

# Editable Spry Table

## Introduction

In this article we will talk about creating "editable Spry Table" (or "editable grid") functionality where user can double-click directly inside a table cell and edit it just like an Excel spreadsheet. Clicking away will return the cell back to the "read-only" state and save the change to the database. In other words the change will be saved as soon as it has been made (NOT for the entire table at once at the end of your edits).

You will learn how to build this functionality in Dreamweaver using standard Update Record Server Behavior and **Spry Server Action** object from [Spry Data Utilities Toolkit](#).



First | Next | Previous | Last

Showing Records 1 to 5 of 27

[1-5](#) | [6-10](#) | [11-15](#) | [16-20](#) | [21-25](#) | [26-27](#)

**Total Records: 5**

First Name	Last Name	Email	Phone Number
Abby	Catcher	abcatch_11@yahoo.com	604-579-6790
Adeline	Kelly	adelkelly@systcom.net	1-888-234-455
Adrian	Brewer	adrianbr34@gmail.com	N/A
Agatha	Well	jagathaw@yahoo.com	55 85 32329201
Alex	Chong	alchong@gmail.com	

To start you will need an HTML table with a row inside a Spry Repeat region i.e. a "Spry Table". The data which populates the table should be coming from a database and must be generated by the [Recordset to XML](#) Server Behavior.

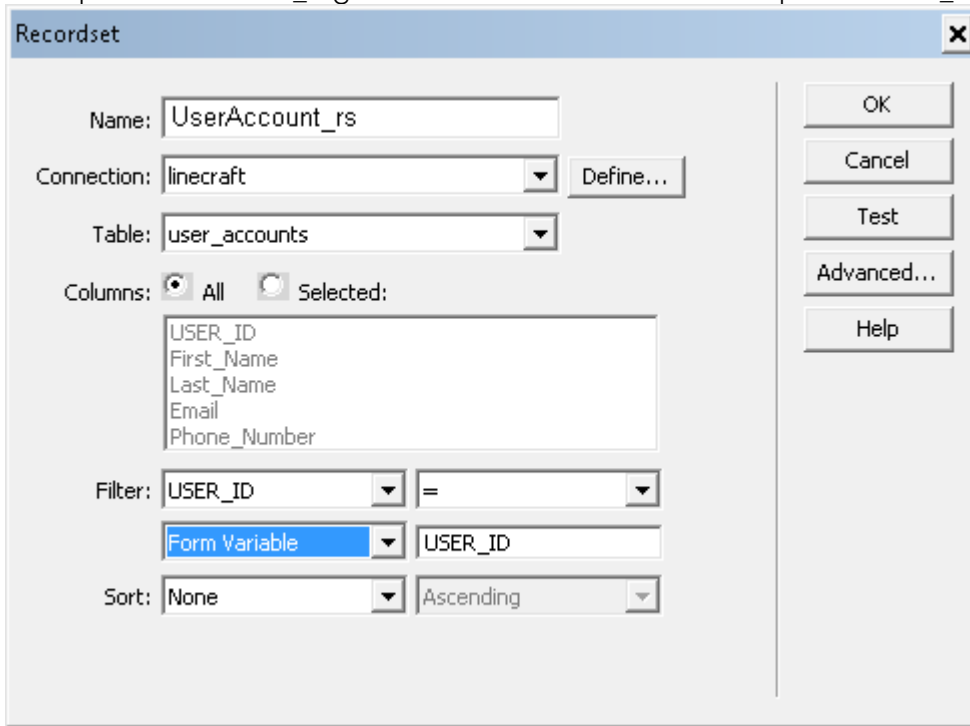
The functionality described in this article has been tested with IE7, IE8, Firefox 3.63, Chrome 2 and Safari 5.02.

## Adding Update Record Server Behavior

1. At this point your page should already contain both the Recordset which gets the records for your table and the **Recordset to XML** Server Behavior to translate this Recordset to XML format (so they can be displayed in your Spry region).

Next you will need a Recordset selecting a **single** record (based, for instance, on a record ID). This would be the Recordset which could get a single row of data that is about to be updated. In this

example UserAccount\_rs gets a user account based on the passed USER\_ID form variable.



