

A 12

Symbiont and Job Management  
Technical Manual



Xerox Corporation  
701 South Aviation Boulevard  
El Segundo, California 90245  
213 679-4511

**XEROX**

# **Xerox Universal Time-Sharing System (UTS)**

**Sigma 6/7/9 Computers**

## **Symbiont and Job Management Technical Manual**

FIRST EDITION

90 19 87A

February 1973

Price: \$3.00

# NOTICE

This publication documents the symbiont and job management functions of the Universal Time-Sharing System (UTS) for Sigma 6/7/9 computers. All material in this manual reflects the C01 version of UTS.

## RELATED PUBLICATIONS

<u>Title</u>	<u>Publication No.</u>
UTS Overview and Index Technical Manual <sup>†</sup>	90 19 84
UTS Basic Control and Basic I/O Technical Manual	90 19 85
UTS System and Memory Management Technical Manual	90 19 86
UTS Operator Communication and Monitor Services Technical Manual	90 19 88
UTS File Management Technical Manual <sup>†</sup>	90 19 89
UTS Reliability and Maintainability Technical Manual	90 19 90
UTS Interrupt Driven Tasks Technical Manual <sup>†</sup>	90 19 91
UTS Initialization and Recovery Technical Manual	90 19 92
UTS Command Processors Technical Manual	90 19 93
UTS System Processors Technical Manual	90 19 94
UTS Data Bases Technical Manual	90 19 95

---

<sup>†</sup>Not published as of the publication date given on the title page of this manual. Refer to the PAL Manual for current availability.

The specifications of the software system described in this publication are subject to change without notice. The availability or performance of some features may depend on a specific configuration of equipment such as additional tape units or larger memory. Customers should consult their Xerox sales representative for details.

# CONTENTS

Symbionts, Cooperatives, and RBBAT, Overview .....	1
Purpose .....	1
Overview .....	1
Symbiont File Attribute Information Maintained by RBBAT .....	3
Symbiont/Cooperative Big Picture .....	4
Description .....	6
SACT - Symbiont Activation Routine .....	8
Purpose .....	8
Usage .....	8
Data Bases .....	8
Subroutines .....	8
Description .....	8
Flowchart .....	10
COOP - User Interface .....	11
Purpose .....	11
Usage .....	11
Data Bases .....	12
Subroutines .....	12
Errors .....	14
Interactions .....	14
Description .....	14
INSYM - Input Symbiont Routine .....	16
Purpose .....	16
Usage .....	16
Operator Communication .....	16
Data Bases .....	16
Subroutines .....	17
Interactions .....	18
Description .....	18
OUTSYM - Output Symbiont Routine .....	19
Purpose .....	19
Usage .....	19
Operator Communication .....	19
Data Bases .....	19
Subroutines .....	19
Interactions .....	20
Description .....	20
ADD - Get a File .....	22
Purpose .....	22
Usage .....	22
Input .....	22
Output .....	22
Errors .....	23
Restrictions .....	23
Description .....	23
Subroutines .....	23

REQDC - Symbiont Core and RAD Management	25
Purpose	25
Description	25
REQDB - Request Symbiont Granule	26
Purpose	26
Usage	26
Data Bases	26
Subroutines	26
Description	26
REQCB - Request Symbiont Core Buffer	27
Purpose	27
Usage	27
Data Bases	27
Description	27
RELCB - Release Symbiont Core Buffer	28
Purpose	28
Usage	28
Data Bases	28
Subroutines	28
Description	29
Data Base	29
SGC - Symbiont Ghost Communication (In Module SACT)	30
Purpose	30
Usage	30
Data Bases	31
Description	31
SYMCOM - Operator-Symbiont Communication	32
Purpose	32
Usage	32
Input	32
Data Base	32
Subroutines	32
Output	32
Errors	33
Description	33
SYMSUBR - Symbiont Subroutines	35
Purpose	35
Data Bases	35
Subroutines	35
SUB - Other Miscellaneous	38
Purpose	38
Data Base	38
Subroutines	38
SUSPTERM - SUSP Suspend Symbiont, TERM Terminate Symbiont	39
SUSP	39
Purpose	39
Usage	39

