

# Actions Tech. Note 7

## Draft 1.0

### Adding Mark-up to Table Rows.

In many situations when working with Databases it is desirable to add mark-up to table rows. Generally you want to add mark-up that will cause a one or more table rows to be repeated for a number of records in a database. In FileMaker's CMDL, for example there is a `[record]` and `[/record]` tag that will repeat the HTML for all records that exist in a selection, this typically encloses a table row in the HTML output.

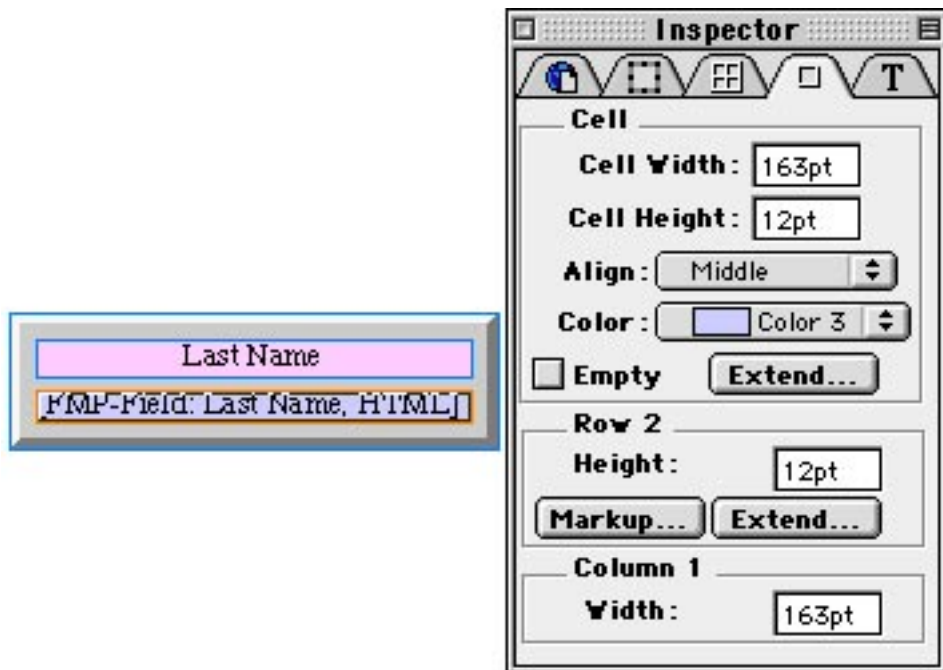
Below is an the sort of code fragment that you might want to generate if you where inserting code for FileMaker.

```
<table border=5 cellpadding=0 cellspacing=5 bgcolor="#cccccc">
  <tr>
    <td><p align=center>Last Name</p></td>
  </tr>

  [FMP-Record]
  <tr>
    <td>[FMP-Field: Last Name, HTML]</td>
  </tr>
  [/FMP-Record]

</table>
```

This shows the `[FMP-Record]` added before the `<tr>` and a `[/FMP-Record]` added after the `</tr>`. If you where inserting these tags manually in Freeway you could just select an item in the row and the required mark-up.



In FW3 it is not possible to add an Freeway Action to a table row, nor will your action get any notification of before-table-row, after-table-row etc. However it is possible to apply an action to a table cell, then through traversing the tag tree it is possible to find the <tr> tag and add mark-up code using JavaScript. If you are not comfortable using JavaScript and would like to use this sort of functionality you can cut and paste the two example actions at the end of this Tech. Note and customise them.

## Finding the <tr> tag(s).

If you apply an action to a table cell then finding the enclosing <tr> tag is quite straight forward.

```
// find the /td/ tag that this item is applied to
var tdTag = fwDocument.fwTags.fwFindAll("td",fwItem);
if (tdTag)
{
    // find the /tr/ tag that the /td/ tag appears within
    trTag = tdTag.fwFindEnclosing("tr")
    if (trTag)
    {
        // do something with the /tr/ tag
    }
}
```

The more general case would make it possible to add mark-up across one or more <tr> tags - it may be that you want mark-up that repeats a number of table rows. The example below shows CMDL mark-up that spans two table rows

```
<table border=5 cellpadding=0 cellspacing=5 bgcolor="#cccccc">
  <tr>
    <td><p align=center>Last Name</td>
  </tr>

  [FMP-Record]
  <tr>
    <td>[FMP-Field: First Name, HTML]</td>
  </tr>
  <tr>
    <td>[FMP-Field: Last Name, HTML]</td>
  </tr>
  [/FMP-Record]

</table>
```

