

Distributing Reports from Batch Jobs via Email — Enhanced

This update to the author's June article examines enhancements made to the REXX program, XMITIP, a freeware utility created by the author and used to send messages and files via email, which now provides support for sending file attachments as rich text format (RTF) documents.

In the June 1999 issue of *Technical Support* I wrote about a simple REXX application that could be used in batch jobs to transmit via email both messages and files to Internet mail recipients. This follow-up article describes an enhancement to that application that increases its applicability and makes it easier to use. See the June 1999 article for more information on the basics of the application.

no parameters, the ISPF dialog interface is invoked. One benefit of this dialog interface is that it generates a report with the %XMITIP command syntax so that you can model batch usage interactively and then just copy the generated command syntax for use in your batch job. This simple interface is shown in Figure 1. Figure 2 shows the fields on the panel. Each entry field is explained as follows:

WHAT'S NEW IN XMITIP?

Several exciting enhancements have been incorporated into the application. These will be presented along with an interesting REXX coding example to provide an application command table "on-the-fly."

Rich Text Format (RTF) Attachments

The REXX application XMITIP now supports new options that allow file attachment(s) to be converted to rich text format (RTF) documents. This simply means that you can define a file that formerly was a printed report so that it arrives in the recipient's email in-basket as a document that will open and appear very close in format to the printed report. Options you can define are:

1. Paper size: letter, legal, or A4
2. Paper orientation: Portrait or Landscape
3. Font size: the pitch of the font

ISPF DIALOG

When the XMITIP command is executed under ISPF with

The REXX application XMITIP now supports new options that allow file attachment(s) to be converted to rich text format (RTF) documents.

◆ **Recipient's Address:** This is the email address where you want to send the message and/or files. Use of a "?" will display an address selection table dialog. See Figure 2. This table will only be displayed once regardless of the use of the "?" in the CC and BCC fields.

◆ **Message DSN:** This field may be an "*", which means that the ISPF Editor will be opened on an empty temporary file to let you enter the message text to be sent. This field may also be a dataset name that contains the text for the message or the field may be blank if you are sending only a file attachment.

◆ **CC Address:** CC, or Carbon Copy, allows you to indicate to what email addresses you want to send copies of this email. If you need to enter multiple addresses, separate each address with a space, and if the field is not large enough enter a "+" to bring up an additional prompting panel. See Figure 9. Use of a "?" will display an address selection table dialog.

◆ **BCC Address:** This is the same as CC except it allows you to send Blind Carbon Copies. These addresses will not appear in the envelope of the email and the primary

recipient and CC recipients will not know the BCC recipients received a copy. The syntax is the same as for CC, including the use of the “+” and “?” special characters.

- ◆ **FROM Address:** This field is used to indicate from whom the email was sent. It will also be used by your recipients if they reply to this email.
- ◆ **SUBJECT:** This field is where you type in the subject of the email. Enter this text without quotes.
- ◆ **Attachment DSN:** This field indicates the dataset name(s) to be sent as file attachments. Code a “+” if you wish to send multiple datasets. Only sequential datasets and members of partitioned datasets may be sent. Be aware that character translation issues may arise if there are special characters in the data.
- ◆ **Carriage Control:** If the dataset has a Record Format (RECFM), which indicates it has ASA carriage control, then this field is ignored. If no carriage control is in the RECFM then you can indicate here with a “YES” that you want to have the carriage control converted.
- ◆ **Report Style:** This field allows you to indicate if you want the report to be: Landscape orientation, Portrait orientation, have a unique file type (e.g., XLS) or null for plain text. Landscape and Portrait will result in RTF encoding of the file. A “+” may be used if more than one file is being transmitted to allow you to specify different options for each file.
- ◆ **Font Size:** The default font defined in the application is Courier. Here you can specify the font size to use to make your report look the best.
- ◆ **Paper Size:** This field indicates your paper size.
- ◆ **Margins:** Specify the margins for each report. Use a “+” for more than one report.

EXAMPLES

To demonstrate how these new features work, I’ve included the following examples.