Data Bases	39
Subroutines	39
Interactions	39
Output	39
Description	39
TERM	40
Purpose	40
Usage	40
Data Bases	40
Subroutines	40
Description	40
RBBAT - Symbiont File Handling	41
Overview	41
Data Bases	42
Symbiont File and Multi-Batch Tables	42
Counters	46
GETI Tables	46
Ghost Communication	47
Operation	47
CONTROL	47
RBBAT Processing Routines	47
AIF, AIFJE	48
AOF, AOFNB, AAFP	49
GOF	51
MBSCALL	52
KPRI0	52
KDEL	53
KDISP	54
JESTAT	55
JEDEL	56
MBS - Multi-Batch Scheduler	57
Entry	57
Data Bases	57
Description	57
Remote Batch Processing	60
Overview	60
RB:FLAG	60
RBSS	63
DSCIO	64
Special Cases	67
Error Handling	67
RBBAT	69
Data Bases	69
General Information	69
Message Files	69
Blocking of RB Input Files	69
Description	70
Remote Batch Processing Effects on Other Routines	75
AIF	75
AOF, AOFNB, AAFP	77
GOF	78

ID

Symbionts, Cooperatives and RBBAT, Overview

PURPOSE

Symbionts transfer information between secondary storage (RAD, DP) and peripheral devices (CR, CP, LP, PL and RBT); cooperatives transfer information between the user's program and secondary storage. A highly significant speed advantage in I/O processing is achieved by substituting a high speed I/O device for a low speed device and by blocking input records so as to require only one I/O wait period by the user for many records. An advantage in resource scheduling is attained because the user program is detached from the device. This allows programs to run at times when devices are otherwise unavailable (RBT not connected; device down; or device busy with previous work).

OVERVIEW

Records sent to and received from the low-speed peripherals (CR, CP, LP, PL, RBT) are buffered to RAD through the symbiont - cooperative routines. Four stages are readily identifiable:

First, input jobs from the CR or RBT are blocked by the input symbiont into disc unit records and written onto secondary storage in the peripheral storage area (PER). This process is carried out asynchronously with respect to other tasks in the system and, once started, is interrupt driven until completion. Initiation is accomplished by operator command for CR and is automatic for RBT. The input symbiont recognizes !JOB cards for CR and RBT and treats them as beginning of file, and end of previous file (if any), recognizes !FIN cards for CR and RBT and treats them as end of stream, and recognizes !RB cards for RBT and treats them as beginning of file/end of previous file as with !JOB cards. At file end the file starting disc address is passed to RBBAT, the symbiont file ghost job, for entry into the batch tables.

Second, when a user issues a read directed to the card reader, the operation is intercepted by the input cooperative. This routine reads from secondary storage and deblocks the records for presentation to the reading program, which is not allowed to read past the end of the symbiont file containing his own job. Initially the multi-batch scheduler selects the job to be run by placing the job and resource information in the GETI tables. The batch user is started and the !JOB card CCI read causes this information to be placed in the users JIT. Thereafter records of the file are passed to the user on subsequent reads.

Third, the output cooperative, which is an intercept routine acting on all output directed to symbiont devices, blocks records into buffers and writes them to secondary storage. Separate symbiont files are built for each type of output. Upon user signal ('superclose', usually at end of job), the file is closed by entering it into the RBBAT queue via the add output file communication.

Fourth is the interrupt - driven output symbiont task, which reads symbiont files and writes the symbiont device. Output symbionts are started automatically when RBBAT senses that there is work to do, the device is idle and otherwise capable of processing the output.

