

```

--
Values(http_Get('http://bhadev.rzkh.de:10010/web/services/CommonSQL/BHA2025SYSTOOLS'
));

CL:CHGLIBL LIBL(QTEMP EPS006BHA COMSQLQRY SYSTOOLS QGPL);

-----
=====
-- Services Info
-----
=====
Select *
  From Services_Info
--  Where Service_Category = 'SPOOL'
--  Where Service_Schema_Name = 'SYSTOOLS'
  Order By Service_Category, Service_Name;

-----
=====
-- Calling User Defined Table Function
-----
=====
-- UDTF without parameters
-----
-----
Select * from Table(USERS()) u;
Select * from Table(Schemas()) x;

-----
-----
-- Using WHERE conditions with UDTFs
-----
-----
Select * from Table(USERS()) u
  Where ODOBNM like '%HAUS%';

  Select * from Table(Message_File_Data('QSYS', 'QSQLMSG'))
  Where Message_Id like 'SQL015%';

-----
-----
-- UDTF with parameters: ListMember_Fnc (List Member)
-----
-----
Select * from Table(COMDBPGM.ListMember_Fnc('QSQLPGM', 'COMDBPGM')) x
  Where      MbrType like '%RPGLE%'
  and Mbr    like '%LIST%'
;

```

```

Select *
  from SysPartitionStat
 Where   Table_Schema = 'COMDBPGM'
        and Table_Name   = 'QSQLPGM'
        and Table_Partition Like '%LIST%'
        and Source_Type   Like '%RPGLE%';

```

-- 1.3. Joining UDTFs

```

-----*
Select a.Object_Name, a.Object_Type, a.Object_Attribute,
      b.Object_Name, b.Object_Type, b.Object_Attribute
  from Table(QSYS2.Save_FILE_Objects(Save_File => 'COMDBMOD', Save_File_Library =>
'BHATRANS')) a
     Full Join
     Table(QSYS2.Save_FILE_Objects(Save_File => 'COMSQLQRY', Save_File_Library =>
'BHATRANS')) b
     on   a.Object_Name = b.Object_Name
        and a.Object_Type = b.Object_Type
        and a.Object_Attribute = b.Object_Attribute
 Order By a.Object_Name, a.Object_Type, a.Object_Attribute,
        b.Object_Name, b.Object_Type, b.Object_Attribute
        ;

```

-- 1.4. Merging UDTFs

```

-----*
Select ObjLib, ObjName, ObjType --, a.*
  from Table(Qsys2.OBJECT_STATISTICS(Object_Schema => 'BXOBJ',
                                     Objtypelist  => '*PGM *SRVPGM',
                                     Object_Name   => '*ALLSIMPLE')) a

Union All
-- Except
  Select ObjLib, ObjName, ObjType --, b.*
  from Table(Qsys2.OBJECT_STATISTICS(Object_Schema => 'BXOBJOLD',
                                     Objtypelist  => '*PGM *SRVPGM',
                                     Object_Name   => '*ALLSIMPLE')) b

Order By ObjName, ObjLib, ObjType;

```

-- 1.5. Joining UDTFs using Lateral

```

-----*
Select *
  from Table(Systools.Related_Objects(Library_Name => 'DIRWEB',
                                     File_Name     => 'DWWEBAPP'));

Select Table_Schema as PFTable_Schema, Table_Name as PFTable,
      Schema_Name as LFSchema,      SQL_Name as LFSQLName, System_Name as
LFSYSName,
      b.*
  from SysTables a cross join

```

```

        Lateral(Select *
                from Table(Systools.Related_Objects(Library_Name =>
a.Table_Schema,
                File_Name => a.Table_Name))
                Where Table_Name like 'DWWEB%' ) b
;

```

```

Select Table_Schema as PFTable_Schema, Table_Name as PFTable,
       Schema_Name as LFSchema,      SQL_Name as LFSQLName, System_Name as
LFSYSName
  from SysTables a cross join
       Lateral(Select *
                from Table(Systools.Related_Objects(Library_Name =>
a.Table_Schema,
                File_Name => a.Table_Name))
                Where      ( SQL_Object_Type like '%LOGICAL%'
                           or SQL_Object_Type Like 'INDEX%')
                           and Schema_Name <> a.Table_Schema)
  Where a.Table_Schema = 'DIRWEB'
        and Table_Type in ('P', 'T');

