

# Operator Command Processing

By definition, the only SMARTS environment capable of receiving and acting on operator commands is the SMARTS server environment, the commands and format for which are documented in this chapter.

SMARTS also provides applications with the ability to accept operator commands when the console communication driver interface (CDI) module is specified to initiate operator communication.

This chapter covers the following topics:

- Communication Driver Interface (CDI) Commands
- SMARTS Server Environment Operator Commands

## Communication Driver Interface (CDI) Commands

When the CDI is active and the application running on SMARTS has opened the console, commands may be issued as follows in the various environments.

The following sections provide information about issuing commands under OS/390. For information about other platforms, see the documentation for the particular Software AG application product that uses SMARTS.

### Commands Issued to an OS/390 Batch Job

```
F jobname,command
```

- where

<i>jobname</i>	is the name of the OS/390 job
<i>command</i>	is the command string to be passed to the application

### Commands Issued to a SMARTS Server Environment OS/390 Job

```
F smarts,SERV,server-name,command
```

- where

<i>smarts</i>	is the name of the SMARTS server environment OS/390 job
<i>server-name</i>	is the name of the application specified on the SERVER configuration statement
<i>command</i>	is the command string to be passed to the application

# SMARTS Server Environment Operator Commands

Once the SMARTS server environment has been initialized, the computer operator can control the various SMARTS server environment facilities and ascertain the status of the SMARTS server environment system by entering one or more of the SMARTS server environment operator commands at the computer operator console.

## Communicating with the SMARTS POSIX Server

Apart from the SMARTS commands to initialize and terminate the POSIX server, the following operator commands may be issued to the POSIX server by issuing the SMARTS operator command

<i>Posix,command</i>
----------------------

- where

<i>Posix</i>	is the name of the POSIX server at startup.
<i>command</i>	is one of the commands QUIESCE or FORCE.

### QUIESCE

Use this command to terminate the POSIX server. Existing users are allowed to run until termination; however, any new request to use the services of the POSIX server is rejected.

Any POSIX server QUIESCE commands after the first issue a message indicating how many users are still using the system. When no users are using the POSIX server system, the command terminates the POSIX server.

### FORCE

Use this command to forcibly terminate the POSIX server and bypass all integrity checks during termination processing. Because command results are unpredictable, Software AG recommends that the entire SMARTS address space be cycled whenever FORCE is used to terminate the server.

## Platform Requirements for Entering Operator Commands

The following sections provide information about entering operator commands under OS/390. For information about other platforms, see the documentation for the particular Software AG application product that uses SMARTS.

### Note:

The MODIFY (F) command and proper ID (OS/390), or the REPLID (VSE) are assumed, and are not shown in command syntax in this chapter.

### OS/390

For OS/390 systems, operator commands are entered via the OS/390 MODIFY (F) and STOP (P) commands. These commands are directed toward the job name.

The general format for entering the OS/390 MODIFY command is:

```
F id,command,argument(s)
```

- where "id" is the job or started task name.

### VSE/ESA

For VSE/ESA systems, every SMARTS server environment has an outstanding reply on the console with the following message:

```
RTSOPC0085-* SMARTS READY FOR COMMUNICATIONS
```

The general format for entering a VSE SMARTS server environment operator command is:

```
nn command,argument(s)
```

- where "nn" is the VSE/ESA outstanding reply number assigned by the system. The message RTSOPC0085-\* is outstanding until the EOJ operator command is entered, at which time operator communications are halted.

In the case of a SMARTS abnormal termination, the operator must respond to the outstanding reply with an "EOB" (end-of-block) operator command.

### VM/CMS

For VM/CMS systems, the VM/CMS machine where SMARTS is started becomes the SMARTS console so operator commands must simply be entered at the VM/CMS console prompt and these will be processed by SMARTS.

The general format for entering a VM/CMS SMARTS server environment operator command is:

```
command,argument(s)
```

## Command Format Requirements

With the exception of the EOJ command, each command may be entered in full or may be abbreviated. The minimum abbreviation required is the number of characters necessary to uniquely identify the command. The characters needed to identify the command are underlined in the description of each command in the sections that follow.

