EXPORTING HTML TABLE DATA TO AN EXCEL DOCUMENT

This is a simple client side JavaScript-based data export functionality that exports cellular data from an HTML table into an Excel document on the fly. In this approach, we'll create an instance of Excel on the client and then pass data and function calls into it. This functionality can provide basis for powerful reporting, in which we can also create calculated/aggregated fields within excel documents. This functionality can be used to create customized Analysis reports/spreadsheets with multiple sheets i.e. data from a single table can be distributed or presented in multiple sheets in same Excel document. New columns and rows can easily be added that can be used to hold calculated values.

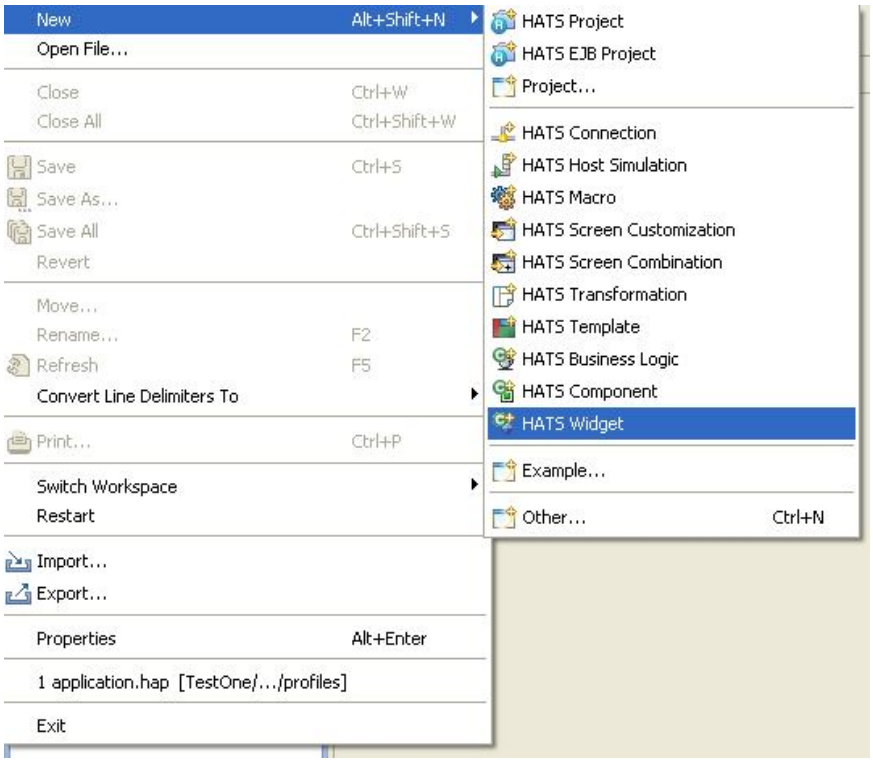
We need to start by identifying a table on our webpage that we need to export in Excel. This can be done in two ways:

A. If our table is displaying data from multiple screens on the host using Screen combination, we can write our custom table widget that will assign an **id** attribute to the table that we'll create inside our widget to hold the cellular data from the host.

