AddonSoftware Report Control

Introduction

Selected Addon documents include an interface to the Barista Document Processing system. The interface updates the Barista Document Archive with information about newly created pdfs, and optionally enters a request to email or fax those pdfs into the Barista Fax/Email Document Queue.

Documents that have been fitted with this interface, as well as output preferences and fax/email information for customers and/or vendors receiving these documents, is maintained via the AddonSoftware Administration menu => Report Control tasks:

- Report Print, Email, Fax Control: lists the Barista alias names of the interfaced documents. Enabling the email/fax interface involves code changes to the document creation program (see the Implementation section). Once a program has been altered, add it to this list. Addon currently ships with the following reports Fax/Email enabled:

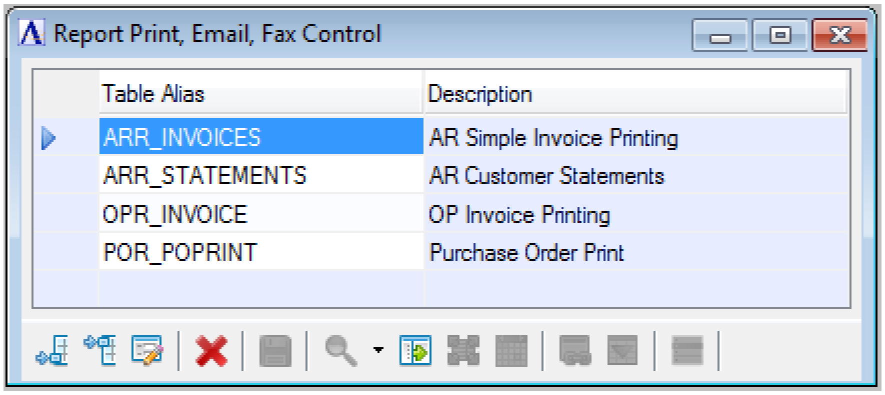
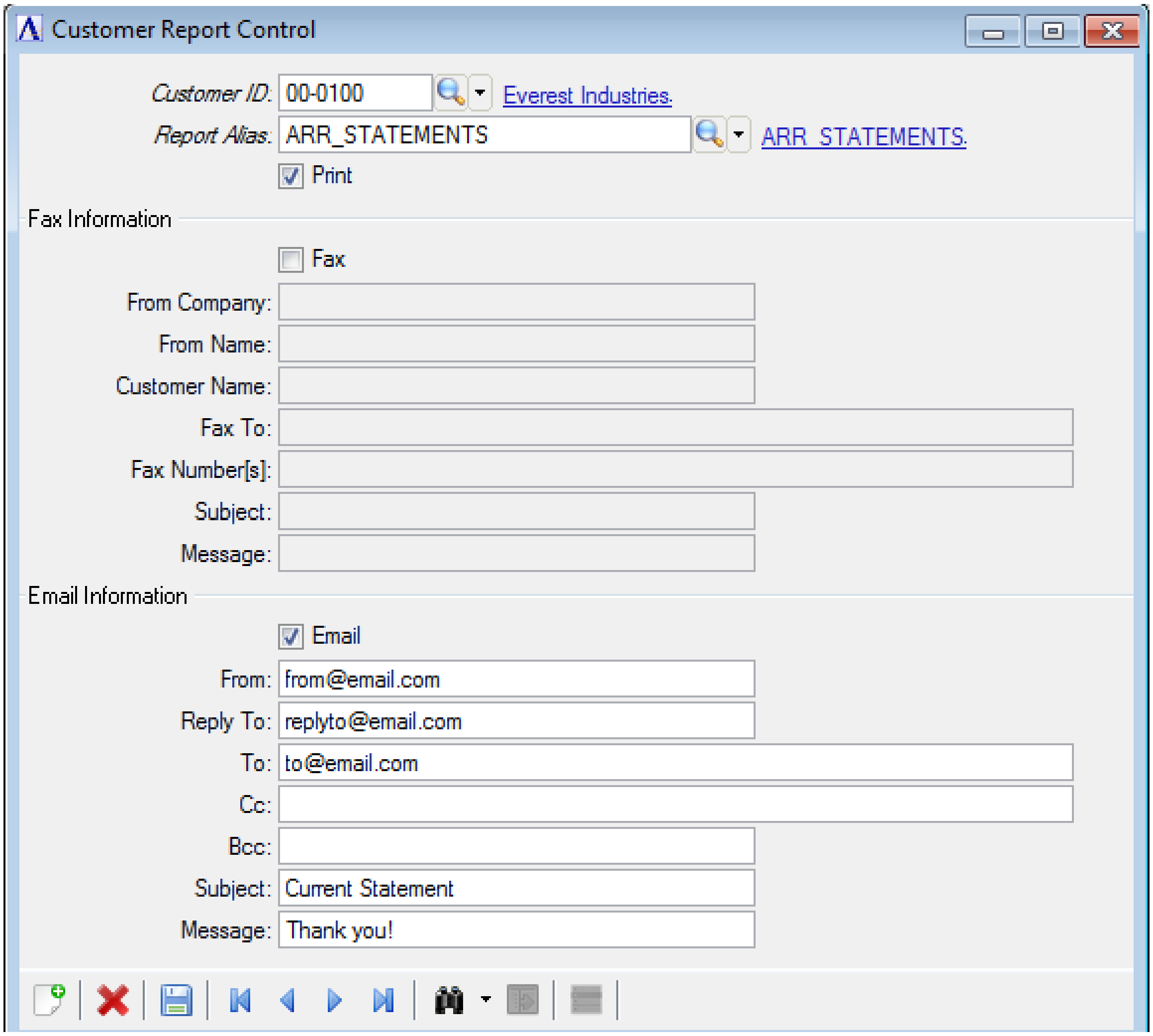


