



Print Solutions from Atac

Newsletter for Asia Pacific Region

First Quarter, 2019

In This Issue

- Output Enrichment VDE Script Editor
- Output Enrichment VDE Script Debugger
- Determining File Formats
- Q & A
- Feedback
- Who else should get this newsletter?
- Unsubscribe

More information at...

<http://www.atac.com.au>

Email David Kirk

or call +61 (419) 962 386

Output Enrichment VDE Script Editor

The Output Manager Viewer is also used to create and update VDE scripts, providing you have a license to do so. Start the Viewer and click “Enrichment Editor”, the right-most button on the button bar. Choose New or Existing to create or update VDE.

The editor environment comprises line numbers on the left and editor-specific buttons on the button bar for Syntax Check, Save, Save All, Save As, Update Profile, UnDo, ReDo, Cut, Copy, Paste, Delete, Print, Sort, Insert (Command or Action), Edit and Tree View.

Edit View (the default) allows for multiple VDE files to be open at once, as a series of tabs across the screen, just under the button bar. Opening any VDE file with one or more INCLUDE statements automatically opens the Included file(s). Or you can just click “Enrichment Editor” to open other VDE files, then switch between them via the tabs.

Large projects often comprise one main VDE script and one or more Included VDE scripts. This allows you to break processing into manageable chunks and re-use generic modules which have already been debugged. Double-clicking any Object or Action highlights that item; pressing F12 then “yellows” all occurrences throughout all open scripts and positions you at the called Action or Object (very useful).

“Syntax Check” or F5 checks your script and takes you to the first offending code line for correction, following a clear explanation of the issue. You should Syntax Check prior to saving, although Save alerts you if there is a syntax error present and asks if you want to proceed Y/N.

You can view a job’s output by dragging and dropping a profile onto the Viewer client, which should be in Output mode. If the Viewer already has a VDE script open, the script edit window shrinks to the left-side of the Viewer and the job’s output appears in the right-hand panel. If the Viewer doesn’t have a script already open, the job output fills the Viewer screen and the button bar fills with navigation buttons. However, you can then open a VDE script and see script and output side by side.

With VDE script and output side by side in the Viewer, there is interactivity between the two panels. If you change an Object’s BEGINWINDOW and ENDWINDOW parameters in the code panel then press F5, the object changes in the output panel accordingly. Double-clicking an Object label in the code panel, then pressing F12, shows the item in the output panel with adjustable handles. Changing the item in the output panel (re-sizing it for example) also adjusts the code panel.

If you click and drag to create a new object in the output panel, a new code snippet is created at the cursor location in the code panel. You can then modify the code snippet and change its name as well as adding code, then pressing F5.

Once you’ve written some code and run it, you can highlight the job in the Job Queue Manager (JQM) and choose right-click | Reprint | select Viewer as Output | Submit. This allows you to move to any page in the job, creating and/or adjusting objects in either the code or output panels.

Finally, remember to always load and save VDE scripts in the Editor prior to putting them into production. You can view and edit VDE scripts just like any text file, using your favourite editor if you wish. However, if you change a VDE script outside the Editor, when it is run there is a check of the script against a hash key on the first script line.

If the script doesn’t match the hash key and you don’t have a license to edit, the script will not run at all. If the script doesn’t match and you do have a license to edit, the script is run but a warning is put into the log. Saving the script in the Editor updates the hash key so it will run both on a system licensed for Editor and one not. Note that most production systems do NOT have a license for Editor; programming is all done on a dedicated development system.

The Output Enrichment Editor provides a GUI-based IDE with integrated Debugger (discussed in the next article).

Contact Atac to arrange VDE training, a pre-requisite to becoming a VDE Developer.

Output Enrichment VDE Script Debugger

The Output Manager Viewer is also a VDE Script Editor (see prior article), and a full-blown Debugger as well. The Debugger can replay a job, stepping line by line through your VDE code, showing the code results in the output panel.

To use the Debugger, first run the job to completion in the Job Queue Manager (JQM) and note the Job Number. Next, Start the Viewer and check it is in Browse mode; then click the left-most button to open your job profile (Open Job). Select your profile and click OK. A dialogue appears for you to select the Job Number (defaults to last run) and run either the Input or Output Debugger.