To do this you will have to find the <table> tag in which the <td> appears, and then to find an arbitrary tag after that. Probably the best thing to do is to then find all the <tr> tags and then to work with that. The following (incomplete) action will find the <tr> tag and an arbitrary <tr> tag after it (specified) in the action's UI.

```
<item-action name="Row Markup">
<action-number name="Amount" title="Rows" default=1 minimum=1>

<action-javascript>

function fwBeforeEndBody()
{
    // get the amount make sure that it is a valid number
    var theAmount=parseInt(fwParameters.Amount.toString());
    if (isNaN(theAmount))
```

```

        theAmount = 1;

var tdTag=fwDocument.fwTags.fwFind("",fwItem);
if (tdTag)
{
    var firstTag = tdTag.fwFindEnclosing("tr");
    var lastTag;

    var table = tdTag.fwFindEnclosing("table");
    if (firstTag && table)
    {
        var rows = table.fwFindAll("tr");

        for (var i = 0 ; i < rows.length ; i++)
        {
            if (rows[i] == firstTag)
            {
                // find the index of the lastTag
                var j = i + theAmount -1;
                if (j >= rows.length)
                    j = rows.length - 1;

                // get the lastTag
                lastTag = rows[j];

                // do something with firstTag & lastTag

                break;
            }
        }
    }
}

</action-javascript>
</item-action>

```

## Adding the Mark-up

Having found the tag or tags you will have to add mark-up to them. Adding mark-up after a specified tag is fairly simple. The `fwAddRaw` method adds raw text and allows you to specify a tag after the text will be added. Adding text before a tag is a little more complex as there is no way in `fwAddRaw` that you can specify a tag before which the text will be added. To add text you before a tag you will have to add it immediately after the tag and then move the tag using the `fwMove` method.

The following function will do this:

```

function EncloseInMarkup(firstTag, lastTag, beforeMarkup, afterMarkup)
{
    if (firstTag && lastTag)
    {
        // add the beforeMarkup if there is any
        beforeMarkup = beforeMarkup.toString();
        if (beforeMarkup)
        {
            // add it after 'firstTag' tag
            var tag = firstTag.fwEnclosing.fwAddRaw(firstTag, beforeMarkup);

            // and move the tag so that it is before
            firstTag.fwEnclosing.fwMove(firstTag, tag);
        }
    }
}

```

```

    }

    // add the afterMarkup if there is any
    afterMarkup = afterMarkup.toString();
    if (afterMarkup)
    {
        // add it after 'lastTag' tag
        lastTag.fwEnclosing.fwAddRaw(lastTag, afterMarkup);
    }
}
}

```

You can then use this function once you have found the tag or tags that you want to enclose in Markup.

## Example 1

The following code will enclose one or more table rows in mark-up. The mark-up is defined in action-mark-up, and fwSubstitute is called first. This means that if you embed parameters of the form &param1; into the mark-up it will be correctly substituted with parameter values. The intention is that if you are not at ease with writing actions in JavaScript you can copy and paste this action, change the name, add parameters to it and just edit the mark-up, adding the code that you would put in before-table-row (if it existed) - if you are familiar with JavaScript you can use this and generate the strings beforeStr and afterStr etc. with whatever recipes are appropriate.

```

<item-action name="Row Markup">
<action-number name="Amount" default=1 minimum=1>

<action-markup name="before" custom>[record]</action-markup>
<action-markup name="after" custom>[/record]</action-markup>

<action-javascript>

function EncloseInMarkup(firstTag, lastTag, beforeMarkup, afterMarkup)
{
    if (firstTag && lastTag)
    {
        // add the beforeMarkup if there is any
        beforeMarkup = beforeMarkup.toString();
        if (beforeMarkup)
        {
            // add it after 'firstTag' tag
            var tag = firstTag.fwEnclosing.fwAddRaw(firstTag, beforeMarkup);

            // and move the tag so that it is before
            firstTag.fwEnclosing.fwMove(firstTag, tag);
        }

        // add the afterMarkup if there is any
        afterMarkup = afterMarkup.toString();
        if (afterMarkup)
        {
            // add it after 'lastTag' tag
            lastTag.fwEnclosing.fwAddRaw(lastTag, afterMarkup);
        }
    }
}

function fwBeforeEndBody()