```

```

-----
=====

```

```

-- List Spooled Files - SPOOLED_FILE_INFO UDTF
-----
=====

```

```

-- List All Spoolfiles of BHAPGMR

```

```

Select *
  from Table(QSYS2.SPOOLED_FILE_INFO(User_Name => 'BHAPGMR'));

```

```

-- List All Spoolfiles older than 1 Day of User Hauser in Status SAVED

```

```

Select *
  from Table(QSYS2.SPOOLED_FILE_INFO(User_Name      => 'HAUSER'
                                     --            , Ending_Timestamp => Current_Timestamp - 1
Day
                                     --            , Status           => 'SAVED'
                                     ));

```

```

-- Spoolfiles Overview

```

```

Select Count(*)           "Nbr Spoolfiles",
       Count(Distinct Spooled_File_name) "Nbr Diff
Spoolfiles",
       Count(Distinct Qualified_Job_Name) "Nbr Diff Jobs",
       VarChar_Format(Sum(Size)/1000000,0, '999,999,990.00') "Size in MB",
       VarChar_Format(Sum(Total_Pages), '9,999,990') "Nbr Pages",
       Count(Distinct Output_Queue_Library concat Output_Queue) "Nbr OutQueues",
       Count(Distinct Date(Creation_Timestamp)) "Diff Days",

```

```

        Count(Case When Date(Creation_Timestamp) < Current_Timestamp - 1 Month
                Then 1 End)                                "Nbr Spoolfiles > 1
Month",
        Count(Case When Date(Creation_Timestamp) < Current_Timestamp - 1 Month
                and Status <> 'SAVED'
                Then 1 End)                                "Nbr Spoolfiles > 1
Month not Saved"
    from Table(QSYS2.Spoiled_File_Info(User_Name => '*ALL'));

-- Spoolfiles Overview per User
Select Job_User,
       Count(*)                                           "Nbr Spoolfiles",
       Count(Distinct Spoiled_File_name)                 "Nbr Diff Spoolfiles",
       Count(Distinct Qualified_Job_Name)               "Nbr Diff Jobs",
       -- VarChar_Format(Sum(Size)/1000000,0, '999,999,990.00')   "Size in MB",
       -- VarChar_Format(Sum(Total_Pages), '9,999,990')           "Nbr Pages",
       Sum(Size/1000000,0) "Size in MB",
       sum(Total_Pages) "Nbr Pages",
       Count(Distinct Output_Queue_Library concat Output_Queue) "Nbr OutQueues",
       Count(Distinct Date(Creation_Timestamp))           "Diff Days",
       Count(Case When Date(Creation_Timestamp) < Current_Timestamp - 1 Month
               Then 1 End)                                "Nbr Spoolfiles > 1
Month",
       Count(Case When Date(Creation_Timestamp) < Current_Timestamp - 1 Month
               and Status <> 'SAVED'
               Then 1 End)                                "Nbr Spoolfiles > 1
Month not Saved"
    from Table(QSYS2.Spoiled_File_Info(User_Name => '*ALL'))
-- Group By Job_User
-- Order By "Size in MB" Desc;
Group By RollUp(Job_User)
Order By Job_User;

-----
-----
-- OUTPUT_QUEUE_ENTRIES
-----
-----

Select * from Table(QSYS2.Output_Queue_Entries(Outq_Lib      => Default,
                                                Outq_Name      => 'QEZDEBUG',
                                                Detailed_Info => 'NO',
                                                Ignore_Errors => Default))
;

Select Spoiled_File_Name, File_Number,      User_Name, 'QPRINT' Output_Queue_Name,
User_Data,
       Total_Pages,      Create_Timestamp, Job_Name
    from Table(QSYS2.Output_Queue_Entries(Outq_Lib      => '*LIBL',

