MEMORANDUM
RM-5044-PR
SEPTEMBER 1966

JOSS: CONSOLE SERVICE ROUTINES
( THE DISTRIBUTOR )
I. D. Greenwald

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND
JOSS: CONSOLE SERVICE ROUTINES
(THE DISTRIBUTOR)
I. D. Greenwald

This research is sponsored by the United States Air Force under Project RAND—Con-
tact No. AF 49(638)-1700—monitored by the Directorate of Operational Requirements
and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF.
Views or conclusions contained in this Memorandum should not be interpreted as
representing the official opinion or policy of the United States Air Force.

DISTRIBUTION STATEMENT
Distribution of this document is unlimited.
PREFACE and SUMMARY

The distributor is that part of the JOSS on-line, time-shared computing service which handles communications with the user consoles. This Memorandum describes the distributor's interface with the consoles and with other elements of the computer program. In addition, some of the design criteria which led to interface decisions are briefly described. An Appendix contains program documentation, consisting of table descriptions and flow diagrams.

This Memorandum is a part of The RAND Corporation's continuing program of research in computer sciences under U.S. Air Force Project RAND.

The JOSS system was originally implemented on the JOHNNIAC computer in 1963 by J. C. Shaw; the present expanded version is implemented on the Digital Equipment Corporation PDP-6 computer.

*JOSS is the trademark and service mark of The RAND Corporation for its computer program and services using that program.
## CONTENTS

**PREFACE and SUMMARY** ...................................................... iii

Section

I. **INTRODUCTION** .......................................................... 1

II. **THE DISTRIBUTOR** .................................................... 3
    Interface with the Monitor ............................................ 3
    Console Characteristics .............................................. 4
    Interface with the Console .......................................... 5

**APPENDIX**

DISTRIBUTOR DOCUMENTATION ............................................... 9

I. **CONSOLE INPUT INTERRUPT ROUTINE** .............................. 11
II. **CONSOLE OUTPUT INTERRUPT ROUTINE** ............................ 17
III. **TURN OFF CONSOLES** .............................................. 21
IV. **INITIALIZE CONSOLES** ............................................ 22
V. **TRANSMIT LINE** ..................................................... 23
VI. **SWITCH TO USER (GREEN)** ....................................... 24
VII. **SPECIAL SEND** .................................................... 25
VIII. **BEEP THE STATIONS** .............................................. 26
IX. **ENABLE STATION** ................................................... 27

**JOSS BIBLIOGRAPHY** ..................................................... 29
I. INTRODUCTION

JOSS software is composed of four major components and several subcomponents (Fig. 1).

The major components are:

1) The monitor, which handles resource allocation (except for disc space), scheduling, queueing, accounting, internally generated signals, and all console and teletype input-output (I/O) buffers;

2) The distributor, which handles all I/O to and from the JOSS consoles and teletypes (except the PDP-6 control teletype);

3) The interpreter, which does all of the processing of user programs;

4) An off-line disc maintenance program, which provides facilities for dump/restore via tape, updates, reallocation, and various other service functions.

The several subcomponents include:

1) The arithmetic processor, which contains all of the arithmetic and function evaluation routines and various other subroutines. This package is a subcomponent of the interpreter and has no interface with the monitor.

2) The on-line disc processor, which handles all I/O to and from the disc. It is a subcomponent of the interpreter. However, communication paths are established via signals to the monitor.
Fig. 1--Overview of JOSS Software Components
II. THE DISTRIBUTOR

INTERFACE WITH THE MONITOR

Input and output to JOSS consoles are handled on a "line" basis: an input line is terminated by a typewriter carriage return or page signal from the user; an output line is terminated by an (internal) terminal character. The main functions of the distributor are to accept input characters, convert them to internal encoding, place them in a monitor-specified internal buffer (see Appendix, Fig. 1), and signal the monitor when a line is complete. Conversely, during output, the distributor accepts characters from a monitor-specified internal buffer, converts them to external encoding, transmits them to the station, and signals the monitor when a line is complete.

To facilitate programming and to improve error detection, we distinguish between the two types of line completion by signals:

1) CR (Carriage Return or Page): input line complete.
2) TO (transmission over): output line complete.

Since the distributor operates via externally generated interrupts (input character waiting, output character transmitted), its only interface with the monitor is via signals.

All signals from the distributor to the monitor are handled via a "signal list." The distributor places the console number and signal type in a relative location (as determined by a count) in a special table, then increments the count (as well as making bounds checks). The monitor,
during selected clock interrupts, picks up the table entries and decrements the count. (During these few microseconds, character interrupts are disabled.)