FIGURE 1: THE BASIC ISPF DIALOG PANEL

```

----- Email Dialog ----- 2.0
Command ==>

Recipient Address ==> ?
Message DSN      ==> *
                  dsn or * to edit a message or null for no message

Optional information
Subject          ==> Test subject
CC Address       ==>
BCC Address      ==>
From Address     ==> lionel.b.dyck@kp.org
Attachment DSN  ==>
Carriage Control ==> Yes or No
Report Style     ==> Landscape, Portrait, File-type or null
Font Size        ==> (default is 9)
Paper Size       ==> Letter, Legal, A4 or null
Margins - in inches (e.g. 8.5 for 8 1/2 inches)
Left ==> Right ==> Top ==> Bottom ==>

Press PF1 for Help
  
```

FIGURE 2: ADDRESS ENTRY/SELECTION TABLE DISPLAY

```

----- XMITIP Email Address List ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Comands: Find and Insert
or S Select (To) B Blind Copy C Carbon Copy D Delete I Insert U Update

Sel Action Recipients Name
          Email Address
-----
          Lionel B. Dyck (Lotus Notes)
          lionel.b.dyck@kp.org
***** Bottom of Data *****
  
```

Figure 3 demonstrates how to send a message to a single recipient and enter the text at the time you create the note. While this example shows you the message text entry, Figure 4 displays the transmission report. The example shown in Figure 5 demonstrates how to send a file with no message, while Figure 6 shows the accompanying delivery report. Figure 7 (see page 14) demonstrates how to send a single file with no message. It also shows how to convert that file to RTF with Portrait orientation, a font size of 10 and on letter size paper. Figure 8 is the accompanying delivery report. An example of using a “+” to extend the CC is displayed in Figure 9.

CODING EXAMPLE FOR APPLICATION ISPF COMMAND TABLE

The sample code in Figure 10 can be used to provide an application ISPF command table for just this application. This is helpful when you want to write an ISPF dialog and don’t have access or authority to update the ISPF command table (ISPCMD5) or either the site or user command tables. This also works so you don’t have to create a command table and add it to the system ISPTLIB,

which you may also be prevented from doing. This sample code is an extract from the XMITIPI REXX code that is the driver for the ISPF Dialog. The basic logic of the code in Figure 10 corresponds as follows:

1. Set the REXX identifier.
2. Obtain any parameters from the user.
3. Obtain the current ISPF APPLID.
4. Test to see if this is the APPLID that we want — XMITIP. If it isn’t, then process the do loop.
5. Set the REXX environment addressing to ISPEXEC.
6. Create the application command table.
7. Set the command (zctverb) name. In this case it is RFIND (repeat find).
8. Set the command length (zcttrunc) minimum. A value of 0 indicates that the entire command name must be entered. A shorter value, for example 2, would accept just the first two

FIGURE 3: TEXT ENTRY

```
EDIT      SYS99184.T100901.RA000.SYSLBD.R0511035          Columns 00001 00080
Command ==>                                           Scroll ==> CSR
***** ***** Top of Data *****
000001 Message text entered via the ISPF Editor.
***** ***** Bottom of Data *****
```

characters of the command (RF if the zctrunc is 2).

FIGURE 4: DELIVERY REPORT

```
Generated Command:
%xmitip lbdyck@pop3.kaiperm.org +
Msgds * +
Subject 'test message'

XMITIP: XMITIP Application level: 2.11
XMITIP:
XMITIP: From: SYSLBDA@NKAISERA.NCAL.KAIPERM.ORG
XMITIP: To: lbdyck@pop3.kaiperm.org
XMITIP: Date: 19 Jul 1999 15:44:35 PST
XMITIP: Subject: test message
XMITIP:
XMITIP: Message maximum lrecl is: 88
XMITIP:
XMITIP: Message successfully transmitted to: lbdyck@pop3.kaiperm.org
XMITIP: Sent 11 records as text, attachments, and control information.
XMITIP: Approximate byte count is: 5
```

9. Set the actual command or action. In this case the action is a variable, &XMTRFIND, which will be set later in the dialog using a VPUT.
10. Set the description of the command. In this case it is optional but helpful.
11. Add the command to the table.
12. Invoke the current command with original options along with the APPLID of XMIT to prevent a loop from occurring now that we have the command table built.
13. Upon return from the recursive execution of this command close and free the command table using TBEND.

FIGURE 5: SENDING A FILE WITH NO MESSAGE

```
----- E-Mail Dialog ----- 2.0
Command ==>
Recipient Address ==> ?
Message DSN ==> *
          dsn or * to edit a message or null for no message
Optional information
Subject ==> test message
CC Address ==>
BCC Address ==>
From Address ==>
Attachment DSN ==> jcl.cntl(tsob)
Carriage Control ==> Yes or No
Report Style ==> Landscape, Portrait, File-type or null
Font Size ==> (default is 9)
Paper Size ==> Letter, Legal, A4 or null
Margins - in inches (e.g. 8.5 for 8 1/2 inches)
Left ==> Right ==> Top ==> Bottom ==>
Press PF1 for Help
```

14. Next, exit the EXEC with a return code of 0.
15. End the loop.

Then within the code the &XMTRFIND is set to PASSTHRU using a VPUT command so that the RFIND command will be seen by this application.