```

```

                                Outq_Name    => 'QPRINT',
                                Detailed_Info => 'YES'))
Where   User_Name in ('BHAPGMR', 'HAUSER')
       and User_Data = 'CRTSRVPGM'
       and Status    = 'READY'
;

Select Spooled_File_Name, File_Number,      User_Name, Output_Queue_Name, User_Data,
       Total_Pages,      Create_Timestamp, Job_Name
from   QSYS2.Output_Queue_Entries_Basic a
Where   User_Name in ('BHAPGMR', 'HAUSER')
       and User_Data = 'CRTSRVPGM'
       and Status    = 'READY';

Select Spooled_File_Name, File_Number,      User_Name, Output_Queue_Name, User_Data,
       Total_Pages,      Create_Timestamp, Job_Name
from   QSYS2.Output_Queue_Entries a
Where   User_Name in ('BHAPGMR', 'HAUSER')
       and User_Data = 'CRTSRVPGM'
       and Status    = 'READY'
--     and Output_Queue_Name = 'QPRINT'
--     and Output_Queue_Library_Name = 'QGPL'
;

=====
-- Delete Spool Files with QCMDEXC
=====

Select Spooled_File_Name, File_Number,      User_Name, Output_Queue_Name, User_Data,
       Total_Pages,      Create_Timestamp, Job_Name,
       QCMDEXC('DLTSPLF FILE('  concat Spooled_File_Name concat ')') concat
              ' JOB('          concat Job_Name             concat ')') concat
              ' SPLNBR('      concat File_Number          concat ')') concat
              ' SELECT('      concat User_Name             concat ')')
-- , a.*
from   Output_Queue_Entries_Basic a
Where   User_Name = 'HAUSER'
       and Create_Timestamp <= Current_Timestamp - 50 Day
;

With x as (Select Spooled_File_Name, File_Number,      User_Name, Output_Queue_Name,
User_Data,
           Total_Pages,      Create_Timestamp, Job_Name,
           QCMDEXC('DLTSPLF FILE('  concat Spooled_File_Name concat ')')
concat
           ' JOB('          concat Job_Name             concat ')')
concat
           ' SPLNBR('      concat File_Number          concat ')')
concat
           ' SELECT('      concat User_Name             concat ')')

```

```

ExcCL
        -- , a.*
        from Output_Queue_Entries_Basic a
        Where      User_Name = 'HAUSER'
                and Create_Timestamp <= Current_Timestamp - 50 Day)
Select Count(*)                               Total_Selected,
        Count(Case When EXCCL = 1 Then 1 End) Deleted,
        Count(Case When EXCCL < 0 Then 1 End) Not_Deleted
From x;
=====
-- Delete Old Spoolfiles
=====
-- Call SysTools.Delete_Old_Spoiled_Files(Delete_Older_Than      => Default,
--                                          P_Output_Queue_Library_Name => Default,
--                                          P_Output_Queue_Name       => Default,
--                                          P_User_Name                => 'HAUSER',
--                                          Preview                    => Default);

Call SysTools.Delete_Old_Spoiled_Files(Delete_Older_Than      =>
Current_Timestamp - 20 Day,
                                          P_User_Name          => 'HAUSER',
                                          Preview                => 'YES');

-----
=====
-- Generate PDF
-----
=====
-- Generate Spoolfiles --> Execute BHAP30CAL (EPS006BHA)
CL: Call BHAP30CAL;

Values(Generate_PDF(Job_Name              => -- '814156/HAUSER/QPRTJOB',
                                          '151639/HAUSER/QPRTJOB',
                                          Spooled_File_Name      => 'PRTRPGFR02',
                                          Spooled_File_Number   => 3,
                                          Path_Name              =>
'/home/Hauser/Examples_SysTools/BHAP300PM_2026-07-02.pdf'));
        -- Path_Name              =>
'/home/Hauser/Examples_SysTools/BHAP300PM_' concat Char(Current_Date, ISO) concat
'.pdf'));

        Commit;

Select Generate_PDF(Job_Name              => a.Job_Name,