## Command Overview

All SMARTS operator commands are described in the next section. The following table summarizes these commands:

Command	Purpose
DUMP	VSE/ESA: switches destination device for SMARTS server environment dumps from SYSLST to COMDMP and vice versa.
EOJ	Causes a logical shutdown of the SMARTS server environment.
PLIST	Displays a list of the current tasks defined in the requested task group and the status of each.
SERV	Enables a server to be stopped or started.
STATS	Writes the current SMARTS server environment statistics to the sysout dataset.
TLIST	Displays a list of the current threads defined in the requested thread group and the status of each.

## Command Descriptions

### DUMP

The DUMP command is only supported under VSE/ESA.

The computer operator uses the DUMP command to switch the device to which SMARTS ABEND dumps are written from SYSLST to COMDMP and vice versa.

Format	Description
DUMP	write snap dump to currently selected device
DUMP,DISK	write dump to VSAM dataset COMDMP
DUMP,NODISK	write dump to SYSLST

### EOJ

The EOJ command causes a logical shutdown of the SMARTS server system.

#### Note:

If the system programmer specified an EOJ verification password using the EOJ,VER system parameter, that password must also be entered with the EOJ command.

Format	Example
EOJ	EOJ
EOJ,VER=password	EOJ,VER=STOPCOMP

Note that a logical shutdown of SMARTS can also be performed with the OS/390 STOP (P) command.

If SMARTS does not come down after you have entered EOJ, try

```
EOJ, FORCE
```

## PLIST

This command provides a list of the current tasks defined in the requested task group and the status of each. If no task group is supplied as a parameter to this command, all tasks of all task groups are displayed. The following is a sample output resulting from this command.

```
RTSOPC0099-T COMMAND RECEIVED AT 9:33:15 FROM CONSOLE - 00 WAS PLIST
RTSOPC0067-T -> GrpName Status Use Wait LastOp Time Program Tid.. L
RTSOPC0067-T -> OC          A-Run
RTSOPC0067-T -> TAM          A-Wait                USTACK          4 0 STC06160
RTSOPC0067-T -> MSGPO        A-Wait
RTSOPC0067-T -> PAGING       A-Wait
RTSOPC0067-T -> FIO          A-Wait
RTSOPC0067-T -> DEFAULT    A-Wait    0    2 Wrtm        USTACK          4 0
RTSOPC0067-T -> DEFAULT    A-Wait    0    2 Coexit      UPDS            4 4
RTSOPC0067-T -> DEFAULT    A-Wait    0    2
RTSOPC0001-T PList command COMPLETED.
```

### GrpName

This is the name of the task group of which the task in question is a member. In the case of system tasks, this is the name of the system task.

### Status

This reflects the current status of the task. The status is a combination of two state indicators separated by a dash ('-'). The primary state indicator is the letter preceding the dash indicates whether the task is Active, Quiescing or Dormant by the letters A, Q and D respectively. Active in this sense indicates that the task is available to do work. When it is quiescing, it will remain active long enough to finish any work which has been started by the task while dormant tasks cannot be used and will have no secondary state associated with them. The secondary states that may occur are as follows:

Status	The task is . . .
Wait	waiting. In this state, the task is waiting on new work or on events requested by programs running in threads associated with it.
HrdW	in a 'hard wait' status caused by the program currently running on it. The task is not available to service other programs that might be waiting for it.
Run	currently running a user program.
Disp	going through its dispatching cycle either finishing off old work or looking for new work.

**Use**

This is the current use count for the task. The use count includes the current user of the task, any users for whom a wait was issued on the task and any users with an affinity for this task.

**Wait**

This is the current wait count for the task. This reflects the number events upon which the task is waiting and includes two standard events those being that work has been queued to the task group work queues or to the task's own work queue.

**LastOp**

This is the last SMARTS server environment operator command that was issued under control of the task.

**Time**

When the task has a secondary status of 'Run', this will reflect the time in seconds that this user has spent under control of the task.

