

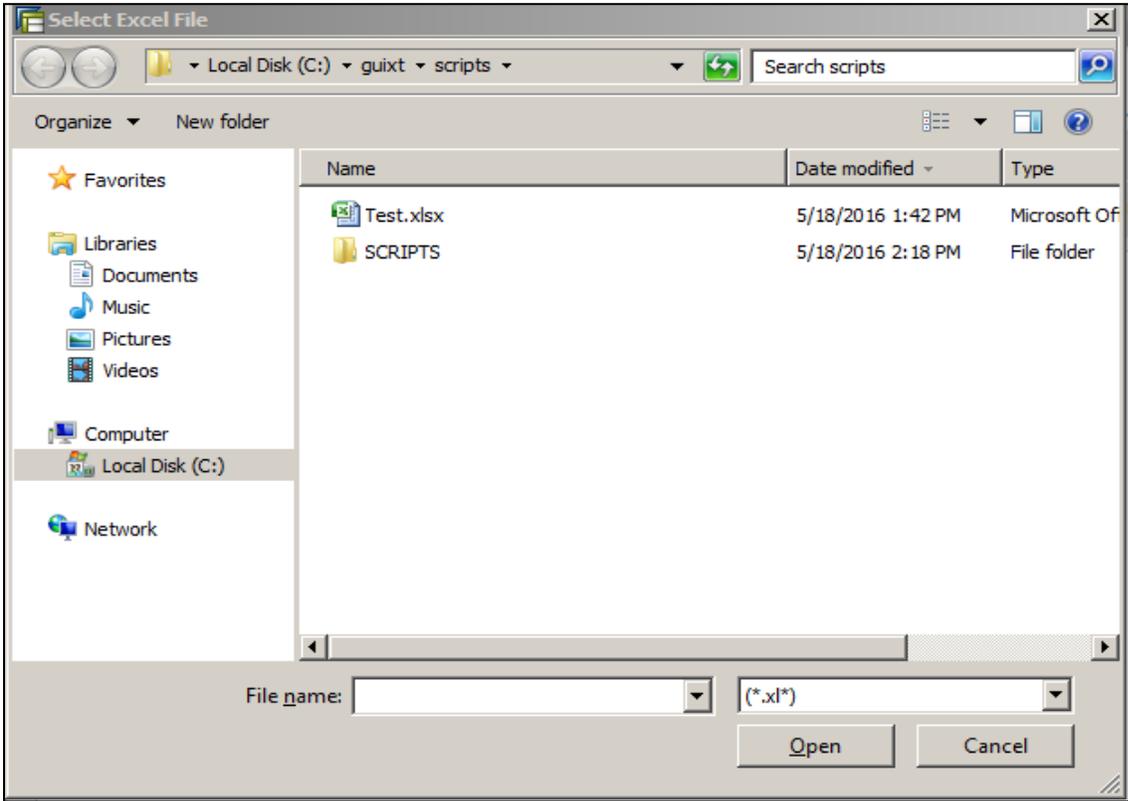
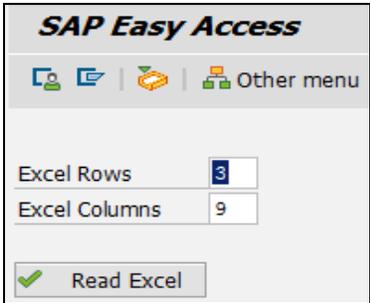
# Liquid UI: Determine Data Rows and Columns in Excel

## Purpose

To determine the data rows and columns from user selected Excel file.

## User Interface

Log into SAP and on the SAP Easy Access Screen click on the 'Read Excel' pushbutton which will bring a pop-up to select an Excel File.



