The Caché MultiValue Spooler

Version 2018.1
2019-09-20

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

The Caché MultiValue Spooler
Caché Version 2018.1 2019-09-20

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

Table of Contents

1

Introduction

Normally, output from **PRINT** statements is directed to the user terminal. However, MultiValue supports mechanisms, such as the (P) option on the **PRINT** statement, that can be invoked to redirect output elsewhere. This is called SPOOLing. (The Jargon File indicates the acronym stands for a method of decoupling output destined for the printer from the actual device used to print it.)

This document describes the spooler used by Caché MultiValue programs and how to make use of it. The spooler makes use of standard Caché facilities to provide functionality comparable to that found on most MultiValue implementations.

1.1 About The Spooler

In Caché, when a user creates the first print job in the global, ^%MV.SPOOL or any other global specified with the SP-GLOBAL command, the runtime automatically initializes and creates the entries necessary to accept spooled data. You can also do this manually with the SP-NEWTAB command.

By default, printed data is collected in the system-wide global, ^%MV.SPOOL. This means that by default there is one spooler table per Caché instantiation, but this can be changed via the SP-GLOBAL command.

Also by default, MultiValue assigns print jobs to the form queue, STANDARD. The SETPTR command is used to manipulate the details of a particular job's output such as its form queue, page dimensions, banner text, and so on. SETPTR can also be used to direct output to a special file called &HOLD&. Output in &HOLD& is not handled by the spooler; administrative action is required before it can be processed.

Note: Caché MultiValue does not support the option of having an application print directly on a printer. If a MultiValue application really wants to print directly to the device, it must use classes inside the MultiValue application that access the Caché facilities for this purpose.

Caché MultiValue does not support the option of having an application print directly on a printer. Instead, Caché supports the "A" option on the SP-ASSIGN statement. This means the despool process will start to transmit the data from ^MV.SPOOL to the printer while the print job is still being created. This gives a timing effect very similar to direct printing except that the application does not communicate directly with the printer.

1.2 Spool Jobs

A spool job can be created in one of three ways:

- By using the (P) option on the command line.
- By executing a PRINTER ON or PRINT ON nnn statement in MultiValue Basic.
- By using the LPTR clause in a CMQL (Caché MultiValue Query Language) statement. For example, **LIST VOC LPTR**.

In each of these instances, output destined for the printer is collected into a Caché global and grouped together to form a print job. This print job is associated with a form queue. The form queue defines a collection of print jobs destined for a specific printer. Administrators can manipulate the form queue and the individual jobs within it. A despooler process will eventually process the queue, writing information from the Caché global to a specific printer, and clearing the global when all the information has been disposed.

Caché supports the MultiValue concept of a file called &HOLD&. The use of the &HOLD& file is initiated by specifying mode 3 in the SETPTR command, after which all printing output will go to the &HOLD& file instead of the spooler global.

The &HOLD& file is created automatically after the first SETPTR verb is executed that specifies mode 3. Caché MultiValue creates &HOLD& as a directory in your installation hierarchy. The &HOLD& file may be located anywhere; by default, it will be placed in the directory associated with the namespace of the user that creates it. In the example that follows the &HOLD& file points to the C:\TMP directory and so all spooler output in mode 3 will go to operating system files in the C:\TMP folder.

```
USER:SETPTR 0,132,66,2,2,3,BRIEF
Creating &HOLD& file.
USER:DELETE-FILE &HOLD&
[440] DICT '&HOLD&' Deleted.
[441] Default Data Section '&HOLD&' deleted.
[443] VOC entry for file '&HOLD&' deleted.
USER:CREATE-FILE &HOLD& DIR C:\TMP
[421] DICT for file '&HOLD&' created. Type = INODE
[429] Default Data Section of '&HOLD&' set to use directory 'C:\TMP'.
[437] Added default record '@ID' to 'DICT &HOLD&'.
[417] CreateFile Completed.
USER:DELETE-FILE &HOLD&
[440] DICT '&HOLD&' Deleted.
[441] Default Data Section '&HOLD&' deleted.
[443] VOC entry for file '&HOLD&' deleted.
```

The &HOLD& file must be of type DIR (an operating system directory/folder) or of type ANODE (a Cache global where each node represents one attribute). For example:

```
USER:CREATE-FILE &HOLD& ANODE
[421] DICT for file '&HOLD&' created. Type = INODE
[418] Default data section for file '&HOLD&' created. Type = ANODE
[437] Added default record '@ID' to 'DICT &HOLD&'.
[417] CreateFile Completed.
USER:
```

Child programs that are started automatically inherit the printer status (unless this has been negated by the SP-CONDUCT command). In the following example, the second program “PROG2” will automatically send its output to the spooler even though it never specifically executed a “PRINTER ON” statement. This is because it inherits the printer status from its parent program “PROG1”:

```
    PROG1
001 PRINTER ON
002 PRINT “Hello”
003 EXECUTE “PROG2”
004 STOP

    PROG2
001 PRINT “World”
```

In the above example, the child program “PROG2” will inherit the same spooler print job started by its parent program. The resulting spooler print job will have 2 lines consisting of the words, “Hello” and “World”.

Any opened spooler print job will remain open until one of the following occurs:

1. The program executes the SP-CLOSE command.

2. The program executes the MVBasic statement: PRINTER CLOSE.
3. The program that originally started the spooler print job terminates.
4. The user logs off.
5. The programs executes a CMQL command such as LIST or SORT. (This might seem an anomaly, but it is UniVerse behavior.)

Note: The default behavior of spool jobs as described in this section depends on the current emulation. Applications that wish explicit control of spooler behavior should do so with the SP-CONDUCT command.

2

Spooler Administration

2.1 Spooler Administration

MultiValue spooler administration involves three classes of actions:

- Using of shell commands such as **SETPTR** to control the sending of print jobs to the printer.
- Issuing shell commands to manipulate the jobs, form queues and despool processes.
- Running the **SP-JOBS** and **SP-STATUS** menu driven utilities to manipulate the jobs, form queues, and despool processes.

2.1.1 Form Queue Names And Numbers

Throughout this manual we refer to form queue names and form queue numbers. A form queue is a collection of globals that define and control a pseudo printer. A despool process later comes along and sends the output to a real printer.

A form queue actually has three identifiers:

- The user-specified form queue name (or form queue number).
- The system-assigned form queue number.
- The system-assigned spooler global subscript.

UniVerse-like MultiValue platforms tend to use form queue numbers. Pick-like MultiValue platforms tend to use form queue names. Caché MultiValue supports both. By default, the form queue globals are indexed under form queue number. Each form queue number has a corresponding form queue name.

If the form queue name is numeric, or starts with a letter (other than F, perhaps followed by Q or N, followed by a number), you can use the form queue name, or any of the formats listed below to identify a form queue to one of the SP- commands:

- A form queue reference in the format *nn*, for example, 99, describes form queue number 99.
- A form queue reference in the format F*nn*, for example, F99, describes form queue number 99.
- A form queue reference in the format FQ*nn* or FN*nn*, for example FN99 or FQ99, describes form queue number 99
- A form queue reference in the format FQ*name* or FN*name*, for example FQHP7210 or FNHP7210, describes a form queue named HP7210.
- Anything else, for example, “STANDARD”, describes the form queue named STANDARD.

In the above rules, the first three result in a reference to a form queue *number*, and the last two result in a reference to a form queue *name*. A form queue number will always have a corresponding form queue name, and a form queue name will always have a corresponding form queue number.

In all the administrative commands that ask for a FORM QUEUE you can specify the form queue number or the form queue name. For example, using the commands SP-CREATE and SP-ASSIGN:

```
SP-ASSIGN 3=F12 (HS
```

means assign print channel 3 to form queue number 12 , with “H” and “S” options.

```
SP-CREATE HP7210 CACHE NULL
```

creates a form queue whose name is HP7210. The form queue number is allocated automatically as the next free unused form queue number.

```
SP-CREATE F15 CACHE NULL
```

creates a form queue whose name is F15 and whose form queue number is 15.

```
SP-FORM F15 HP7210
```

Referring to the preceding example, this renames the form queue named F15, which is also form queue number 15, to HP7210 and it retains form queue number 15.

2.1.2 Form Queue Groups

you can optionally create one or more form queue groups. A form queue group consists of a list of existing form queues that can be used as a collection of form queues. A form queue group may be used by the despool process to send spooler output to all form queues in the group, rather than just one form queue. form queues that are grouped together should share some commonality, such that grouping them together makes administration easier.

You use SP-CREATE to create a form queue group by specifying GROUP as the second argument, then listing multiple existing form queues. Illegal and duplicate form queue names are ignored. You can create a form queue group without specifying its form queues by specifying the empty string ("") as the third argument, as follows:

```
SP-CREATE MYFQGRP GROUP ""
```

You can use the SP-DEVICE command to add, remove, or replace form queues from the form queue group.

2.1.3 MultiValue De-Spool Devices

The MultiValue spooler allows for a despool process to send print jobs from the spooler tables to an external resource, for example a printer. The despool process is controlled using the commands: SP-START, SP-STOP, SP-SUSPEND, SP-RESUME and SP-KILL.

The despool process writes to a device as defined by the form queue. This device is specified when creating a form queue with the SP-CREATE command. You can change the device associated with a form queue using the SP-DEVICE command.

When defining an output device for the despooler, you use a similar syntax as defined for the ObjectScript OPEN statement: Device{:parameters}{:timeout}. You can specify any device supported by Caché and so you can write spooler output to printers, files, tapes, terminal screens, programs, and so on. For further information on devices supported see the Cache I/O Device Guide.

2.1.4 The &HOLD& File

You implicitly create the &HOLD& file the first time you use the SETPTR command and specify mode 3 (the sixth argument). The spooler creates &HOLD& as a directory-type file in the current working directory (identified in the @PATH variable). Normally &HOLD& should be a directory-type file, but it can be pre-created as an anode-type if that is preferred. For example, on a new account, the following commands show this happening:

```
NEWACCOUNT:SETPTR 0,100,30,2,2,3,BRIEF,BANNER NEXT,INFORM
  Creating &HOLD& file.

  NEWACCOUNT:ED VOC &HOLD&
  &HOLD&
  6 lines long.
  ----:P
  0001: F
  0002: &HOLD&
  0003: ^DICT.HOLD
  0004:
  0005:
  0006:
  Bottom at line 6.
  ----:EXK
  NEWACCOUNT:
```

In the definition of &HOLD&, attribute 2 means a directory in the Mgr/namespace installation directory will have been created where the file resides. For example, the account NEWACCOUNT, by default this uses namespace NEWACCOUNT and so the &HOLD& file will be a directory called something like C:\InterSystems\Cache\Mgr\NEWACCOUNT\&HOLD&.