The Input Debugger is used to debug a VDEINPUTSOURCE script, run by the Input client during Input processing i.e. AFP-IN, PDF-IN, XRX-IN etc. The Output Debugger is used to debug a VDEOUTPUTSOURCE script, run by the Output client during Output processing i.e. AFP-OUT, PDF-OUT, XRX-OUT etc.

Choose either "Run Input Debugger" or "Run Output Debugger" and click OK to open the debugger. The Viewer screen maximises and divides itself into several panels. The right-hand panel shows your output; the left-hand side shows a code panel at top and Debugger panel at bottom. All panels can be re-sized. As you move through the script, a pointer in the code panel shows the next statement to be executed.

The Debugger panel includes a button bar with buttons for:

Step-In – step one line of code at a time through your script, for each click

Step-Over – step over the current line of code, executing it but not displaying any called lines

Step-Out – execute code until the current object is exited

GoToCursor – execute code until the cursor location is reached

Go – run till completion, or until a Breakpoint is reached

Set-Breakpoint – puts a Breakpoint at the current line (red dot in left-hand column)

Edit-Breakpoints – invokes a dialogue where you can add, delete or disable Breakpoints based on Location, Data or Test values in the job

Show-Next-Statement – takes you to the next code line to be executed (arrow in left-hand column)

Pause-Debugging – pause execution temporarily

Restart-Job – start the Job debugging again from the start

Exit-Debugger – prompts you to exit the Debugger and finish debugging

The Debugger panel also includes tabs as follows:

Watch – select and examine values of any variable used in your code

Text – displays info about any text present in the current window of interest; rotation, position, number of items etc.

Information – displays info about comments, page and group elements

Time – information re day, month, year, minute, second, time zone etc.

Number – numeric system variables PageNumber, SideID, BeginPageNumber etc.

Misc – HOSTJOBNAME, HOSTJOBNUMBER, VIPJOBNUMBER, FEED, VIPCLIENTNAME etc.

Test – monitor the tests you create in your scripts

Auto-Test – monitor inbuilt tests NOTEXT, FRONT, ENDOFJOB, BACK etc.

The Debugger allows you to step line by line through your code or execute until a condition is met, a line is reached, a value changes etc. You can see program flow and examine variables and their content. As each line is executed, you see immediate results in the output panel. It's an essential tool for any developer and a bit of an eye-opener for trainees too.

Atac provides structured training in all aspects of VDE programming, including the Debugger.

Determining File Formats

Why do we need to determine a print file's FORMAT?

- In order to perform an ad-hoc print file import into BHCS, you need to tell Print Utility what FORMAT the file is stored in.

More information at...

<http://www.atac.com.au>

Email David Kirk

or call +61 (419) 962 386

- If you have several permanently polled “watch” directories for BHCS, you need to know which FORMAT is assigned to each “watch” dir so you can drop your print file into the correct one.
- If you’re writing an Output Manager (Emtex) profile, you’ll need some knowledge of the file’s FORMAT to utilise the appropriate INPUT client and options.
- If you’re in the print bureau business, you’ll be given sample data in a variety of FORMATS to try and work with.

If you haven’t been told specifically how a file is formatted, *you’ll have to work it out for yourself.*

BASICS

When we refer to a file’s FORMAT here, we’re not talking about margins, tab setup etc. We’re talking about the basic structure of the data file, as stored on disk.

- Essentially, most print files are RECORD based, where each record is usually a single line of printable text. A print record can comprise printable characters, comments (data not sent to a printer), printer instructions and graphical information (pictures) among other items.
- Each record may be the same (fixed) length or different from others (variable). If a file has VARIABLE length records, then a mechanism must exist to tell us where each record starts and how long it is. This usually comprises at least two characters (length bytes) at the start of each record, sometimes also repeated at the end of each record. These two characters are interpreted in a special way which allows us to calculate the record length, up until we find the next record/set of length bytes.
- Each record usually has an associated CARRIAGE CONTROL character. This tells us whether we send the print data then advance the paper or advance the paper first. How far do we advance the paper (1, 2 or 3 lines)? Is the carriage control character the first character in the print record or the last? We can even have programmable vertical tab stop positions (up to 12) to which our carriage control character can refer. One permanently defined tab stop is Channel 1 or TOP OF FORM. A print record may therefore say “Print the following data and then advance to the top of the next form”.
- In mainframe terminology, raw carriage control is called MACHINE carriage control and includes over two dozen different paper movement commands. Some of these values are replicated as viewable carriage control characters called ANSI or ASA. A mainframe print file will have either MACHINE or ASA carriage control which PRECEDES the print data in each record.
- Laser printers allow us to select the FONT and SIZE of printed data. The mechanism to achieve this is printer specific and is achieved many different ways. Some print files contain a FONTINDEX character which follows the carriage control byte. This character may refer to a specific position in a list of fonts previously nominated. There may even be a COLOUR byte following the FONTINDEX byte which tells the printer which colour to use for that record.
- Printable TEXT may be in EBCDIC (mainframe language) or ASCII (PC world). ASCII files are “viewable” in NOTEPAD. Although they’ll probably not look exactly like they’ll print, you can read most of the information which ends up on paper. EBCDIC files require a special Viewer to read the print text, such as HEXEDIT, “V” or ULTRAEDIT, via their VIEW | EBCDIC option. Foreign languages such as Chinese, Japanese and Korean with extensive alphabets are represented by a pair of characters for each foreign language character; this is called DOUBLE-BYTE data.