Table Alias	Description
ARR_INVOICES	AR Simple Invoice Printing
ARR_STATEMENTS	AR Customer Statements
OPR_INVOICE	OP Invoice Printing
POR_POPRINT	Purchase Order Print

(Figure 1: Fax/Email enabled reports)

- Customer Report Control: for a given customer, select a report from the Report Print, Email, Fax Control list, and then select one or more of the output checkboxes and corresponding recipient information (fax to name/number, email address, etc.).
- Vendor Report Control: for a given vendor, select a report from the Report Print, Email, Fax Control list, and then select one or more of the output checkboxes and corresponding recipient information (fax to name/number, email address, etc.).



Customer Report Control

Customer ID: 00-0100 Everest Industries.

Report Alias: ARR_STATEMENTS ARR_STATEMENTS.

☑ Print

Fax Information

□ Fax

From Company:

From Name:

Customer Name:

Fax To:

Fax Number[s]:

Subject:

Message:

Email Information

☑ Email

From: from@email.com

Reply To: replyto@email.com

To: to@email.com

Cc:

Bcc:

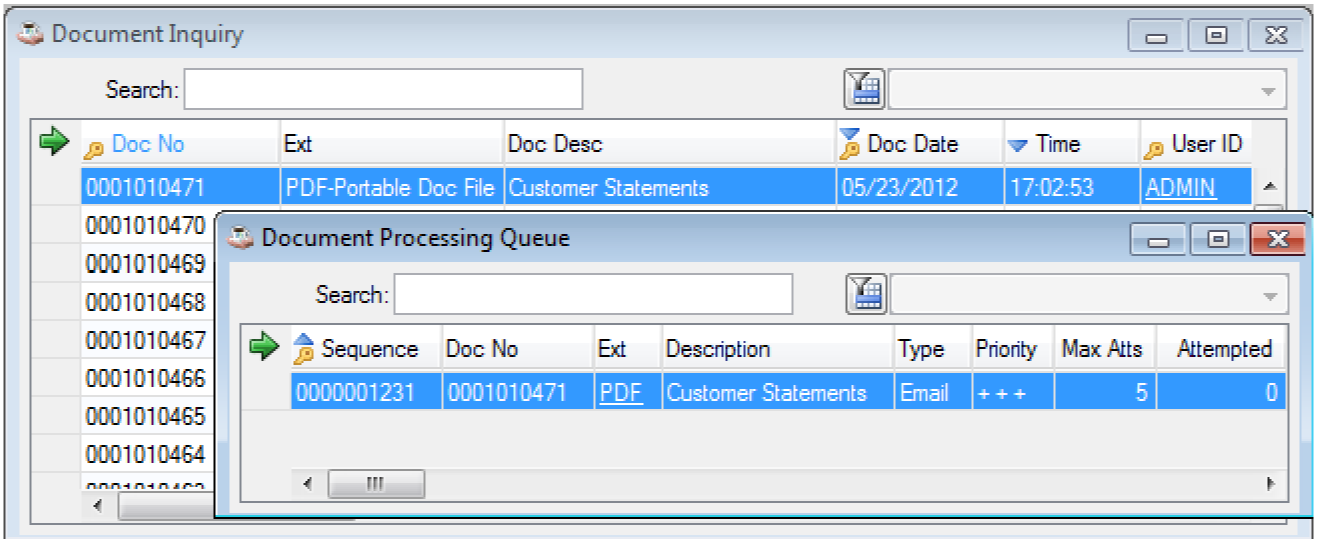
Subject: Current Statement

Message: Thank you!

(Figure 2: Customer Report Control showing email information for customer 000100 statements)

Operation

The Addon documents listed in figure 1 all follow a similar flow for creating the documents and handling the email/faxing. Each process creates two BBj Form objects. The first form object contains pages for all customers/vendors in the document run. This object is presented on screen for review, saving, or printing. The second form object creates a separate .pdf file for each customer/vendor document and saves the .pdf in the Barista Document warehouse. As the individual .pdf's are created, the document program also reads the Customer or Vendor Report Control file, and adds the document path/name and recipient information to the Fax/ Email queue as specified. The .pdf file remains in the document archive, so it can be retrieved again later if desired.



