



# The Caché Multi Value Spooler

Version 2008.2  
10 October 2008

*The Caché MultiValue Spooler*  
Caché Version 2008.2 10 October 2008  
Copyright © 2008 InterSystems Corporation  
All rights reserved.

This book was assembled and formatted in Adobe Page Description Format (PDF) using tools and information from the following sources: Sun Microsystems, RenderX, Inc., Adobe Systems, and the World Wide Web Consortium at [www.w3c.org](http://www.w3c.org). The primary document development tools were special-purpose XML-processing applications built by InterSystems using Caché and Java.



Caché WEBLINK, Distributed Cache Protocol, M/SQL, N/NET, and M/PACT are registered trademarks of InterSystems Corporation.



InterSystems TrakCare, InterSystems Jalapeño Technology, Enterprise Cache Protocol, ECP, and InterSystems Zen are trademarks of InterSystems Corporation.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

The copying, use and disposition of this document and the software programs described herein is prohibited except to the limited extent set forth in the standard software license agreement(s) of InterSystems Corporation covering such programs and related documentation. InterSystems Corporation makes no representations and warranties concerning such software programs other than those set forth in such standard software license agreement(s). In addition, the liability of InterSystems Corporation for any losses or damages relating to or arising out of the use of such software programs is limited in the manner set forth in such standard software license agreement(s).

THE FOREGOING IS A GENERAL SUMMARY OF THE RESTRICTIONS AND LIMITATIONS IMPOSED BY INTERSYSTEMS CORPORATION ON THE USE OF, AND LIABILITY ARISING FROM, ITS COMPUTER SOFTWARE. FOR COMPLETE INFORMATION REFERENCE SHOULD BE MADE TO THE STANDARD SOFTWARE LICENSE AGREEMENT(S) OF INTERSYSTEMS CORPORATION, COPIES OF WHICH WILL BE MADE AVAILABLE UPON REQUEST.

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

For Support questions about any InterSystems products, contact:

**InterSystems Worldwide Customer Support**

Tel: +1 617 621-0700  
Fax: +1 617 374-9391  
Email: [support@InterSystems.com](mailto:support@InterSystems.com)

# Table of Contents

<b>1 Introduction .....</b>	<b>1</b>
1.1 About The Spooler .....	1
1.2 Spool Jobs .....	2
<b>2 Spooler Administration .....</b>	<b>5</b>
2.1 Spooler Administration .....	5
2.1.1 Form Queue Names And Numbers .....	5
2.1.2 MultiValue De-Spool Devices .....	6
2.1.3 The &HOLD& File .....	7
2.1.4 Form Queue Control .....	7
2.1.5 Auxiliary Printing .....	8
<b>3 Spooler Commands .....</b>	<b>11</b>
3.1 SETPTR .....	12
3.2 SP-ASSIGN .....	16
3.3 SP-AUX .....	17
3.4 SP-CLEAR .....	17
3.5 SP-CLOSE .....	18
3.6 SP-CONDUCT .....	18
3.7 SP-COPY .....	19
3.8 SP-COPIES .....	20
3.9 SP-CREATE .....	20
3.10 SP-DELETE .....	20
3.11 SP-DEVICE .....	21
3.12 SP-EDIT, SPEDIT .....	21
3.13 SP-EJECT .....	24
3.14 SP-FORM .....	24
3.15 SP-FQDELETE .....	24
3.16 SP-GLOBAL .....	25
3.17 SP-JOBS .....	25
3.18 SP-KILL .....	27
3.19 SP-LOOK .....	27
3.20 SP-MOVEQ .....	27
3.21 SP-NEWTAB .....	27
3.22 SP-OPEN .....	28
3.23 SP-OPTS .....	28
3.24 SP-PAGESIZE .....	28
3.25 SP-POSTAMBLE .....	28

3.26 SP-PREAMBLE .....	29
3.27 SP-PURGEQ .....	29
3.28 SP-RESUME .....	29
3.29 SP-SKIP .....	29
3.30 SP-START .....	30
3.31 SP-STATUS .....	30
3.32 SP-STOP .....	31
3.33 SP-SUSPEND .....	32
3.34 SP-SWITCH .....	33
3.35 SP-TESTPAGE .....	33
3.36 SP-VERBOSE .....	34
<b>4 Spooler Command Examples .....</b>	<b>35</b>
4.1 A Simple Example .....	35
4.2 A SETPR Example .....	36
4.3 An SP-JOBS Example .....	37
4.4 An SP-STATUS Example .....	37
4.5 An SP-AUX Example .....	38
4.6 Examples Specifying An Output Device .....	40
4.6.1 Writing To A Null Printer .....	40
4.6.2 Writing To A UNIX/Linux Printer .....	41
4.6.3 Writing To A Program On A UNIX/Linux System .....	41
4.6.4 Writing To A Printer On A Windows System .....	41
4.6.5 Writing To A File On A UNIX System .....	41
4.6.6 Debugging The Despool Program .....	41
<b>Appendix A: Setting Up A Windows Shared Printer .....</b>	<b>43</b>
A.1 Setup .....	43
A.2 Testing .....	44

