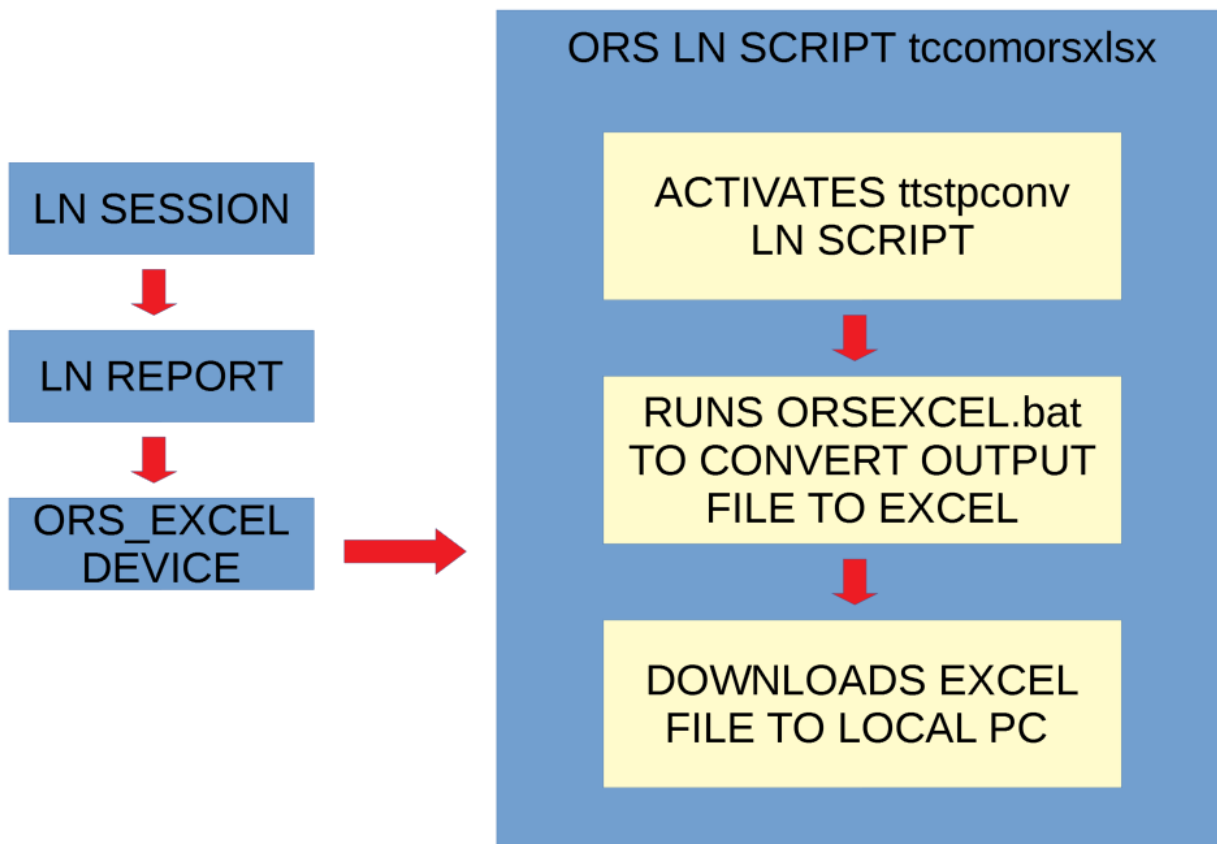


## Open Reporting System.

### What is it?.

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

## HOW ORS WORKS



## ORS INSTALL

### 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 → Y  
It also may say that you are installing from untrusted repositories, answer yes to install the module.

Download OpenReportingSystem as a zip file from github:

<https://github.com/LaranIkal/OpenReportingSystem>

When opening the zip file, there is a main folder: OpenReportingSystem-master

Open OpenReportingSystem-master folder and copy folder OpenReportingSystem to your root c:\

Now you have installed Open Reporting System. Follow the next steps.

For any help, you can reach me at my email in the header of this document or in baan board:

Author: Carlos Kassab, E.Mail: [laran.ikal@gmail.com](mailto:laran.ikal@gmail.com), Date: 2019/June/06

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

File View Tools Specific Help

General Statistics

Package tc Common

VRC B61C a prod

Module com Common Data

Program Script orslsx

Description Use ttstpconv & convert delimited file report to xlsx format

Script Type 3GL (Without 4GL Engine)

Compile Flags

Tools Interface Version 2020

☐ Expired

History		User	Date
Created	ckassab	06/06/19	02:17:03 PM
Last Update	ckassab	06/06/19	02:28:41 PM

Documentation

☐ Tech.Document.

☐ Release Notes

Key Words

Search Key 1

Search Key 2

Search Key 3

Tools interface version depends on your tools version.