```

```

{
  var beforeStr = fwSubstitute(fwMarkups["before"]);
  var afterStr = fwSubstitute(fwMarkups["after"]);

  // get the amount
  var theAmount=parseInt(fwParameters.Amount.toString());
  if (isNaN(theAmount))
    theAmount = 1;

  var tdTag=fwDocument.fwTags.fwFind("",fwItem);
  if (tdTag)
  {
    var firstTag = tdTag.fwFindEnclosing("tr");
    var lastTag;

    var table = tdTag.fwFindEnclosing("table");
    if (firstTag && table)
    {
      var rows = table.fwFindAll("tr");

      for (var i = 0 ; i < rows.length ; i++)
      {
        if (rows[i] == firstTag)
        {
          // find the index of the lastTag
          var j = i + theAmount -1;
          if (j >= rows.length)
            j = rows.length - 1;

          // get the lastTag
          lastTag = rows[j];

          // add the markup to the firstTag & lastTag
          EncloseInMarkup(firstTag, lastTag, beforeStr, afterStr);
          break;
        }
      }
    }
  }
}
}
}
}
}

</action-javascript>
</item-action>

```

## Example 2

As with Example 1 this is intended for you to copy & paste. This action is more general in that it will add mark-up around item that is added to the page. If it is applied to a table cell then the user has an option for it to be added to a row or a table cell, as well as the number of rows or cells to enclose.

```

<item-action name="Item Markup">
  <action-popup name="Repeat" title="Apply To"> <value name="Row"> <value name="Cell">
  </action-popup>
  <action-number name="Amount" default=1 minimum=1>

  <action-markup name="before" custom>[record]</action-markup>
  <action-markup name="after" custom>[/record]</action-markup>

</action-javascript>

```

```

function EncloseInMarkup(firstTag, lastTag, beforeMarkup, afterMarkup)
{
    if (firstTag && lastTag)
    {
        // add the beforeMarkup if there is any
        beforeMarkup = beforeMarkup.toString();
        if (beforeMarkup)
        {
            // add it after 'firstTag' tag
            var tag = firstTag.fwEnclosing.fwAddRaw(firstTag, beforeMarkup);

            // and move the tag so that it is before
            firstTag.fwEnclosing.fwMove(firstTag, tag);
        }

        // add the afterMarkup if there is any
        afterMarkup = afterMarkup.toString();
        if (afterMarkup)
        {
            // add it after 'lastTag' tag
            lastTag.fwEnclosing.fwAddRaw(lastTag, afterMarkup);
        }
    }
}

```

```

function fwBeforeEndBody()
{
    var beforeStr = fwSubstitute(fwMarkups["before"]);
    var afterStr = fwSubstitute(fwMarkups["after"]);
    var tags=fwDocument.fwTags.fwFindAll("",fwItem);
    if (tags.length == 0)
        return;

    var isTableCell = fwItem.fwIsTableCell;

    if (isTableCell)
    {
        // make sure that we
        var theAmount=parseInt(fwParameters.Amount.toString());
        if (isNaN(theAmount))
            theAmount = 1;

        var table = tags[0].fwFindEnclosing("table");
        if (fwParameters.Repeat=="Row")
        {
            var rows = table.fwFindAll("tr");
            for (var i = 0 ; i < rows.length ; i++)
            {
                if (rows[i].fwFind("",fwItem))
                {
                    var j = i + theAmount -1;
                    if (j >= rows.length)
                        j = rows.length - 1;
                    EncloseInMarkup(rows[i], rows[j], beforeStr, afterStr);
                    break;
                }
            }
        }
        else
        {
            var cells = table.fwFindAll("td");
            for (var i = 0 ; i < cells.length ; i++)
                if (cells[i].fwOwner==fwItem)

```

```

        {
            var j = i + theAmount - 1;
            if (j >= cells.length)
                j = cells.length - 1;
            EncloseInMarkup(cells[i], cells[j], beforeStr, afterStr);
            break;
        }
    }
}
else
{
    for (var i in tags)
    {
        var tag = tags[i];
        EncloseInMarkup(tag, tag, beforeStr, afterStr);
    }
}

}

function fwInterface()
{
    var isTableCell = fwItem.fwIsTableCell;
    fwParameters.Repeat.fwVisible=isTableCell;
    fwParameters.Amount.fwVisible=isTableCell;
    fwParameters.Amount.fwTitle=fwParameters.Repeat+"s";
}

</action-javascript>
</item-action>

```