```

```

        Spooled_File_Name => a.Spoiled_File_Name,
        Spooled_File_Number => a.File_Number,
        Path_Name => '/home/Hauser/Examples_SysTools/' concat
        Trim(a.Spoiled_File_Name) concat
'_ ' concat
        Right(Digits(a.File_Number), 2) concat
'_ ' concat
        Case When a.User_Data > ''
            Then Trim(a.User_Data) concat
        Else ''
        End concat
        Char(Current_Date, ISO) concat
'_ ' concat
        Replace(a.Job_Name, '/', '-') concat
'.pdf') GenPDF,
        Spooled_File_Name, File_Number, User_Name, User_Data,
        Total_Pages, Create_Timestamp, Job_Name
from Table(QSYS2.Output_Queue_Entries(Outq_Lib => Default,
        Outq_Name => 'QPRINT')) a
Where User_Name = 'HAUSER'
and User_Data like 'BHAP30%'
and Status = 'READY'
and Job_Name = -- '814156/HAUSER/QPRTJOB'
                '151639/HAUSER/QPRTJOB'
-- and Job_Name = Qsys2.Job_Name
Order By Create_Timestamp Desc, User_Data;

-----
-----
-- System Global Variable JOB_NAME
-----
-----
Values(Job_Name);

-----
-----
-- View with Global Variables
-----
-----
Create Or Replace Variable COMSQLQRY.GblPath
    VarChar(256)
    Default '/home/Hauser/';

Create Or Replace Variable COMSQLQRY.GblUser
    VarChar(10)
    Default Session_User;

```

```
Create Or Replace Variable COMSQLQRY.GblUsrDta
  VarChar(10)
  Default '';
```

```
Create Or Replace Variable COMSQLQRY.GblOutQ
  VarChar(10)
  Default 'QPRINT';
```

```
Create Or Replace Variable COMSQLQRY.GblJobQual
  VarChar(28)
  Default Job_Name;
```

```
Create Or Replace Variable COMSQLQRY.GblSplF
  VarChar(10)
  Default '';
```

```
Create Or Replace Variable COMSQLQRY.GblSplFNo
  Integer
  Default 0;
```

```
Commit;
Stop;
```

```
-----
-----
-- Generate PDF Examples
-----
```

```
cl: Call BHAP30CAL;
```

```
-- Display Output Queue Entries
-----
```

```
Select *
  from Table(QSYS2.Output_Queue_Entries(Outq_Lib      => Default,
                                         Outq_Name    => GblOutQ)) a
  Where Create_Timestamp >= Current_Timestamp - 4 Hours
  ;
```

```
-- Set Global Variables
-----
```

```
Set GblPath      = '/Home/Hauser/Examples_SysTools',
  GblOutQ        = 'QPRINT',
  GblUser         = Session_User,
  GblJobQual     = -- '814156/HAUSER/QPRTJOB'
                  -- '151639/HAUSER/QPRTJOB'
                  '151639/HAUSER/QPRTJOB'
                  -- '834148/HAUSER/QPRTJOB'
  ;
```

```
Stop;
```

Commit;

-- Create View

Create Or Replace View COMSQLQRY.VWCRTPDF as
With x as (Select SysTools.Generate_PDF(

Job_Name => a.Job_Name,
Spooled_File_Name => a.Spooled_File_Name,
Spooled_File_Number => a.File_Number,
Path_Name => '/' concat

RTrim(LTrim(GblPath, '/'), '/') concat '/' concat

Trim(a.Spooled_File_Name)

concat '_' concat

Right(Digits(a.File_Number),

2) concat '_' concat

Case When a.User_Data > ''
Then Trim(a.User_Data)

concat '_'

Else ''

End

concat

Char(Current_Date, ISO)

concat '_' concat

Replace(a.Job_Name, '/', '-')

concat '.pdf') GenPDF,

a.*,

'PDF for SpoolFile ' concat Trim(Spooled_File_Name) concat

' SpoolFileNo ' concat File_Number concat

' User ' concat Trim(User_Name) concat

' UserData ' concat Trim(User_Data) concat

' TotalPages ' concat Total_Pages concat

' Job ' concat Trim(Job_Name) concat

' SpoolCreation ' concat Create_Timestamp as MsgText

from Table(QSYS2.Output_Queue_Entries(Outq_Lib => Default,
Outq_Name => GblOutQ)) a