This action can be overridden. On a new account, simply create the type of &HOLD& file you want, for example:

```
NEWACCOUNT:CREATE-FILE &HOLD& DIR C:\TEMP\MYHOLDFILE
```

or

```
NEWACCOUNT:CREATE-FILE &HOLD& ANODE
```

If the &HOLD& file has already been created, you can simply delete it with DELETE-FILE and manually re-create it as shown.

The &HOLD& file can therefore be any regular MultiValue file of type DIR (points to an operating system file path), or of type ANODE (a special Caché type suitable for large items). A &HOLD& file cannot be an INODE file.

2.1.5 Form Queue Control

Caché allows for a subroutine to be called either before, or after, or both before and after a job is printed on a form queue. The purpose of the subroutine is to allow an arbitrary sequence of characters to be sent to the spool device, most typically for fine control of the device, such as changing the device from portrait mode to landscape mode.

The general name of the subroutine to be called is contained in the variable, *MVCACHEPRINTER*. If a subroutine by this name exists, it will be called once at the start and at the end of every print job. If the subroutine named by *MVCACHEPRINTER* does not exist, no action is taken.

Note: If the print job is printed more than once, for example when copies is set to a value greater than 1, then this subroutine is called multiple times, once per copy of each print job.

Caché also provides for finer control over the routines called by allowing them to be specified separately per form queue via the SP–PREAMBLE and SP–POSTAMBLE commands.

The API of the pre/post–amble routine called is:

```
SUBROUTINE AMBLER (OCCASION, JOBNO, FQNAME, GLOBAL, OUTPUT)
```

where:

- “AMBLER” is the name of the routine which will handle the queue
- “OCCASION” is the string “PRE” or “POST” indicating why the routine was called; this allows the same routine to handle both actions
- “JOBNO” is the job number being output
- “FQNAME” is the spooler global subscript. To convert this subscript to a form queue name or number, do the following:

```
$XECUTE 'SET %EXTERNALNAME = ' : GLOBAL : '("' : FQNAME : '","NAME")'
```

- “GLOBAL” is the name of the Caché global holding the spooler data
- “OUTPUT” is the string of characters that should be sent to the spool device before the job starts (if OCCASION is “PRE”) or after the job finishes (if it is “POST”).

By default, all accounts use the same spooler. Since the preamble and/or postamble subroutine might be called from an account other than where it was defined, the subroutine should be cataloged globally. This is easier to manage than cataloging locally or normally in every account that might need to use it.

2.1.6 Auxiliary Printing

Auxiliary printing occurs when a spool job is directed to a printer connected directly to the terminal or personal computer rather than to that controlled by the spooler. Using this facility, an application can print data to the spooler and the job will appear on the normal spooler tables in the assigned form queue. When the print job is closed, the spool job that was created will be sent to the user’s auxiliary printer.

Auxiliary printing is initiated by the **SP-AUX** command using the “A” option:

```
USER:SP-ASSIGN =HP7210 A
```

To check if the spool job will be directed to the local printer, look for the “AUX” option on the job in the form queue. For example:

```
USER:SP-LOOK

            FORM QUEUES

Chan Q#   Q name        Width Lines Top Bot P#  Options


0   1    HP7210        132   66    3   3   1    AUX, INFORM
```

Caché must know the control code sequences to be sent to the terminal to turn on and off auxiliary printing. These are contained in the terminal definition file, TERMDEFS, documented elsewhere. Each terminal type has an entry in the TERMDEFS file. The control definitions used by Caché are “mc5” to turn on auxiliary printing and “mc4” to turn off auxiliary printing. These match those used by most terminal emulators. Without these definitions in the terminal definition file, auxiliary printing will not work.

The **SP-AUX** command can be used to print existing print jobs to the auxiliary printer. For example, the following command causes Caché to print existing jobs 22 and 23 to the own auxiliary printer:

```
USER:SP-AUX 22 23
```

If your terminal definition does not include codes to turn on and off the printer an error message will be displayed, such as:

```
Error. No auxiliary printer control sequences available for terminal.
```

2.1.7 Printer Control Characters

When a print job is sent to a physical printer, a new page is specified by an ASCII 12 character, the form feed control character. A new line is specified by a two-character sequence: an ASCII 13 (carriage return) and an ASCII 10 (line feed). These are the default values. Some printers require other printer control character sequences. For example, a new page may require both a form feed (ASCII 12) and a carriage return (ASCII 13) character. A new line may require just a line feed (ASCII 10) character. If your printer is not formatting correctly, you can use the **SP-CONTROL** command to change the printer control character defaults.

3

Spooler Commands

The list of supported commands have been mainly derived from the Pick/jBase/Reality platforms but can appear to some extent on other platforms. The following is a summary of the commands available from the command line (and also embedded into applications).

Most (but not all) of the commands take the format

```
USER:SP-XXXXXX [ ARG1  [ ARG2 ... ]]
```

and if the arguments are not entered on the command line, they will be prompted for. The following example shows an invocation of **SP-DEVICE** with all the arguments being specified on the command line:

```
USER:SP-DEVICE STANDARD "|PRN|HPLJ80"
```

The following example shows two invocations of **SP-DEVICE** issuing prompts for unspecified arguments:

```
USER:SP-DEVICE

FORM-QUEUE   DEVICE: STANDARD    HPLJ80
USER:SP-DEVICE

FORM-QUEUE   DEVICE: STANDARD

DEVICE: HPLJ80
USER:
```

The available spooler commands and their implementation status follow.

3.1 SETPTR

The **SETPTR** command lists and sets the current printer settings.

```
SETPTR [chan,width,depth,topmargin,botmargin,mode,option[,option]]
```

SETPTR with no arguments lists the current printer settings.

To change one or more printer settings, specify the desired positional argument(s) with the appropriate leading commas.

Positional Argument	Description

Positional Argument	Description
chan	The affected print channel (unit number). The default is 0.
width	Page width in characters. The default is 132. You can also use **TERM** to change *width*.
depth	Page depth in lines (number of lines per page). The default is 66. Setting *depth* to 0 disables pagination. You can also use **TERM** to change *depth*.
topmargin	Size of top of page margin, in lines. The default is 3.
botmargin	Size of bottom of page margin, in lines. The default is 3.
mode	Printing mode: 1 directs output to the spooler where it can later be manipulated (The default). 3 sends the output to the &HOLD& file.

Positional Argument	Description
option	

Positional Argument	Description
	One or more of the following in any order. Multiple *option* values must be separated by commas. AT [FORM QUEUE] – when using print mode 1, sends the output to the specified form queue. AUX – turns on auxiliary printing and hence all print jobs will not go to the spooler or the &HOLD& file, but will instead be sent to the auxiliary printer attached to the user's terminal. BANNER [text] – In print mode 1, it specifies the text to be printed on a one-page "banner" preceding the job output. In print mode 3, it sets the item ID in &HOLD&: BANNER name. The *text* is a quoted string, enclosed with either single quotes or double quotes. This string may include commas and multiple blank spaces. BANNER NEXT [text] – For print mode 3, it designates that the item id of the entry in the &HOLD& file be incremental; that is, each successive print job creates a new entry in the &HOLD& file. The optional [text] specifies the text to be printed on a one-page "banner" preceding the job output. BANNER UNIQUE [text] – For print mode 3, it designates that the item id of the entry in the &HOLD& file be constant; that is, successive print jobs will overwrite the previous entry because they all have the same item id in the &HOLD& file. The optional [text] specifies the text to be printed on a one-page "banner" preceding the job output. BRIEF – specifies that changes to the current printer settings are implemented without confirmation prompts. COPIES [n] – sets the number of times the job will be printed (applicable only for print mode 1). EJECT [[m-]n] – EJECT specifies that the spooler should add one formfeed at the end of each print job. EJECT *n* specifies that the spooler should add *n* formfeeds at the end of each print job. EJECT *m-n* specifies that the spooler should add *n* formfeeds at the end of each print job, and *m* formfeeds at the beginning of each print job. FHEADN – synonym for HEADON. HEADN – synonym for HEADON. HEADON – prints banner output at the start of a print job. The opposite of NOHEAD. HOLD – indicates the job should be held in the queue after printing. INFORM – indicates that the print number of the job should be sent to the user terminal when the job is created. KEEP – the print job is kept open until the user exits the MultiValue Shell, a subsequent SETPTR command is executed for this job, or the job is closed via the **SP-CLOSE** command. NFMT – synonym for NOFMT. NHEAD – synonym for NOHEAD. NOEJECT – specifies that no formfeeds are to be issued at the end of a print job. NOFMT– suppresses all formatting for a print job, including pagination and top and bottom margins. To set this option for all print channels for the account, use **SP-CONDUCT** bit 128.

Positional Argument	Description
	NOHEAD – suppresses banner output at the start of a print job. This is the Caché default. The opposite of HEADON. NOINFORM – indicates that the print number of the job should *not* be sent to the user terminal when the job is created. OPEN – This is the same as the KEEP option. SKIP – the job output should be skipped, that is, not sent to any physical device. UNPROTECT – cancels any security controls for the job (useful only for print mode 1; once set it cannot be unset). DEFAULT (or DFLT) — causes all unspecified positional arguments to revert to the initial systemwide default values, regardless of emulation. NODEFAULT (or NODFLT) — causes all unspecified positional arguments to retain their current values, regardless of emulation.

After specifying these settings, **SETPTR** prompts you to confirm them with a Y or N, unless you specified the BRIEF option. If you specify an invalid *option*, **SETPTR** informs you with a Warning, then sets all of the valid specified arguments.

In Caché MultiValue, unspecified values for *width*, *depth*, *topmargin*, *botmargin*, or *mode* default to the systemwide defaults. This behavior for unspecified values is emulation-dependent. For example, **SETPTR 0,,,,,3** reverts these four settings to the defaults in Caché MultiValue. In other MultiValue emulations, these four settings retain their previously set values. This behavior can be overridden for any emulation by specifying the DEFAULT or NODEFAULT *option* keyword.

If the &HOLD& file does not exist, **SETPTR** with *mode*=3 creates &HOLD& as a directory-type file in the current working directory (identified in the @PATH variable). For example, Mgr/namespace. Normally &HOLD& should be a directory-type file, but you can pre-created it as an anode-type file if that is preferred. To do this, you can use **CREATE-FILE &HOLD& ANODE** to create &HOLD& as a MultiValue global. In this case, a subsequent **SETPTR** with *mode*=3 writes to this existing &HOLD& ANODE global file. You must specify ANODE; by default **CREATE-FILE** creates an INODE file. An INODE file cannot be used by **SETPTR**.