Symbionts use, for buffer memory, pages obtained from the pool of the user memory. This restricts maximum user size in that a user must not be allowed to exceed the available physical space left while symbionts are active. The cooperatives use similar buffer and control memory pages from the user's virtual space. The buffer management routines get memory and restrict size appropriate to the mapped/unmapped condition on entry.

Symbiont files are selected by RBBAT for input and output by resource, priority, system id., and control information maintained by RBBAT. Priority of symbiont files which originates from the job card (or on-line user default) may be changed by the operator, who may also delete files. Control information (e.g. remote batch hold) is specified by the user. Table GA-1 lists the RBBAT maintained file information.



Table GA-1: Symbiont File Attribute information maintained by RBBAT

File Type Device Type Attribute (table)	INPUT	OUTPUT					
		LP	CP	TY	PL	PR	PP
sysid (BH:SID)	N	N	N	N	N	N	N
file starting disc address (BW:SDA)	N	N	N	N	N	N	N
Remote ID (BB:RID)	N	R	R	R	R	R	R
PRIORITY INCREMENT (BB:PI)	N						
file's ACCOUNT (BD:ACCOUNT)	N						
JOB RUN time (BH:TIME)	N						
JOB RESOURCES: SP, 7T, 9T, CO (BW:RES)	N						
target partitions (BH:PART)	N						
shareable volumes	N						
exclusive volumes	N						
'ORDER' restriction	N						
'ACCOUNT' restriction	N						
'HOLD' output signal	N	R	R	R	R	R	R
message file flag		N					

N - determined by system (SYSGEN/CONTROL defaults or user JCL)