Where User_Name = Upper(Trim(GblUser))

and User_Data like Upper(Trim(GblUsrDta)) concat '%'

and Job_Name = GblJobQual

and Status = 'READY')

Select x.*,

Trim(MsgText) concat Case When GenPDF = 1 Then ' successfully generated'
Else ' not

generated'

End as MsgTxt2

From x;

Commit;


```
Spooled_File_Number => 4));
```

```
-----  
-----  
-- Read needed Spoolfile information, split them into columns and convert into  
numeric values (if necessary)
```

```
--> Create View  
-----  
-----
```

```
Set GblJobQual = '814156/HAUSER/QPRTJOB',  
    GblSplf     = 'PRTRPGFR02',  
    GblSplfNo  = 69;
```

```
Set GblJobQual = --'788336/HAUSER/QPADEV0002',  
                 '151639/HAUSER/QPRTJOB',  
    GblSplf     = 'PRTRPGFR02',  
    GblSplfNo  = 4;
```

```
select * from Library_List_Info;
```

```
-- Create Or Replace View COMSQLQRY.VWSplfDta as
```

```
With a as (Select * from Table(SysTools.Spooled_File_Data(Job_Name => GblJobQual,  
                                                           Spooled_File_Name      => GblSplf,  
                                                           Spooled_File_Number    => GblSplfNo))),
```

```
    -- Heading
```

```
    x1 as ( Select Ordinal_Position, Spooled_Data, Row_Number() Over(Order By  
Ordinal_Position) LfdX1,
```

```
                Lead(Ordinal_Position, 1, 999999999) Over(Order By
```

```
Ordinal_Position) MaxOrd
```

```
    from a
```

```
    Where Left(Spooled_Data, 5) = ' Jahr'),
```

```
    -- Sales Information
```

```
    X2 as (Select Ordinal_Position, Spooled_Data,  
                Left(Spooled_Data, 10) SalesYear,
```

```
                Try_Cast(Replace(Trim(Substr(Spooled_Data, 11, 26)), '.', '')) as  
Dec(31, 2))
```

```
                Sales,
```

```
                Try_Cast(Replace(Trim(Substr(Spooled_Data, 37, 29)), '.', '')) as  
Dec(31, 2))
```

```
                Costs,
```

```
                Case When Locate('-', Substr(Spooled_Data, 66, 29)) > 0 Then -1
```

```
else +1 End *
```

```
                Try_Cast(RTrim(Replace(Trim(Substr(Spooled_Data, 66, 29)),
```

```
                '.', ''), '- ') as Dec(31, 2)) Diff,
```

```
                Case When Locate('-', Substr(Spooled_Data, 96, 9)) > 0 Then -1
```

```
Else +1 End *
```

```
                Try_Cast(RTrim(Replace(Trim(Substr(Spooled_Data, 96, 9)),
```

```
                '.', ''), '- ') as Dec(11, 2)) Percent
```

```
                --, Substr(x2.Spooled_Data, 105, 10) Col6
```

```
    from a
```

```
Where Left(Spooled_Data, 5) between ' 1990' and ' ' concat
Digits(Dec(Year(Current_Date), 4, 0))),
```

```
-- Address Information
```

```
x3 as ( Select Row_Number() Over(Order By Ordinal_Position) LfdX3 ,
        Try_Cast(Trim(Left(Spooled_Data, 3)) as Dec(5, 0)) Company,
        Try_Cast(Trim(Substr(Spooled_Data, 5, 7)) as Dec(9, 0)) CustNo,
        Trim(Substr(Spooled_data, 12, 34)) CustName,
        Trim(Substr(Spooled_Data, 46, 33)) Address,
        Trim(Substr(Spooled_Data, 79, 2)) Country,
        Trim(Substr(Spooled_Data, 81, 10)) ZipCode,
        Trim(Substr(Spooled_Data, 91)) City
```

```
from a
```

```
Where Left(Spooled_Data, 3) between ' 1' and ' 9'
and Substr(Spooled_Data, 5, 7) between '0000000' and '9999999'
and Locate('.', Left(Spooled_Data, 10)) = 0),
```

```
-- Heading and Sales Information
```

```
x4 as ( Select Distinct
        Case When x2.Ordinal_Position between x1.Ordinal_Position and
x1.MaxOrd Then lfdx1 End lfdX4,
        x2.*
        from x1 cross join x2
        Where Case When x2.Ordinal_Position between x1.Ordinal_Position and
x1.MaxOrd Then lfdx1 End is not NULL)
```

```
-- Select * from x4;
```

```
Select x4.Ordinal_Position,
        Company, CustNo, CustName, Address, Country, ZipCode, City,
        SalesYear, Sales, Costs, Diff, Percent
        from x4 join x3 on lfdx4 = lfdx3
        Where CustNo is not NULL ;
-- Order By x4.Ordinal_Position
;
```

```
Commit;
```

```
Set GblJobQual = -- '814156/HAUSER/QPRTJOB',
                 '151639/HAUSER/QPRTJOB',
GblSplf       = 'PRTRPGFR02',
GblSplfNo    = 3;
```

```
-----
-- Read View to get the spool filed data
-----
```

```
-----
Set GblJobQual = -- '814156/HAUSER/QPRTJOB',
                 '151639/HAUSER/QPRTJOB',
GblSplf       = 'PRTRPGFR02',
```

```

GblSplfNo = 4;

Select * from VWSplfDta;

-- Save the spooled file Data into a (temporary) Table
-----
Drop Table if exists qtemp.PRTRPGFR02;
Drop Table if exists comsqlqry.PRTRPGFR02;
Drop Table if exists EPS006BHA.PRTRPGFR02;
commit;
Stop;

-- Declare Global Temporary Table PRTRPGFR02
Create Table comsqlqry.PRTRPGFR02
as (Select GblJobQual Job, GblSplf Splf, GblSplfNo SplfNo,
          a.*, Current_Timestamp as WrtDate, Session_User as WrtUser, Job_Name as
WrtJob
      from VWSplfDta a
      Order By Ordinal_Position)
With data
--With replace
--ON COMMIT PRESERVE ROWS
;
Commit;

-- Insert Into PRTRPGFR02
--      Select GblJobQual, GblSplf, GblSplfNo,
--      a.*, Current_Timestamp as WrtDate, Session_User as WrtUser,
Job_Name as WrtJob
--      from VWSplfDta a
--      Order By Ordinal_Position;

-- Delete from Prtrpgfr02;
-- Commit;

CL:CHGLIBL LIBL(QTEMP COMSQLQRY EPS006BHA SYSTOOLS QGPL);

Select *
  from PRTRPGFR02;

Select * from Library_List_Info;

Select * from PRTRPGFR02
Order By WrtDate, Ordinal_Position;

-- Generate XML from the spooled file Data (out of the View)
-----
Select XMLGroup(Company, CustNo, CustName, Address, Country, ZipCode, City,
SalesYear, Costs, Diff, Percent
              Order By Ordinal_Position