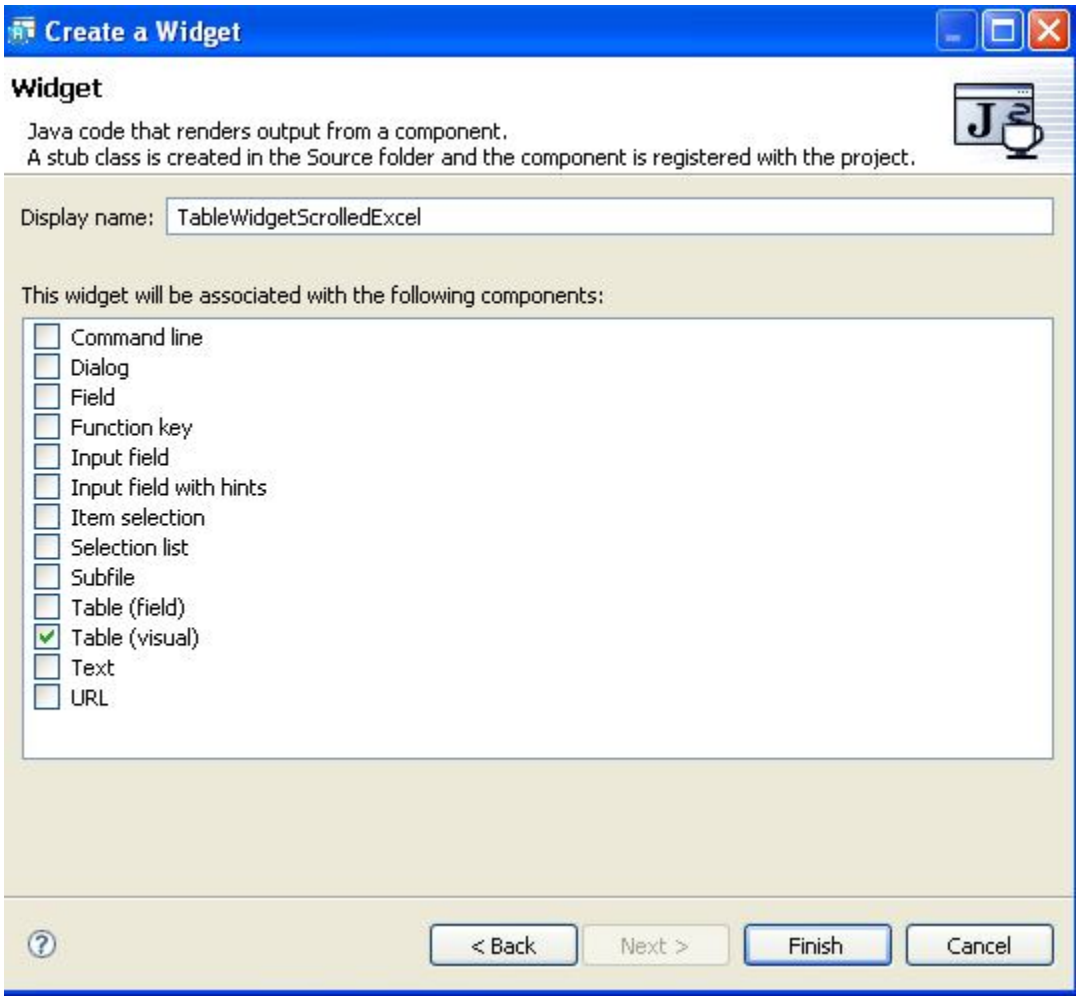
B. We can use the default table widget in our screen combination and find the table that contains our data by a hit and trial method that involves searching the elements in the JSP by tag name and once our required table is found, we assign it the **id** at runtime using client side JavaScript.

In both the above cases, our JavaScript function that implements the Excel exporting functionality will read the **id** of the table and then start with the exporting process.

Using option **A**, we can create our custom widget in HATS:



We associate our widget with the **Visual Table Component**:



As we modify the **drawHTML** method of our widget, we can assign the **id** attribute among others as highlighted below:

```
public StringBuffer drawHTML() {
    // TODO Auto-generated method stub
    StringBuffer buffer = new StringBuffer(256);
    int oldSID = 0;

    HTMLElementFactory factory = HTMLElementFactory.newInstance(
            contextAttributes, settings);

    //Button to Print the Table
    buffer.append("<INPUT TYPE=\"button\" NAME=\"ExcelGenerator\" CLASS=\"HostButton\" VALUE=\"Excel View\" ONCLICK=\"exportToExcel();\"> ");

    //Opening the DIV Tag
    buffer.append("<DIV ID=\"scrollDiv1\" STYLE=\"overflow:auto; HEIGHT:310px; WIDTH:100% \"> ");

    //Opening TABLE Tag
    buffer.append("<TABLE ID='detailsTable' CLASS=\"HATSTABLE\" CELLSPACING=0 CELLPADDING=0 WIDTH=\"100%\" BORDER=1> ");
```

Using option **B**, we can search for the table that contains our data from the host screen by using:

document.getElementsByTagName("table")[0].setAttribute("id","detailsTable").

Now a JavaScript method can be developed to parse the HTML table using its ID. In this method, we would create and call an ActiiveXObject for Excel application at client-side. After that an Excel work book will be added along a spread-sheet to that object. Then the HTML table will be parsed for data elements, and will placed in Excel document at specified positions/cells. This is a completely JavaScript based functionality that would require MS Office setup at client machine, that will work fine in Internet Explorer 6 browser environment. This method can be placed in the template or individually in customization depending on the requirements.