blank - not applicable or not maintained

R - determined by system (user JCL) & only maintained for remote files

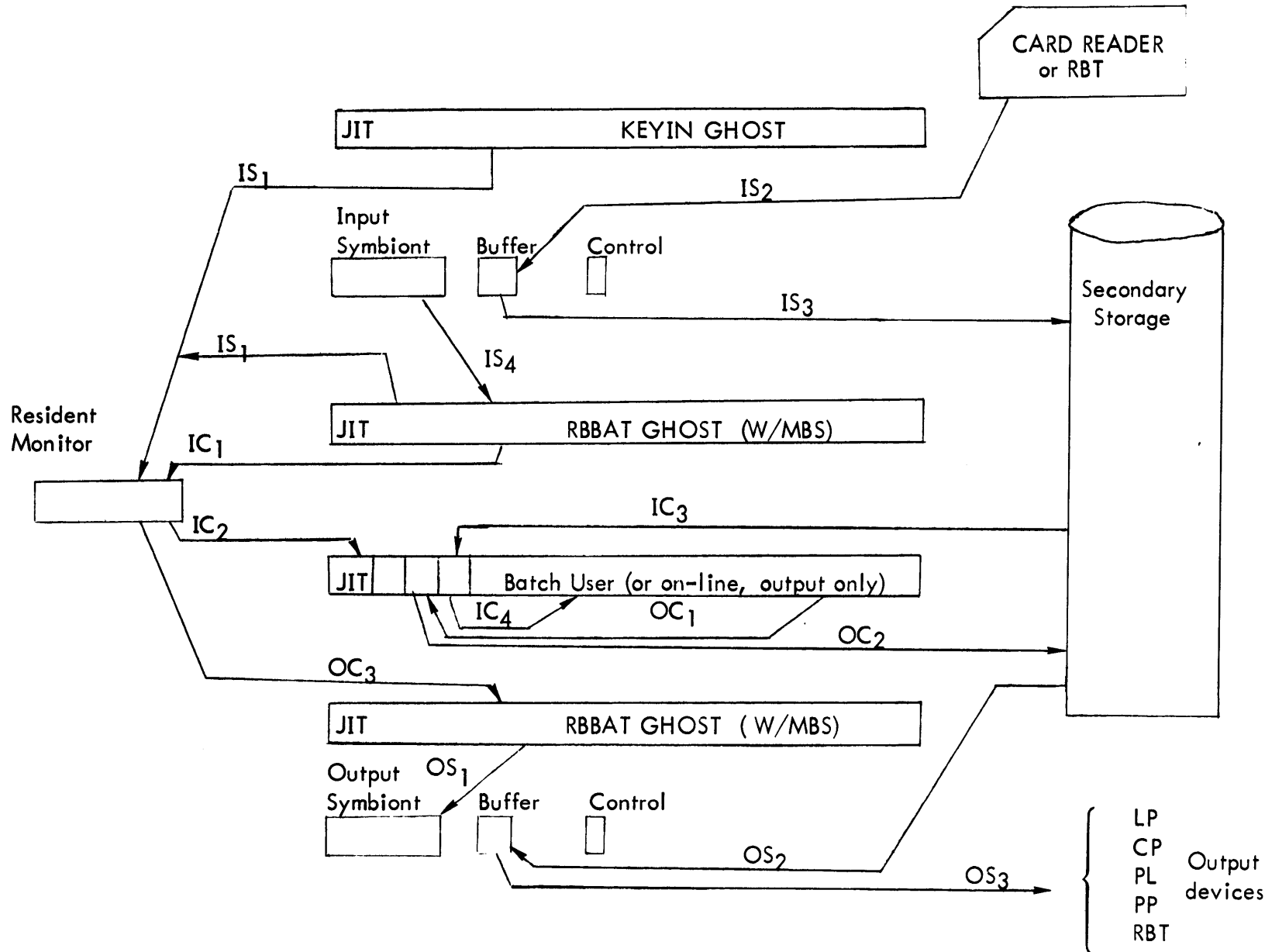


Figure GA-1 Symbiont/Cooperative Big Picture

Symbiont/Cooperative Big Picture

- IS1. Input symbiont activated by KEYIN or RBBAT via resident table.
- IS2. Card images read into symbiont buffer.
- IS3. When buffer is full, contents written to secondary storage linked to previous blocks of same file.
- IS4. Continue until end of file, then tell RBBAT. Terminate when !FIN read.
- IC1. RBBAT/MBS selects job to run by putting job information into resident tables
- IC2. When batch user reads first card, job information is transferred to JIT.
- IC3. Input symbiont file blocks are read into users cooperative buffer.
- IC4. Records are transferred, one at a time, in response to user reads.
- OC1. User symbiont output is intercepted and put into cooperative buffers.
- OC2. When buffer is full, contents are written to secondary storage.
- OC3. User issues 'superclose' and file is queued by RBBAT.
- OS1. RBBAT initiates output symbiont.
- OS2. Output symbiont blocks are read into symbiont buffer.
- OS3. Records are transferred, one at a time, to output device. Continue processing blocks until end of file, then start another file or terminate.

UTS-C01 TECHNICAL MANUAL

Description

The means of communicating with a symbiont is a collection of resident tables called SYMTAB (Section VI.02); all messages addressing the symbiont from the operator are examined by the subroutine, SYMCOM (Section GA.03). The symbiont tables in SYMTAB are created of system generation time.

The symbiont communicates with RBBAT via a resident message area, SGCBUF. The symbiont for a local device communicates with its operator via the system console typewriter. The symbiont for a remote device (RBT) communicates with its operator via one of the remote devices. Some communications require action by the operator and his response is handled by KEYIN or RBBAT and passed to the symbiont in SYMTAB.

A single symbiont may drive many similar devices. For example, the output symbiont (OUTSYM, Section GA.01.04) may drive many printers and card punches. The symbiont is given its device specific information when initialized, and it maintains this information in a context buffer (Section VI.03). The location of this context buffer is known to the symbiont whenever it is operating on that device.