## Liquid UI Code [Script]

### User Interface file -

```
SAPLSMTR_NAVIGATION.E0100.sgs
1
2 load('wsoffice'); // Need to load this file to display File Selection pop-up
3 // User Interface
4 clearscreen();
5 inputfield([1,0], "Excel Rows", [1,15], {"size":3, "name":"z_mm01_rows"});
6 inputfield([2,0], "Excel Columns", [2,15], {"size":3, "name":"z_mm01_cols"});
7 pushbutton([4,0], "@0V@Read Excel", "?", {"process":excelRowColumnCount});
8
9 // Remove blank spaces
10 String.prototype.trim=function(){return this.replace(/^\s+|\s+$/g, '');}
11
12 // To validate if the variable holds blank or null value
13 function isBlank(jvar){
14     if(typeof jvar == 'string') {
15         jvar = jvar.trim();
16     }
17     return(jvar == 'undefined' || jvar == null || jvar == "" || jvar == void 0);
18 }
19
20 // Function to Show File Open Dialog
21 function selectFileDialog(szPrompt){
22     if(szPrompt==void 0)
23         szPrompt = 'Select Excel File';
24     var dialog = new ActiveXObject('MsComDlg.CommonDialog');
25     // dialog.Filter='All Files (*.*)|*.*';
26     dialog.Filter='(*.xl*)|*.xl*'; // BD
27     dialog.MaxFileSize=32767;
28     //dialog.AllowMultiSelect = true;
29     dialog.DialogTitle=szPrompt;
30     dialog.Flags=0x200|0x80000|0x800|0x4|0x200000
31     dialog.ShowOpen();
32     //var ret = dialog.FileTitle;
33     var ret = dialog.FileName;
34     dialog = void 0;
35     return ret;
36 }
37
38 // Function to open Excel File From The File Selection Dialog
39 function openExcel(filename) {
40     if(excelObj == void 0)
41         excelObj = new ActiveXObject('Excel.Application');
42     excelBook = excelObj.Workbooks.Open(filename);
43     excelObj.Visible = true;
44     excelObj.ScreenUpdating = true;
45 }
46
47 // Determine Total Number of Columns with Data in Excel Spreadsheet
48 function determineNoOfDataColumns(excelActiveSheet, nColumnHeadingRow) {
49     var excelColumnCount = excelActiveSheet.Columns.Count;
50     for(var i = 1; i<excelColumnCount;i++) {
51         if (excelActiveSheet.Cells(nColumnHeadingRow, i).Value == undefined || typeof(excelActiveSheet.Cells(nColumnHeadingRow, i).Value) == 'undefined') {
```

```

52         break;
53     }
54 }
55 return i;
56 }
57
58 // Determine Total Number of Rows with Data in Excel Spreadsheet
59 function determineNoOfDataRows(excelActiveSheet, nTotalDataColumns, nStartDataRow) {
60     var excelRowCount = excelActiveSheet.Rows.Count;
61     for(var i = nStartDataRow; i<excelRowCount;i++) {
62         lastRowFound = false;
63         for(var j = 2; j<nTotalDataColumns;j++) {
64             if (excelActiveSheet.Cells(i, j).Value == undefined || typeof(excelActiveSheet.Cells(i, j).Value) == 'undefined') {
65                 lastRowFound = true;;
66             } else {
67                 lastRowFound = false;
68                 break;
69             }
70         }
71         if(lastRowFound == true) {
72             // Last Row with Data + 1;
73             break;
74         }
75     }
76     return i;
77 }
78
79 // Function to determine rows and columns
80 function excelRowColumnCount() {
81     OPEN_EXCEL_FILE;
82     if(excelObj == void 0) {
83         excelFileName = selectFileDialog('Select Excel File');
84         if(excelFileName.length) {
85             openExcel(excelFileName);
86             excelSheet = excelBook.ActiveSheet;
87         } else {
88             message('E: No Excel File Selected');
89             return; // If Problem opening selected excel file, stop the process
90         }
91     } else { // Excel is already open (Manually Opened or Re-run for Error Processing)
92         try {
93             excelSheet = excelBook.ActiveSheet;
94             // Check to see if we can read cell value, if not then Re-Open Excel File
95             var cellCheckValue = excelSheet.Cells(1, 2).Value;
96         }
97         catch(err) {
98             delete excelObj;
99             goto OPEN_EXCEL_FILE;
100     }
101 }

```

```

102
103     totalDataColumns = determineNoOfDataColumns(excelSheet,1);
104     totalDataRows = determineNoOfDataRows(excelSheet,totalDataColumns,1);
105     totalDataColumns = totalDataColumns-1;
106     totalDataRows = totalDataRows-1;
107     set('V[z_mm01_rows]','&V[totalDataColumns]');
108     set('V[z_mm01_cols]','&V[totalDataRows]');
109 }

```