```

```

                Option Row "Sales"
                    Root "SalesReport")
from VWSp1fDta;

Select * from COMSQLQRY.VWSp1fDta Order By Company, CustNo, SalesYear;

-- Generate JSON from the spooled file Data (out of the View)
-----
Select JSON_Object('SalesReport':
    JSON_ArrayAgg(
        JSON_OBJECT('Sales':
            JSON_OBJECT('COMPANY':    Company,
                        'CUSTNO':     CustNo,
                        'CUSTNAME':   CustName,
                        'ADDRESS':    Address,
                        'COUNTRY':    Country,
                        'ZIPCODE':    ZipCode,
                        'SALESYEAR':  Trim(SalesYear),
                        'COSTS':      Costs,
                        'DIFF':       Diff,
                        'PERCENT':    Percent)))
            Order By Company, CustNo, SalesYear))
from VWSPLFDTA;

-- Values(Systools.Generate_Spreadsheet(Path_Name      =>
'/Home/Hauser/Examples_SysTools/VWSp1fDta_01',
--                               Spreadsheet_Query => 'Select * from
COMSQLQRY.VWSp1fDta',
--                               Spreadsheet_Type   => 'xlsx',
--                               Column_Headings   => 'COLUMN'));

Values(Systools.Generate_Spreadsheet(Path_Name      =>
'/Home/Hauser/Examples_SysTools/VWSp1fDta_01',
--                               Spreadsheet_Query => 'Select * from
COMSQLQRY.PRTRPGFR02',
--                               Spreadsheet_Type   => 'xlsx',
--                               Column_Headings   => 'COLUMN'));

Select * from COMSQLQRY.PRTRPGFR02;
Commit;

Select *
  from TAb1e(Joblog_Info('*'))
  Where Message_Timestamp >= Current_Timestamp - 10 Minutes;