# 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, 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:

- Using the (P) option on the command line;
- By executing "PRINTER ON" or "PRINT ON nnn" statement in MultiValue Basic;
- By using the LPTR token in a CMQL statement: "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 specifying mode 3, and will create &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 will 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 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 these notes on the spooler 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.

Universe-like MultiValue platforms tends to use form queue numbers and 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. There also exists a form queue name that corresponds to the form queue number.

Because some MultiValue platforms use numeric form queues, and others used named form queues, Caché uses the following rules to resolve the ambiguity:

- A form queue reference in the format *nn*, for example, 99, describes form queue number 99.
- A form queue reference in the format *Fnn*, for example, F99, describes form queue number 99.

- A form queue reference in the format `FQnn` or `FNnn`, for example `FN99` or `FQ99`, describes form queue number 99
- A form queue reference in the format `FQname` or `FNname`, 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 also 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 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](#).

## 2.1.3 The &HOLD& File

You implicitly create the &HOLD& file the first time you use the [SETPTR](#) command. The command creates a &HOLD& file in the MGR/namespace directory. For example, on a new account, the 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: ^D.7
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\BuildXXX\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).

## 2.1.4 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-ambler 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 form queue 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.

## 2.1.5 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.



# 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 the SP-DEVICE being used, the first with all the arguments on the command line, the second and third with the command prompting for them:

```
USER:SP-DEVICE STANDARD HPLJ80
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 full format of the command is:

```
SETPTR chan,width,depth,topmargin,botmargin,mode,option{,option{,option  
&}}
```

where the meaning of the various arguments follows. Empty arguments, indicated by successive commas, do not change the value of their respective setting.

---

Position	Description
----------	-------------

Position	Description
chan	the affected print channel
width	page width in characters
depth	page depth in lines
topmargin	size of top margin in lines
botmargin	size of bottom margin in lines
mode	printing mode: 1 directs output to the spooler where it can later be manipulated; 3 sends the output to the &HOLD& file.

Position	Description
option	<p>disposition:</p> <p>AT [FORM QUEUE] – when using a printing mode of 1, sends the output to the specified form queue</p> <p>HOLD – indicates the job should be held in the queue after printing</p> <p>SKIP – the job output should be skipped, that is, not sent to any physical device</p> <p>AUX – Turns on auxiliary printing and hence all print jobs will not go to the spooler or the &amp;HOLD&amp; file, but will instead be sent to the auxiliary printer attached to the user’s terminal</p> <p>OPEN – This is the same as the KEEP option</p> <p>UNPROTECT – cancels any security controls for the job (useful only for printing mode 1; once set it cannot be unset)</p> <p>BANNER [text] – in printing mode 1, it specifies the text that will be printed on a one-page “banner” preceding the job output.</p> <p>BANNER UNIQUE [text] – For printing mode 3, it designates that the item id of the entry in the &amp;HOLD&amp; file be constant i.e. successive print jobs will overwrite the previous entry because they all have the same item id in the &amp;HOLD&amp; file. The optional [text] specifies the text that will be printed on a one-page “banner” preceding the job output.</p> <p>BANNER NEXT [text] – For printing mode 3, it designates that the item id of the entry in the &amp;HOLD&amp; file be incremental i.e. successive print jobs will create new and different entries in the &amp;HOLD&amp; file. The optional [text] specifies the text that will be printed on a one-page “banner” preceding the job output.</p> <p>COPIES [N] – sets the number of times the job will be printed (applicable only for printing mode 1)</p> <p>EJECT – specifies that the spooler should add an extra formfeed at the end of the job in addition to the one it normally sends</p> <p>NOEJECT – specifies that no formfeeds are to be printed after the job.</p> <p>INFORM – indicates that the print number of the job should be send to the user terminal when the job is created</p> <p>KEEP – the print job is kept open until the user logs off, a subsequent SETPTR command is executed for this job, or the job is closed via the <b>SP-CLOSE</b> command.</p>

## 3.2 SP-ASSIGN

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

This command assigns a printer form queue and printer options to a printer channel. It 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 creation of a new form queue. Where specified, the leading equal sign is mandatory; a space after the equal sign is optional. The *formspecs* are:

- =FORMNAME  
Specifies a form queue name other than the default of STANDARD for print channel 0
- nn=FORMNAME  
Specifies a form queue name other than the default of STANDARD for print channel nn
- FFORMNAME  
Specifies a form queue name other than the default of STANDARD for print channel 0
- FFORMNUMBER  
Specifies a form queue number for print channel 0
- QFORMNAME  
Specifies a form queue name other than the default of STANDARD for print channel 0
- QFORMNUMBER  
Specifies a form queue number for print channel 0
- nnn  
Changes the number of copies to print from the default of 1 to nnn

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
- 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 and printing suppressed.

Unless M is 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.

## 3.3 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.

## 3.4 SP-CLEAR