This all affects how we are able to read each record and send it to a printer.

TOOLS

If you perform the task of determining file formats regularly, you’ll end up using a variety of tools. Here are a few Windows tools I use to examine and determine file formats:

- Hexedit – Hex view of a file, can display ASCII and EBCDIC character set, can BINARY edit files, search/replace, can only display/print in hex mode.
- “V” – www.fileviewer.com – Plain text view of a file as well as hex mode. ASCII/EBCDIC, search/replace, line/column info on-screen, many options. This is a commercial product which you must pay for after the evaluation period.
- DEBUG – Don’t look surprised! DEBUG is part of every DOS/Windows computer and is run from a Command Prompt. DEBUG <filename> will invoke a very unfriendly “-” prompt only. Type “d100” and press ENTER to see a hex view of the first 128 bytes of <filename>. Type “q” and press ENTER to quit.

More information at...

<http://www.atac.com.au>

Email David Kirk

or call +61 (419) 962 386

- UltraEdit – Powerful commercial product, heavily biased towards programmers. ASCII/EBCDIC, search/replace full editing, many display modes, “huge” file support.
- Notepad – Free, present on any Windows system, small ASCII files only, search/replace.
- Beyond Compare – Instantly compare any two files and see differences.

FILE TYPES

Here are the main “standard” print file types, although this list is not in any way exhaustive:

- ASCII – VARIABLE LENGTH, ASCII text records followed by carriage return / line feed / form feed CARRIAGE CONTROL characters.
- ASCII with ASA CARRIAGE CONTROL. Exactly as ASCII but with a single leading ASA CARRIAGE CONTROL character, which can be “`“`”, “`1`”, “`2`”, “`3`”, “`+`”, “`+`” and controls the vertical print paper movement. Each record is terminated with carriage control / line feed; these are print record delimiters only and are not sent to the printer. Typically produced by legacy mainframe (COBOL) apps.
- Barr/DOS – Produced by and for the original Barr DOS program, also known as Barr S/370 format. Data can be ASCII or EBCDIC. Comprises four byte hex signature “`76 1A FF FF`” implicitly identifying the file. Subsequent records have a pair of length bytes, followed by the print record and then a matching pair of length bytes at the end of the record. The length bytes refer to the record length EXCLUDING the length bytes themselves. So a Barr/DOS file which starts with `76 1A FF FF 00 03 01 44 45 00 03` tells us the first data record is three bytes long and comprises characters “`01`”, “`44`” and “`45`”.
- VBM – VARIABLE, BLOCKED, MACHINE Carriage Control. This is a common format for Xerox print files which typically contain BINARY positioning and resource information. Each record is preceded by FOUR length bytes; only the first two are used, with the remaining two set to NUL. A machine CARRIAGE CONTROL byte follows, then the data. The record length bytes INCLUDE themselves in the length calculation. There are no trailing length bytes. Records may be gathered together into BLOCKs which have their own four byte BLOCK LENGTH bytes; once again only the first two bytes are used. Therefore a VBM file which starts like this: `24 63 00 00 00 06 00 00 89 40...` tells us that the first BLOCK of records is 2,463 (HEX) long, and the first record is 6 (HEX) bytes long, including the four length bytes. The CARRIAGE CONTROL byte is “`89`” and the data is 40 HEX, or the EBCDIC SPACE character.
- AFP print data is IBM’s document format, which contains large amounts of data about a print file. With fully composed AFPDS each and every data record starts with a hex value of “`5A`” followed by a two byte length field (for example `00 19`) and then a three byte structured field which defines the function of the record (for example `D3 B1 8A`) and finally the data. The length bytes INCLUDE themselves in the length calculation. Carriage control may be Machine or ASA. AFP data can be ASCII or EBCDIC, FIXED or VARIABLE length.
- MAINFRAME IP FORMAT (EPS) – Very similar to VBM except the length bytes are in reverse order and a special 128 byte header record contains job attribute info prior to the print file date. This format is known as LCDS mode to CA-SPOOL and VPS. It is also known as EPS format to Xerox and Barr.
- Postscript and PCL print files are effectively print STREAMS with no fixed record structure. Positioning commands within the stream can place data anywhere on the page, with no specific carriage control necessary. Typically, the text is ASCII. Can contain IMAGES, generic drawing commands, COLOUR and positioning information via ESCAPE SEQUENCES which the printer recognises and obeys rather than prints. Postscript files are viewable in NOTEPAD whereas PCL files can contain non-viewable characters as well as ASCII text.