BRIEF means that changes to the current printer settings will be made without displaying the changes and prompting you to confirm them. Thus, `SETPTR ,,64` prompts you for confirmation before it sets the page width to 64 lines; `SETPTR ,,64,,,,BRIEF` sets the page width to 64 lines without prompting for confirmation.

- BANNER name: item ID is always *name*. Each subsequent job overwrites the previous version of *name*.
- BANNER NEXT: item ID is an incremented number, P#0000_nnnn, where *nnnn* is incremented for each entry to &HOLD&, *nnnn* increments from 0001 through 9999.
- BANNER NEXT name: item ID is an incremented number, name_nnnn, where *nnnn* is incremented for each entry to &HOLD&. *nnnn* increments from 0001 through 9999.
- BANNER UNIQUE: item ID is an incremented number, P#0000_nnnn, where *nnnn* is incremented each time the **SETPTR** command is executed. *nnnn* increments from 0001 through 9999.

  In UniData emulation, BANNER UNIQUE is a synonym for BANNER NEXT.

You can establish the **SETPTR** settings for print channel 0 as the systemwide default settings for print channel 0 by issuing the **SETPTR.DEFAULT** command.

3.2 SETPTR.DEFAULT

```
SETPTR.DEFAULT [ (D) ]
```

The **SETPTR.DEFAULT** command takes the current print channel 0 settings and establishes them as the print channel 0 default settings. **SETPTR.DEFAULT** must be run from the SYSPROG account. Before issuing **SETPTR.DEFAULT** you define the print channel 0 settings using **SETPTR**. **SETPTR.DEFAULT** makes these settings the print channel 0 defaults for all future **SETPTR** commands systemwide. **SETPTR.DEFAULT** has no effect on print channels other than print channel 0. These **SETPTR.DEFAULT** settings remain in effect across system reboots until you issue a **SETPTR.DEFAULT (D)** command. The (D) option reverts all settings to the initial printer default settings.

3.3 SP-ASSIGN

```
SP-ASSIGN ?
SP-ASSIGN {formspecs} {options}
```

This command assigns a printer form queue and printer options to a printer channel. It can also be used to clear existing printer channel assignments. **SP-ASSIGN** is one of the commands that does not fit the usual command format.

The ? argument displays the current spooler options.

The *formspecs* argument specifies the assignment (or deassignment) of a form queue. This form queue is created using SP-CREATE. In Caché MultiValue, and several MultiValue emulations, the specified form queue must already exist. In D3, MVBase, R83, POWER95, and Ultimate emulations, **SP-ASSIGN** automatically creates the named form queue if it doesn't exist, and then assigns it. This behavior can be changed using **SP-CONDUCT** bit position 8192.

The *formspecs* are as follows. Where specified, the leading equal sign is mandatory; a space after the equal sign is optional:

- *=formidentifier*

  Specifies a form queue identifier for print channel 0. *formidentifier* can be either a name or a number, as displayed by the **LISTPTR** command. By default, print channel 0 is named STANDARD.

- *nn=formidentifier*

  Specifies a form queue identifier for print channel *nn*. *formidentifier* can be either a name or a number, as displayed by the **LISTPTR** command.

- *nn=*

  Clears the existing form queue identifier from print channel *nn*.

- F*formname*

  Specifies a form queue name for print channel 0. By default, print channel 0 is named STANDARD.

- F*formnum*

  Specifies a form queue number for print channel 0. By default, print channel 0 is assigned form queue 0.

- Q*formname*

  Specifies a form queue name for print channel 0. By default, print channel 0 is named STANDARD.

- *printnum*

An integer that specifies the number of copies to print. The default is 1. When no other *formspecs* item is specified, *printnum* changes the number of copies for print channel 0. When specified with *nn=formname*, *printnum* changes the number of copies for the specified print channel. Print channel assignment and the number of copies must be separated by a blank space.

In D3 and related emulations, you can use **SP-ASSIGN** to create a new form queue with a user-chosen number. For example, `SP-ASSIGN F3` will create a queue numbered 3 and named F3 if queue #3 doesn't already exist. If you do that, you still need to either use **SP-DEVICE** to set the queue's despooler, or use other commands to move jobs from the queue to other queues for printing.

You can specify one or more *options* values in any order (for example, `(AMU`). The following *options* values are supported:

- A – Auxiliary printing
- F – Create form queue
- H – Hold job after being printed
- K – Kill (clear) the form queue assignments for all print channels. Parenthesis required.
- M – Suppress the display of "Entry #" message when hold job created
- O – Keeps the print job open over multiple programs
- Q – Create form queue
- S – Suppress automatic printing when job created
- U – Unprotect the spool job

For example,

```
USER:SP-ASSIGN =MYFORMQUEUE 2 (HS
```

assigns print channel 0 to the form queue MYFORMQUEUE, with 2 copies printed per print job. The print job will be held (H) and printing suppressed (S).

Assigning *options* values deletes any prior *options* values. The M option is initially provided by default. When M is not specified, the form queue informs by default.

To view the options you have assigned, use **SP.LOOK** or **SP.ASSIGN ?**. These two commands display other additional information and display the *copies* and *options* values in different formats.

3.4 SP-AUX

```
SP-AUX JobNumber[-JobNumber] [JobNumber[-JobNumber]] [(S)]…
```

This command takes a number of print jobs and sends them to the auxiliary printer attached to a user terminal. The terminal must have the codes defined to control an auxiliary printer. Assuming these codes exist, the **SP-AUX** command turns on auxiliary printing attached to the terminal, transmits the specified jobs, and then turns off the auxiliary printer.

The jobs to be transmitted can be any number of jobs, a range of jobs such as 25-28, or any combination (97 100 104-109).

The (S) option is the "silent" option. When this is set, notification will be sent to the terminal that auxiliary jobs are being printed.

3.5 SP-CLEAR

```
SP-CLEAR [form-queue]
```

This command clears all the jobs from the specified form queue.

3.6 SP-CLOSE

```
SP-CLOSE {(Rnnn}
```

SP-CLOSE closes print jobs that are currently open due to the KEEP option specified in **SETPTR**. Without a job number, this command will close ALL print jobs that are currently open for the current user. The use of the (Rnnn) option means only print job nnn will be closed.

3.7 SP-CONDUCT

```
SP-CONDUCT [(V)]
SP-CONDUCT ?
SP-CONDUCT nnn [nnn [...]] [(V)]
```

The **SP-CONDUCT** command allows you to control the conduct of the spooler to accurately reflect your own application's needs. **SP-CONDUCT** settings will override default settings established for the current emulation. Users who wish this to happen in every session should add **SP-CONDUCT** to the login command.

SP-CONDUCT has three syntactical forms:

- **SP-CONDUCT** with no arguments (except the optional (V) verbose option) returns the current settings as the integer total of the bit positions. The verbose option lists the component bit positions that make up this integer total.
- **SP-CONDUCT ?** lists all of the available bit integers and their symbolic names.
- **SP-CONDUCT nnn** allows you to set bit positions for various spooler behaviors. You can specify a single *nnn* value or several *nnn* values separated by blank spaces.

  There are six ways to specify spooler behavior settings:

  – `nnn`: an integer that sets the specified bit positions. Any prior bit settings are eliminated. For example, 129 sets bits 1 and 128.

  – `+nnn`: a signed integer that sets a single specified bit position. All other bit positions remain unchanged. For example, +256 sets the 256 bit; it has no effect on other bit settings.

  – `-nnn`: a signed integer that resets (clears) a single specified bit position. All other bit positions remain unchanged. For example, –256 clears the 256 bit, setting it to 0; it has no effect on other bit settings.

  – `+name`: a keyword that sets a single bit position by symbolic name. All other bit positions remain unchanged. For example, +NO_INHERIT sets the 32 bit (see table below); it has no effect on other bit settings.

- `-name`: a keyword that resets (clears) a single bit position by symbolic name. All other bit positions remain unchanged. For example, –NO_INHERIT clears the 32 bit, setting it to 0 (see table below); it has no effect on other bit settings.
- `DEFAULT`: a keyword that resets all bit positions to the default settings for the current emulation.

The following are the available bit positions and their symbolic names:

Bit Position	Symbolic Name	Description
1	CLOSE_CMQL	Any CMQL command that terminates will automatically close any opened printer spooler job (unless SP-OPEN is in effect).
2	CLOSE_INTERNAL	Same action as 1, except it applies to Caché emulation commands such as **WHO** or **WHERE**.
4	CLOSE_PROC	Same action as 1, except it applies to PROCs and PARAGRAPHs.
8	CLOSE_PROGRAM	Same action as 1, except it applies to user-written applications written in MVBasic and CATALOGed.
16	UNIQUE_JOB	All new programs will automatically get their own printer spool job and will close that printer spool job upon termination (unless SP-OPEN is in effect).
32	NO_INHERIT	New programs will NOT inherit the printer flag from their parent program.
64	UNIQUE_CMQL	Each MultiValue query will have a unique print spooler job.
128	PAGINATE	Default Pagination. Spooler print jobs will have pagination set by default. That is, by default a print job issues a form feed every 66 lines and adds a 3 line top margin and a 3 line bottom margin. This is the default for Caché, and for UniVerse and UniData emulations. Pagination is not the default for other emulations; to paginate a print job, they must specify a HEADING or FOOTING.
256	HEADING_FIRST_IMMEDIATE	If set, the first HEADING statement will be executed immediately. Otherwise, HEADING is not executed until the application comes to a new page.

Bit Position	Symbolic Name	Description
512	HEADING_SUBSEQUENT_IMMEDIATE	Same as 256, except second and subsequent HEADING statements are executed immediately.
1024	PAGE_LAST_LINE	A HEADING statement is executed when the application writes to the last line of the old page. Otherwise, HEADING is executed when the application writes to the first line of the new page. This setting is ignored by CMQL.
2048	NO_INITIAL_FF	If this is set and HEADING is the first output of the spooler job, a leading form feed (FF) is not issued. Otherwise, the first HEADING executed issues a leading form feed.
4096	CLOSE_ALL	If set, a PRINTER CLOSE with no argument closes all print channels. If not set, a PRINTER CLOSE with no argument closes only print channel 0.
8192	CREATE_FQ	Automatically create a form queue. If **SP-ASSIGN** is used to assign to a form queue, and the form queue does not exist, this bit determines the behavior. If set, **SP-ASSIGN**automatically creates the form queue. If this bit is not set, an error message is displayed and the **SP-ASSIGN** fails.
16384	FF_INHIBIT	Inhibits the form feed inside a heading when the heading is generated with the MVBasic EXECUTE statement with either a CAPTURING or an OUTPUT clause.
32768	SP_DEFAULT	Assigns default page size. Specifies the behavior if you use **SP-ASSIGN** to assign a printer to print channel 0 and no page size has been assigned for that form queue. If set, the printer page size defaults to the system default. If not set, the printer page size defaults to the most recent setting for print channel 0. See **SP-PAGESIZE**.