```
SP-CLEAR [ FORM-QUEUE ]
```

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

## 3.5 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.

## 3.6 SP-CONDUCT

```
SP-CONDUCT { nnn }
```

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

Without a command argument, **SP-CONDUCT** display the current conduct value. Otherwise, the value of the argument selects from the following options:

- 1: Any CMQL command that terminates will automatically close any opened printer spooler job (unless **SP-OPEN** is in effect).
- 2: Same action as 1, except it applies to Caché emulation commands such as **WHO** or **WHERE**.
- 4: Same action as 1, except it applies to PROCs and PARAGRAPHS.
- 8: Same action as 1, except it applies to user-written applications written in MVBasic and CATALOG'ed.
- 16: 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: New programs will NOT inherit the printer flag from their parent program.
- 64: Each MultiValue query will have a unique print spooler job.

The selection values are additive. The defaults setting by emulations is:

Emulation	Value
Cache	1
D3	47 (1, 2, 4, 8, 32)

Emulation	Value
IN2	15 (1, 2, 4, 8)
Information	15 (1, 2, 4, 8)
jBase	48 (16, 32)
MVBase	47 (1, 2, 4, 8, 32)
PIOpen	15 (1, 2, 4, 8)
Pick	15 (1, 2, 4, 8)
Prime	15 (1, 2, 4, 8)
Reality	32
R83	47 (1, 2, 4, 8, 32)
R95	47 (1, 2, 4, 8, 32)
Ultimate	47 (1, 2, 4, 8, 32)
UniData	64
Universe	1

## 3.7 SP-COPY

```
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. 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} fn_) {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 (D) option shows that SP.COPY will delete the original job once the copy has completed successfully.

The (O) option shows that SP.COPY will overwrite any existing target output. Without this option, if the target already exists an error will be reported.

The (V) option shows that SP.COPY will report in Verbose mode giving additional information about what is copied and what is deleted.

## 3.8 SP-COPIES

```
SP-COPIES [JOB [COPIES] ]
```

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

## 3.9 SP-CREATE

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

This command creates a form queue and allocates a device to it. 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, 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. The *device-name* must exist and be known to the system. 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.

## 3.10 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.

## 3.11 SP-DEVICE

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

This command allows you to change the device for an existing form queue. It assigns the specified printer device to a spooler form queue. It assigns the form queue a form number (FQ) in the global `^%MV.SPOOL`.

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`, `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. The *device-name* must exist and be known to the system. 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 mid-way through.

## 3.12 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.

Command	Description
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.
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.

Command	Description
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 despoller 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
User name 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.

## 3.13 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.

## 3.14 SP-FORM

```
SP-FORM [OLD-FORM-QUEUE [NEW-FORM-QUEUE] ]
```

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

## 3.15 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.

## 3.16 SP-GLOBAL

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

Without an operand, this command simply displays the name of the global currently being used as the spooler table. With an operand, it changes the name of the global being used as the spooler table.

The *global-name* argument must be a valid Caché global variable name, preceded by a ^ character. By default, the name of the global and therefore the spooler table is ^%MV.SPOOL, which is a system-wide global. This means all users share the same spooler. This command changes the name of the global where output will be collected. This allows different users to share a separate global.

In the example following, it means that in the future all users of account GREG will use the cache global called SPOOLER in the namespace ADMIN. (Since most Caché MultiValue account names have the same namespace name as account name, the namespace ADMIN is the same as the MultiValue account name 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.

## 3.17 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 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.

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 <b>SP-MOVEQ</b> command.
2	MOVE PRINT JOB	Move a print job from one form queue to another.