In addition to the two signals generated by the distributor (CR and TO), there are three others:

1) ON: The console has been turned ON.
2) OFF: The console has been turned OFF.
3) IN: The user wishes to interrupt his processing.

All signals from the monitor to the distributor are handled via subroutine calls. Where necessary, the subroutines disable character interrupts to accomplish their purpose. Monitor-generated signals are:

1) SHUT: Turn off all consoles.
2) INIT: Initialize all consoles; turn on those waiting.
3) TL (transmit line): Initiate transmission to a console.*†
4) SU (switch to user): Give control of console to user.*†
5) BEEP: Send one beep to each console that is on.
6) ENB (enable): Enable input from a specified console.*

**CONSOLE CHARACTERISTICS**

When attached to the JOSS system, the consoles can be in one of two mutually exclusive states (which the computer simulates for teletype stations):

---

*Console number is part of calling sequence.
†Buffer number is implicit in calling sequence.
1) GREEN: The user controls the console.

2) RED: The computer controls the console.

In the RED state, the typewriter keyboard is locked and the user is limited to three controls:

1) Power OFF, which sends a unique signal to the central processing unit (CPU).

2) Interrupt request (IN), which sends a unique signal to the CPU. This signal is interpreted as a request to interrupt processing and return the console to the GREEN state.

3) Typewriter READY/HOLD, which sends two unique signals to the CPU. The hold signal is interpreted as a request to discontinue output typing temporarily (without losing characters); the ready signal is interpreted as a request to resume output typing.

The RED state prevails whenever the carriage return or page key is struck, as well as when power is turned on. In addition, the computer can signal the console to "switch to RED." The console GREEN state can occur only as a result of a computer signal to "switch to GREEN."

Further computer control over the console includes attaching or detaching it from the system, and turning indicator lights on and off.

INTERFACE WITH THE CONSOLE

One of the design criteria for JOSS was to minimize the probability of a "stare-down problem," i.e., a situation in which the console is in one state but the computer thinks it is in another state. For example, if the console is RED but the computer thinks it is GREEN, each will be waiting for the other to take some action.
To meet this criterion, several status toggles were built into the logic of the console and made available for computer inspection:

1) ON state.
2) RED/GREEN state.
3) READY/HOLD state.
4) IN-REQUEST state (which shows the state of a computer-generated acknowledgment to user's interrupt request).

The computer can send the following signals:

1) NO-OP: no operation.
2) Set the ON state on or off.
3) Set the RED/GREEN state to RED or GREEN.
4) Set the IN-REQUEST state on or off.

At the same time, the computer may request the console to respond with the state of the status toggles.

The distributor contains a two-word station control register (SCR) for each console (see Appendix, Table 1). Among other things, the SCR reflects the computer's view of the state of the console. On any computer- or console-generated change of state or on any detected errors on incoming characters, the state toggles (except READY/HOLD) are checked against the SCR. If the check disagrees, the computer sends the appropriate signal to place the console in the same state as the SCR, then rechecks. (In addition, incoming bad characters are converted to a special code which, upon line completion, result in the interpreter's retyping the input line with overstrikes (#) replacing the bad characters. Besides informing the user of the errors, this supplies information for maintenance personnel.)
A final device for avoiding stare-down is user intervention. If the console is GREEN when it should be RED, the user can overcome the problem by hitting CR. This will place the console in RED (the distributor ignores a CR from any station whose SCR indicates RED). Conversely, if the console is RED when it should be GREEN, the user can hit INTERRUPT. An INTERRUPT from a station whose SCR indicated GREEN results in a request for the state of the status toggles, since this condition is logically impossible. (The user cannot interrupt a program in progress when he has no program in progress.)

Further interfaces with the console concern non-printing typewriter functions which take longer than one character time to perform (i.e., TAB, Carriage Return, Page). As a result, succeeding (output) characters would type "on the fly." To avoid this problem, the console goes into a "hold" state during these functions and sends a "ready" signal upon completion. The distributor inhibits typing until the ready is received by setting the SCR to hold when it sends a nonprinting function. (The conversion table from internal to external form has a bit to indicate such a function.)

Shifting case on the typewriter during input results in changing to different look-up tables for conversion. The location of the current look-up table for a console is carried in its SCR, as is a bit to indicate "current" case. On input line completion, the distributor ensures that the typewriter is in lower case. The output conversion table carries a bit to indicate case; if this disagrees with "current" case, the latter is inverted and the appropriate shift character generated and transmitted.
The distributor also controls allowable line length, inhibiting the user from backspacing beyond the beginning of a buffer or typing over the end. In the latter case, a special character alerts the interpreter to output an error message.