Bit Position	Symbolic Name	Description
65536	SHARE_HEADING	

These bit position values are additive. The default setting for each emulation is:

Emulation	Value
Cache	129 (1 + 128)
D3	95535 (1 + 2 + 4 + 8 + 32 + 256 + 1024 + 4096 + 8192 + 16384 + 65536)
IN2	15 (1 + 2 + 4 + 8)
Information	15 (1 + 2 + 4 + 8)
jBase	38704 (16 + 32 + 256 + 512 + 1024 + 4096 + 32768)
MVBase	96047 (1 + 2 + 4 + 8 + 32 + 256 + 512 + 1024 + 4096 + 8192 + 16384 + 65536)
PIOpen	15 (1 + 2 + 4 + 8)
Pick	5903 (1 + 2 + 4 + 8 + 256 + 512 + 1024 + 4096)
Prime	15 (1 + 2 + 4 + 8)
Reality	104224 (32 + 256 + 512 + 1024 + 4096 + 32768 + 65536)
R83	96047 (1 + 2 + 4 + 8 + 32 + 256 + 512 + 1024 + 4096 + 8192 + 16384 + 65536)
POWER95	96047 (1 + 2 + 4 + 8 + 32 + 256 + 512 + 1024 + 4096 + 8192 + 16384 + 65536)
UDPICK	192 (64 + 128)
Ultimate	96047 (1 + 2 + 4 + 8 + 32 + 256 + 512 + 1024 + 4096 + 8192 + 16384 + 65536)
UniData	192 (64 + 128)
Universe	129 (1 + 128)

3.8 SP-CONTROL

```
SP-CONTROL form-queue [ff,nl[,lff]]
```

This command is used to define what control characters are used by the despool process when outputting a print job to a printer. By default, a new page is defined by an ASCII 12 form feed character, and a new line is defined by a two-character sequence: an ASCII 13 (carriage return) and an ASCII 10 (line feed). These defaults are not appropriate for all printers. A new page on some printers requires both a form feed and a carriage return character (ASCII 12 and 13). A new line on some printers requires just a line feed character (ASCII 10).

Define the new page sequence using the *ff* parameter. Define the new line sequence using the *nl* parameter. The available values are: LF = line feed (ASCII 10); FF = form feed (ASCII 12); CR = carriage return (ASCII 13); nnn = a single character, specified by the corresponding ASCII decimal integer value. You can specify multiple characters by using the underscore symbol, for example, `CR_LF`.

To leave the existing value unchanged, omit the parameter, leaving the placeholder comma when necessary. For example, `SP-CONTROL HP7210 ,CR_LF`. To revert a value to the system default, use the value DE. For example, `SP-CONTROL HP7210 DE,DE`.

On UNIX® platforms, **SP-CONTROL** converts a CR+LF sequence into a LF sequence. This is intended to aid developers programming applications that write to both Windows systems and UNIX systems: Windows systems require a CR+LF sequence for a new line and UNIX systems require simply a LF sequence. Therefore, if you use a CR+LF sequence in a **SP-CONTROL** command for a form queue that outputs to a UNIX file, Caché converts this sequence to LF only.

The optional *lff* parameter specifies the form feed sequence for any leading form feeds at the start of a line. If *lff* is defined, the *ff* parameter is ignored, and only form feeds at the start of a line are translated. A form feed found in the middle of a line is not translated. This parameter is used to support binary data in a print job. Once *lff* is set, you can print binary data as follows:

```
PRINT CHAR(255):"BINARY":
PRINT binarydata
PRINT "Hello World"
```

This program translates the form feed in the first line. The *binarydata* value is then output without translation. This is followed by the “Hello World” character string without any intervening new line sequences.

3.9 SP-COPY

Copies one or more spooler jobs. There are two syntactical forms: the first copies a spooler job to another spooler job; the second copies a spooler job to a MultiValue file defined in the VOC.

```
SP-COPY jn1 {jn2 {jn3 ...}} {(DOV)} TO: {jna {jnb {jnc ...}}}
```

With this command format, one or more spooler jobs (jn_) are copied to new spooler jobs on the spooler. Multiple job numbers are separated by spaces. If an alternate job number is specified in the TO: prompt, then it is copied to this print job number. If no alternate job number is specified, the next available job number is automatically used.

```
SP-COPY jn1 {jn2 {jn3 ...}} {(DOV)} TO: ({DICT} fna) {itemidA {itemidB ...}}
```

This form of the command, where the output file name is given by (fn_) or (DICT fn_), the jobs are copied to the MultiValue file. If an alternate item id is specified in the TO: prompt, then the job is copied to this item id. If no alternate item id is specified, the output item id becomes the job number. The MultiValue file can be either a Caché global or a directory as defined by your computer's file system. Note that a Caché global (INODE file) has a maximum size limit of roughly 3.5 million characters; output beyond that limit is truncated and warning message is issued. To copy spooler jobs larger than that, you can use **CREATE-FILE** to create a target file of type ANODE; however, an ANODE file larger than 3.5 million characters encounters the same maximum size limit when being accessed or edited.

The optional DOV letter codes perform the following operations: the (D) option deletes the original job once the copy has completed successfully. The (O) option overwrites any existing target output; without this option, if the target already exists an error is reported. The (V) option reports in Verbose mode, giving additional information about what is copied and what is deleted.

The following example copies spooler job number 4 and 5 to other spooler jobs. Because no target job numbers are supplied, they are automatically allocated:

```
USER:SP-COPY 4 5
TO:
Job 4 copied to job 21
Job 5 copied to job 22
USER:
```

The following example copies spooler jobs numbered 4 and 5 to specified spooler jobs numbered 1001 and 1011. The D option means that once the copy executes successfully, jobs 4 and 5 are deleted:

```
USER:SP-COPY 4 5 (DV
TO:1001 1011
Job 4 copied to job 1001
Job number 4 deleted.
Job 5 copied to job 1011
Job number 5 deleted.
USER:
```

The following example copies spooler jobs numbered 4 and 5 to the Windows directory C:\TMP. Because the program specifies no item ids, the file names default to the job numbers:

```
USER:CREATE-FILE C_TMP DIR C:\TMP
[421] DICT for file 'C_TMP' created. Type = INODE
[429] Default Data Section of 'C_TMP' set to use directory 'C:\TMP'
[437] Added default record '@ID' to 'DICT C_TMP'.
[417] CreateFile Completed.
USER:SP.COPY 4 5 (VO
TO:(C_TMP
Job 4 copied to OS file C:\TMP/4
Job 5 copied to OS file C:\TMP/5
USER:
```

3.10 SP-COPIES

```
SP-COPIES [job [copies] ]
```

This command changes the number of copies to be printed for the specified job.

3.11 SP-CREATE

```
SP-CREATE [form-queue [device-type [device-name] ] ]
```

SP-CREATE creates a form queue and assigns a device to it. (**SP-DEVICE** is used to assign a device to an existing form queue.) All three arguments are optional. If you omit an argument, **SP-CREATE** prompts you for an argument value.

The supported *device-type* values are: CACHE, DEBUG, GROUP, LPTR, NULL, PORT, PROG, TAPE, and UNIX®.

A *device-type* of DEBUG directs output to the user terminal (device 0), with 0.2 of a second delay between lines.

A *device-type* of GROUP creates a form queue group named *form-queue*. This form queue group contains those existing form queues that you specify as a series of *device-name* values (separated by blank spaces). You can specify a *device-name* of "", then later assign existing form queues to the form queue group using **SP-DEVICE**.

A *device-type* of PROG despools the print job to a process-private global, where it can be accessed by the specified MVBasic program or an ObjectScript program. You specify the name of the routine to call in *device-name*. Following the *device-name* routine name, you can specify one or more argument values to pass to this routine. Argument values are specified in the correct order, separated by blank spaces. The PROG process-private global contains the spooled job data, including leading and trailing form feeds, banners, preamble and postamble. This process-private global also includes the following subscripts:

- `^||PPGMVDESPOOL("JOBNO")` The original spooler job number.
- `^||PPGMVDESPOOL("GLOBAL")` The name of the global with the original job.
- `^||PPGMVDESPOOL("FQNAME")` The form queue name that held the job.
- `^||PPGMVDESPOOL("PARAMETERS")` The list of parameters passed to the *device-name* routine.

A *device-name* must exist and be known to the system. When *device-name* is specified as a command argument, it must be specified as a quoted string. If you specify no device values at the prompts and press return, **SP-CREATE** creates an empty form queue.

SP-CREATE does not start the printing on that form queue. **SP-START** is used for this purpose.

3.12 SP-DELETE

```
SP-DELETE [joblist]
```

This command allows you to delete one or more print jobs. The optional *joblist* argument accepts a list of jobs to delete separated by spaces, as shown in the following example: `SP-DELETE 66 68 71`. You can also delete a range of print jobs, as shown in the following example: `SP-DELETE 66-70`. If you omit the *joblist* argument, **SP-DELETE** prompts you for a print job list.

3.13 SP-DEVICE

```
SP-DEVICE [form-queue [device-type [device-name] ] ]
```

This command allows you to:

- Change the device for an existing form queue. **SP-DEVICE** assigns the specified printer device to a spooler form queue. It assigns the form queue a form number (FQ) in the global ^%MV.SPOOL.
- Change the form queue list for an existing form queue group by specifying *device-type* GROUP. **SP-DEVICE** can assign multiple form queues to a form queue group. By default, **SP-DEVICE** overwrites any existing form queues in the form queue group. The `(A` letter code option causes it to append the specified form queues to the form queue group, rather than replacing them. The `(R` letter code option causes it to remove the specified form queues from the form queue group.

(**SP-CREATE** is used to create a form queue and assigns a device to it.)

All three arguments are optional. If you omit an argument, **SP-DEVICE** prompts you for the argument value. The *form-queue* default is STANDARD. The supported *device-type* values are: CACHE, DEBUG, GROUP, LPTR, NULL, PORT, PROG, TAPE, and UNIX®.

