

# newsletter



**PGCG**

Pretty Good Consulting Group

Got a problem? We create solutions.

**EOM News and Views**  
**Issue 5, September 2008**

## Keyword substitution in File Masks, (take 2)

Using EOM's Keyword substitution feature in your File Masks reduces the number of File Masks and saves you time.

The hate mail started arriving almost immediately after last month's newsletter was posted. "Why would anyone use <DAY> as a Print Attribute?", "Our printer names aren't that consistent", "Do you have any REAL keyword substitution examples?" and so on. Points taken, so lets look a little closer at a couple of real-life examples:

### 1) Using the <DAY> keyword substitution for a Print Attribute

Believe it or not this use of keyword substitution just solved a real-life customer issue as the August newsletter went out. The customer has a report printing every day, but only needs to print/email that report certain days of the week. So the question is : How can EOM print/email the report on certain days of the week?

Answer: Create seven Print Attributes, one for each day (i.e. MONDAY, TUESDAY, WEDNESDAY,...). Then for

the days you want don't want to print, set the page range to 99999-99999 (or something beyond the last page). Make sure that you also uncheck the EOM Header/Trailer in the Print Attribute for the days that do not print. Your Print Job in the File Mask would like something like " <DAY> | Default" . Even though you have a Print Attribute for every day it is a good idea to have a default Print Attribute just in case. (By the way, another way to accomplish this task is to create a DDA with logic in a DDA to check the day, then immediately do a "End Report", which stops printing)

### 2) Using <USERTAGx> to control both the printer and the print attribute

There are multiple customers already using the values in the USERTAGx keywords to control how (Print Attribute) and where (Physical Printer) the data stream is printed.

Setting up File Masks for every Print/Email/Transfer/Custom/Backup/Index Job can get tedious some times. Worse, if there are 12 different "how" data streams are to be printed (Print Attributes) and 100 "where" (Physical printers) the data streams could be printed, then we are talking about 1,200 File Masks - clearly this is a lot to manage. So, if you sweet-talked your application developers into preceding each data stream with values that helped figure out the "how" and "where" your life could be a lot simpler. There is also EOM specific syntax that could make your life easier (go to the On-line Help and search on \$DEPHDR\$):

```
$DEPHDR$ STOREx C85L66DUP  
{rest of file}
```

The File Mask would have a Print Job with a printer of <USERTAG1> and the Print Attribute would be <USERTAG2>.

## Contents

Keyword Substitution.....	1
DDA Feature.....	1
How do I...? .....	2
EOM for MCP.....	3
Quick Hits.....	8
Contact information .....	8

## DDA Feature (you may not know about)

Ever need to do some processing after the last page has printed? There are times when it would be very handy to be able to have a set of DDA commands execute after the last page has printed. The problem is that DDA does not have an obvious way to tell you when the last page is done. One way to handle this is to use a Trailer, Print Attribute for that Trailer, and a DDA that the Print attribute references:

Print Attribute:	MAIN
DDA:	MAIN DDA
Print Trailer:	Yes
Trailer Attribute:	MyTrailer
Trailer Print Attr:	FinishUp
DDA:	TheEnd

The Trailer Attribute "MyTrailer" does not have to have any Line Items, but the Print Attribute used

for the Trailer does need to reference a DDA ("TheEnd"). This DDA then can do whatever is required to finish the Print Job. Keep in mind that any variables used in the original DDA are still available for the DDA referenced by the Trailer Print Attribute/DDA. Also, the Physical Printer must have "Print Trailer Page" set to "Yes".

One real-life use of this technique is at a bank that processes check information. The original DDA reformats the data for output, and keeps a running total of the number of items (checks) and check totals in DDA variables. When the Trailer DDA executes, creates a human readable sentence using those values, and then uses the Get User Input command to display the sentence so that the operator can confirm the number of items and the total amount from that file.

# How do I ...?

This section of the newsletter will discuss solutions to questions that come from real customers trying to solve real problems. The question came up: What do I do when a printer hangs in EOM?

Every once in a while we hear from a customer that is testing a new Windows printer driver, usually related to creating files, that gets hung up in EOM. The usual symptom is that the printer appears to start printing, or is in pre-processing/post-processing and just doesn't change. The good news is that since EOM 7.0 every Print Job printing on a printer is one its own thread so the rest of EOM keeps processing. The bad news is that there are a few cases where EOM does not recognize that a printer is hung, hence doesn't automatically interrupt the thread.