No.	Name	Description
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 <b>SP-STOP</b> command.
7	RESUME PRINTING	Resume printing on a form queue that has been suspended with the <b>SP-SUSPEND</b> command (or option 9). Same as the <b>SP-RESUME</b> command.
8	EDIT PRINT JOB	Add or remove pages from the given print job.
9	SUSPEND PRINT	Suspend printing on a form queue. Same as the <b>SP-SUSPEND</b> command.
10	CLOSE JOB	Close the specified print job.
11	SP-STATUS	Changes the menu to that provided by the <b>SP-STATUS</b> command.
12	KILL PRINTING	Kill printing immediately (even mid-job) and stop the despool process. Same as the <b>SP-KILL</b> command.
14	CLEAR QUEUE	Delete all inactive print jobs on the form queue. Same as the <b>SP-CLEAR</b> command.
16	DETAIL DISPLAY	Given a job number, display the details about the job in verbose mode.
99	EXIT	Exit the program.

## 3.18 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.

## 3.19 SP-LOOK

```
SP-LOOK
```

Display the assignment options for each assigned print channel. These options are assigned using **SP-ASSIGN**.

## 3.20 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.

## 3.21 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. If the spooler global is ^SPOOL then you should be careful of using this in case non-MultiValue spool jobs are deleted.

## 3.22 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.

## 3.23 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.

## 3.24 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.

## 3.25 SP-POSTAMBLE

```
SP-POSTAMBLE FORM-QUEUE SUBROUTINE-NAME
```

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

## 3.26 SP-PREAMBLE

```
SP-PREAMBLE FORM-QUEUE SUBROUTINE-NAME
```

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.

## 3.27 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.

## 3.28 SP-RESUME

```
SP-RESUME [FORM-QUEUE]
```

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

## 3.29 SP-SKIP

```
SP-SKIP [FORM-QUEUE [PAGES] ]
```

Defines how many pages to skip at the end of a printing a job.

## 3.30 SP-START

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

The **SP-START** command starts a despool process as a background job. 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.

## 3.31 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 <b>SP-CREATE</b> command.
2	CHANGE QUEUENAME	Change the name of the form queue. Same as the <b>SP-FORM</b> command.
4	CHANGE DEVICE	Change the device associated with an existing form queue. Same as the <b>SP-DEVICE</b> command.
5	CHANGE PAGE SKIP	Define how many pages to skip at the start of despooling a print job. Same as the <b>SP-SKIP</b> command.
6	LIST PRINT JOBS	Changes the menu to that provided by the <b>SP-JOBS</b> command.
7	DELETE FORM QUEUE	Delete a form queue and all the print jobs that were in that queue. Same as the <b>SP-FQDELETE</b> command.
8	VERBOSE DISPLAY	Displays detailed information about the specified form queues.
10	PURGE	Removes jobs from the form queue.
99	EXIT	Exit the program.

## 3.32 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.

## 3.33 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, re-position 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:

```

## 3.34 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.

## 3.35 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|” .
- **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.

## 3.36 SP-VERBOSE

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

This command provides detailed information on the specified jobs or form queues. It allows you to supply one or more job numbers or form queue names for verbose display.

If you specify no arguments, you are prompted to specify a print job / form queue name pair.

# 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.

## Spooler Command Examples

---

```
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](#)
- Option 3 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:

## Spooler Command Examples

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
2. CHANGE QUEUENAME
3. CHANGE DEVICE
4. CHANGE PAGE SKIP
5. CHANGE PAGE SKIP
6. LIST PRINT JOBS
7. DELETE FORM QUEUE
8. VERBOSE DISPLAY
9. EXIT

Enter action code / Page number (P#)

## 4.5 An SP-AUX Example

To enable auxiliary printing, there are 3 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 ([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 despoiled 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
User name 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 auxilliary printer) and mc4 (turns OFF the auxilliary 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 Caché ObjectScript [OPEN](#) command can be used. For example, on Windows systems, the device name looks like

```
|PRN|printername
```

and on UNIX systems, like

```
"lp -dprintername:QW:3:
```

**Note:** For Windows printers, the sequence “PRN” must always appear in uppercase.

**CAUTION:** When attempting to use a Windows printer, Caché will not be able to locate the printer if it is shared on a network. This is a known limitation.

The workaround is to direct printer data to a file and use the Windows spooling facilities to direct it to the proper device.

### 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 NUL
```

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 NUL
```

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
```

## 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
```

## 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 start a `despool` process from the terminal prompt as a

foreground job to aid in debugging problems. The following example runs a despool routine for form queue FQNAME in debug mode.

```
USER>do MVDespool^%SYS.MVSP2("FQNAME",1)
MultiValue Despool opening device '|PRN|HP7210' , mode '' , timeout ''
```

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 [System 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, in a terminal session, 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: OFF
USER> do ^MVDespool^%SYS.MVSP2("FQDEBUG",1)
```

# 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 iterative process of repeated trials and tests.

### A.1 Setup

#### 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 \\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 \\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.MultiValueSP("MYFQ",1)
MultiValue Despool opening device '\\DELLHOME\HP7210' , mode 'rw' , timeout ''
```