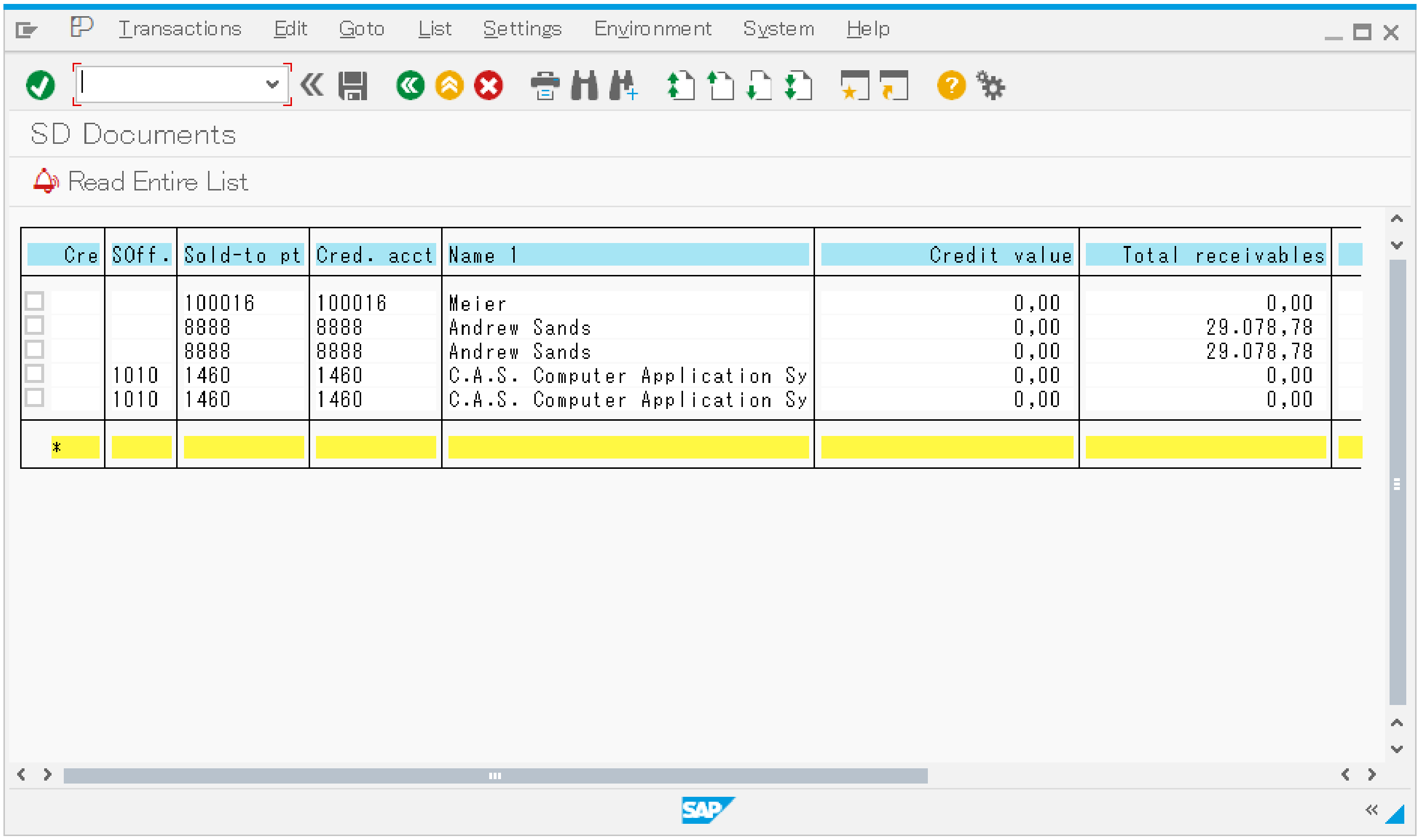
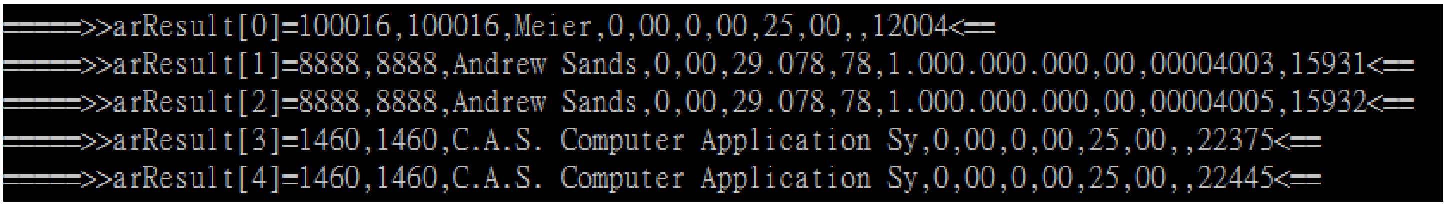
Liquid UI: Read specific columns from a list screen

User Interface (VKM1, result screen)



Cre	SOff.	Sold-to pt	Cred. acct	Name 1	Credit value	Total receivables
□		100016	100016	Meier	0,00	0,00
□		8888	8888	Andrew Sands	0,00	29.078,78
□		8888	8888	Andrew Sands	0,00	29.078,78
□	1010	1460	1460	C.A.S. Computer Application Sy	0,00	0,00
□	1010	1460	1460	C.A.S. Computer Application Sy	0,00	0,00
*						

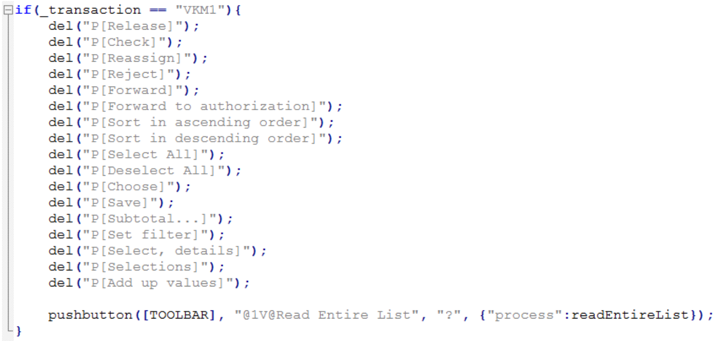
Read result in Cornelius Window



```
=====>>arResult[0]=100016,100016,Meier,0,00,0,00,25,00,,12004<==
=====>>arResult[1]=8888,8888,Andrew Sands,0,00,29.078,78,1.000.000.000,00,00004003,15931<==
=====>>arResult[2]=8888,8888,Andrew Sands,0,00,29.078,78,1.000.000.000,00,00004005,15932<==
=====>>arResult[3]=1460,1460,C.A.S. Computer Application Sy,0,00,0,00,25,00,,22375<==
=====>>arResult[4]=1460,1460,C.A.S. Computer Application Sy,0,00,0,00,25,00,,22445<==
```

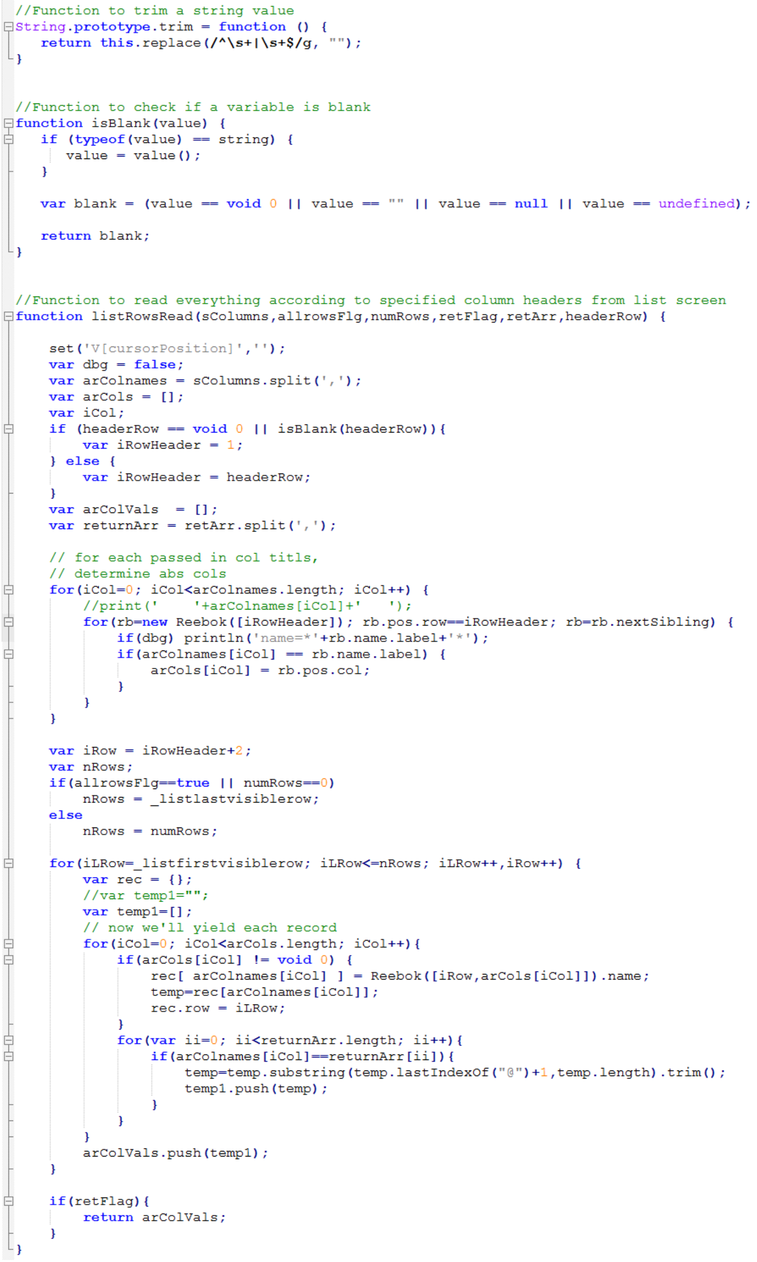
Liquid UI Code [Script]

User Interface



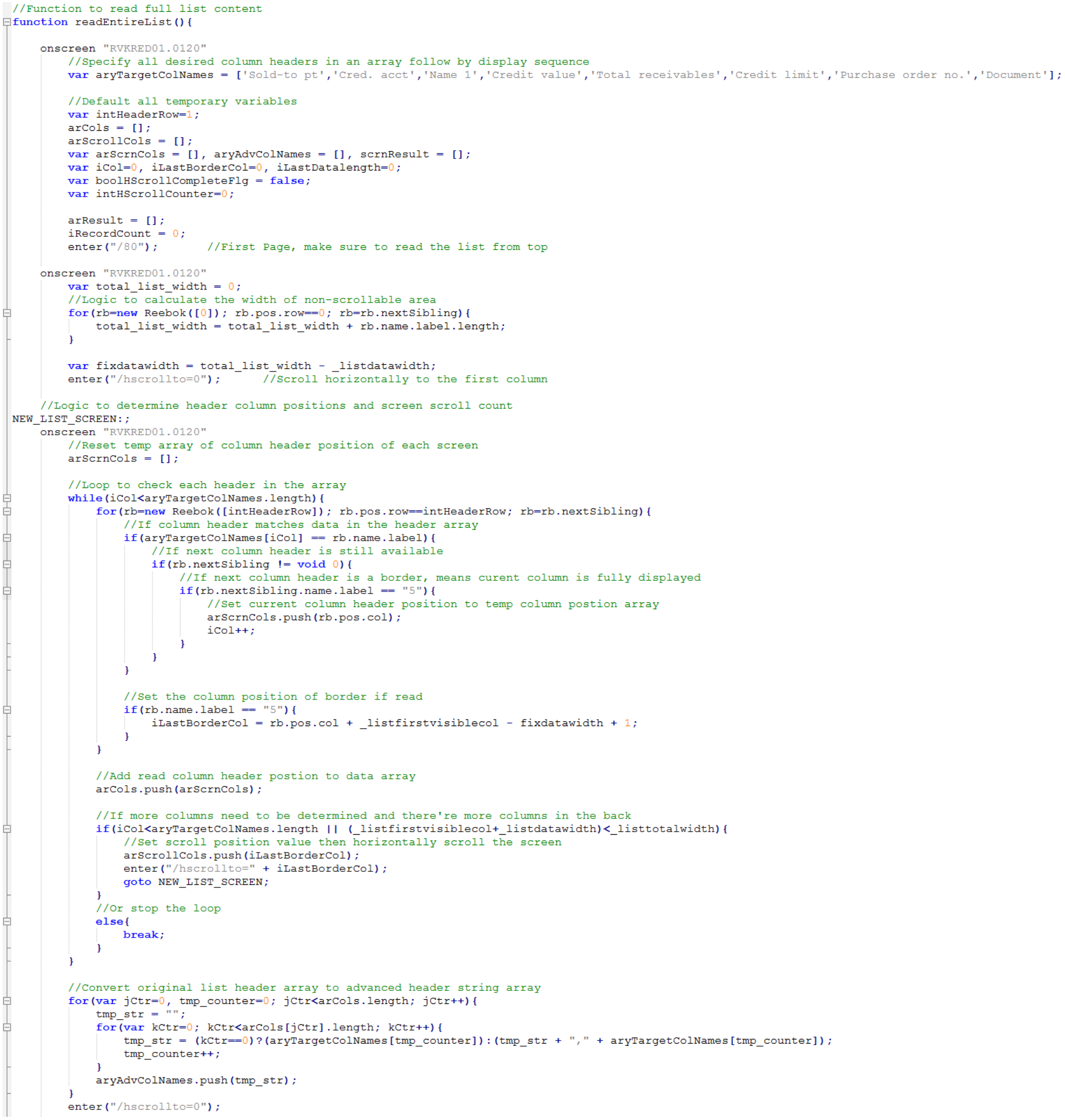
```
if(_transaction == "VKM1"){
    del("P[Release]");
    del("P[Check]");
    del("P[Reassign]");
    del("P[Reject]");
    del("P[Forward]");
    del("P[Forward to authorization]");
    del("P[Sort in ascending order]");
    del("P[Sort in descending order]");
    del("P[Select All]");
    del("P[Deselect All]");
    del("P[Choose]");
    del("P[Save]");
    del("P[Subtotal...]");
    del("P[Set filter]");
    del("P[Select, details]");
    del("P[Selections]");
    del("P[Add up values]");

    pushbutton([TOOLBAR], "@1V@Read Entire List", "?", {"process":readEntireList});
}
```

Generic Functions



```
//Function to trim a string value
String.prototype.trim = function () {