There is a way out ... Since you usually can't live with a perpetually hung-up printer, the way to resolve this problem is:

- 1) Go to the server that runs the EOM service
- 2) If the EOM client is running: close Print Preview, close the Configuration Explorer, and finally close the client
- 3) Under Computer Management / Services, find "Unisys Enterprise Output Manager", right-click, and stop the service (it may not come down)
- 4) Get to the Task Manager (Ctrl-Alt-Del) "Processes" tab, find the "EomService.exe" process, right-click and select "End Process"
- 5) Restart EOM

But there is a catch: it could be the case that the offending Print Job starts up again, hanging the printer exactly the same way. This calls for more drastic measures - find the name of the file for this Print Job and delete it after step "4)" then restart EOM. Before you take the EOM client down simply right-click the file in the File or Print Job Management window and select "Properties..." then remember the value found in "Data Directory/File Name".

# EOM notes for MCP systems

While we do not want to show favoritism for one operating system over another, there have been quite a few contacts and questions regarding EOM on the MCP. So, let us spend some time to walk through some of the basics:

1) EOM installation on the MCP system is a matter of copying the files from the release CD to disk, setting some privileges, and then modifying the MCP EOM configuration for your particular site. The EOM library has to run privileged and have access to the TCP/IP stack. Example console syntax is:

```
MP *OBJECT/DEPCON ON DISK + PU + SERVICE "TCPIP".
```

You also have to allow virtual servers and IOHandlers to execute by setting the ACTIVE SITE value: PS ACTIVE SITE=10 which allows for 10 concurrent Virtual Server and/or IOHandler library calls. The OBJECT/DEPCON/CONFIG configuration file is not required, but does have some useful settings. FORMIDISPAPERTYPE= set to TRUE tells the EOM library to send the FORMID text value to the EOM PC overwriting the Print Attribute Paper "type field". Configuration parameter FILENODEMASK tells the EOM library which parts of the MCP filename to the EOM PC. For example, FILENODEMASK= FIRSTTWO means that for a filename of (usercode)NODE1/NODE2/NODE3/NODE4 on PACK the EOM library would send "NODE1" for the "FILE NODE A" File Mask field and "NODE2" or the "FILE NODE B" File Mask field. Probably the most important parameter in the EOM library configuration file is the ROUTE statement. ROUTE is used to relate a text value with a text value or actual IP address of a remote EOM instance:

				LOCAL	REM	PING
. ROUTE	DESTINATION	HOSTNAME	SOCKET	TSAP	TSAP	BLOCKS
-----	-----	-----	-----	-----	-----	-----
ROUTE	EOM-PC1	EOM-PC1	102	DEPCON	DEPCON	7
ROUTE	EOM-PC2	depcon.PGCG.com	102	DEPCON	DEPCON	25
ROUTE	EOM-PC3	192.161.161.161	102	DEPCON	DEPCON	7

Note that the HOSTNAME field can be a DNS resolved name, fully qualified DNS name, or a static IP address. By the way, the PING BLOCKS value is the number of data blocks sent from the MCP EOM to the remote EOM, followed by a confirmation from the remote EOM. This is for flow control, normally the value of 7 is sufficient.

2) Next, we need to get the PrintS device configured. There are two choices for EOM: Virtual Server or an IOHandler. The difference between the two, at a high level, is that when a print request is submitted to a PrintS Virtual Server, PrintS calls the Virtual Server Library entry point (DEPCONSERV for EOM library) for each "piece" of the print request. So, the library will be called for the header, banner, each file in the print request, and finally the trailer. The Virtual Server does whatever it is supposed to do with each piece of the print request and then returns a status to PrintS, usually saying that it completed the piece of the print request successfully. For an IOHandler, PrintS can call a few entry points depending on which part of print request is currently being handled. Example entry points in the EOM library are "DEPIO\_OPEN" to open a connections, "DEPIO\_CLOSE" to close, "DEPIO\_SEND" for each record, and so on. Status is returned to PrintS on each call to the library entry point. The major difference between the IOHandler and the Virtual Server is that the "DEPIO\_SEND" entry point is called for each line of data and/or carriage control in the print stream whereas the Virtual Server is expected to handle the whole file itself. One of the major benefits of using IOHandlers is that PrintS does all the formatting (like handling of PAGECOMP, Device Drivers, and so on).