Got an unrecognisable print file format? Atac can help you work out what it is.

Q & A

Q. How do I determine what format an LPD file is in so I can receive it into BHCS correctly?

A. It’s best to see the data before Barr manipulates it in any way. You can turn on an LPD TRACE which captures unadulterated data as it arrives at the Barr PC. Choose TOOLS | TCP/IP CONFIGURATION | START (LPDTRACE). Receive your LPD print data, then click STOP (LPDTRACE). Close the TCP/IP CONFIG. Utility. You’ll find a file called LPDTRACE.TXT in the C:\Program Files\Common Files\Barr\Trace directory.

More information at...

<http://www.atac.com.au>

Email David Kirk

or call +61 (419) 962 386

More information at...

<http://www.atac.com.au>

Email David Kirk

or call +61 (419) 962 386

Examine this file closely to determine the format yourself OR send the first couple of pages of LPDTRACE.TXT to Atac for free analysis. Once you've determined the correct FORMAT, then you can modify the relevant TCP/IP LPD Queue File Format settings to expect that particular FORMAT. Receive your file again; if all works correctly, your file is viewable as TEXT in the Barr Viewer.

Q. How does the "SEND TO" feature work on the right-click menu for a SPOOL job?

A. You can send documents to programs of your choice. Destination programs are stored in C:\Program Files\Barr\Spooler\SendTo. If the SendTo folder does not already exist, it will be automatically created when you right-click a document in the Spool Window.

You must be granted the **Document – View** user access right to use this option.

Once you have established destination programs, they will appear in the Spool Window's **Send To** list. A maximum of 20 destination programs can display in the **Send To** list. To set up destination programs, either create program shortcuts or copy programs to the SendTo folder.

To send a document to a destination program, use one of the following methods in the Spool Window.

On the menu bar, select Document | Send To | **Program**.

Right-click a document and select Send To | **Program**.

Files sent from the spool to other programs are read-only. If you want to modify and save a file, use the program's **Save As** option.

The "SEND TO" feature can be used, for example, to setup a link to Acrobat PDF reader to enable rapid viewing of PDF files.

Q. When I receive an encrypted or compressed file from another BHCS unit, my BHCS tries to re-encrypt or re-compress it when I send it to my local printer. Why does this occur and how can I fix it?

A. Transfers between BHCS units are performed in BARR ARCHIVE format which preserves all settings, **including the encryption and compression field contents**. The file therefore arrives on your BHCS with both these fields (still) set. In order to print to a local printer, the fields need to be cleared. This is normally done at receive time by an Override Table rule, unconditionally clearing the NDHBCMPL and NDHBKEYN fields. (If the file is later required to be on-forwarded to another BHCS, you'll need to manually re-populate either or both of these fields again.)

Feedback

Your feedback is valuable to Atac and helps us provide the high standard of service which you have come to expect from us, and which we are proud to deliver. Whether you have questions about our product range, a story to share regarding your experiences, or you would like to comment on our newsletter, just send us an **email**.

Subscribe

If you would like us to send a copy of this newsletter to additional people, send us an **email** containing their email address(es).

Unsubscribe

If you would prefer not to receive this newsletter, send us an **email**.

Atac's Privacy Policy

At Atac we value your input, and we keep it private. We will never make any of this information available to any other company, nor will it be sold or traded in any way.