    return this.replace(/^\s+|\s+$/g, "");
}


//Function to check if a variable is blank
function isBlank(value) {
    if (typeof(value) == string) {
        value = value();
    }

    var blank = (value == void 0 || value == "" || value == null || value == undefined);

    return blank;
}


//Function to read everything according to specified column headers from list screen
function listRowsRead(sColumns,allrowsFlg,numRows,retFlag,retArr,headerRow) {

    set('V[cursorPosition]','');
    var dbg = false;
    var arColnames = sColumns.split(',');
    var arCols = [];
    var iCol;
    if (headerRow == void 0 || isBlank(headerRow)){
        var iRowHeader = 1;
    } else {
        var iRowHeader = headerRow;
    }
    var arColVals  = [];
    var returnArr = retArr.split(',');

    // for each passed in col titls,
    // determine abs cols
    for(iCol=0; iCol<arColnames.length; iCol++) {
        //print('    '+arColnames[iCol]+'   ');
        for(rb=new Reebok([iRowHeader]); rb.pos.row==iRowHeader; rb=rb.nextSibling) {
            if(dbg) println('name=*'+rb.name.label+'*');
            if(arColnames[iCol] == rb.name.label) {
                arCols[iCol] = rb.pos.col;
            }
        }
    }

    var iRow = iRowHeader+2;
    var nRows;
    if(allrowsFlg==true || numRows==0)
        nRows = _listlastvisiblerow;
    else
        nRows = numRows;

    for(iLRow=_listfirstvisiblerow; iLRow<=nRows; iLRow++,iRow++) {
        var rec = {};
        //var temp1="";
        var temp1=[];
        // now we'll yield each record
        for(iCol=0; iCol<arCols.length; iCol++){
            if(arCols[iCol] != void 0) {
                rec[ arColnames[iCol] ] = Reebok([iRow,arCols[iCol]]).name;
                temp=rec[arColnames[iCol]];
                rec.row = iLRow;
            }