This version of XMITIP further enhances the usability of the application, adding features and functions that the various users have requested. To download the latest version of XMITIP, visit www.share.org/mvs.html. Planned enhancements in the near future include support for binary attachments and the ability to define a signature file. 

FIGURE 6: DELIVERY REPORT

```
Generated Command:
%xmitip lionel.b.dyck@kp.org +
Msgds * +
Subject 'test message' +
File 'syslbd.jcl.cntl(tsob)'

XMITIP: XMITIP Application level: 2.11
XMITIP:
XMITIP: From: SYSLBDA@NKAISERA.NCAL.KAIPERM.ORG
XMITIP: To: lionel.b.dyck@kp.org
XMITIP: Date: 19 Jul 1999 15:47:27 PST
XMITIP: Subject: test message
XMITIP: File: 'syslbd.jcl.cntl(tsob)'
XMITIP: -records: 11
XMITIP:
XMITIP: Message maximum lrecl is: 88
XMITIP:
XMITIP: Message successfully transmitted to: lionel.b.dyck@kp.org
XMITIP: Sent 33 records as text, attachments, and control information.
XMITIP: Approximate byte count is: 1,238
```



NaSPA member Lionel B. Dyck is a lead MVS systems programmer at a large HMO in California. He has been in systems program-

ming since 1972 and has written numerous ISPF dialogs over the years. He is an active member of SHARE and is also a technical reviewer for Technical Support. He can be contacted at Lionel.B.Dyck@kp.org.

© 1999 Technical Enterprises, Inc.
For reprints of this document contact editor@naspa.net.

FIGURE 7: SENDING A FILE AS AN RTF ATTACHMENT

```

----- Email Dialog ----- 2.0
Command ==>

Recipient Address ==> ?
Message DSN ==> *
                dsn or * to edit a message or null for no message

Optional information
Subject ==> test message
CC Address ==>
BCC Address ==>
From Address ==>
Attachment DSN ==> 'syslbd.jcl.cntl(tsob)'
Carriage Control ==> Yes or No
Report Style ==> p Landscape, Portrait, File-type or null
Font Size ==> 10 (default is 9)
Paper Size ==> let Letter, Legal, A4 or null
Margins - in inches (e.g. 8.5 for 8 1/2 inches)
    Left ==> .5 Right ==> .5 Top ==> .5 Bottom ==> .5

Press PF1 for Help
    
```

FIGURE 10: LOCAL TABLE CODING EXAMPLE

1. /* REXX */
2. arg options
3. Address ISPEXEC "VGET ZAPPLID"
4. if zapplid <> "XMIT" then do
5. Address ISPEXEC
6. "TBCreate xmitcmds names(zctverb zcttrunc zctact zctdesc)", "replace share nowrite"
7. zctverb = "RFIND"
8. zcttrunc = 0
9. zctact = "&XMITRFIND"
10. zctdesc = "RFIND for XMITIP"
11. "TBAdd xmitcmds"
12. "Select CMD(%'sysvar('sysicmd') options *) Newappl(XMIT)"
13. "TBEnd xmitcmds"
14. Exit 0
15. end

FIGURE 8: RTF DELIVERY REPORT

```

Generated Command:
%xmitip lionel.b.dyck@kp.org +
Msgds * +
Subject 'test message' +
File 'syslbd.jcl.cntl(tsob)' +
Report p/10/let

XMITIP: XMITIP Application level: 2.11
XMITIP:
XMITIP: From: SYSLBDA@NKAISERA.NCAL.KAIPERM.ORG
XMITIP: To: lionel.b.dyck@kp.org
XMITIP: Date: 19 Jul 1999 15:50:21 PST
XMITIP: Subject: test message
XMITIP: File: 'syslbd.jcl.cntl(tsob)'
XMITIP: -records: 11
XMITIP:
XMITIP: Message maximum lrecl is: 89
XMITIP:
XMITIP: Message successfully transmitted to: lionel.b.dyck@kp.org
XMITIP: Sent 40 records as text, attachments, and control information.
XMITIP: Approximate byte count is: 1,478
    
```

FIGURE 9: CC EXTENDED ENTRY PANEL

```

----- * Email Dialog -----
Command ==>

Enter Carbon Copy addresses:

Address:
Address:
Address:
Address:
Address:
Address:
Address:
Address:
Address:
Address:
Address:
Address:

Enter CLEAR to clear, PF3 when complete to continue or CANCEL to Cancel
    
```