On receipt of an OFF signal from a console, the distributor sets a flag in the SCR to "disable" that console, which prevents the generation of signals to the monitor until the latter has had time to "clean up," whereupon it generates the ENB (enable) signal described above (p. 4).

Finally, the BEEP signal alerts the user to an administrative message. Normally, it is sent once per second for as long as the machine operator desires.
Appendix

DISTRIBUTOR DOCUMENTATION

Fig. 1--Station Buffer
Table 1

THE STATION CONTROL REGISTER (SCR): A TWO-WORD-PER-STATION TABLE DESIGNATED T75 AND T76

a. Format of T75

<table>
<thead>
<tr>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0=enabled; 1=disabled.</td>
</tr>
<tr>
<td>1-13</td>
<td>Not used.</td>
</tr>
<tr>
<td>14</td>
<td>0=RED</td>
</tr>
<tr>
<td>15</td>
<td>0=HOLD</td>
</tr>
<tr>
<td>16</td>
<td>0=NO IN-REQUEST</td>
</tr>
<tr>
<td>17</td>
<td>0=OFF</td>
</tr>
<tr>
<td>18-35</td>
<td>Location of attached buffer.</td>
</tr>
</tbody>
</table>

b. Format of T76

<table>
<thead>
<tr>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1=transmitting.</td>
</tr>
<tr>
<td>1</td>
<td>1=awaiting done.</td>
</tr>
<tr>
<td>2</td>
<td>1=last character was upper case.</td>
</tr>
<tr>
<td>3-8</td>
<td>Not used.</td>
</tr>
<tr>
<td>9-17</td>
<td>One character stack bit 9=1:character to be sent bits 10-17=external form of character.</td>
</tr>
<tr>
<td>27</td>
<td>1=upper case table; 0=lower case table.</td>
</tr>
</tbody>
</table>
I. CONSOLE INPUT INTERRUPT ROUTINE

Entered via channel trap (INP)
Note INP must be less than OUP

C25: Read in station number = n
     Pick up SCR(n)
     Read in character & release scanner
     Station disabled?                      YES  NO
     TTY station?                           .90  TC25
     SCR = ON?                             .06

.03 Convert character to internal form

Normal: SCR = Green?
     Advance current pointer via C25.63
     Store character in buffer
     Dismiss break and exit

.05 Send NO-OP status request via C25.80
     Dismiss break and exit

.06 Is character an ON signal?
     Is character a status response?
     Dismiss break and exit

.05

.06 Is character an ON signal?                      YES  NO
     Is character a status response?             .19

.90 ON signal?
     Set SCR = ON + READY                      YES

.90A Dismiss break and exit

.90B OFF signal?                                    YES  NO
     Set SCR = OFF + HOLD
     Dismiss break and exit

HOLD Signal

.11 Set SCR = HOLD
     Dismiss break and exit
READY Signal

.13 Set SCR = READY
   Was SCR = HOLD?
.13A Dismiss break and exit
.14 SCR = awaiting done?
   SCR = character to be sent?
   Reset SCR
.17 SCR = transmitting?
   Set up character for output via C26.50
   Transmission over?

IN Signal

.15 SCR = RED?
   Send light IN, status via C25.80
.16 Set SCR = IN
   Was SCR = IN?
   Dismiss break and exit
.16A Place on signal list via C25.50
   Dismiss break and exit
.23 Send NO-OP, status via C29
   Dismiss break and exit

ON Signal

.19 SCR = OFF?
   SET SCR = ON + HOLD + RED
   Send light system, status via C29
   Place on signal list via C25.50
   Dismiss break and exit
.20 Send light system, status via C29
   Dismiss break and exit
OFF Signal

.22  Set SCR =
     All flags off
     DUMBFR
     Disable flag
     Place on signal list via C25.50
     Dismiss break and exit

IN + HOLD Signal

.24  Set SCR = HOLD
     To .15 (IN Signal)

UPSHIFT Signal

.35  Set SCR = Upper Case Table
     Dismiss break and exit

DOWNSHIFT Signal

.37  Set SCR = Lower Case Table
     Dismiss break and exit

BACKSPACE Signal

.60  Is "current pointer" > 0?  .60A
    Decrement current pointer
    Backspace byte pointer via C25.55
.60A  Dismiss break and exit

SPACE Signal

.62  Advance current pointer via C25.63
     Advance byte pointer
     Dismiss break and exit
PAGE Signal