- Time to add the “update” fields and the server code. Place your cursor somewhere outside your Spry Region and click **Record Update Form Wizard** button located on the **Data** tab of the **Insert** bar to run the Wizard.



In the **Record Update Form Wizard** dialog make sure to include ALL table columns. Clicking OK will insert the HTML form with all the fields. Dreamweaver will also add the server-side

code required for updating the record.

✕
Record Update Form

Connection:

Table to update:

Select record from:

Unique key column:   Numeric

After updating, go to:

---

Form fields:

Column	Label	Display As	Submit As	
USER_ID	USER_ID:	Text		▲
First_Name	First_Name:	Text field	Text	
Last_Name	Last_Name:	Text field	Text	
Email	Email:	Text field	Text	
Phone Number	Phone Number:	Text field	Text	▼

Label:

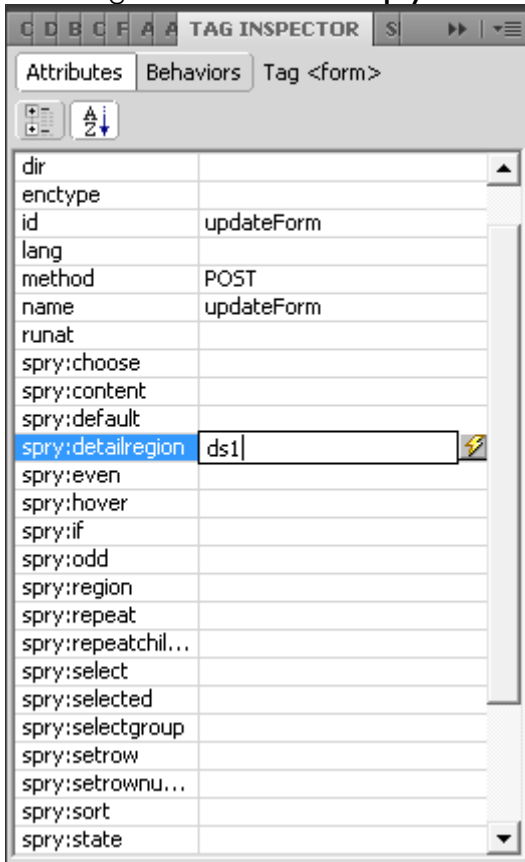
Display as:  Submit as:

Text:

## Form adjustments

Now let's make some adjustments to the form.

1. We need to tie the form to your Spry Region. To do so open your **Attributes panel**, select the form in the Design view and set the **spry:detailregion** parameter to the name of your Spry Data Set.



2. At the moment every field on your update form has its value set to the corresponding value from your UserAccount\_rs Recordset. We need to change that so that the values point to the Spry Data Set values instead.

Select every field (including your hidden USER\_ID field) and using **Properties Inspector** change its value to the corresponding value from your Spry Data Set.

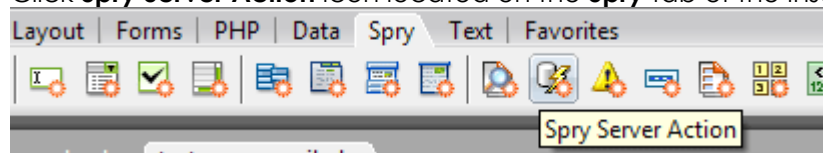
For example, change `<?php echo $row_UserAccount_rs['Email']; ?>` to `{Email}`, `<?php echo $row_UserAccount_rs['First_Name']; ?>` to `{First_Name}`, etc. You can also use the dynamic value button (which looks like a lightning bolt - right beside the value field in the Properties Inspector) to pick the correct values from the Spry Data set.

Lastly you need to make sure that when a row in your Spry Table is selected the form fields change their values to those in the selected row. Switch to the Code view and add `spry:setrow="ds1"` to the `<tr>` tag containing values from the Spry Data Set (where ds1 is the name of your Data set).