```

```

-----
=====
-- GENERATE_SPREADSHEET
-----

```



```

Sum(Total) Total
from COMSQLQRY.MONSUMX
Group By

RollUp(SalesYear, CustNo)
CustNo',
                                Spreadsheet_Type => 'xlsx',
                                Column_Headings => 'COLUMN'));

-- With Totals and Summary Text
Values(Systools.Generate_Spreadsheet(Path_Name =>
'/Home/Hauser/Examples_SysTools/MONSUMX03',
                                Spreadsheet_Query => 'Select Case When
Grouping(SalesYear) = 1
                                Then ''Grand
Total''
                                When
Grouping(CustNo) = 1
                                Then ''Total
Year '' concat SalesYear
                                Else ''''
                                End Summary_Text,
                                SalesYear, CustNo,
                                Sum(Jan) Jan,
Sum(Feb) Feb, Sum(Mar) Mar,
                                Sum(Apr) Apr,
Sum(May) May, Sum(Jun) Jun,
                                Sum(Jul) Jul,
Sum(Aug) Aug, Sum(Sep) Sep,
                                Sum(Oct) Oct,
Sum(Nov) Nov, Sum(Dec) Dec,
                                Sum(Total) Total
                                from COMSQLQRY.MONSUMX
                                Group By
                                Order By SalesYear,
CustNo',
                                Spreadsheet_Type => 'xlsx',
                                Column_Headings => 'COLUMN')));

```

```

-----
=====
-- SEND_EMAIL
-----
=====
Values(Systools.Send_Email(To_Email => 'Hauser@ModEdCon.Com',
Subject => 'Sales Report',
Body => 'Attached you will find the monthly sales

```

```

report',
                                Attachment =>
'/Home/Hauser/Examples_SysTools/MONSUMX03.xlsx'));

-----
-----
-- Combine Generate_Spreadsheet and Send_eMail
-----
-----

Begin
  Declare LocCRLF      Char(2)          Default '';
  Declare LocBody     VarChar(1024)    Default '';
  Declare LocCustNo   VarChar(15)      Default '10003';
  Declare LocType     Varchar(4)       Default 'xlsx';

  Declare LocPath     Varchar(256)     Default '/Home/Hauser/Examples_SysTools/Monsum_';
  Declare LocQuery    VarChar(4000)    Default '';
  Declare LocResult   Integer          Default 0;

  Set LocPath        = Trim(LocPath) concat Trim(LocCustNo);
  Call Lprintf(LocPath);
  Set LocQuery       = 'Select Case When Grouping(SalesYear) = 1
                        Then ''Grand Total''
                        When Grouping(CustNo) = 1
                        Then ''Total Year '' concat SalesYear
                        Else ''''
                        End Summary_Text,
                        SalesYear, CustNo,
                        Sum(Jan) Jan, Sum(Feb) Feb, Sum(Mar) Mar,
                        Sum(Apr) Apr, Sum(May) May, Sum(Jun) Jun,
                        Sum(Jul) Jul, Sum(Aug) Aug, Sum(Sep) Sep,
                        Sum(Oct) Oct, Sum(Nov) Nov, Sum(Dec) Dec,
                        Sum(Total) Total
                        from COMSQLQRY.MONSUMX
                        Where CustNo = '''' concat Trim(LocCustNo) concat ''''
                        Group By RollUp(SalesYear, CustNo)
                        Order By SalesYear, CustNo';

  Call LPrintF(LocQuery);

  Set LocCRLF        = Cast(x'0D25' as Char(2));
  Set LocBody        = 'Dear Customer ' concat Trim(LocCustNo) concat ', ' concat
Repeat(LocCRLF, 2) concat
                        'attached you will find the Sales Report per Month. ' concat
Repeat(LocCRLF, 2) concat
                        'Best Regards ' concat
LocCRLF concat
                        'Birgitta Hauser';
  Call LPrintF(LocBody);