A *device-type* of DEBUG directs output to the user terminal (device 0), with 0.2 of a second delay between lines.

A *device-type* of PROG despools the print job to a process-private global, where it can be accessed by the specified MVBasic program or an ObjectScript program. You specify the name of the routine to call in *device-name*. Following the *device-name* routine name, you can specify one or more argument values to pass to this routine. Argument values are specified in the correct order, separated by blank spaces. The PROG process-private global contains the spooled job data, including leading and trailing form feeds, banners, preamble and postamble. This process-private global also includes the following subscripts:

- `^||PPGMVDESPOOL("JOBNO")` The original spooler job number.
- `^||PPGMVDESPOOL("GLOBAL")` The name of the global with the original job.
- `^||PPGMVDESPOOL("FQNAME")` The form queue name that held the job.
- `^||PPGMVDESPOOL("PARAMETERS")` The list of parameters passed to the *device-name* routine.

The *device-name* must exist and be known to the system. The *device-name* of a printer must be prefaced by |PRN|. When *device-name* is specified as a command argument, it must be specified as a quoted string. Any change in device name assignment takes effect when the despool process starts writing a new print job; it will not affect a print job midway through.

The following example shows several invocations of SP-DEVICE:

```
USER:SP-DEVICE STANDARD "|PRN|HPLJ80"
USER:SP-DEVICE CANON CACHE "|PRN|Canon MP530:(/WRITE:/APPEND:/DATATYPE="TEXT"):0"
USER:SP-DEVICE HP20 CACHE "|PRN|\\MACHINE\HP32:wan:0"
USER:SP-DEVICE FILEOUT CACHE E:\DATA\FILES:wa
```

3.14 SP-DISPLAY

```
SP-DISPLAY job1/formqueue [job1/formqueue [...jobn/formqueue]]
```

This command provides detailed information on the specified jobs or form queues. **SP-DISPLAY** is a synonym for SP-VERBOSE.

3.15 SP-EDIT, SP.EDIT

```
SP-EDIT [job[-job]]

SP.EDIT [job[-job]] {L} {MD} {MS}
```

This command allows an administrator to manipulate spooler jobs. The **SP-EDIT** form is similar to the jBase, Pick, and Reality implementations. The **SP.EDIT** form is similar to UniVerse.

SP-EDIT

The **SP-EDIT** command allows you to edit pending print jobs. It is called from the command line as

```
SP-EDIT [JOB[-JOB]]
```

which allows editing of the characteristics of a single job or a range of jobs. Once invoked, the administrator enters a series of commands which are applied to the identified jobs. The available commands are:

Command	Description
nn	Moves the line pointer to line nn and displays that line in the print job.
+nn	Moves the line pointer down nn lines and displays that line in the print job.
–nn	Moves the line pointer up nn lines and displays that line in the print job.
A	Repeat the previous “locate” command.
B	Move line pointer to the bottom of the print job.
CP	Moves the line pointer to the same position as the despooling position. This allows you to easily move to where the last line that was printed.
Dnn	Moves the line pointer down nn lines and displays that line in the print job.
EX	Exit editing this print job.
EXK	The EXK command is similar but exits editing ALL print jobs that were specified.

Command	Description
FD	Delete the print job.
HELP	Display the help screen.
HEX	Toggles the display of the lines of data from normal displayable characters to display of the lines of data as the hexadecimal value of each character.
L	List a single line and advance the line pointer Lnn.
Lnn	List nn lines of the print job and advance the line pointer.
L{nn} {str}	Locate all occurrences of the string “str” in the print job starting at the current line position.
N{nn}	Moves the line pointer down nn lines and displays that line in the print job.
N{nn}P	Moves the line pointer down by nn pages and displays that page in the print job.
P	Display a screen of data.
Pnn	Display a screen of data, but first of all move the line pointer to page nn.
SP	Sets the despool position to whatever the line pointer currently is positioned at. (This is a good way of restarting a print job at a different position.)
T	Move line pointer to top of print job.
U{nn}	Moves the line pointer up nn lines and displays that line in the print job.
U{nn}P	Moves the line pointer up nn pages and displays that page in the print job.
W	Display the previous screen of lines in the print job.

SP.EDIT

The **SP.EDIT** command allows you to perform simple administration on pending print jobs. It is called from the command line as

```
SP.EDIT [JOB[-JOB]] {L} {MD} {MS}
```

which allows editing of the characteristics of a single job or a range of jobs. If no jobs are specified, the commands apply to ALL jobs. The meaning of the command options are:

Option	Description
L	Display the first 512 characters of each selected job in turn.
MS	The selected jobs will be set to spool. This changes the setting for any jobs that were set to SKIP. If there is an active despooller process running, all existing jobs are now candidates to be printed.
MD	The select jobs will be deleted from the spooler.

If none of the options is specified on the command line, the characteristics of each selected job will be displayed in a form similar to

```
------- Details of Print Job # 45 in ^MV.SPOOL("45") -------

Form queue number :              00000
Form queue name :                STANDARD
Job status :                     CLOSED
Time of last status change :     Dec 08 2006 16:22:29
Number of lines in job :         6
Number of pages in job :         1
Time job created :               Dec 08 2006 16:22:29
Time job closed :                Dec 08 2006 16:22:29
Number copies to print :         1
Namespace of job creator :       %SYS
Account name of job creator :    SYSPROG
Username of job creator :       Greg
Port number of job creator :     5700
Despool page position :          0,0
Options :                        HOLD, INFORM, SKIP
```

The administrator is then prompted for a series of commands drawn from the following list:

Option	Description
D	Delete the print job.
E	Edit the print job; begin a session similar to the **SP-EDIT** command for that print job so the administrator can examine and further manipulate the print job.
N	Do not display anything from the print job.
S	Set the status of the print job to be ready to be spooled. That is, remove any SKIP option from the print job. If a despool process is running, this print job becomes eligible for printing.
X	Exit this print job and all subsequent print jobs you might have specified.
Y	Display the first 512 characters of each selected job in turn.

3.16 SP-EJECT

```
SP-EJECT [pages]
```

This command creates a print job that begins with the specified number of blank pages. It spools the specified number of form feeds (page ejects). The optional *pages* argument must be an integer from 0 through 10. The default is 1.

3.17 SP-FORM

```
SP-FORM [old-form-queue [new-form-queue] ]
```

Change the name of a form queue. The form queue number remains the same.

3.18 SP-FQDELETE

```
SP-FQDELETE [form-queue]
```

Deletes a form queue and all the jobs on the queue. If there are any print jobs currently being printed, it will leave that print job as-is and not delete the form queue.

If you use **SP-FQDELETE** to delete a form queue which was a member of a form queue group, the form queue is deleted from the form queue group. Therefore, if you re-create the same form queue, you will need to add it again to the form queue group.

3.19 SP-GLOBAL

```
SP-GLOBAL [global-name] [(S]
```

SP-GLOBAL without an operand displays the name of the spooler table global for the current account. **SP-GLOBAL** with an operand changes the name of the spooler table global for the current account.

By default, the name of the global (and therefore the spooler table) is `^%MV.SPOOL`, which is a system-wide global. Thus by default, all users share the same spooler table. **SP-GLOBAL** changes the name of the global where output will be collected for the current account. This allows each account to maintain a separate global, or for multiple accounts to share a global.

The *global-name* argument must be a syntactically valid Caché global variable name. Caché global variable names begin with the ^ character. **SP-GLOBAL** rejects a global name that fails global naming conventions with an appropriate error message. For further details on naming conventions for globals, refer to the "Variables" chapter of *Using Caché ObjectScript*.

The *global-name* should be either a nonexistent global or an existing MultiValue spooler global. If *global-name* refers to an existing data global or a MultiValue file, **SP-GLOBAL** displays an appropriate error message, then prompts you before proceeding, as shown in the following examples:

Existing global:

```
USER:SP-GLOBAL ^MYGLOBAL
Warning, the global ^MYGLOBAL already contains non-spool data.
If you continue, this data will be lost.
Do you wish to continue (Y/N) ?
```

Existing MultiValue file:

```
USER:CREATE-FILE MYSPOOLER
USER:SP-GLOBAL ^MYSPOOLER
Warning, the global ^MYSPOOLER is already allocated to the MV file MYSPOOLER.
If you continue, this file will be deleted.
Do you wish to continue (Y/N) ?
```

You can use the `(S` letter code option to suppress this prompt and assign the specified global as the spooler table.

The following example sets the Caché global ^SPOOLER in the namespace ADMIN as the spooler table for the current namespace GREG. Because Caché namespace names map to MultiValue account names, this means that all future users of account GREG will use SPOOLER in account ADMIN:

```
GREG:SP-GLOBAL ^|"ADMIN"|SPOOLER
Setting spooler global name to '^|"ADMIN"|SPOOLER'
```

The name of the global is part of the account metadata. It persists once set, so it remains set for all future logins for that user until explicitly changed.

3.20 SP-JOBS

```
SP-JOBS
```

The **SP.JOBS** command shows the status of print jobs queued to the spooler, and prompts you to enter a numeric action code to control a specified print job. It lists jobs in the sequence in which they were created, with the most recently created job shown first. **SP.JOBS** lists the job number, the queue name, the line number, the account name, the date and time created, the status, and the number of pages printed.

Print jobs are assigned sequential integer numbers. Once a day Caché MultiValue resets the assignment sequence to 1, so that print job numbers can be reused. However, numbers already assigned to pending print jobs are skipped over. For example, if the job number sequence is reset when there are pending print jobs numbered 1, 2, and 4, new print jobs will be assigned job numbers 3, 5, and so forth.

The following action codes allow manipulation of print job options and status.

No.	Name	Description
1	MOVE FORM QUEUE	Move all jobs from one form queue to another. Same as the **SP-MOVEQ** command.
2	MOVE PRINT JOB	Move a print job from one form queue to another.
3	CHANGE OPTIONS	Modify the options associated with a print job.
4	CHANGE #COPIES	Alter the number of copies of the specified job printed.
5	DELETE JOB	Remove the jobs listed from the queue.
6	STOP PRINTING	Stop printing on a form queue once the current job has completed and stop the despool process. Same as the **SP-STOP** command.
7	RESUME PRINTING	Resume printing on a form queue that has been suspended with the **SP-SUSPEND** command (or option 9). Same as the **SP-RESUME** command.
8	EDIT PRINT JOB	Add of remove pages from the given print job.
9	SUSPEND PRINT	Suspend printing on a form queue. Same as the **SP-SUSPEND** command.
10	CLOSE JOB	Close the specified print job.
11	SP-STATUS	Changes the menu to that provided by the **SP-STATUS** command.
12	KILL PRINTING	Kill printing immediately (even mid-job) and stop the despool process. Same as the **SP-KILL** command.
14	CLEAR QUEUE	Delete all inactive print jobs on the form queue. Same as the **SP-CLEAR** command.
16	VERBOSE DISPLAY	Given a job number, display the details about the job in verbose mode.
98	FILTER/SORT	Allows for filtering (using the WITH clause) and sorting (using the BY clause).