.65 Set flag = PG
To .67

CR Signal

.66 Set flag = CR
.67 Set SCR = RED
Was SCR = RED?
Dismiss break and exit
.67A Set SCR = lower case, hold
Send NO-OP, status request via C29
Place CR or PG adjacent to last
input character
Set byte pointer to point to CR or PG
Place on signal list via C25.50
Dismiss break and exit

STATUS Signal

.70 Add to appropriate counters
Mask out all but GREEN/IN/ON
Error?
Dismiss break and exit
.71 Add to error counter
RED/GREEN error?
ON/OFF error?
SCR = IN?
Send light IN, status via C25.80
Dismiss break and exit
.71A Send turn off IN, status via C25.80
Dismiss break and exit
.76 SCR = RED?
Send switch RED, status via C29
Dismiss break and exit
.77 Send switch to GREEN, status
via C25.80
Dismiss break and exit
PARITY Error

.78     Add to counter
.78A    Set illegal character
        To "normal"

Illegal Character

.79     To .78A

Place on Signal List

C25.50  Signal count < signal max?  HALT
        Store (Signal, station #)
        in SIGTBL
        Increment count
        Signal count < signal limit?  Exit
        Increment COMEBACK
        Exit

Backspace the Byte Pointer

C25.55  Decrement address
        Increment byte four times
        Exit

Advance Current Pointer

C25.63  Current < 78?  .61
        Increment current
        .64  Current < farthest right?
        Set farthest right = current
        .64A  Exit
        .61  Place "line too long" as first
        character in buffer
        Dismiss break and exit
Send

C25.80 SCR = character to be sent? .83
New character = switch to GREEN? .82
Put character to be sent into
  new character
  .82 Clear character to be sent
  .83 SCR = READY?
  SCR = Awaiting done? .85
  Transmit character
  Set SCR = awaiting done
  .84 Store SCR
  Exit
  .85 Place new character in character
go to be sent
  To .84
## II. CONSOLE OUTPUT INTERRUPT ROUTINE

Entered via Channel Trap (OUP)
Note that OUP must be greater than INP

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C26</td>
<td>Disable input traps</td>
</tr>
<tr>
<td></td>
<td>Read in station # = n</td>
</tr>
<tr>
<td></td>
<td>Set SCR = not awaiting done</td>
</tr>
<tr>
<td></td>
<td>SCR = READY?</td>
</tr>
<tr>
<td></td>
<td>SCR = character to be sent?</td>
</tr>
<tr>
<td></td>
<td>Send character, release scanner</td>
</tr>
<tr>
<td></td>
<td>Set SCR = no character to be sent</td>
</tr>
<tr>
<td></td>
<td>Set SCR = awaiting done</td>
</tr>
<tr>
<td></td>
<td>Enable input traps</td>
</tr>
<tr>
<td></td>
<td>Dismiss break and exit</td>
</tr>
<tr>
<td></td>
<td>SCR = transmitting?</td>
</tr>
<tr>
<td></td>
<td>TTY station?</td>
</tr>
<tr>
<td></td>
<td>Set up character for output via C26.50</td>
</tr>
<tr>
<td></td>
<td>Terminal character?</td>
</tr>
<tr>
<td></td>
<td>Send character, release scanner To .05</td>
</tr>
<tr>
<td></td>
<td>SCR = transmitting?</td>
</tr>
<tr>
<td></td>
<td>Is next character terminal?</td>
</tr>
<tr>
<td></td>
<td>Set transmission over via C26.25</td>
</tr>
<tr>
<td></td>
<td>Dismiss station</td>
</tr>
<tr>
<td></td>
<td>Enable input traps</td>
</tr>
<tr>
<td></td>
<td>Dismiss break and exit</td>
</tr>
<tr>
<td>Transmission over</td>
<td><strong>0.21</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C26.25</td>
<td>Set SCR = not transmitting</td>
</tr>
<tr>
<td></td>
<td>Set SCR for dummy buffer</td>
</tr>
<tr>
<td></td>
<td>Place on signal list via C25.50</td>
</tr>
<tr>
<td></td>
<td>Exit</td>
</tr>
</tbody>
</table>
Set up character for output

C26.50 Pick up character according to byte pointer
   Terminal? .60
   Convert by table look-up Character = space? .51
   Case shift required? .51
   Set appropriate shift character
   Exit
   .51 Advance byte pointer
   Long function? .59
   Set SCR = HOLD
   Exit
   .59 Set SCR = lower case
   Was SCR = lower case? .58
   Set downshift character
   Exit
   .58 Set transmission over via C26.25
   Exit
TTY Input routine

TC25
SCR = RED?
Character = ON?
Character = OFF?
SCR = ON?
Character = IN?