Now if you test the page in the browser you will notice that the values of the form fields change every time you select a row in your Spry Table. If you were to change the values now and submit the form the page would refresh and the selected record would be updated in the database. That's nice, but not exactly what we are aiming for. Since we are building an AJAX application we want no page refreshes. It's easy to accomplish with **Spry Server Action** object.

## Adding Spry Server Action object

Click **Spry Server Action** icon located on the **Spry** tab of the insert panel.



In the dialog select your update form from the **Form** dropdown, the **Display progress** indicator (if you are using one) and the name of your Spry Data set from the **Refresh Spry data set** dropdown to reload your Spry Table if you want your data to be fresh (for example if your Spry Table is sorted by the value in the edited column).

A screenshot of the 'Spry Server Action' dialog box. The 'Form' dropdown is set to 'updateForm'. The 'Display progress indicator' dropdown is set to 'div "div\_progress"'. Under 'Actions upon completion', the 'Refresh Spry data set' dropdown is set to 'ds1'. There are checkboxes for 'Display server message' and 'If success invoke function', and text input fields for 'If error invoke function'. A 'How to use this dialog?' link is present. At the bottom, there is a 'LINECRAFT' logo with a red cursor icon. On the right side, there are 'OK', 'Cancel', and 'Help' buttons.

Click **OK** to insert the object.

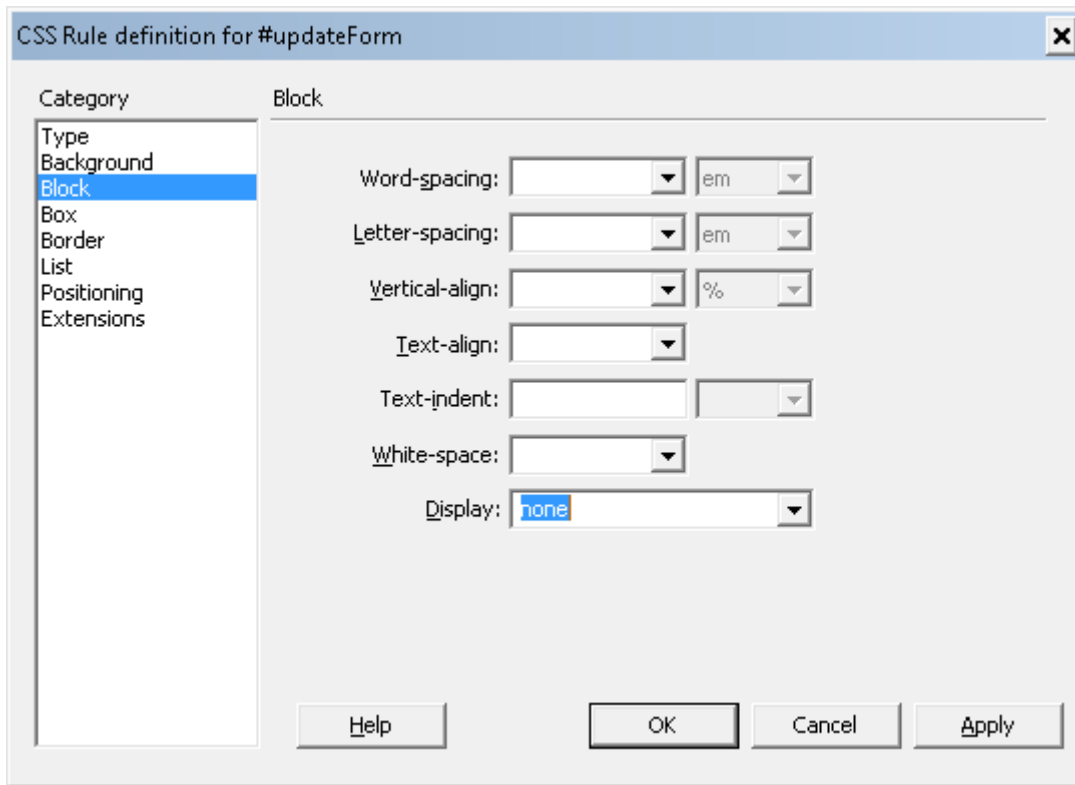
Test your page now and you will notice that the refresh is gone. Definitely a progress, but let's not stop at that!