No.	Name	Description
99	EXIT	Exit the program.

3.21 SP-KILL

```
SP-KILL [form-queue]
```

This command will kill printing of the current job on the form queue and then stop the despool process. The difference between this command and **SP-STOP** is that **SP-STOP** waits for the current job to finish printing, this one does not. You can specify * as the form queue name/number in which case we kill all running print despool processes.

3.22 SP-LOOK

```
SP-LOOK
```

Displays the assigned print channels with their form queue numbers (Q#), form queue names, specifications, number of copies to print (P#), and their assignment options. **SP-ASSIGN** assigns form queues and their options to a print channel. **SETPTR** assigns the Width, Lines, Top, and Bot specifications to a print channel. The **SP-LOOK** option keywords correspond to the **SP-ASSIGN** letter codes as follows:

SP-LOOK	SP-ASSIGN
AUX	A
COPIES *n*	*printnum*
HOLD	H
INFORM	M *not* assigned
KEEP	O
SKIP	S
UNPROTECT	U

3.23 SP-MOVEQ

```
SP-MOVEQ [from-form-queue [to-form-queue] ]
```

This command moves all the jobs from one form queue to another. Any job that is currently being printed is not moved.

3.24 SP-NEWTAB

```
SP-NEWTAB
```

This command completely re-initializes all form queues and adds a single default form queue called STANDARD. All print jobs and form queues are lost. **SP-NEWTAB** kills the current spooler global, as assigned by SP-GLOBAL. By default, the name of the MultiValue spooler global is `^%MV.SPOOL`, which is a system-wide global. Thus by default, all users share the same spooler global.

If you use **SP-GLOBAL** to change the spooler global to another value (for example, the Caché general-purpose ^SPOOL spooler global) then you should be careful using **SP-NEWTAB** to avoid deleting non-MultiValue spool jobs.

SP-NEWTAB is performed independent of transaction status.

3.25 SP-OPEN

```
SP-OPEN
```

This command causes any new print job to remain open until either the user logs off or until the SP-CLOSE command is executed. It has the same effect as using the O option in the SP-ASSIGN command or the OPEN or KEEP option in the SETPTR command.

3.26 SP-OPTS

```
SP-OPTS [job [options] ]
```

This allows you to change the options on a job. The options are those items that can appear in position 7 of the SETPTR command: HOLD, SKIP, BANNER and so on.

3.27 SP-PAGESIZE

```
SP-PAGESIZE form-queue [width [ depth [ topmargin [ bottommargin]]]]
```

The **SP-PAGESIZE** command defines the page size for a MV spooler form queue. These values that can be set are page width , page depth, top margin and bottom margin. When a form queue is assigned using either **SETPTR** or **SP-ASSIGN**, these values will be used as necessary. You can delete these values by executing the **SP-PAGESIZE** with all the values set to 0.

The SETPTR command also allows defining a page width, depth, top and bottom margin. If these values are omitted in SETPTR, and the form queue has values set by **SP-PAGESIZE**, then these are the values will be used.

3.28 SP-POSTAMBLE

```
SP-POSTAMBLE form-queue subroutine
```

Sets the name of the subroutine name to be called after a job finishes printing on the named form queue.

3.29 SP-PREAMBLE

```
SP-PREAMBLE form-queue subroutine
```

Sets the name of the subroutine name to be called before a job begins printing on the named form queue. See Form Queue Control for a discussion of the preamble.

3.30 SP-PURGEQ

```
SP-PURGEQ [form-queue [job-list]]
```

This command removes jobs in the list from the specified form queue. The job-list is made up of a space-separated list of job numbers, or job number ranges, for example, “6–11”. Use “*” to remove all jobs from the queue.

3.31 SP-RESUME

```
SP-RESUME [form-queue]
```

If a despool process has been suspended with the **SP-SUSPEND** command, then this will resume the printing.

3.32 SP-SHOW

```
SP-SHOW job1/formqueue [job1/formqueue [...jobn/formqueue]]
```

This command provides detailed information on the specified jobs or form queues. **SP-SHOW** is a synonym for SP-VERBOSE.

3.33 SP-SKIP

```
SP-SKIP [form-queue [[lead-]trail] ]
```

Defines how many pages to skip after and (optionally) before printing a job. You can specify an integer number of *lead* leading form feeds before a print job and *trail* trailing form feeds after a print job. For example, `SP-SKIP myqueue 1-3`. If you specify a single integer, it defines the number of trailing form feeds, with a default of zero leading form feeds.

3.34 SP-START

```
SP-START [printer-name | *] [(B | F | I)] [(L)]
```

The **SP-START** command starts a despool process. This despool process monitors the spooler form queue and sends jobs from the spooler to the printer.

If you specify no argument, you are prompted to specify a form queue name. An * argument defaults to the STANDARD form queue. If you supply *, Caché will try to start all the defined printers.

By default, **SP-START** runs despool as a background process. To run despool as an interactive process, specify (I). To run despool as a background process, specify (B) or specify no letter option. To start despool as a foreground process which occupies a terminal, specify (F). The (F) option is useful for debugging, as described in "Debugging the Despool Program".

The (L) option logs some of the activities of the despool process to the MultiValue log file, located at Cache/mgr/mv.log. You also can use the **%SYS.System.WriteToMVLog()** method to write to the mv.log file. Refer to the TRAP-EXCEPTIONS command for further details on mv.log.

To start printer spoolers as part of Caché start-up, do the following:

1. Write a paragraph in the voc of someaccount that starts the printers:

   ```
   0001 PA
   0002 SP-START PRINTER1 CACHE |PRN|PRINTER1
   0003 SP-START PRINTER2 CACHE |PRN|PRINTER2
   ```

2. Have Caché %ZSTART schedule a background job to run this using the MV command:

   ```
   ZN "someaccount" MV "PHANTOM START.PRINTERS"
   ```

   %ZSTART is an ObjectScript routine you create and save (%ZSTART.mac) in the %SYS namespace. For further details, refer to "Customizing Start and Stop Behavior with ^%ZSTART and ^%ZSTOP Routines".

3.35 SP-STATUS

```
SP-STATUS
```

The **SP-STATUS** command shows the status of the currently defined spooler form queues. It then prompts you to enter one of the following numeric action codes to control a queue or printer assignment.

No.	Name	Description
1	CREATE FORMQUEUE	Create a new form queue. Same as the **SP-CREATE** command.
2	CHANGE QUEUENAME	Change the name of the form queue. Same as the **SP-FORM** command.
3	PAGE SIZE	
4	CHANGE DEVICE	Change the device associated with an existing form queue. Same as the **SP-DEVICE** command.
5	CHANGE PAGE SKIP	Define how many pages to skip at the start of despooling a print job. Same as the **SP-SKIP** command.
6	LIST PRINT JOBS	Changes the menu to that provided by the **SP-JOBS** command.
7	DELETE FORM QUEUE	Delete a form queue and all the print jobs that were in that queue. Same as the **SP-FQDELETE** command.
8	VERBOSE DISPLAY	Displays detailed information about the specified form queues.
9	DESPOOL CONTROL	Changes the new page and/or new line control character sequences. Same as the **SP-CONTROL** command.
10	PURGE	Removes jobs from the form queue.
98	FILTER/SORT	Allows for filtering (using the WITH clause) and sorting (using the BY clause) of information.
99	EXIT	Exit the program.

3.36 SP-STOP

```
SP-STOP [printer-name | * ]
```

The **SP-STOP** command stops printing on the printer at the end of the current job and then stop the despool process. Stopping a despool process causes jobs to wait on the spooler form queue and not be sent from the spooler to the printer.

If you specify no argument, you are prompted to specify a form queue name. An * argument defaults to the STANDARD form queue. Specifying * will stop all executing print spool jobs.

3.37 SP-SUSPEND

```
SP-SUSPEND [form-queue]
```

This command will suspend printing of a job. The despool process simply loops waiting for a **SP-STOP**, **SP-RESUME** or **SP-KILL** command to be issued.

This command is useful so the operator can, for example, adjust the printer on a long print job. If the operator notices a problem (ink running out, paper becoming skewed) the operator can suspend the printing, correct the problem, reposition the print job to where the problems first began and then use **SP-RESUME** to continue printing.

Repositioning the print job follows these steps:

- Suspend the despool process, if necessary, with the **SP-SUSPEND** command.
- Edit the print job with the SP-EDIT command (note: For legacy reasons the **SP-EDIT** and **SP.EDIT** commands differ, always use **SP-EDIT** for this task).
- Once in the edit, position the editor to the line you want to reposition at.
- In the editor execute the SP command.
- Restart the printer with the **SP-RESUME** command.

For example, the following transcript shows repositioning print job 2 , currently printing on the STANDARD form queue, to resume at line 402:

```
USER:SP-SUSPEND STANDARD
SUSPEND command initiated on form queue STANDARD running on job 2508
USER:SP-EDIT 2
PRINT JOB # 2
WARNING: Job status is PRINTING
TOI
.401
000401 XXXX YYYY ZZZZZZ aaaaa bbbbb cccc
.SP
Print position set to 402,0
.EX
USER:SP-RESUME STANDARD
RESUME command initiated on form queue STANDARD running on job 2508
USER:
```

3.38 SP-SWITCH

```
SP-SWITCH [new-form-queue [job] ...
```

The **SP.SWITCH** command allows the administrator to switch one or more print jobs to the specified spooler form queue.

If you specify no arguments, you are prompted to specify a form queue name and one or more print jobs.

3.39 SP-TESTPAGE

```
SP-TESTPAGE [device | form-queue]
```

This command generates a standard test page and sends it to the designated destination.

- `SP-TESTPAGE` sends the test page to the standard print device, `"|PRN|:(/WRITE:/APPEND:/DATATYPE="TEXT"):0"`.
- `SP-TESTPAGE device` sends the test page to the specified Caché print device. This can be in the form, "devicename", or optionally with an added open mode and timeout, for example, `"|PRN|:rwn:2"`.
- `SP-TESTPAGE form-queue` sends the test page to the print device associated with the given form-queue. If the form queue is of type DEBUG, or the device name is NULL or DEBUG, then Caché will generate an error message.