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

```
*****
* Title   : tccomorsxlsx - ttstpconv report output delimited file to xlsx Excel format
* Author   : Carlos Kassab
* Date     : June,06,2019
*****
* Script Type : 3GL
*****
***** DECLARATION SECTION *****

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)
                           string      local.path(1024)      | Client Location for report.
                           string      file_to_email(1024)
                           string      start(1024)           | Start command
                           string      application(132)       | Application to start.
                           long        app_id                 | Application id.
                           string      device.parameters(50)

extern domain tcmcs.long    retval
extern domain tcmcs.long    fileid1
extern domain tcmcs.long    err
extern string               monthnum(2),daynum(2)            |Actual date
extern long                 utc.func.ret, ret
extern long                 _yearno, _monthno, _month_dayno, _hours, _minutes, _seconds

#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$(1)) | tmp file created from report output
|message("argument2:" & argv$(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](mailto: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](mailto: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 Specific Help

Papertype by D... Systems

Device	ORS_EXCEL
Description	xlsx Excel format file output
Device Type	Rewrite file
Locale	ISO-8859-1 Western Europe - 8 bit
	<input type="checkbox"/> Intermediate File in XML Format

Printer

Driver	
Device Queue	
Paper Type	
Left Margin	0 <input type="checkbox"/> Form Feed Every Page

File

Driver	
Shell Command	
4GL Program	otccomorsxlsx
Argument	ASCII
Path	\$\$ - ORS_EXCEL - \$\$ - \${LOGNAME}
	<input type="checkbox"/> Change Output file allowed
Page Length	0

Select Report Server

Select Device

Run your sql query and use the new device: ORS\_EXCEL, you must have an output like this:

	A	B	C	D
	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
4	1729	04-10-2018 08:36	In Process	10-10-2018 08:36
5	1730	04-10-2018 08:44	In Process	06-04-2019 08:44
6	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
8	1850	10-15-2018 08:55	In Process	12-27-2018 08:55
9	1858	10-22-2018 08:28	In Process	05-05-2019 08:28

## Reports with PDF Output:

This is mainly used for reports with imagens and special design like invoices and checks.

Download OpenReportingSystem as a zip file from github if you have not done it yet:

<https://github.com/LaranIkal/OpenReportingSystem>

Whe opening the zip file, there is a main folder: OpenReportingSystem-master

Open OpenReportingSystem-master folder and copy folder OpenReportingSystem to your root c:\

Now you have installed Open Reporting System. Follow the next steps.

For any help, you can reach me at my email in the header of this document or in baan board:



Author: Carlos Kassab, E.Mail: [laran.ikal@gmail.com](mailto:laran.ikal@gmail.com), Date: 2019/June/06

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

File View Tools Specific Help

General Statistics

Package tc Common

VRC B61C a prod

Module com Common Data

Program Script orspdf

Description Use ttstpconv & convert html file report to pdf format

Script Type 3GL (Without 4GL Engine) v

Compile Flags

Tools Interface Version 2020

☐ Expired

History

	User	Date
Created	ckassab	06/08/19 08:30:46 AM
Last Update	ckassab	06/08/19 01:09:38 PM

Key Words

Search Key 1

Search Key 2

Search Key 3

Documentation

☐ Tech.Document.

☐ Release Notes

Tools interface version depends on your tools version.

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

\*\*\*\*\*

\* Title : tccomorspdf - ttstpconv report output html file to PDF format

\* Author : Carlos Kassab

\* Date : June,06,2019

\*\*\*\*\*

\* Script Type : 3GL

\*\*\*\*\* DECLARATION SECTION \*\*\*\*\*

```
extern domain tcmcs.str132    program_name
extern domain tcmcs.str132    prog_arguments
                                string      tmp.file1(1024), tmp.pdf.file1(1024)      | Temporary file names.
                                string      tmp.file_r(1024)
                                string      local.path(1024)      | Client Location for report.
                                string      file_to_email(1024)
                                string      start(1024)      | Start command
                                string      application(132)    | Application to start.
                                long        app_id      | Application id.
                                string      device.parameters(50)