**Program**

This is the name of the program currently active under control of the task, or the last program to be active under control of the task if it has a secondary status of 'wait'. If the task has never been used, this will be blank, however, once it has been used, this will always contain a value.

**Tid..**

This is the tid of the current TIB active under control of the task, or the last TIB to be active under control of the task if it has a secondary status of 'wait'. If the task has never been used, this will be blank, however, once it has been used, this will always contain a value.

**L**

This is the level number on which one of the following users is running:

- the user currently active under control of the task; or
- the last user that was active under control of the task if it has a secondary status of 'wait'.

If the task has never been used, this is blank; once it has been used, this always contains a value.

**SERV**

The SERV command enables the computer operator to pass commands to a server. Servers can be started and terminated using this command, and requests can be sent to servers. These servers must be specified in the SMARTS server environment sysparm SERVER. See the chapter *Configuring SMARTS Environments*.

**Examples**

```
SERV TERM,server-id
```

- to terminate the specified server.

```
SERV INIT,server-id,server-parameters
```

- to initialize the specified server. The string "*server-id,server-parameters*" is exactly the same as it would be specified in a SYSPARM SERVER=(*server-id,server-parameters*).

```
SERV server-id,server-command
```

- to send a command to the specified server. As long as the *server-id* does not conflict with any valid operator command name, the term "SERV" can be omitted in this notation.

## STATS

The STATS command writes the SMARTS server environment EOJ statistics to the dataset specified by the SYSPRINT DDNAME in the SMARTS server environment start-up procedure.

Format	Example
STATS	STAT

## TLIST

The TLIST command lists the current threads defined in the requested thread group and the status of each. If no thread group name is provided as a parameter to the request, all threads of all thread groups are displayed.

### Subgrp

The name of the thread subgroup to which the thread in question belongs.

### Status

The current status of the thread. The status is a combination of two state indicators separated by a dash ('-'). The primary state indicator is the letter preceding the dash and indicates whether the thread is active, quiescing, or dormant by the letters A, Q, and D, respectively. "Active" indicates that the thread is available to do work. When it is quiescing, it remains active long enough to finish any work that was started in the thread, while a dormant thread cannot be used and has no secondary state associated with it. The secondary states that may occur are as follows:

Status	Description
Free	Indicates that the thread is free to run other work. If there was a previous user of the thread, this state indicates that this user's program ended or was rolled out.
Occ	The 'occupied' status indicates that the thread is available to do work; however, the user program currently occupying the thread must be rolled out before any new work can be started in the thread.
Disp	Indicates that the thread is reserved and the dispatcher is currently in the process of either starting a new user program or rolling in a user program that was previously rolled out.
Run	Indicates that the user program in the thread is currently running.
Susp	Indicates that the user program has been temporarily suspended as a wait was issued either directly by the user program or indirectly by a function used by the program. In this state, the user program may not be rolled out. Internally, it indicates that the operating system task associated with the work is active elsewhere. Once the condition for the wait is satisfied, the task continues processing this work.

**Use**

The current use count for the thread. The use count includes the current user of the thread plus any other non-relocatable users previously rolled out from this thread.

**Wait**

The current wait count for the thread. This reflects the number of users waiting to run in the thread at the present time.

**LastOp**

The last SMARTS server environment operator command that was issued in the thread.

**Time**

When the thread has a secondary status of 'Susp' or 'Run', this reflects the time in seconds that this user spent in the thread.

**Program**

The name of the program currently active in the thread, or the last program to be active in the thread if the thread has a status of 'free' or 'occ'. If the thread has never been used, this is blank; however, once the thread has been used, this always contains a value.

**Tid..**

The TID of the current TIB active in the thread, or the last TIB to be active in the thread if the thread has a status of 'free' or 'occ'. If the thread has never been used, this is blank; however, once the thread has been used, this always contains a value.

**Active L**

The level number on which the one of the following users is running:

- the user currently active in the thread; or
- the last user to be active in the thread if the thread has a status of 'free' or 'occ'.

If the thread has never been used, this is blank; however, once the thread has been used, this always contains a value.