Example syntax to define EOM devices to PrintS are:

```
PS CONFIG + EOMPRINTER1 IOHANDLER="DEPCONSERV (OMAHAEOM/EOMPRINTER1 ) IN *OBJECT/DEPCON ON DISK"
```

```
PS CONFIG + EOMPRINTER2 SERVER="DEPCONSERV (OMAHAEOM/EOMPRINTER2 ) IN *OBJECT/DEPCON ON DISK"
```

[WFL1](#)

[COBOL](#)

[ALGOL](#)

[WFL2](#)

# EOM notes (continued)

Note that both IOHandler and Virtual Server entry points are supported and can be used simultaneously. The first parameter ("OMAHAEOM") is the name of the remote EOM instance, which either resolves via DNS or has a ROUTE statement definition. This means that the value could be an IP address too - be careful though, using IP addresses in PrintS definitions means that you must change all definitions if the IP address changes as opposed to changing one line in the OBJECT/DEPCON/CONFIG configuration file. The second parameter ("EOMPRINTER1") is the value sent to the remote EOM instance for the "Destination" File Mask Field. Usually the value is identical to the PrintS device name, but it does not have to be.

There is additional (optional) EOM library syntax allowed on the PrintS definition. One common example is:

```
PS CONFIG + EOMPRINTER3 SERVER="DEPCONSERV (OMAHAEOM/EOMPRINTER1/NOCTRLFF ) IN *OBJECT/DEPCON ON DISK"
```

The "NOCTRLFF" syntax tells the EOM library to send the print request without any print control information. The only non-printable characters included are the carriage return, line feed (CR-LF) at the end of each line, and translates page ejects into the formfeed (FF) character.

There are a variety of PrintS supported settings for Virtual Servers and IOHandlers, the ones we are most concerned are:

```
PS CONFIG EOMPRINTER1 SERVERPRIORITY=50          (often the default is 80, which is too high and conflicts with
other processing)
PS CONFIG EOMPRINTER1 BLOCKSTRUCTURE=UNBLOCKED  (if EOMPRINTER1 is defined as an IOHandler)
```

Please refer to the Enterprise Output Manager for ClearPath OS 2200 and ClearPath MCP Configuration and Operations Guide for a lot more information on syntax options.

3) Finally, we are ready to submit print requests to PrintS defined EOM devices. There are a variety of ways to submit print requests: CANDE, WFL, console, and programmatically. More complete examples can be viewed by selecting the icons on the right hand side of the page.

A CANDE session user might use syntax along the lines of:

```
PDEF(DESTINATION="EOMPRINTER1")
PRINT (ACCOUNTING)MONTHLY/STATEMENT/REPORT/= ON PACK
```

WFL provides a boat-load of capability to for PrintS interaction, a simple WFL syntax example:

```
...
REMOVE BLUECROSS/DEMO/= ON PACK;
RUN OBJECT/BLUECROSS;
  BDNAM=BLUECROSS/DEMO;
```

Or

```
PRINT WFL/BLUECROSS;
  PRINTDEFAULTS=(DESTINATION="EOMPRINTER2",
  BANNER=FALSE,
  PAGECOMP="LANDSCAPE LPP=66 LPI=6 CPL=132 BORDER=(0.25 0 0.25 0) LM=0",
  HEADER=SUPPRESSED, TRAILER=SUPPRESSED);
```

Or an example WFL from Kester that parameterizes the email syntax ([click here](#)).

# *EOM notes (continued)*

Operator interaction via the console could be a WFL PRINT command or, more likely, using the PrintS interface to redirect print requests to PrintS defined EOM devices (PS MODIFY ...).

Programmatic submission of print requests to PrintS allow for the PrintS syntax, but the additional capability lies with the EOM-specific syntax available.

A COBOL example:

```
...
101400 WORKING-STORAGE SECTION.
101500 77 CR          PIC X(01) VALUE @0D@.
101600 77 ESC        PIC X(01) VALUE @27@.
...
102300 PARA-2.
102400     STRING
102500     ESC          DELIMITED BY SIZE
102600     "60"         DELIMITED BY SIZE
102700     ESC          DELIMITED BY SIZE
102800     "S  "        DELIMITED BY SIZE
102900     "DEPCON-EMAIL=" DELIMITED BY SIZE
103000     "STEVED@PRETTYGOODCONSULTINGGROUP.COM" DELIMITED BY SIZE
103100     CR           DELIMITED BY SIZE
103200     INTO WHOLELINE.
103300     WRITE PRINT-REC FROM WHOLELINE BEFORE ADVANCING 0 LINES.
103400
103500     MOVE SPACES TO WHOLELINE.
103600     STRING
103700     ESC          DELIMITED BY SIZE
103800     "60"         DELIMITED BY SIZE
103900     ESC          DELIMITED BY SIZE
104000     "S  "        DELIMITED BY SIZE
104100     "DC-ES="      DELIMITED BY SIZE
104200     "This is an example subject." DELIMITED BY SIZE
104300     CR           DELIMITED BY SIZE
104400     INTO WHOLELINE.
104500     WRITE PRINT-REC FROM WHOLELINE BEFORE ADVANCING 0 LINES.
...
```