We can now safely hide the form from the user.

Click the "+" button at the bottom of the **CSS Styles** panel to add a new declaration.

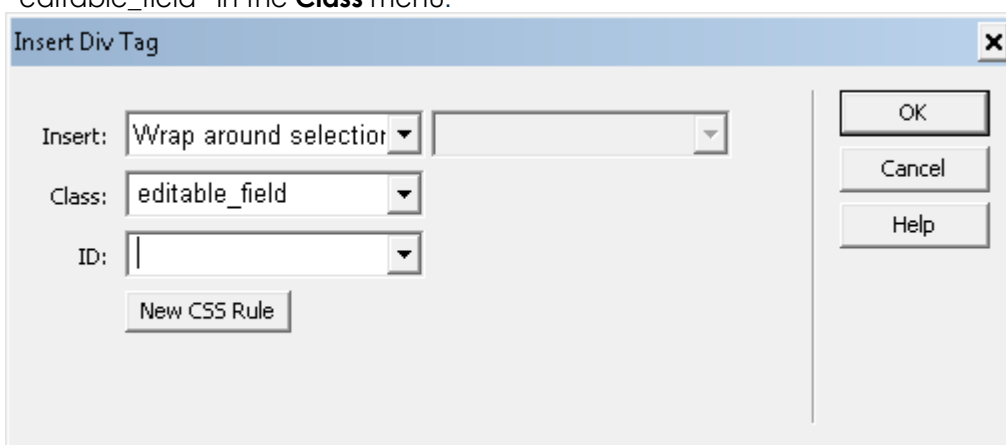
From the **Selector Type** dropdown pick "ID (applies to only one HTML element)" and from the **Rule Definition** "(This document only)".

Click "OK" and switch to the **Block** tab in the pop-up dialog. Select "none" from the **Display** dropdown. Your form will now be hidden.

