

## **IES PRINT WRITER: CASE STUDY – A/R CUSTOMER STATEMENTS**

### **Purpose**

The purpose of this Document is to discuss, as a case study, the implementation of A/R Statement generation in IES Business, and how it includes functionality to employ PrintWriter to perform a pdf mail run as part of the Statement generation process.

This document is targeted at IES Developers.

### **Introduction**

A/R Debtor (Customer) Statement generation in IES Business is an interesting case study in the context of Print Writer, because it offers the User multi-level output options for execution, and it is useful for the IES Developer to glean how this is done. For example, when this option is selected from the Menu, the User can choose each time between a combination of the following 3 ways of producing Statements: -

- ✓ Printing a Statement as a predefined and formatted (programmatically) Statement via the Print Spooler, either on plain paper or pre-printed stationery;
- ✓ Printing a Statement from a Document Writer object (which is easily customized);
- ✓ Formatting the Statement from Document Writer and converting it to PDF, then attaching it to an e-mail and dispatching it automatically.


The Statement generation is controlled from the program "do.debtprint" which is found in file "cbp" in an IES Business DataMart, and which is called from a Menu Process. The source code for this program is available in the indicated file, and can be copied to "client.bp" for further customization.

In this document, we will show sections of the program to illuminate certain points, but we will not show the entire program (you can view the program itself in it's source file).

### **The User Options at Runtime**

When selected from the Menu, the A/R Statement Generation process offers the User options as shown below.

In the 1<sup>st</sup> instance, the User may choose from a range of different Data Selectors, and may choose to select Customer Accounts managed in a specific Currency, or where the current Balance is not = 0, or where there is in fact an e-mail address recorded for Statement dispatch by e-mail, etc.

Be sure to use your  today!

IES PRINT WRITER: CASE STUDY – A/R CUSTOMER STATEMENTS

The User also selects the Financial Period to process the Statements for, and some other instructions, but the specific option that we wish to focus on, is the one called 'Statement Print Type'.



## Generate AR Statements

**please adjust the Data Selector if necessary**

Data Selector

**adjust Period and Statement Date if necessary**

Period   
Statement Date

**select the preferred execution settings**

- Pre-Printed Stationery
- Print If Bal = 0

Statement Print Type

The 'print type' as explained as follows: -

For this Statement run, please choose between the following options: -

- 1: Legacy Printing  
Printing all selected Statements via the print spooler on plain or pre-printed stationery.
- 2: Legacy + PDF e-mail Statements  
Printing all selected Statements via the print spooler on plain or pre-printed stationery, except when an e-mail Statement address is found on the AR Master Account, in which case the statement is formatted PDF style and automatically dispatched by e-mail.
- 3: Document Layout Printing  
Use the Document Layout for Windows printing.
- 4: Document Layout + PDF e-mail Statements  
Use the Document Layout for Windows printing, except when an e-mail Statement address is found on the AR Master Account, in which case the statement is formatted PDF style and automatically dispatched by e-mail.

Be sure to use your today!

**How the execution program acts on the User choice(s) –**

The logic shown below is from "do.debtprint", and reveals how the User choice is initially gleaned from the pre-run screen, and validated.

\* By default, 'state.choice' is selected by the User at runtime.

```
state.choice = i<37>[1,1]
```


```
chk = '1234' ; if index(chk,state.choice,1) else state.choice = '1'
```

```
if state.choice = '2' or state.choice = '4' then
  yn = " ; call pdfcreator.yn(yn)
  if yn = 'y' else
    begin case
      case state.choice = '2'
        state.choice = '1'
      case state.choice = '4'
        state.choice = '3'
    end case
  end
end
```

Above, the runtime record is "i", and we find the User's selected choice on field 37, but we are interested only in the 1<sup>st</sup> character, which must be 1, 2, 3 or 4. If it is 2 or 4, then PDFCreator is required, and therefore we use an internal IES call to check whether PDFCreator is present on the current workstation. If not, we change the setting either to 1 or 3, i.e. not to have to use PDF format. As such, if PDF cannot be used, no e-mailing will be performed, and Statements which would otherwise have been mailed will be printed like the others.

After selecting the Accounts to process, i.e. according to the selected Data Selector and User response to selection criteria, the program loops through the selected Accounts and performs a Statement in each valid case. Before executing the statement itself, the program now checks the operating choice, and whether an e-mail address for statement dispatch is present on the Account Master record. (Hint: 'dkey' is the Customer Account number in the logic shown here.)

```
* which method will be used?
echk = oconv(dkey,'tents;x1;;140')
begin case
  case state.choice = '2'
    * mail if mail address present, else legacy
    if echk = " then
      * stay right here for legacy printing
    end else
      gosub pdf.mail
      go slip.2
    end
  case state.choice = '3'
    * docwriter
    gosub do.docwriter
```

Be sure to use your  today!

## IES PRINT WRITER: CASE STUDY – A/R CUSTOMER STATEMENTS

```

    go slip.2
  case state.choice = '4'
    * docwriter or mail
    if echk = "" then
      gosub do.docwriter
      go slip.2
    end else
      gosub pdf.mail
      go slip.2
    end
  case 1
    * legacy, stay right here and fall through
  end case

```

Essentially, this logic determines whether –

- (a) to stay in the current subroutine, which will perform legacy Statement printing
- (b) or to use DocWriter to execute the Statement and then use PrintWriter to print the Statement directly to a Windows printer
- (c) or to use PrintWriter (which will use DocWriter to execute the Statement) to format the Statement into a PDF and dispatch it by e-mail.