extern domain tcmcs.long      retval
extern domain tcmcs.long      fileid1
extern domain tcmcs.long      err
extern string                  monthnum(2),daynum(2)      |Actual date
extern long                    utc.func.ret, ret
extern long                    _yearno, _monthno, _month_dayno, _hours, _minutes, _seconds
```

#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.pdf.file1 = tmp.file1 & ".pdf" | This will be the final converted file name.

| Arguments description

|message("argument1:" & argv\$(1)) | tmp file created from report output

|message("argument2:" & argv\$(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

tmp.file1 = tmp.file1 & ".html"

wait.and.activate( "ttstpconv", argv\$(1), tmp.file1, argv\$(3), argv\$(4) )

| Running script to convert html file to pdf format, file name in variable tmp.pdf.file1

program\_name = "/OpenReportingSystem/PDFOutput/ORSPDF.bat"

prog\_arguments = tmp.file1 & " " & tmp.pdf.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](mailto: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) & ".pdf"
file.cp( tmp.pdf.file1, file_to_email )

else

if tc.is.html.ui() then

    client.download.file( tmp.pdf.file1 )

else

    local.path = "${BSE_TMP}/" & str$(utc.num()) & ".pdf"

    err = server2client( tmp.pdf.file1, local.path, false, false,false )

    if (err) then
        message("Could not copy file to client")
    else
        |https://www.sumatrapdfreader.org/free-pdf-reader.html
        | Set the next line according to your app path in the client machine.
        application = "/OpenReportingSystem/PDFOutput/SumatraPDF.exe"
        start = application & " " & get.local.filename()
        app_id = start.application.local(start,0,err)
    endif

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](mailto: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:

Papertype by D... Systems

Device	ORS_PDF
Description	PDF format file output
Device Type	Rewrite file
Locale	ISO-8859-1 Western Europe - 8 bit
	<input type="checkbox"/> Intermediate File in XML Format
Printer	
Driver	
Device Queue	
Paper Type	
Left Margin	0 <input type="checkbox"/> Form Feed Every Page
File	
Driver	
Shell Command	
4GL Program	otccomorspdf
Argument	ASCII
	<input type="button" value="Select Report Server"/>
	<input type="button" value="Select Device"/>
Path	\$\$ - ORS_PDF - \$\$ - \${LOGNAME}
	<input type="checkbox"/> Change Output file allowed
Page Length	155

Modify ttaad3100s000 000

Author: Carlos Kassab, E.Mail: [laran.ikal@gmail.com](mailto:laran.ikal@gmail.com), Date: 2019/June/06

## ORS\_PDF device testing:

1.- let's copy out firs query tcsqlorssales to tcsqlorsslspdf, now we have our new query:

Query | Documentation

Query: orsslspdf

Owner:

Description: Sales Report - ORS PDF

Created by: ckassab

Date: 06/06/19

Output Type: Report list

Report: tcsqlorsslspdf

Print Options: A3-SMALL-MANUAL

Columns label:

Rows label:

Labels horizontal:

Query Select Statement

```
select
    tds400.orno, | Sales Order
    tds400.odat, | Order Date
    tds400.hdst, | Sales Order Status
    tds400.ddat  | Planned Delivery Date
from
    tds400      | Sales Orders
where
    tds400.hdst = tds400.in.process
```

Generate the query report in order to have our report like this:

File | Edit | View | Group | Tools | Specific | Help

Package: tc Common

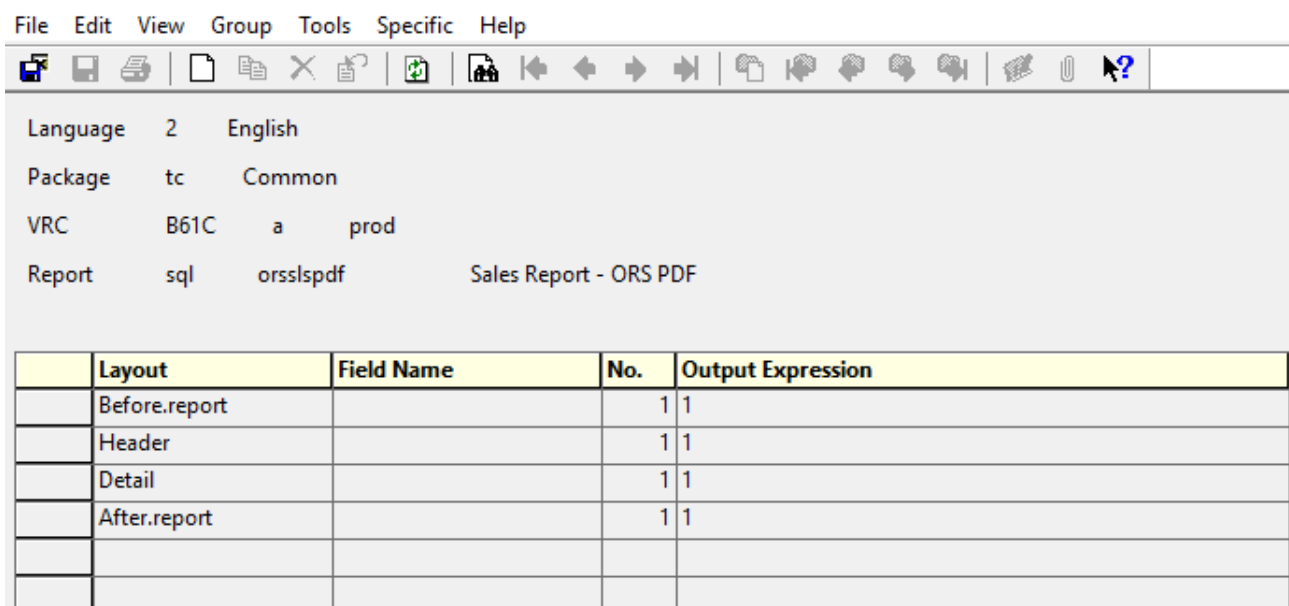
Language: 2 English

Report	VRC	Report Type	Expired	Description
sql orssales	B61C a prod	4GL Report	<input type="checkbox"/>	Sales Report - ORS Test
sql orsslspdf	B61C a prod	4GL Report	<input type="checkbox"/>	Sales Report - ORS PDF

Right click on the orsslspdf report and go to layouts.

Insert two new layouts in order to have it like this:

Author: Carlos Kassab, E.Mail: [laran.ikal@gmail.com](mailto:laran.ikal@gmail.com), Date: 2019/June/06



Go to menu Specific → Edit All Layouts

Edit your report layouts in order to look like this:

[illegible]

Author: Carlos Kassab, E.Mail: [laran.ikal@gmail.com](mailto:laran.ikal@gmail.com), Date: 2019/June/06

[illegible]