```

```

If Systools.Generate_Spreadsheet(Path_Name      => LocPath,
                                Spreadsheet_Query => LocQuery,
                                Spreadsheet_Type  => LocType,
                                Column_Headings  => 'COLUMN') = 1
Then Call LPrintf('Spreadsheet successfully generated');

If SysTools.Send_eMail(To_eMail   => 'Hauser@ModEdCon.com',
                      Subject     => 'Sales Report',
                      Body        => LocBody,
                      Attachment  => Trim(LocPath) concat '.' concat
Trim(LocType)) = 1
Then Call LPrintf('eMail successfully sent');
Else CALL LPrintf('Error Sending eMail');
End If;
Else Call LPrintf('Error Generating Spreadsheet');
End If;
Return;
End;

```

```

-----
-- Create Multiple Generate_Spreadsheet for different users and send them directly
as eMail Send_eMail
-- in a Single SELECT-Statement
-----

```

```

-----
-- Create or Replace View COMSQLQRY.VW_Send_SalesReport
-- For System Name VWSndSalR
-- as

```

```

With x as (Select Distinct s.CustNo, CustName1, 'Hauser@ModEdCon.com' eMail_Addr,
                        '/Home/Hauser/Examples_SysTools/Monsum_' concat
Trim(s.CustNo) Path,

```

```

                        Cast(x'0d25' as Char(2)) CRLF,
                        'xlsx' Extention
From Sales s Join AddressX a on a.CustNo = s.CustNo),
y as (Select x.*,
        'Select Case When Grouping(SalesYear) = 1
            Then ''Grand Total''
            When Grouping(CustNo) = 1
            Then ''Total Year '' concat SalesYear
            Else ''''
        End Summary_Text,
        SalesYear, CustNo,
        Sum(Jan) Jan, Sum(Feb) Feb, Sum(Mar) Mar,
        Sum(Apr) Apr, Sum(May) May, Sum(Jun) Jun,
        Sum(Jul) Jul, Sum(Aug) Aug, Sum(Sep) Sep,
        Sum(Oct) Oct, Sum(Nov) Nov, Sum(Dec) Dec,
        Sum(Total) Total
from COMSQLQRY.MONSUMX

```

```

        Where CustNo = '' concat Trim(CustNo) concat ''
        Group By RollUp(SalesYear, CustNo)
        Order By SalesYear, CustNo' QueryText,
        Trim(Path) concat '.' concat Trim(Extension) IfsFile,
        'Sales Report for CustNo ' concat Trim(CustNo) concat '
' concat Trim(CustName1) eMail_Subject,
        'Dear Customer ' concat Trim(CustName1) concat
', ' concat Repeat(CRLF, 2) concat
        'attached you will find the Sales Report '
concat Repeat(CRLF, 2) concat
        'Best Regards '
concat CRLF concat
        'Birgitta Hauser' eMail_Body

        From x),
        z as (Select SysTools.Generate_SpreadSheet(Path_Name => Path,
                                                    Spreadsheet_Query => QueryText,
                                                    Spreadsheet_Type => Extension,
                                                    Column_Headings => 'COLUMN')
Spreadsheet,
            y.*
        from y)

Select CustNo, Spreadsheet,
        Case When Spreadsheet = 1
            Then SysTools.Send_eMail(To_eMail => eMail_Addr,
                                     Subject => eMail_Subject,
                                     Body => eMail_Body,
                                     Attachment => IFSFile)
        Else '0'
        End eMail
        From z;

Commit;

Select Trim(CustNo),
        'CustNo: ' concat Trim(CustNo) concat
        Case When Spreadsheet = 1
            Then ' Spreadsheet generated'
            Else ' Spreadsheet NOT generated'
        End,
        'CustNo: ' concat Trim(CustNo) concat
        Case When eMail = 1
            Then ' eMail sent'
            Else ' eMail NOT Sent'
        End
        from ComSQLQRy.VW_Send_SalesReport;

```