The standard test page consists of the block letters “Cache Test Page”, followed by the Caché version string, the date and time that the test page was written, the output device name (for example |PRN|), and the Cache I/O name (for example |TRM|:|5740).

SP-TESTPAGE performs this test page operation using the following steps:

1. Creates a temporary form queue.
2. Adds a test page as a job to that form queue.
3. Initiates a despool process using **SP-START**.
4. Waits for the job to finish printing (or for timeout).
5. Stops the despool process using **SP-STOP**.
6. Deletes the temporary form queue.

Accordingly, the output of a **SP-TESTPAGE** command looks something like the following:

```
USER:SP-TESTPAGE File
Form queue STANDARD created as form number FQ00000000 in global ^%MV.SPOOL
Temporary form queue 'SPTESTPAGE_4616' created
Test print job number 1 created, starting despool process
Spooler STARTED in BATCH mode on form queue SPTESTPAGE_4616 at job 10012
The test page appears to have printed successfully.
STOP command initiated on form queue SPTESTPAGE_4616 running on job 10012
Temporary form queue 'SPTESTPAGE_4616' deleted
Full log written to file c:\intersystems\cache\mgr\mv.log
USER:
```

3.40 SP-VERBOSE

```
SP-VERBOSE {job|formqueue} [{job2|formqueue2} [...]]
```

This command provides detailed information on the specified print jobs or form queues. It allows you to supply one or more print job numbers or form queue identifiers (form queue names or form queue numbers as displayed by the **LISTPTR** command) for verbose display. If specifying multiple print jobs or form queues, separate them with blank spaces.

If you specify no arguments, you are prompted to specify a print job number or form queue name (or number).

The following is a print job display:

```
USER:SP-VERBOSE 5

------- Details of Print Job # 5 in ^%MV.SPOOL("5") -------

Form queue number : FQ00000000
Form queue name : STANDARD
Job status : CLOSED
Time of last status change : Mar 08 2011 13:53:41
Number of lines in job : 10
Number of pages in job : 1
Time job created : Mar 08 2011 13:53:41
Time job closed : Mar 08 2011 13:53:41
Number copies to print : 1
Namespace of job creator : USER
MV Account name of job creator : USER
OS Username of job creator : glenn
Cache Username of job creator : UnknownUser
Port number of job creator : 20
Despool page position : 0,0
Unique Job Identifier : 12
System name : DELLHOME
IP Address : 127.0.0.1
Cache Instance : DELLHOME:CACHE
Options :
```

The following is a form queue display:

```
USER:SP-VERBOSE STANDARD

------- Details of form queue STANDARD in ^%MV.SPOOL("FQ00000000") -------

Form queue number :              FQ00000000
Form queue name :                STANDARD
Time formq created :             Mar 08 2011 10:36:26
Status of formq :                INACTIVE
Namespace of formq creator :     USER
Account name of formq creator :  USER
Username of formq creator :     glenn
Number of jobs on formq :        0
Job number being despooled :     --none--
Job ID of despool process :      --none--
Device name for despool :        |PRN|DOC2
Leading and trailing page skips :1
Notional device type :           CACHE
Despool job pre-amble routine :  --undefined--
Despool job post-amble routine : --undefined--
Page Size :                      --undefined--
Despool control chars :          FF,CR_LF
USER:
```

4

Spooler Command Examples

4.1 A Simple Example

This example shows printing 3 lines of output on a Windows printer. The Windows name of the printer was HP7210.

Assign a printer device name to the default STANDARD form queue.

```
SOMEACCT:SP-DEVICE STANDARD CACHE "|PRN|HP7210"
Form queue STANDARD created as form number FQ00000000 in global ^MV.SPOOL
SOMEACCT:
```

This has the effect of creating a new spooler table in the Caché global called ^MV.SPOOL.

Now write to the spooler. The program TEST12 writes 3 lines to the default print spool form queue.

```
SOMEACCT:ED GBP TEST12
TEST12
4 lines long.
----:p
0001: PRINTER ON
0002: FOR I = 1 TO 3
0003:     PRINT "Line ":I:" of 3"
0004: NEXT I
Bottom at line 4.
----:FIBC
"TEST12" filed in file "GBP".
TEST12
[B0] Compilation completed.
[241] 'TEST12' Cataloged.
SOMEACCT:TEST12
```

At this point, an SP-JOBS will show the print job is now CLOSED and waiting to be printed.

```
SOMEACCT:SP-JOBS
Dec 20 06 13:40:00  ^MV.SPOOL           PRINT JOBS                PAGE 1 OF 1

JOB   QUEUE        LINE    ACCOUNT  CREATED      STATUS   OPT PRINTED
1     STANDARD     1088    SYSPROG  20 Dec 13:39 CLOSED       0,0 OF 67,1

1. MOVE FORM QUEUE 6.  STOP PRINTING    11. SP-STATUS      16. VERBOSE DISPLAY
2. MOVE PRINT JOB  7.  RESUME PRINTING  12. KILL PRINTING
3. CHANGE OPTIONS  8.  EDIT PRINT JOB
4. CHANGE #COPIES  9.  SUSPEND PRINT    14. CLEAR QUEUE
5. DELETE JOB      10. CLOSE JOB                           99. EXIT

Enter action code / Page number (P#)
```

Start a despool process. These jobs run in the background and monitor the spooler tables. When they see a completed and closed print job, they write it to the printer.

```
SOMEACCT:SP-START *
Spooler STARTED on form queue STANDARD at job 3076
```

If nothing appears on the printer:

- Execute the SP-JOBS menu command and see if the job still exists on the form queue.
- If the job exists, the SP-STATUS command displays the status of the STANDARD form queue. If the status displayed is I/O Error, an incorrect device name may be assigned to the form queue (see the SP-DEVICE command).
- If the job has been removed and no longer exists, the despool process was able to send the job to the physical device. In this case, you may need assistance from your system administrator to determine what happened to the output.

Stop the despool process. This stage not really necessary, but shown for completeness. With SP-STOP, it will stop the background despool process and so further print jobs will simply wait on the form queue for further action.

```
SOMEACCT:SP-STOP *
STOP command initiated on form queue STANDARD running on job 3076
SOMEACCT:
```

4.2 A SETPR Example

The following SETPTR command sets channel 3 to be a printer with a page 132 characters wide and 64 lines deep. Two lines are left blank at the top and bottom for margins. The print output is to be directed to the spooler (printing mode 1). Three copies of the output should be printed with a banner page of “FINAL” and an additional page ejected at the end of each copy.

```
SETPTR 3,132,64,2,2,1,AT HP7000, COPIES 3,EJECT,BANNER FINAL
```

4.3 An SP-JOBS Example

The SP-JOBS command displays a menu to display the print jobs and allows lots of manipulation of the jobs. The purpose of this section is just to talk about the options that are available. This is a sample **SP-JOBS** output and menu:

```
Dec 05 2006 13:56:55  ^MV.SPOOL          PRINT JOBS               PAGE 1 OF 3

JOB   QUEUE        LINE    ACCOUNT  CREATED      STATUS   OPT PRINTED
16    DEBUG        5700    SYSPROG  05 Dec 10:58 CLOSED   H   6,3 OF 6,3
17    DEBUG        5700    SYSPROG  05 Dec 10:59 CLOSED   H   6,3 OF 6,3
18    DEBUG        5700    SYSPROG  05 Dec 11:14 CLOSED   H   6,3 OF 6,3
19    DEBUG        5700    SYSPROG  05 Dec 11:14 CLOSED   H   6,3 OF 6,3
20    STANDARD     5700    SYSPROG  05 Dec 11:15 CLOSED       0,0 OF 9,1
21    DEBUG        5700    SYSPROG  05 Dec 11:15 CLOSED   H   6,3 OF 6,3
22    DEBUG        5700    SYSPROG  05 Dec 11:16 CLOSED   H   6,3 OF 6,3
23    DEBUG        5700    SYSPROG  05 Dec 11:16 CLOSED   H   6,3 OF 6,3
24    DEBUG        5700    SYSPROG  05 Dec 11:22 CLOSED   H   6,3 OF 6,3
25    DEBUG        5700    SYSPROG  05 Dec 11:22 CLOSED   H   6,3 OF 6,3
26    DEBUG        5700    SYSPROG  05 Dec 11:22 CLOSED   H   6,3 OF 6,3

1. MOVE FORM QUEUE 6.  STOP PRINTING    11. SP-STATUS      16. VERBOSE DISPLAY
2. MOVE PRINT JOB  7.  RESUME PRINTING  12. KILL PRINTING
3. CHANGE OPTIONS  8.  EDIT PRINT JOB
4. CHANGE #COPIES  9.  SUSPEND PRINT    14. CLEAR QUEUE
5. DELETE JOB      10. CLOSE JOB                          99. EXIT

Enter action code / Page number (P#)
```

The column marked OPT shows up to 3 characters, H, S, and U. If H is shown, the HOLD option is used (it won’t be deleted after printing); the S option indicates skip printing; the U option states that the output is unprotected, hence removing security.

Regarding the options:

- Option 2 is the same as SP-SWITCH

- Option3 is the same as SP-OPTS
- Option 4, the same as SP-COPIES
- Option 8, the same as SP-EDIT
- Option 10 is the same as SP-CLOSE
- Option 16, the same as SP-VERBOSE

4.4 An SP-STATUS Example

The **SP-STATUS** command shows the jobs in the queue and allows a single option:

```
Dec 14 2006 11:55:17  ^MV.SPOOL          PRINT JOBS               PAGE 1 OF 1

QUEUE NAME     DEVICE                STATUS           #Q   SKIP
STANDARD                             NO DEVICE        4
DEBUG          DEBUG:0.1             INACTIVE         6    2
HP7200         HP7200                INACTIVE         5    1

1. CREATE FORMQUEUE  4. CHANGE DEVICE      7. DELETE FORM QUEUE

2. CHANGE QUEUENAME  5. CHANGE PAGE SKIP   8. VERBOSE DISPLAY

                     6. LIST PRINT JOBS                            99. EXIT

Enter action code / Page number (P#)
```

4.5 An SP-AUX Example

To enable auxiliary printing, there are three mechanisms you can use:

1. Use the A option to the SP-ASSIGN command, for example:

   ```
   USER:SP-ASSIGN 3=FORMNAME HSA
   ```

2. Use the AUX option to SETPTR:

   ```
   USER:SETPTR 3,132,66,0,0,1,banner,hold,aux
   ```

3. Use the SP-AUX command to print an existing print job:

   ```
   USER:SP-AUX 3-4 8
   ```

In cases 1) and 2), when a print job is created it will be built up on the spooler as usual. When the print job is closed for whatever reason, Caché will attempt to print the job on the printer attached to the user's terminal (the auxiliary printer). Once the job has been printed, it will behave as though it were a normal print job, That is, it will become ready for printing by any despool process (unless the SKIPJOB option was set), and may also be deleted.