```
File Edit Mode Options Form Help
<tr>
  <td>FFFFFFF </td><td> FFFFFFFF </td><td> <b>FFFFFFFFFFFFFFFFFFFFFF</b> </td><td> FFFFFFFF </td>
</tr>
```

LAYOUT: Detail 1 LINES: 5

ottadvformedit col:126/row: 1

```
File Edit Mode Options Form Help
</table>
</body>
</html>
LAYOUT: after_report 1 LINES: 1
ottadvformedit col: 19/row: 1
```

Save and compile your report.

Run and print your query using device: ORS\_PDF

Your pdf file should look like below.

VERY IMPORTANT NOTE: Pay attention to the logo size, it impacts page size.

## Sales Good Looking Report



Date: 06-08-2019 [20:48] Sales Report - ORS PDF Page: 1  
My Great Company comp: 999

Order	Order Date	Sales Order Status	Plnd Dlv Date
10001675	02-16-2018 11:58	In Process	06-01-2018 11:58
10001729	04-10-2018 08:36	In Process	10-10-2018 08:36
10001730	04-10-2018 08:44	In Process	06-04-2019 08:44
10001731	04-10-2018 08:46	In Process	07-30-2018 08:46
10001843	10-10-2018 09:13	In Process	12-26-2018 09:13
10001850	10-15-2018 08:55	In Process	12-27-2018 08:55
10001858	10-22-2018 08:28	In Process	05-05-2019 08:28
10002056	01-31-2019 08:24	In Process	05-24-2019 08:24
10002115	03-04-2019 09:57	In Process	06-01-2019 09:57
10002214	04-30-2019 08:32	In Process	04-30-2019 08:32
10002215	04-30-2019 08:33	In Process	04-30-2019 08:33
20002027	01-10-2018 16:17	In Process	03-29-2018 16:17
20002218	01-16-2019 07:25	In Process	03-14-2019 13:34
20002219	01-16-2019 07:26	In Process	03-12-2019 09:24
20002225	02-01-2019 07:16	In Process	03-26-2019 13:04
20002226	02-01-2019 07:18	In Process	03-12-2019 10:58
20002229	02-13-2019 08:12	In Process	03-08-2019 07:56
20002230	02-13-2019 08:14	In Process	04-05-2019 09:24
20002237	02-27-2019 09:45	In Process	04-23-2019 13:04
20002240	03-13-2019 07:47	In Process	05-03-2019 13:21
20002242	03-13-2019 07:53	In Process	05-07-2019 09:24
20002246	04-01-2019 15:07	In Process	05-29-2019 15:07
20002248	04-14-2019 13:12	In Process	06-04-2019 08:06
20002249	04-14-2019 13:14	In Process	06-04-2019 09:43
20002250	04-17-2019 07:32	In Process	06-12-2019 00:00
20002251	05-01-2019 07:34	In Process	06-10-2019 10:58
20002254	05-08-2019 07:42	In Process	07-01-2019 13:04
20002255	05-08-2019 07:44	In Process	07-01-2019 09:24
20002258	05-09-2019 11:18	In Process	06-06-2019 11:18
20002260	05-13-2019 09:36	In Process	05-24-2019 09:36
20002262	05-21-2019 07:57	In Process	05-21-2019 09:36
30012672	11-03-2015 08:39	In Process	01-26-2016 08:39



Author: Carlos Kassab, E.Mail: [laran.ikal@gmail.com](mailto:laran.ikal@gmail.com), Date: 2019/June/06

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