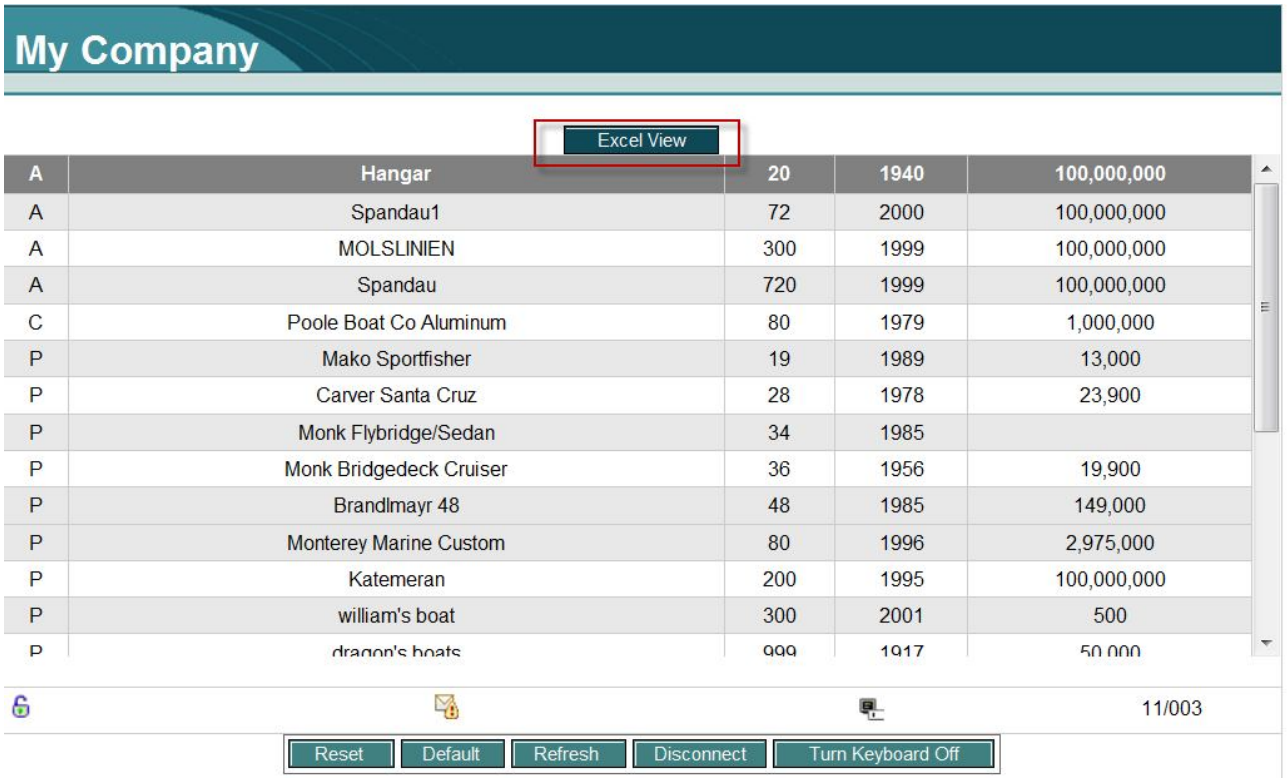
Following is the JavaScript code snippet for exporting table data into Microsoft Excel:

```
<script language=javascript>

    function exportToExcel()
    {
        contentType="application/vnd.ms-excel";

    var oExcel = new ActiveXObject("Excel.Application");
        var oBook = oExcel.Workbooks.Add;
        var oSheet = oBook.Worksheets(1);

        :   .....
        .   .....

            Code to populate cells of the OSheet object with the table cells

            .....
            .....

             }
        }

        oExcel.Visible = true;
        oExcel.UserControl = true;
    }

</script>
```