In case 3), it allows you to print any job, or list of jobs, or range of list of jobs, to the printer attached to your own terminal. The use of the SP.AUX command does not affect the job in any way.

For example, the command

```
USER:SP-ASSIGN 4=FORMNAME HSA
```

sets print channel 4 to form queue FORMNAME, and sets the HOLD, JOBSKIP and AUX options. The AUX option means any print jobs are sent to the user's attached printer through the terminal. Once printed, it becomes a regular print job. The

HOLD and JOBSKIP options mean it will not be despooled by any despool process but will be retained on the spooler form queue.

The SETPTR command is enhanced to support the AUX option as in

```
USER:SETPTR 4,132,66,0,0,1,hold,skip,aux
```

and is similar to the SP-ASSIGN example and sets the same options.

The SP-LOOK command is enhanced to show the AUX option, for example:

```
USER:SP-LOOK
 FORM QUEUES
 Chan Q#   Q name          Width Lines Top Bot P#  Options

0    0    STANDARD        132   66    0   0   1    AUX, HOLD, INFORM, SKIP
```

The SP-VERBOSE command is similarly enhanced to show the AUX option:

```
USER:SP-VERBOSE 6

    ------- Details of Print Job # 6 in ^MV.SPOOL("6") -------

    Form queue number :              FQ00000000
    Form queue name :                STANDARD
    Job status :                     CLOSED
    Time of last status change :     May 25 2007 14:41:03
    Number of lines in job :         1
    Number of pages in job :         1
    Time job created :               May 25 2007 14:41:03
    Time job closed :                May 25 2007 14:41:03
    Number copies to print :         1
    Namespace of job creator :       USER
    Account name of job creator :    USER
    Username of job creator :       UnknownUser
    Port number of job creator :     22
    Despool page position :          0,0
    Options :                        AUX, HOLD, INFORM, SKIP
```

And so is **SP-JOBS**:

```
USER:SP-JOBS
    May 25 2007 14:42:26  ^MV.SPOOL         PRINT JOBS         PAGE 1 OF 1

    JOB   QUEUE        LINE    ACCOUNT  CREATED      STATUS   OPT PRINTED
    6     STANDARD     22      USER     25 May 14:41 CLOSED   AHS 0,0 OF 1,1
```

In all cases, the terminal definition for the current terminal in use needs to support auxiliary printing, otherwise an error message is displayed.

The command strings to support this are the mc5 string (turns ON the auxiliary printer) and mc4 (turns OFF the auxiliary printer).

When a print job is created with the AUX option in effect, the print job will be tagged with the AUX flag, as shown in the SP-VERBOSE and SP-JOBS commands detailed earlier. While this flag is set, the despool process will not despool the job. Once a print job closes, and the print job has been printed to the users terminal, the AUX flag is reset and the despool process can continue.

If there is confusion why a print job won't print, it may be that it was created with the AUX flag but an error occurred before the job could be fully printed to the users auxiliary printer, hence the AUX flag is still set on the job. In this case, you can use the SP-OPTS command (or option 3 from the **SP-JOBS** menu) to reset the AUX option.

4.6 Examples Specifying An Output Device

As noted in the SP-CREATE and SP-DEVICE commands, there are three arguments:

- FORM-QUEUE
- DEVICE-TYPE
- DEVICE-NAME

The FORM-QUEUE represents a logical collection point in the spooler for all output of a similar format.

The DEVICE-TYPE argument is present for compatibility and is not used at this time. However, it must be one of the following values: CACHE, DEBUG, LPTR, NULL, PORT, PROG, TAPE, UNIX®.

The DEVICE-NAME is the name of the physical device. Its form is operating system dependent. Any value acceptable to the ObjectScript OPEN command can be used. For example, on Windows systems, the device name looks like:

```
|PRN|printername
```

and on UNIX® systems, like:

```
/usr/bin/lp -dprintername:QW:3:
```

Note: For UNIX®, a full path name is required for the device or program. For Windows printers, the sequence “PRN” must always appear in uppercase.

For Windows, using a shared printer across a network has additional requirements. These are described in Appendix A of this manual: Setting Up a Windows Shared Printer.

4.6.1 Writing to a Null Printer

At the time of installation, or after using the SP-NEWTAB command, Caché always creates a queue named STANDARD as queue #0. Since STANDARD is the default queue when a user enters the MV shell, this queue may end up with a lot of unwanted output. If no one starts a despool process on it, unwanted jobs fill up the spooler queue.

To deal with this issue, an instance can create a despooler process for a queue to send the output to the “null” device. On Windows, issue the command:

```
SP-DEVICE STANDARD CACHE NULL
```

On Linux systems, the following command is equivalent:

```
SP-DEVICE STANDARD CACHE /dev/null
```

As an alternative, the administrator may create a new queue, called “VOID” in this example, for this purpose:

```
SP-CREATE VOID CACHE NULL
```

Output that is destined to be discarded can be directed to this queue.

4.6.2 Writing to a UNIX®/Linux Printer

This command writes to the UNIX® device name /dev/usb/lp0 and uses the WN parameters to open the device in write mode and allow for the device to be new.

```
USER:SP-CREATE QNAME CACHE /dev/usb/lp0:WN
```

4.6.3 Writing to a Program on a UNIX®/Linux System

This command writes to the UNIX® spooler, the **lp** verb, specifies a destination queue on the UNIX® spooler, The QW parameters state to open a queue and send input to another process, and gives a 3 second timeout before it fails.

```
USER:SP-DEVICE QNAME CACHE /usr/bin/lp -dlaserjet:QW:3
```

Note that in UNIX® a full path name is required for the device or program.

4.6.4 Writing To A Printer On A Windows System

This command writes to the default printer:

```
USER:SP-DEVICE STANDARD CACHE "|PRN|"
```

while this one writes to a named printer:

```
USER:SP-DEVICE STANDARD CACHE "|PRN|HPLJ7210"
```

Depending upon the printer type and printer driver, you might find that the output goes to the Windows spooler but it is then lost by Windows. In this case, you need to open the Windows device in TEXT mode, as shown in the following example:

```
USER:SP-DEVICE CANON CACHE "|PRN|Canon MP530:(/WRITE:/APPEND:/DATATYPE="TEXT"):0"
```

4.6.5 Writing to a File on a UNIX® System

This command directs output to a file:

```
USER:SP-DEVICE FQNAME CACHE /tmp/fileout:WN
```

4.6.6 Debugging the Despool Program

The despool program is normally started with SP-START which starts a background job. This causes terminal output to be lost. An administrator can issue **SP-START** with the (F) option to start a despool process from the terminal prompt as a foreground job to aid in debugging problems.

In this mode, should an error occur, it will not trap automatically and clear held locks. Despool locks need to be cleared manually for this process. Use the Management Portal to look for locks for this process.

As a further debug tool, the device can be called DEBUG. In the following example, the administrator creates a form queue called FQDEBUG with a device specification of DEBUG:0.2, assigns themselves to the new print queue and does a simple LIST, hence creating a job to output. Then, using **SP-START** with the (F) option to start the despool process in foreground mode. This will then display to the screen the print jobs for the FQDEBUG form queues with a delay of 0.2 seconds between each line.

```
USER:SP-CREATE FQDEBUG CACHE DEBUG:0.2
USER:SP-ASSIGN =FQDEBUG
USER:LIST VOC (P)
USER:SP-START FQDEBUG (F)
```

A

Setting Up a Windows Shared Printer

This section contains some hints for avoiding common pitfalls associated with setting up shared network printers on Windows for use by MultiValue applications. The suggestions here are not foolproof. Often getting a shared printer set up is an interactive process of repeated trials and tests.

A.1 Setup

Caché needs to have access to the network resource for jobs other than Terminal (|TRM| device) processes to be able to access the resource.

Use The Full Network Name

The most reliable and consistent way to refer to network printers is to use the full network name. Thus, “\\THATHOST\SALESHP7200” is preferred over just “SALESHP7200”.

For example, to create a form queue called MYFQ which is associated with a Windows shared printer named HP7210 on a remote system whose network name is DELLHOME, and then print to it, issue the following:

```
USER:SP-CREATE MYFQ CACHE "|PRN|\\DELLHOME\HP7210"
USER:SP-START MYFQ
USER:SP-ASSIGN =MYFQ
USER:LIST VOC BASICLPTR
```

Use NET Commands

The following makes LPT1 the default windows printer. This can then be referenced by the Caché device name, `"|PRN|"`. The net command is run from a windows shell, and then the **SP-CREATE** from a MultiValue shell.

From a DOS command shell:

```
NET USE LPT1: \\DELLHOME\HP7210
```

and from the MultiValue shell:

```
SP-CREATE MYFQ STANDARD CACHE "|PRN|"
```

A.2 Testing

There are two mechanisms to help establish a good printer on the MultiValue spooler.

Use SP-TESTPAGE

The command SP-TESTPAGE outputs a test page to a Cache device. You can specify either the Cache device name, or the form queue name (and we extract the device name). For example:

```
USER:SP-TESTPAGE "|PRN|\\DELLHOME\HP7210"
Opening device name \\DELLHOME\HP7210, open mode rw, timeout 0
Device \\DELLHOME\HP7210 opened successfully!
Test writing to device \\DELLHOME\HP7210 reported no error.
```

One can use alternate names to **SP-TESTPAGE** varying the form-queue name and target device. For example, using `SP-TESTPAGE 0` selects Cache device 0, the current terminal, so the output is written to the terminal. The command `SP-TESTPAGE "|PRN|"` directs output to the Caché default printer.

Run the DESPOOL process manually

Normally, an administrator will start a despool process in the background using the **SP-START** command. However, for debugging purposes, a developer can start it from the COS command shell and watch for errors to the screen. For example,

```
USER>DO MVDespool^%SYS.MVSP2("MYFQ",1)
MultiValue Despool opening device '\\DELLHOME\HP7210' , mode 'rw' , timeout ''
```