Author: Carlos Kassab, E.Mail: laran.ikal@gmail.com, Date: 2019/June/06

Open Reporting System.

What is it?.

It is an easy way to get your own good looking reports with output comming directly from ERP LN report sessions as file downloads in Excel xlsx or PDF format.

Prerequisites:

- PowerShell(https://docs.microsoft.com/en-us/powershell/)
- ERP LN Windows Server
- VisualStudio Code IDE (https://code.visualstudio.com), For development machine.
- wkhtmltopdf (https://wkhtmltopdf.org/index.html)

Notes:

- Powershell must be at least version 5.
- Check PowerShell on the server and the development machine are the same versions. To check PowerShell version open a PowerShell console and run: \$PSVersionTable.PSVersion

Reports with Excel Output:

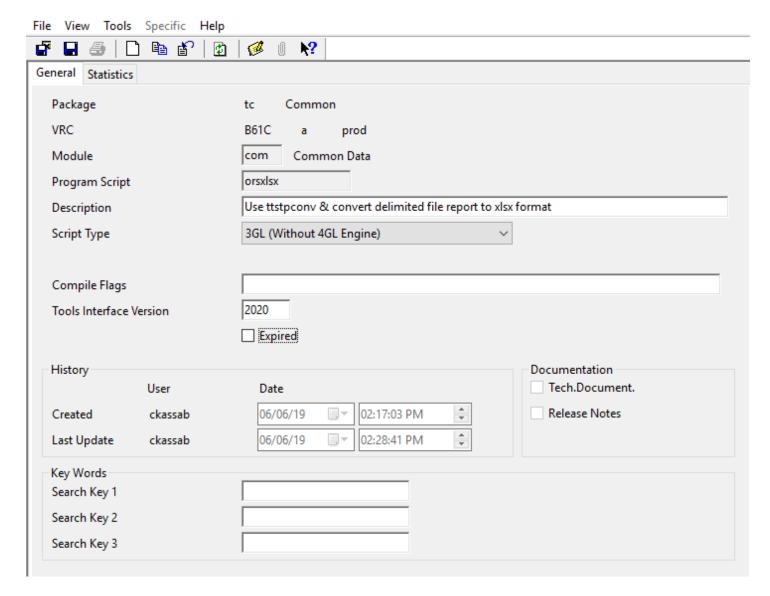
Install PowerShell ImportExcel Module:

- Open PowerShell Console as administrator
- Run this command: Find-Module ImportExcel | Install-Module

If when running this command, PowerShell asks to install nuget provider, answer yes \rightarrow Y It also may say that you are installing from untrusted repositories, answer yes to install the module.

Author: Carlos Kassab, E.Mail: laran.ikal@gmail.com, Date: 2019/June/06

Login to LN, navigate to Tools \rightarrow Application Development \rightarrow Program Scripts / Libraries, start session Program Scripts / Libraries and create the next library:



Tools interface version depends on your tools version.

Copy the below the script source code, save and compile your library:

```
* Title : tccomorsxlsx - ttstpcony report output delimited file to xlsx Excel format
              : Carlos Kassab
|* Author
* Date
              : June,06,2019
************************************
* Script Type
extern domain tcmcs.str132
                             program_name
extern domain tcmcs.str132
                             prog_arguments
              string
                             tmp.file1(1024), tmp.xlsx.file1(1024)
                                                                  | Temporary file names.
              string
                             tmp.file_r(1024)
                             local.path(1024)
                                                   | Client Location for report.
              string
                             file_to_email(1024)
              string
                             start(1024)
                                                   | Start command
              string
                             application(132)
                                                   | Application to start.
              string
              long
                             app id
                                                   | Application id.
                             device.parameters(50)
              string
extern domain tcmcs.long
                             retval
extern domain tcmcs.long
                             fileid1
extern domain tcmcs.long
                             err
                             monthnum(2),daynum(2)
                                                           Actual date
extern string
extern long
                             utc.func.ret, ret
                             _yearno, _monthno, _month_dayno, _hours, _minutes, _seconds
extern long
#pragma used dll ottdllbw
#pragma used dll ottdllhtml
#include <bic_desktop>
function main()
For development and testing, use temporary file tmp.file1 on development machine.
tmp.file1 = creat.tmp.file$( bse.tmp.dir$() )
tmp.xlsx.file1 = tmp.file1 & ".xlsx" | This will be the final converted file name.
 | Arguments description
 |message("argument1:" & argv$(1)) | tmp file created from report output
 |message("argument2:" & argy$(2)) | path to file in device setup
 |message("argument3:" & argv$(3)) | arguments parameter in device setup
 |message("argument4:" & argv$(4)) | This is wt value, still waiting to know description.
 |message("argument5:" & argv$(5)) | at this time this value is comming empty
wait.and.activate( "ttstpconv", argv$(1), tmp.file1, argv$(3), argv$(4))
 | Running script to convert delimited file to xlsx format, file name in variable tmp.xlsx.file1
program name = "/OpenReportingSystem/ExcelOutput/ORSEXCEL.bat"
prog_arguments = tmp.file1 & " " & tmp.xlsx.file1 & " " & spool.report
retval = run.prog( program_name, prog_arguments, RP_WAIT )
if job.process then
  utc.func.ret = utc.to.date(utc.num(), _yearno, _monthno, _month_dayno, _hours, _minutes, _seconds)
  monthnum = str$(_monthno)
  if _monthno < 10 then
  monthnum = "0" & str$(_monthno)
```