25A
Dismiss break and exit

ON1
Set SCR = ON, READY
To IN1 + 1, signal = ON

OFF1
SCR = ON?
Set SCR = RED, OFF, Disabled,
      Not transmitting,
      not awaiting done
To IN1 + 1, signal = OFF

IN1
Signal = IN

IN1+1
Place on signal list via C25.50
Dismiss break and exit

.10
Execute the TTY look-up table

.12
Bump character count
To CR1 on 72nd character
Store character in buffer
Send character back via C25.80
Dismiss break and exit

PG1
Send CR via C25.80
      To .20 with signal = PG

CR1
Send CR via C25.80
Set signal = CR

.20
Store signal in buffer
Set SCR = RED
Send line feed via C25.80
To IN1 + 1

RUBOUT
Decrement count and go to E·BUF
      if < 0
Backspace byte pointer
Send backslash via C25.80
Dismiss break and exit

E·BUF
Send CR, LF via C25.80
Dismiss break and exit
TTY Output Routine

TC26  Get character (byte pointer)
       Terminal
       Set trans. over via C26.25
       Dismiss station via C26.20

26A   Convert character by table
       look up (XCT)

26A5  Bump byte pointer
       .10 Send character, release scanner
           Exit via C26.21

       .30 (PAGE) Set CR in buffer
           Set line feed
           TO .10

       .40 (CR) Set LF in buffer
           Set CR
           TO .10
III. TURN OFF CONSOLES

Entered by JSR SHUT

SHUT:  Turn off 630 traps
       Delay about 30 ms
       (to complete any output)
       Send 271g to stations 0(1)N*S-1
       [N*S=6010; note that TTY's receive a "9"
       JOSS consoles receive: "turn off
       system lite"]
       Delay about 30 ms
       Exit
IV. INITIALIZE CONSOLES

Entered by PUSHJ PP C20

C20   Turn off PI system
      Turn off system lites by JSR SHUT
      Clear all receive scanner flags that are up
      by DATAI (delay about 90 μs between flags)
For station 0(1) N.S-1:
      Send (via C25.80) 3708
      ["On, status" for consoles; "X" for TTY]
      Set the SCR for:
      DUMBFR, READY, RED, OFF, NØ-IN
Assign channels to 630 and release scanners
      (OUP=4; INP=3)
      Turn on PI system; Enable channels 2(1)7
      Exit
V. TRANSMIT LINE

Entered by PUSHJ PP, C27
Station # in S(18-35)
Ptr. to buffer header in S.BUF(S)(18-35)

C27 Disable 630 traps
Pick up SCR words
  [S.BUF(3)]+1(= Pointer to buffer)−SCR
   Set up pointer word in buffer
   Set SCR = transmitting
   SCR = awaiting done? .20
   TTY station? (SL CONSOL) .10
   SCR = READY? .20
   Set up character for output by C26.50
     Terminal character? HALT
     Send via C29 .05
   Exit
   .10 (TTY station)
     Send a null via C25.80
     To .05
   .20 Store SCR words
     To .05
VI. SWITCH TO USER (GREEN)

Entered by PUSHJ PP, C28
Station # in S(18-35)
Ptr. to buffer header in S.BUF(S)(18-35)

C28 Disable 630 traps
Set SCR for lower case look-up table
(T78 or T90 for console or TTY)
Set SCR, to point to buffer
Set buffer pointer word
Set counts = 0
Set SCR = GREEN, NØ-IN
TTY station? .10
Send, via C25.80; switch green, status
.05 Enable 630 traps
Exit
.10 [TTY station]
   Send, via C25.80: Ding
   To .05
VII. SPECIAL SEND

Entered by JSR  C29

C29  Send the contents of RM to station
designated by RN
Set SCR = awaiting done
Store SCR words
Exit
VIII. BEEP THE STATIONS

Entered by JSR  C30

C30  Set i = N.S-1
     .1 Disable the 630 traps
        SCR (i) = ON?
        Send, via C25.80: Beep
     .3 Enable the 630 traps
        i≤CONSOL?
        Exit
     .3A  i = i-1
         To .1
IX. ENABLE STATION

Entered by JSR C31
Station # in S(18-35)

C31

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>.2</td>
<td>.2</td>
</tr>
</tbody>
</table>

Disable 630 traps
Reset SCR disable bit
SCR = ON? (Set OFF)
Send, via C25.80: Lite system, status
.2 Enable 630 traps
 Exit
JOSS BIBLIOGRAPHY

Publications of Historical Interest:


"The JOSS System: Time-Sharing at RAND," Datamation, Vol. 10, No. 11, November 1964, pp. 32-36. (This article is based on Baker, above.)


Publications of Current Interest:
