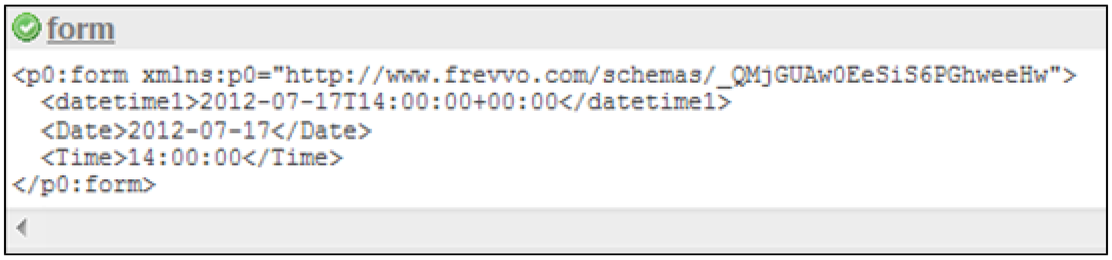
You **must specify the _formTz=<tz>** parameter appended to the share URL when initializing a form with date/time controls from an XML init document, the _data parameter or a business rule that uses the rule identifier,form.load. This is because the form server needs to know the timezone in which to return the date and time. Successful initialization cannot be guaranteed if the &_formTz parameter is not supplied. This URL parameter is not needed if your form/flow only contains date or time controls.

Here is an example of a url that initializes a date only, time only and date/time control. Notice the appended &_formTz parameter supplying the timezone, America/New York.

```
http://localhost:8082/frevvo/web/tn/doctenant/user/designer/app/_2CnmgAwnEeSqSNaJd2kfF
Q/formtype/_QMjGUAw0EeSiS6PGhweeHw/popupform?_data=Date:'2012-07-17',Time:'14:00:00',d
atetime1:'2012-07-17T14:00:00'&_formTz=America/NewYork
```



The local time will be converted to and saved in a valid UTC format for the time portion of the date/time control in the submission XML document. You may see the time stored with a trailing "Z" or with an offset from UTC time. See Initializing Forms with XML Documents and Viewing XML Documents for further information.


form

```
<p0:form xmlns:p0="http://www.frevvo.com/schemas/_QMjGUAw0EeSiS6PGhweeHw">
  <datetime1>2012-07-17T14:00:00+00:00</datetime1>
  <Date>2012-07-17</Date>
  <Time>14:00:00</Time>
</p0:form>
```

Checkbox Controls

In the current release selecting multiple check box options is not supported via _data. Also when using Url templates only the 1st item from the named template will used.

Unbound Controls

Variables passed into your form via _data need not be bound to actual form fields. Since _data is used as a way to initialize form fields why would you want to do this?

One place this is very useful is in business rules. See the section on templatized URls in business rules for more details. However anywhere that you can use templates in your form: labels, help, hint; display message, etc... the value can either come from the value in an actual form field or from _data values that are not in any way bound to the name of a control.

onInit/onSubmit

The onInit feature allows you to populate your embedded frevvo form with the data from your own HTML form. So if you want to pass some data from your web page to the frevvo form embedded into that same web page, you will use onInit. onInit function will execute when the frevvo form loads.

The onSubmit function allows you to copy the data from the frevvo form (when that frevvo form is submitted) into your HTML form. Then you can use your own JavaScript on your HTML page to process that data (for example save it somewhere else at your own convenience and time, etc.)

The embedding page and Live Forms must be in the the same domain/origin.Typically this means same uri scheme, host and port.

The value of onInit must be an in scope javascript function. This onInit javascript function will be called before the form is instantiated. For example &onInit=getData will cause Live Forms to call the getData() javascript function. The onSubmit function will be called after the submit button is clicked.

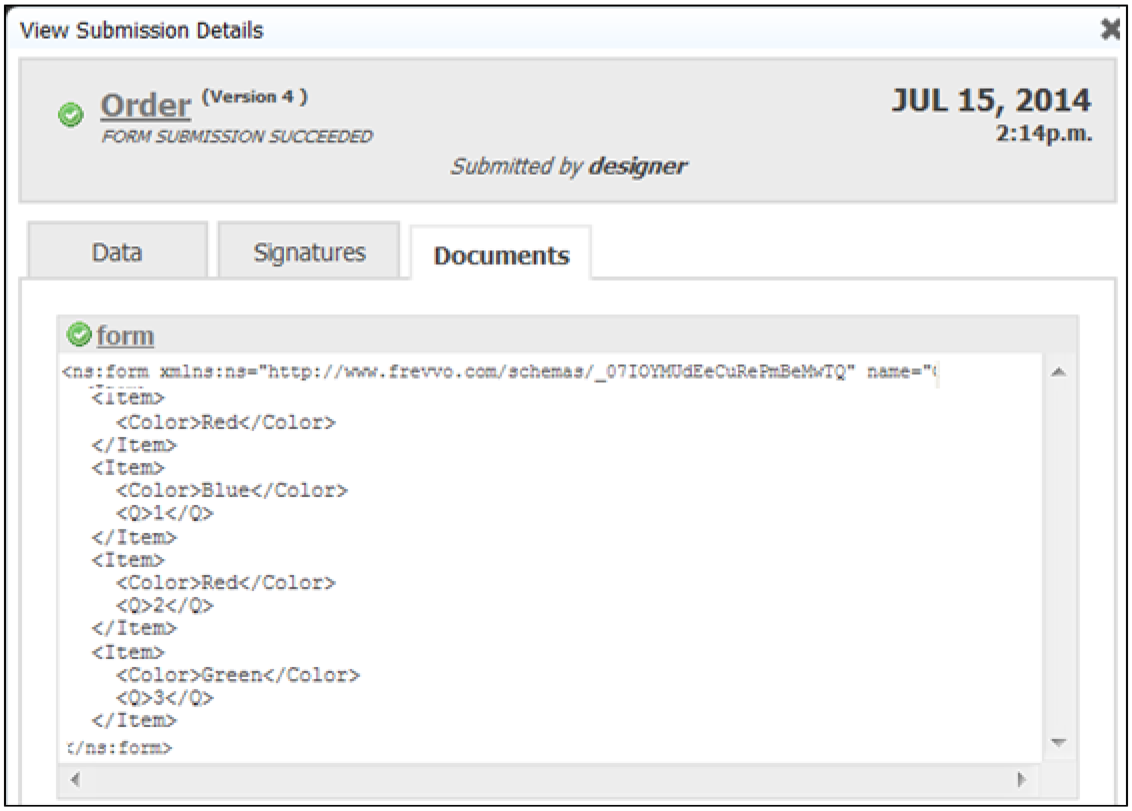
Here is a sample html page an embedded Live Forms script tag and an onInit javascript function. The Live Forms script tag copied was from the S hare dialog and the &onInit Url parameter was appended. Each xml document is placed inside a container element (input, textarea or any other element such as a div) and assigned and xml id. In this example there is a single xml document placed inside a <textarea> tag.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
        <head>
                <script language="JavaScript" type="text/javascript">
                        /**
                                Called before the form is rendered in the iframe
                                Return - docs - [{
                                                xml: *
                                        }]
                        **/
                        function getData(){
                                alert('onInit called');
                                var textarea = document.getElementById('form');
                                return [{
                                        xml: textarea.value
                                }];
                        }
                        /**
                                Called after the form is submitted
                                Parameters:
                                        docs - [{
                                                element:{
                                                        name: *,
                                                        namespace: *
                                                        },
                                                        xml: *
                                        }]
                        **/
                        function onSubmit(docs){
                                alert('onSubmit called');
                                var textarea = document.getElementById('form');
                                if( textarea )
                                        textarea.value = docs[0].xml;
                        }
            </script>
        </head>
        <body>
         <textarea id="form" name="doc1" rows="4" cols="100">
             <ns:form xmlns:ns="http://www.frevvo.com/schemas/_07IOYMUdEeCuRePmBeMwTQ"
name="Order">
               <Item>
                    <Color>Red</Color>
              </Item>
               <Item> <Color>Blue</Color>
              </Item>
               <Item> <Color>Red</Color>
              </Item>
               <Item> <Color>Green</Color>
              </Item>
            </ns:form>
        </textarea>

       <script xmlns=http://www.w3.org/1999/xhtml

src="http://localhost:8082/frevvo/web/tn/nancy.com/user/designer/app/_yVkhYcUdEeCuRePm
BeMwTQ/
formtype/_07IOYMUdEeCuRePmBeMwTQ/embed?container=true&resize=true&onInit=getData"
type="text/javascript" language="Javascript">
```