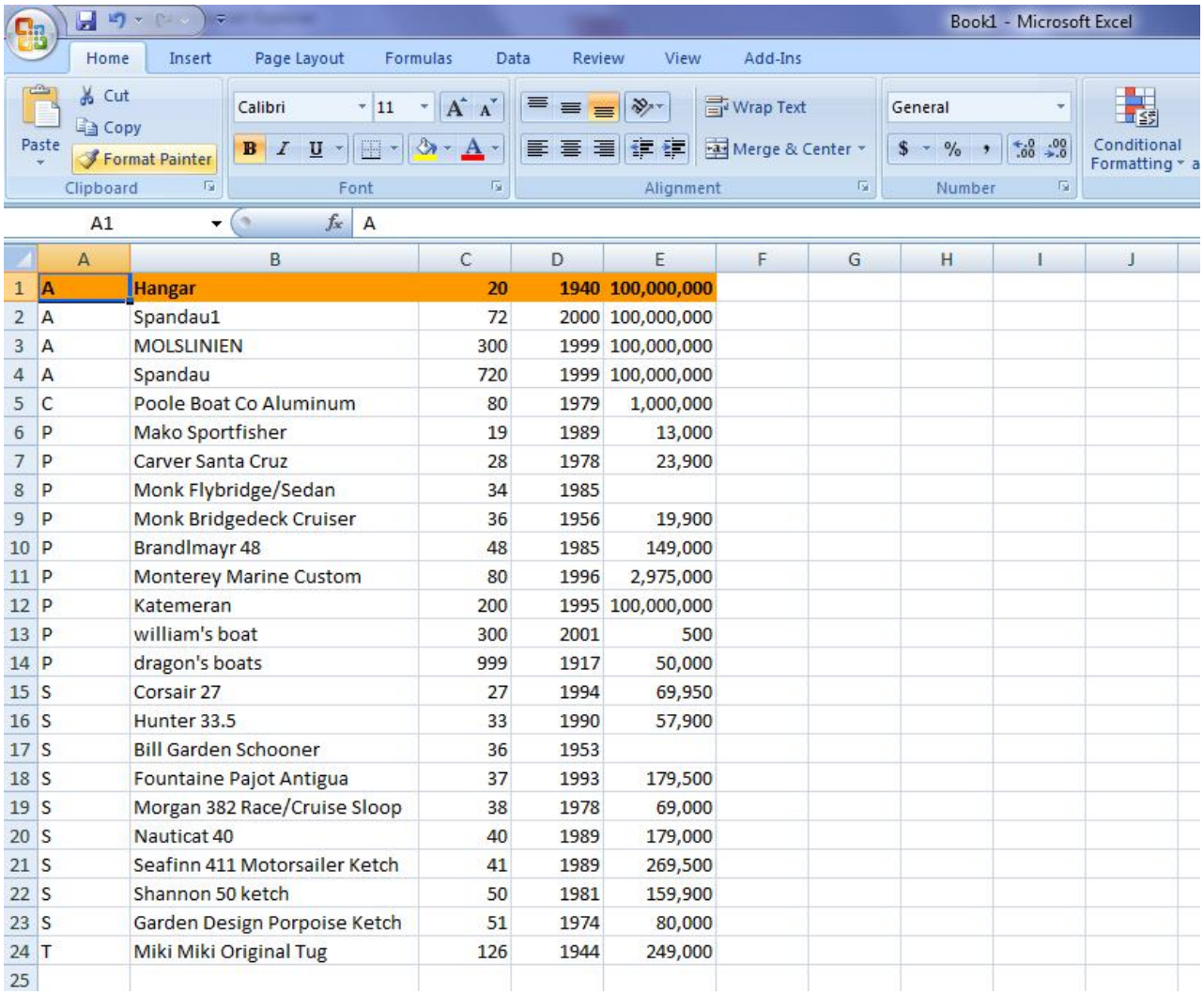
A button can be displayed on the JSP that triggers the Excel export functionality. For example:

<button onclick="exportToExcel();">Export to Excel File</button>

The custom table widget renders the table as shown in the screen below:



When we click the button labeled “Excel View” in the above screen, an new instance of Microsoft Excel is created and the table data is exported to a new sheet in Excel as shown below:



	A	B	C	D	E	F	G	H	I	J
1	A	Hangar	20	1940	100,000,000					
2	A	Spandau1	72	2000	100,000,000					
3	A	MOLSLINIEN	300	1999	100,000,000					
4	A	Spandau	720	1999	100,000,000					
5	C	Poole Boat Co Aluminum	80	1979	1,000,000					
6	P	Mako Sportfisher	19	1989	13,000					
7	P	Carver Santa Cruz	28	1978	23,900					
8	P	Monk Flybridge/Sedan	34	1985						
9	P	Monk Bridgedeck Cruiser	36	1956	19,900					
10	P	Brandlmayr 48	48	1985	149,000					
11	P	Monterey Marine Custom	80	1996	2,975,000					
12	P	Katemeran	200	1995	100,000,000					
13	P	william's boat	300	2001	500					
14	P	dragon's boats	999	1917	50,000					
15	S	Corsair 27	27	1994	69,950					
16	S	Hunter 33.5	33	1990	57,900					
17	S	Bill Garden Schooner	36	1953						
18	S	Fountaine Pajot Antigua	37	1993	179,500					
19	S	Morgan 382 Race/Cruise Sloop	38	1978	69,000					
20	S	Nauticat 40	40	1989	179,000					
21	S	Seafinn 411 Motorsailer Ketch	41	1989	269,500					
22	S	Shannon 50 ketch	50	1981	159,900					
23	S	Garden Design Porpoise Ketch	51	1974	80,000					
24	T	Miki Miki Original Tug	126	1944	249,000					
25										

As mentioned previously, this functionality can be further enhanced to create customized Analysis reports/spreadsheets with multiple sheets and various calculations can be done on the exported data as per requirement.

Note: - To run this functionality at client browser, security level must be customized that is,

* Set the security level of Internet Explorer to **Medium-Low**

* Go to **iexplorer** security settings and select the "**Enable**" radio button for security option "**Initialize and script ActiveX controls not marked as safe**".

Sources:
HATS Hotspot : http://www-949.ibm.com/software/rational/cafe/community/hats
IBM: www.ibm.com

IBM Global Services
Route 100
Somers, NY 10589
U.S.A.
Produced in the United States of America
08-10