            for(var ii=0; ii<returnArr.length; ii++){
                if(arColnames[iCol]==returnArr[ii]){
                    temp=temp.substring(temp.lastIndexOf("@")+1,temp.length).trim();
                    temp1.push(temp);
                }
            }
        }
        arColVals.push(temp1);
    }

    if(retFlag){
        return arColVals;
    }
}
```

Read List Screen Function



```
//Function to read full list content
function readEntireList(){

    onscreen "RVKRED01.0120"
        //Specify all desired column headers in an array follow by display sequence
        var aryTargetColNames = ['Sold-to pt','Cred. acct','Name 1','Credit value','Total receivables','Credit limit','Purchase order no.','Document'];

        //Default all temporary variables
        var intHeaderRow=1;
        arCols = [];
        arScrollCols = [];
        var arScrnCols = [], aryAdvColNames = [], scrnResult = [];
        var iCol=0, iLastBorderCol=0, iLastDatalength=0;
        var boolHScrollCompleteFlg = false;
        var intHScrollCounter=0;

        arResult = [];
        iRecordCount = 0;
        enter("/80");       //First Page, make sure to read the list from top

    onscreen "RVKRED01.0120"
        var total_list_width = 0;
        //Logic to calculate the width of non-scrollable area
        for(rb=new Reebok([0]); rb.pos.row==0; rb=rb.nextSibling){
            total_list_width = total_list_width + rb.name.label.length;
        }

        var fixdatawidth = total_list_width - _listdatawidth;
        enter("/hscrollto=0");      //Scroll horizontally to the first column

    //Logic to determine header column positions and screen scroll count
NEW_LIST_SCREEN:;
    onscreen "RVKRED01.0120"
        //Reset temp array of column header position of each screen
        arScrnCols = [];

        //Loop to check each header in the array
        while(iCol<aryTargetColNames.length){
            for(rb=new Reebok([intHeaderRow]); rb.pos.row==intHeaderRow; rb=rb.nextSibling){
                //If column header matches data in the header array
                if(aryTargetColNames[iCol] == rb.name.label){
                    //If next column header is still available
                    if(rb.nextSibling != void 0){
                        //If next column header is a border, means curent column is fully displayed
                        if(rb.nextSibling.name.label == "5"){
                            //Set current column header position to temp column postion array
                            arScrnCols.push(rb.pos.col);
                            iCol++;
                        }
                    }
                }

                //Set the column position of border if read
                if(rb.name.label == "5"){
                    iLastBorderCol = rb.pos.col + _listfirstvisiblecol - fixdatawidth + 1;
                }
            }

            //Add read column header postion to data array
            arCols.push(arScrnCols);

            //If more columns need to be determined and there're more columns in the back
            if(iCol<aryTargetColNames.length || (_listfirstvisiblecol+_listdatawidth)<_listtotalwidth){
                //Set scroll position value then horizontally scroll the screen
                arScrollCols.push(iLastBorderCol);
                enter("/hscrollto=" + iLastBorderCol);
                goto NEW_LIST_SCREEN;
            }
            //Or stop the loop
            else{
                break;
            }
        }

        //Convert original list header array to advanced header string array
        for(var jCtr=0, tmp_counter=0; jCtr<arCols.length; jCtr++){
            tmp_str = "";
            for(var kCtr=0; kCtr<arCols[jCtr].length; kCtr++){
                tmp_str = (kCtr==0)?(aryTargetColNames[tmp_counter]):(tmp_str + "," + aryTargetColNames[tmp_counter]);
                tmp_counter++;
            }
            aryAdvColNames.push(tmp_str);
        }
        enter("/hscrollto=0");
    // List of SD Documents
CONTINUE_TO_CHECK_LIST:;
    onscreen 'RVKRED01.0120'
        //If horizontal scroll is completed, continue to check vertical scroll
        if(boolHScrollCompleteFlg){
            goto CONTINUE_TO_V_SCROLL;
        }

        //Read detail based on current screen
        scrnResult = listRowsRead(aryAdvColNames[intHScrollCounter],true,0,true,aryAdvColNames[intHScrollCounter],intHeaderRow);

        //If the screen has not horizontal scrolled
        if(intHScrollCounter == 0){
            for(var jCtr in scrnResult){
                //If data matches bottom border line, set amount of record and stop adding new record to data array
                if(scrnResult[jCtr][0].substring(0,7) == '6444444' || scrnResult[jCtr][0].substring(0,7) == '1444444' || scrnResult[jCtr][0].substring(0,7) == '0444444'){
                    iRecordCount = arResult.length;
                    break;
                }

                //Add new record to data array
                arResult.push(scrnResult[jCtr]);
            }
        }
        //If the screen is horizontal scrolled
        else{
            //Add current screen detail to each row of new added records
            for(var iCtr=iLastDatalength, jCtr=0; iCtr<arResult.length; iCtr++, jCtr++)
                arResult[iCtr] = arResult[iCtr].concat(scrnResult[jCtr]);
        }

        //If the screen has not been horizontally scrolled
        if(!boolHScrollCompleteFlg){
            //If next scoll col position is still available
            if(intHScrollCounter < aryAdvColNames.length){
                //Increment the horizontal scroll counter, then scroll
                intHScrollCounter++;

                if(intHScrollCounter == aryAdvColNames.length){
                    //Mark horizontal scroll is completed and scroll back to 0 col
                    boolHScrollCompleteFlg = true;
                    enter("/hscrollto=0");
                    goto CONTINUE_TO_CHECK_LIST;
                }
                else{
                    enter("/hscrollto=" + arScrollCols[intHScrollCounter-1]);
                    goto CONTINUE_TO_CHECK_LIST;
                }
            }
            //If no next scroll col position
            else {
                //Mark horizontal scroll is completed and scroll back to 0 col
                boolHScrollCompleteFlg = true;
                enter("/hscrollto=0");
                goto CONTINUE_TO_CHECK_LIST;
            }
        }

CONTINUE_TO_V_SCROLL:;

        //If record counter is not set and read record length is less than system value, continue to read next page
        if((arResult.length < _listlastrow) && (iRecordCount==0)){

            //Reset counter and flag then go to next page of list screen
            intHScrollCounter = 0;
            boolHScrollCompleteFlg = false;
            iLastDatalength = arResult.length;
            enter("/82");        //Next Page
            goto CONTINUE_TO_CHECK_LIST;
        }
        //If record counter is set or read record length matches system value, means ends of list read
        else {
            for(var jCtr in arResult){
                println("=====>>arResult[" + jCtr + "]=" + arResult[jCtr] + "<==");
            }

            enter("?");
        }

}
```