```
        </script>
       </body>
</html>
```

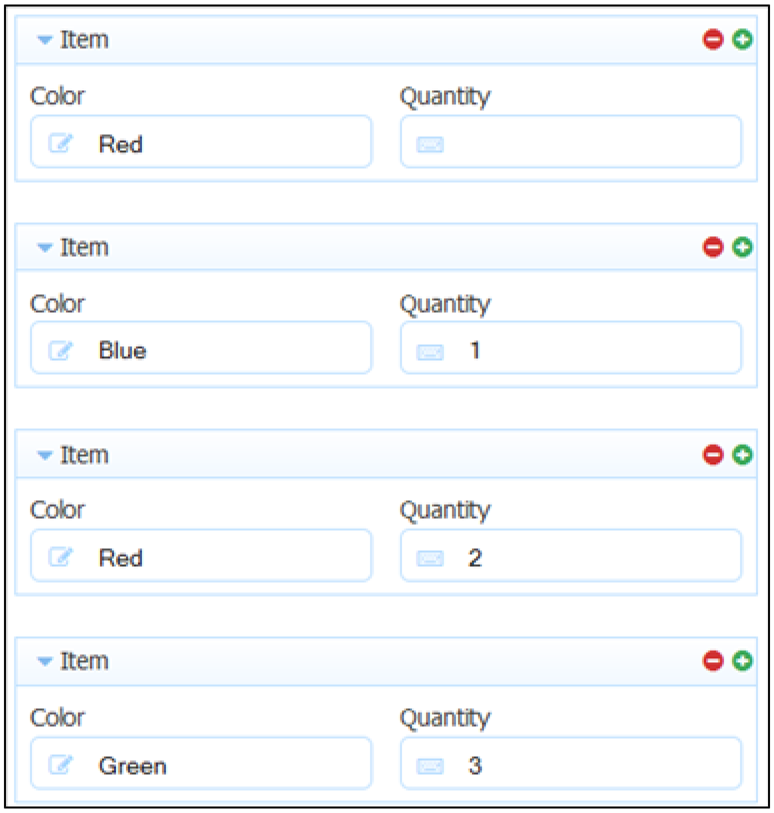
Things to note:

1. The Url parameter &onInit=getData used matches exactly the name of a javascript method in the html page also named getData().
2. The xml data is also embedded in the html page in the <textarea> tag.
3. The <ns:form> content came directly from a prior form submission and was copied out of the Live Forms submission repository's Documents tab.



```
<ns:form xmlns:ns="http://www.frevvo.com/schemas/_07IOYMUdEeCuRePmBeMwTQ" name="
  <Item>
    <Color>Red</Color>
  </Item>
  <Item>
    <Color>Blue</Color>
    <Q>1</Q>
  </Item>
  <Item>
    <Color>Red</Color>
    <Q>2</Q>
  </Item>
  <Item>
    <Color>Green</Color>
    <Q>3</Q>
  </Item>
</ns:form>
```

This is how the form looks when initialized via the onInit method. The particular form used in this sample has a repeat control. By default this form has a single repeating item. When Live Forms initializes the form with repeating data returned from the getData() method it auto creates the additional items. This form also contained a Item Added Rule which is why there is also a value in the Quantity field.


Item

Color: Red
Quantity:

Item

Color: Blue
Quantity: 1

Item

Color: Red
Quantity: 2

Item

Color: Green
Quantity: 3

Post and XML Documents

If form A and form B both contain the same data source, and the Form Action in form A is set to "'Post'" to the form Url of form B, then form B's fields from schema will be initialized with the values entered into form 1.

This occurs because the Post request to the Form Server contains the XML document(s) from form A. If one data source was added from XSD schema to form A. Then it will contain two documents. The first is the form's from-scratch document containing elements for all controls added from palette. The second is the schema document.

Form B's from schema document is in the same namespace as form A's since we added the same XSD data source to both. So the values entered into form A's fields will be used as an initial instance document to initialize form B's fields.Refer to Multi page forms for the details about posting data between forms using Form Action wizards.

See default field values to understand the precedence that XML documents sent in the Post request to form B take over the other possible methods of defaulting form field values.

See Initializing Forms with XML Documents for more information.