Author: Carlos Kassab, E.Mail: laran.ikal@gmail.com, Date: 2019/June/06

```
endif
 daynum = str$(_month_dayno)
 if _month_dayno < 10 then
  daynum = "0" & str$(_month_dayno)
 endif
 file_to_email = bse.tmp.dir$() & "/"
                                         & spool.report & "_" & monthnum & daynum & str$(_yearno) & ".xlsx"
 file.cp( tmp.xlsx.file1, file_to_email )
else
 if tc.is.html.ui() then
  client.download.file( tmp.xlsx.file1 )
 else
  local.path = "${BSE_TMP}/" & str$(utc.num()) & ".xlsx"
  err = server2client( tmp.xlsx.file1, local.path, false, false,false )
  if (err) then
   message("Could not copy file to client")
  else
   application = "excel.exe"
   start = application & " " & get.local.filename()
   app_id = start.application.local(start,0,err)
  endif
 endif
endif
retval = seq.unlink(tmp.file1) |For development and testing, comment this line to avoid file deletion.
```

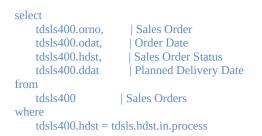
Author: Carlos Kassab, E.Mail: laran.ikal@gmail.com, Date: 2019/June/06

Create your printing device, login to LN and navigate to:
Tools → Device Management → Device Data run Device Data Session and add this device:

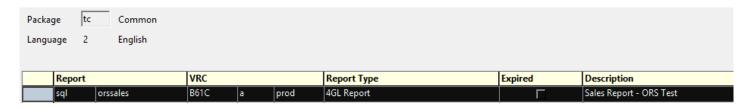
File View Tools S	Specific Help							
Papertype by D Sy	rstems							
Device	ORS_EXCEL ORS_EXCEL							
Description	xlsx Excel format file output							
Device Type	Rewrite file V							
Locale	ISO-8859-1 ▶ Western Europe - 8 bit							
	ntermediate File in XML Format							
Printer								
Driver								
Device Queue								
Paper Type								
Left Margin	0 Form Feed Every Page							
File								
Driver								
Shell Command								
4GL Program	otccomorsxlsx							
Argument	ASCII	Select Report Server						
		Select Device						
Path	\$\$ - ORS_EXCEL - \$\$ - \${LOGNAME}							
	☐ Change Output file allowed							
Page Length	0							

Author: Carlos Kassab, E.Mail: laran.ikal@gmail.com, Date: 2019/June/06

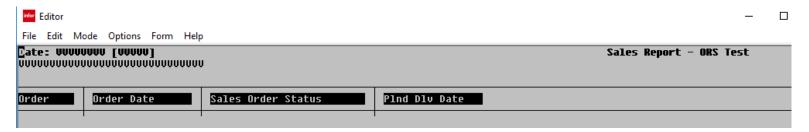
To test your new reporting system, create a query, name it tosqlorssales with the next sql code:



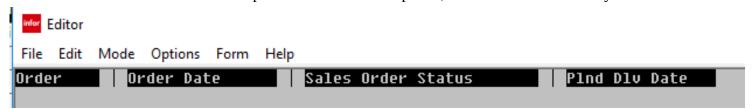
Once the report has been generated, go to maintain reports and find your new query report:



Right click on the report name and edit the layout header, it should be like this:

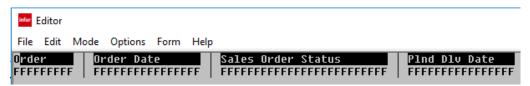


Delete all header lines but keep header column descriptions, it must be like this after your deletion:



Note. You must also resize the header to keep just the column descriptions line.

If you edit Header and Detail layouts, they must look like this:



Here we are creating the header and detail for our delimited file. Save, exit and compile your report.

Author: Carlos Kassab, E.Mail: laran.ikal@gmail.com, Date: 2019/June/06

Execute your sql query, you must have an output like this:

4	Α			В		С	D			
L	rtcsqlorssales									
2	Order	~	Order	Date	-	Sales Order Status	Plnd Dlv Date			
3		1675	02-16-	2018	11:58	In Process	06-01-2018 11:58			
Ļ		1729	04-10-	2018	08:36	In Process	10-10-2018 08:36			
j		1730	04-10-	2018	08:44	In Process	06-04-2019 08:44			
j		1731	04-10-	2018	08:46	In Process	07-30-2018 08:46			
7		1843	10-10-	2018	09:13	In Process	12-26-2018 09:13			
3		1850	10-15-	2018	08:55	In Process	12-27-2018 08:55			
)		1858	10-22-	2018	08:28	In Process	05-05-2019 08:28			

All sources and documentation can be found on github: https://github.com/LaranIkal/OpenReportingSystem