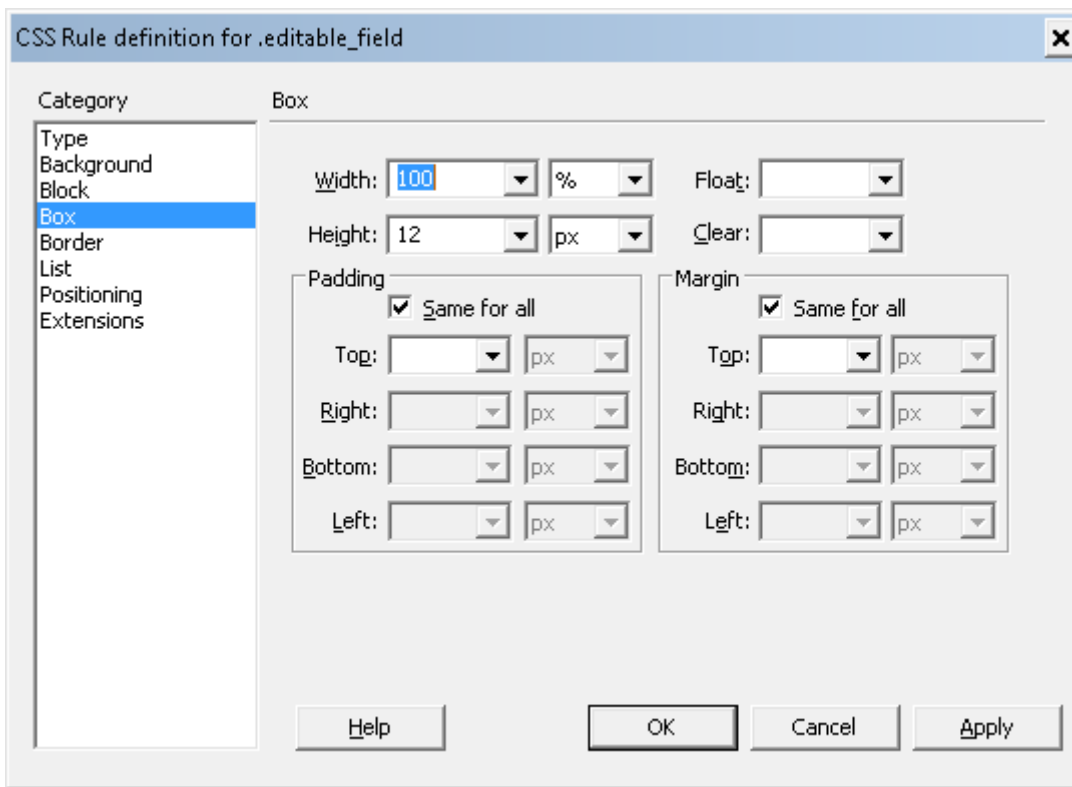


## Making editable cells

1. First of all we need to make sure that even if a cell in your table contains no data it can still be clicked on and edited. To achieve that we need to wrap the content of every cell you want to be editable inside a <div> tag. Go ahead and select the content of the first cell, then click **Insert DIV tag** button on the **Layout** tab of the **Insert** bar. In the dialog that will pop-up make sure "Wrap around selection" option is selected and enter "editable\_field" in the **Class** menu.



Click **New CSS** Rule button to open the dialog for defining CSS properties for the class. Click "OK" and enter 100% for the **Width** and 12px (or whatever you want the height of your cell to be) for the **Height** and click "OK" again:



Add the following block of JavaScript inside the <head> tag of your page:

```
<script type="text/javascript">
<!--
function editField(what){
    what.contentEditable=true;
}
function updateField(what, fld){
    what.contentEditable=false;
    field=eval("document.updateForm."+fld);
    if(document.all){
        new_value=what.innerText;
    }else{
        new_value=what.textContent;
    }
    //Update ONLY if the new value is different from the old one
    if(new_value!=field.value){
        field.value=new_value;
        document.updateForm.submit();
    }
}
//-->
</script>
```

Notice that "updateForm" in this script is the name of your update form - change it to your own value if needed.

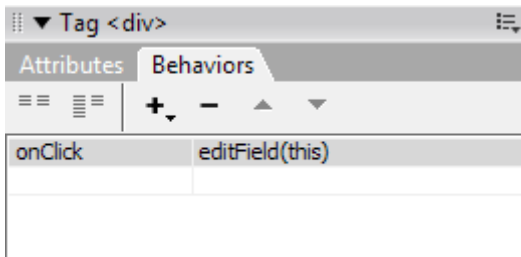
The script contains two functions: **editField()** and **updateField()**.

First one (**editField()**) will make the cell editable and will be triggered by user's **onclick** event.

The second function (**updateField()**) will set the value of the corresponding field inside our hidden

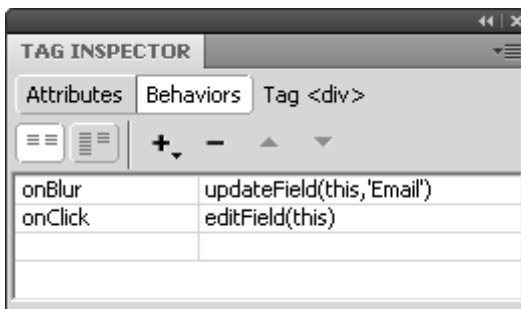
update form to the text of the cell and trigger the form submission to update the record and reset the view. This will occur when user clicks away from the cell and ONLY if the new value is different from the old one.

Now select each "editable\_field" div tag and use **Behaviors** panel and to add "editField(this)" (without quotes) to each **div** tag making sure **onclick** event is selected in event handler dropdown.



**Behaviors** panel in Dreamweaver CS3

2. After that repeat the procedure and add "updateField(this,' field\_name')" (without quotes) to each of the div tags making sure **onblur** event is selected in event handler dropdown. Instead of the field\_name enter the name of the corresponding update form field. Notice the single quotes surrounding the field's name:



**Behaviors** panel in Dreamweaver CS4 and CS5

## Conclusion

That's all! You can upload the files to your server and test your editable grid.