Document Inquiry

Search:

Doc No	Ext	Doc Desc	Doc Date	Time	User ID
0001010471	PDF-Portable Doc File	Customer Statements	05/23/2012	17:02:53	ADMIN
0001010470					
0001010469					
0001010468					
0001010467					
0001010466					
0001010465					
0001010464					

Document Processing Queue

Search:

Sequence	Doc No	Ext	Description	Type	Priority	Max Atts	Attempted
0000001231	0001010471	PDF	Customer Statements	Email	+ + +	5	0

(Figure 3: Barista Document Inquiry and Document Processing Queue)

Implementation

Altering a program for automated fax/email is basically a three-step process. Before creating a new .pdf file, call Barista's bac_documents public to retrieve a document ID for the file, and make an entry in the Document Inquiry table. The complete enter list is shown in Appendix A.

```
rem --- new document; call bac_documents.bbj to get back a barista document
ID

    new_doc_id$=""
    pdf_doc$=""
    call stbl("+DIR_SYP")+"bac_documents.bbj",new_doc_id$,"","","I","PDF",
:       docdir$,"pdf",option!.getAlias(),"C",arm01a.customer_id$,
:       rd_table_chans$[all],"DOC_ID"
```

Next, create the new .pdf file. This sample simply uses the document ID (new_doc_id$) returned from Barista as the document name. If you like, you could create a more intuitive (but unique) document name and supply that name when calling bac_documents, and then use that name for the pdf_doc$.

```
header: rem --- create the pdf and print the statement header

rem --- create pdf file

    while pdf_doc$ = ""
        prtr_dev = unt
        pdf_doc$ = docdir$ + new_doc_id$ + ".pdf"
        start_block = 1
        if start_block
            open (prtr_dev,err=*endif)pdf_doc$
            close (prtr_dev)
            erase pdf_doc$
        endif
        docPDF! = myBBjPDFPrinter!.getBBjPDFForm(pdf_doc$, requestAttributes!
)
    wend
```

Finally, when the .pdf is complete, use the docPDF!.print() method to save it to disk, and then call Addon's adc_rptcontrol public. This program will retrieve fax and/or email settings from the Vendor and/or Customer Report Control files described above, and create an entry in Barista's Document Processing Queue.

```
rem --- pdf completed

    docPDF!.print()
    call stbl("+DIR_PGM")+"adc_rptcontrol.aon",option!.getAlias(),
:       arm01a.customer_id$,"C",new_doc_id$,"pdf"
```

Appendix A

Call/enter arguments for Barista bac_documents.bbj:

```
rem  rd_doc_no$            - Sequential document number.  Assigned if
rd_audit_action$ contains "DOC_ID"
rem  rd_doc_date$          - Document creation date.  Current date if null.
rem  rd_doc_time$,         - Document creation time.  Current time if null.
rem  rd_doc_source$        - Document origin: (I)nternal Barista application,
(E)xternal reporting (Jasper, etc)
rem  rd_doc_ext$           - Document extension/type
rem                            CSV - Comma separated values
rem                            DSP - Display (DocOut Viewer)
rem                            HTM - HTML page (Jasper)
rem                            ODT - OpenOffice Text Document (Jasper)
rem                            PDF - Portable Document File
rem                            PRT - Printer output
rem                            RAW - Raw format printer output
rem                            RTF - Rich text format (Jasper)
rem                            TXT - Tab delimited text
rem                            XML - XML formatted
rem                            XLS - Spreadsheet (Jasper)
rem                            GCS - Google Docs comma separated
rem                            GHT - Google Docs HTML Document
rem                            GPD - Google Docs PDF document
rem                            GTX - Google Docs TXT document
rem                            GXL - Google Docs XLS document
rem                            GXM - Google Docs XML document
rem  rd_doc_path$          - Document path/URI
rem  rd_source_type$,      - Document system source. (B)arista for table
listings, or any other user defined 1 byte code.
rem  rd_source_alias$      - Barista table alias
rem  rd_source_id$         - Document application source reference (O)rder
Proc, (G)eneral Ledger, etc.
rem  rd_source_ref$        - Related to source id (above).  May include order
number, journal number, etc.
rem  rd_table_chans$[all] - Opened table/channel/template array
rem  rd_audit_action$      - Additional actions:
rem                            DOC_PATH   - Retrieve base document group
output path
rem                            DOC_ID     - Retrieve next sequential document
number.
rem                            NOREPRINT  - Do not allow reprint of document
rem                            INCOMPLETE - Report did not finish
rem                            ARCHIVE    - Archive document record
rem  rd_doc_name$          - Document file name on disk. Defaults to document
number + extension
rem  rd_doc_title$         - Document title.  Defaults to task_desc retrieved
from Barista SysInfo record.
```

```
rem  rd_doc_keywords$     - Document search keywords.  Additional keywords
for searching.
```