# *EOM notes (continued)*

An ALGOL example:

```
DEFINE CTRLSEQ = 48"27", "60", 48"27", "K "#;

OPEN(OUTFILE);

REPLACE SCRATCH BY " " FOR 80;
WRITE(OUTFILE[SPACE 6], 80,SCRATCH);
REPLACE SCRATCH BY CTRLSEQ, "FONT1.PCL "; % Initial font
WRITE(OUTFILE, 80,SCRATCH);
REPLACE SCRATCH BY " " FOR 80;
REPLACE SCRATCH BY " Welcome to the world of DEPCON!";
WRITE(OUTFILE[SPACE 4], 80,SCRATCH);

REPLACE SCRATCH BY " " FOR 80;
REPLACE SCRATCH BY CTRLSEQ, "FONT2.PCL"; % second font
WRITE(OUTFILE, 80,SCRATCH);
REPLACE SCRATCH BY " " FOR 80;
REPLACE SCRATCH BY "We will show how easy it is to change fonts ";
WRITE(OUTFILE, 80,SCRATCH);
REPLACE SCRATCH BY "in the middle of a print request. This page ";
WRITE(OUTFILE, 80,SCRATCH);
```

# Upgrade now!

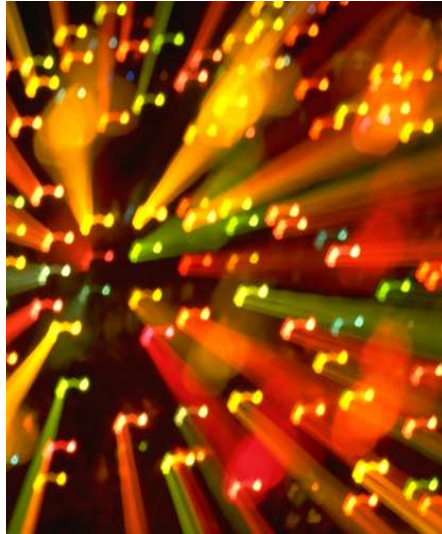
I know, you have heard it before: upgrade the software because the vendor says it is the thing to do. Well, .... they are right in this case. EOM versions 6.1 and below are no longer supported. However, the major reason to upgrade to EOM 7.1 is that the latest and greatest software provides latest and greatest capability -- that you can really use. This is not bells and whistles, this is functionality that increases your productivity and likely reduce cost of output management. In addition, Interim Correction 7.1.6 fixes a significant Windows print driver issue.

## Who are we?

Highly skilled, creative, solution provider focused on the Unisys Enterprise Output Manager product (formerly known as DEPCON) with a sense of urgency sums up who we are and what we do. We provide general Enterprise Output Manager consulting, migrations, upgrades, configuration, training, and custom programming.

On-site services, remote services, and general consulting are available now.

Why use PGCG? Deep knowledge of the EOM product integrated into a variety of customer environments sets us apart. Our customers production environment depends on solid, working solutions that we provide.



## Quick Hits

A few of you had issues with the EOM Administrator program installation. Contact Unisys for an updated install if you are still having trouble, or try the workaround by found by WHO???: If a previous version of the Administrator program was already installed then just remove the previous version Windows Add/Delete, and then run the install from the 7.1 CD. This avoids the upgrade from 6.1 to 7.1, which appears to caused the installation to fail.

---

When upgrading from EOM 6.1, make sure you replace User Jobs with updated Custom Jobs.

---

Have a suggestion for "How do I ...?" Write a brief description and send it to [SteveD@PrettyGoodConsultingGroup.com](mailto:SteveD@PrettyGoodConsultingGroup.com) for future newsletter discussion.

---

Interested in EOM training? We can either do custom training on-site or arrange for a formal class through Unisys. Please contact us for details.

---

Pretty Good Consulting Group, LLC  
4235 Ivy Court North  
Lake Elmo, Minnesota 55042  
[www.PrettyGoodConsultingGroup.com](http://www.PrettyGoodConsultingGroup.com)