The logic for legacy printing has no further relevance here. It is visible in “do.debtprint” where it can be changed to suit Customer requirements.

The logic for using PrintWriter to PRINT or MAIL the Statement is very relevant, and we will show the PRINT option 1<sup>st</sup>.

```


do.docwriter: *
  tpe = 1
  periodf = period
  periodt = period
  outx = ""
  outx<8> = 'Preparing Statement Print on Account: ':oconv(dkey,'mcu')
  docwritespre = outx
  write docwritespre on port

  call get.ar.state(dkey,periodf,periodt,tpe)
  ref = 'arstate-1-pp'
  lom = ""
  call s.logi(ref,lom)
return

```

Notice that we pass a record to “docwritespre” with a message to display on screen while this Statement is prepared. The display message is written to field 8 (please consult the PrintWriter Introduction and Shell Program manuals in conjunction with this Manual ...) as indicated for the ‘Instruction Key’ on the PrintWriter object definition. No further instructions are necessary for printing.

The call that we make to “get.ar.state” uses the parameters we pass, i.e. Account Number (dkey), Period From (periodf), Period To (periodt) and Type (tpe) to

Be sure to use your  today!

## IES PRINT WRITER: CASE STUDY – A/R CUSTOMER STATEMENTS

prepare data (statement contents) that it writes to "docwritespre", where DocWriter can access any of the details for inclusion in the Statement layout as specified for DocWriter. (Hint: "get.ar.state" is an internal IES program that is not shown in detail here.)

Once the data has been prepared for DocWriter, and the instruction passed for PrintWriter, we can call the PrintWriter Menu Process to perform the Statement. In the above case, the process we call is "arstate-1-pp", and if we should look at it's definition in the Print Writer register, it looks like this: -

PrintWriter Key	arstate-1-pp	use DocWriter Key + "-p" or any unique key ...
Name	AR Statement, Print HTML	
DocWriter Object	arstate-1 AR Debtor Statement - English	
Action	Print	
Instruction Key	*port	


Above, we can see that the DocWriter object that will actually prepare the Statement is called "arstate-1", that the action = "print" and that the Key we use to pass instructions for PrintWriter on "docwritespre" = '\*port' which means the network port number – all of this holds true to the logic shown above.

NEXT, we look at the logic for doing a Statement by e-mail with PDF format.

pdf.mail: \*

```
tpe = 1
periodf = period
periodt = period
outx = ""
outx<1> = echk
outx<2> = 'Statement for ':perdescr
outx<4> = 'S_':perdescr
outx<8> = 'Preparing Statement to Mail on Account: ':oconv(dkey,'mcu')
docwritespre = outx
write docwritespre on port
```

```
call get.ar.state(dkey,periodf,periodt,tpe)
ref = 'arstate-1-pm'
lom = ""
call s.logi(ref,lom)
read docwritespre from port else docwritespre = ""
dwp = docwritespre
if dwp<5> = "" then
  * failed
  dwp1 = date()
  dwp2 = time()
  dwp3 = perdescr
  dwp4 = 'Failed: '
```

Be sure to use your  today!

## IES PRINT WRITER: CASE STUDY – A/R CUSTOMER STATEMENTS

```

if dwp<7> = " then
  dwp4 = dwp4 : 'Unknown Error'
end else
  dwp4 = dwp4 : dwp<7>
end
* update pdffails
pdferr = dwp<7> ; if pdferr = " then pdferr = 'Unknow Error'
pdfkey = " ; call get.unq.11(pdfkey)
pdfk = date()
pdfk<2> = time()
pdfk<3> = pdferr
pdfk<4> = dkey
pdffails = pdfk
write pdffails on pdfkey
pdffrep = 'y'
end else
* succeeded
dwp1 = dwp<5> ; if dwp1 = " then dwp1 = date()
dwp2 = dwp<6> ; if dwp2 = " then dwp2 = time()
dwp3 = perdescr ; if dwp3 = " then dwp3 = 'Unknown'
dwp4 = 'OK'
end
* update lentsmail
readu lentsmail from dkey else lentsmail = "
lmx = lentsmail
lmx = insert(lmx,1,1;dwp1)
lmx = insert(lmx,2,1;dwp2)
lmx = insert(lmx,3,1;dwp3)
lmx = insert(lmx,4,1;dwp4)
lmxlim = dcount(lmx<1>,vm)
if lmxlim > 20 then
  lmx = delete(lmx,1,lim)
  lmx = delete(lmx,2,lim)
  lmx = delete(lmx,3,lim)
  lmx = delete(lmx,4,lim)
end
lentsmail = lmx
write lentsmail on dkey

return


```

In this case, we also pass instructions for PrintWriter, but this time we include the necessary e-mail address and some others ... (see the **green** lines)

Then we also call 'get.ar.state' to prepare data content for DocWriter to use (see the **red** line).


And then we call the PrintWriter process (see the **black** lines).

The rest of the logic (**purple** lines) is where we inspect the result, and in this example we actually record success or failure on the 'lentsmail' file, where from a Customer record the User can drill into PDF Statement history, and in the event of failure, we also record an error on a file called "pdffails". At the end of the Statement Generation sequence, if there were indeed any PDF mail errors, then the program links to an error Report that lists the errors.

Be sure to use your  today!

© Infolab, 2006

This Documentation is copyrighted by Infolab (Pty) Ltd. [ [www.infolab.cc](http://www.infolab.cc) ] All rights are reserved. Licensed INFOLAB Users are granted permission, for internal use ONLY, to reproduce the Documentation, and to include amendments dealing with specific instructions local to your installation.

Be sure to use your  today!