The symbionts coexist with one another and the rest of the system by means of queued access to resources. As each symbiont no longer needs a particular buffer it is released and another acquired. Buffers are acquired by waiting in line on a buffer queue, so that each symbiont gets fair access to available buffers. Optimally there are enough buffers for all symbionts, but the symbionts will operate in a degraded fashion with fewer buffers. A similar technique is used with RBBAT communication buffers, and with secondary storage granules. Also in acquiring initial memory for the symbiont and its context a queue is used. Description of the performance of these functions appears later in Section GA.

A symbiont performs only one I/O operation at a time and gives up control until the operation is complete. All I/O calls made by a symbiont will be to QUEUE1 (Section DA). This call requests I/O with end action. When the I/O has completed, control is returned to the symbiont at the specified end-action location with register SR3 loaded with the context buffer address. The context buffer provides the symbiont with the necessary device-dependent information. The symbiont gives up control until end-action occurs. Exit is to the last caller of the symbiont, either the symbiont activate routine SACT (Section GA.01.01) or the end action processing routine REQCOM in IOQ (Section DA). The flow of symbiont activity is roughly as follows:

1. First entry from SACT.
2. Queue an I/O request with end-action: device I/O if input, file I/O if output.

3. Exit to SACT.
4. SACT initiates any other symbiont activity or exits to caller: IOINT in IOQ or idle loop in execution scheduler SSS.
5. Entry from REQCOM at symbiont end-action address.
6. Next I/O request is queued by symbiont.
7. Exit to REQCOM and to process interrupted by I/O completion above.
8. Repeat 5-7 until buffer full and written (input) or buffer empty (output).
9. Release buffer and queue symbiont for another, exiting to REQCOM.
10. Next entry will be from IOINT via SACT.
11. Repeat 2-10 until a) !FIN card input or b) last output buffer encountered.

Two special cases of symbiont usage by the system can be identified as follows:

When RBBAT detects erroneous !JOB, !LIMIT or !RESOURCE commands for a job or must communicate with a remote operator or the central computer operator, a message file is built. The case is special because it is system created output and is handled on a next up basis by OUTSYM. The display of files in the system or belonging to a remote user is handled this way.

When a symbiont file is deleted by its remote submitter or by the computer operator, it is given to OUTSYM (whether an input or output file). In this context OUTSYM reads the blocks from secondary storage, ignoring content and checking only the file's linkage, and releases the secondary storage space a block at a time.

UTS-C01 TECHNICAL MANUALID

SACT - Symbiont activation routine

PURPOSE

To initiate (enter), or reactivate (re-enter) symbionts that are queued for execution.

USAGE

BAL, SR4      SACT

DATA BASES

SYMTAB - symbiont table (Section VI.01)

Context block (Section VI.03)

SUBROUTINES

CNTXTSET is used to clear and set up a context block (Section GA.05.01).

GSC is used to get a context block (Section GA.03.02)

SAQNSERT is used to queue the symbiont for reactivation in case a context block is not available (Section GA.05.01).

DESCRIPTION

SACT is called by the I/O interrupt routine in IOQ and by the idle loop in the execution scheduler (SSS). When SACT is entered, it first checks if any symbionts are queued in SQHD. If no symbiont devices are queued, SACT immediately returns to the calling routine. If symbiont devices are queued, SACT scans the queues in sequential order; first the symbiont activate queue, second the symbiont core buffer queue, and last the symbiont RAD granule queue. For each symbiont device queued in SQUE, SACT performs the following operations:

As a symbiont device is found in a queue, it is removed from that queue.

The symbiont context block address and symbiont return address are found in SCNTXT and SRET, subtables of SYMTAB, for any symbiont device found in the buffer queue or the granule queue. In this case the symbiont is entered immediately.

When a symbiont device is found in the activate queue, the device status in SSTAT, a subtable of SYMTAB, is checked. If the device status is active, the same conditions exist as described for the buffer and granule queues above.

For a device status of inactive, the device signal character in SSIG, a subtable of

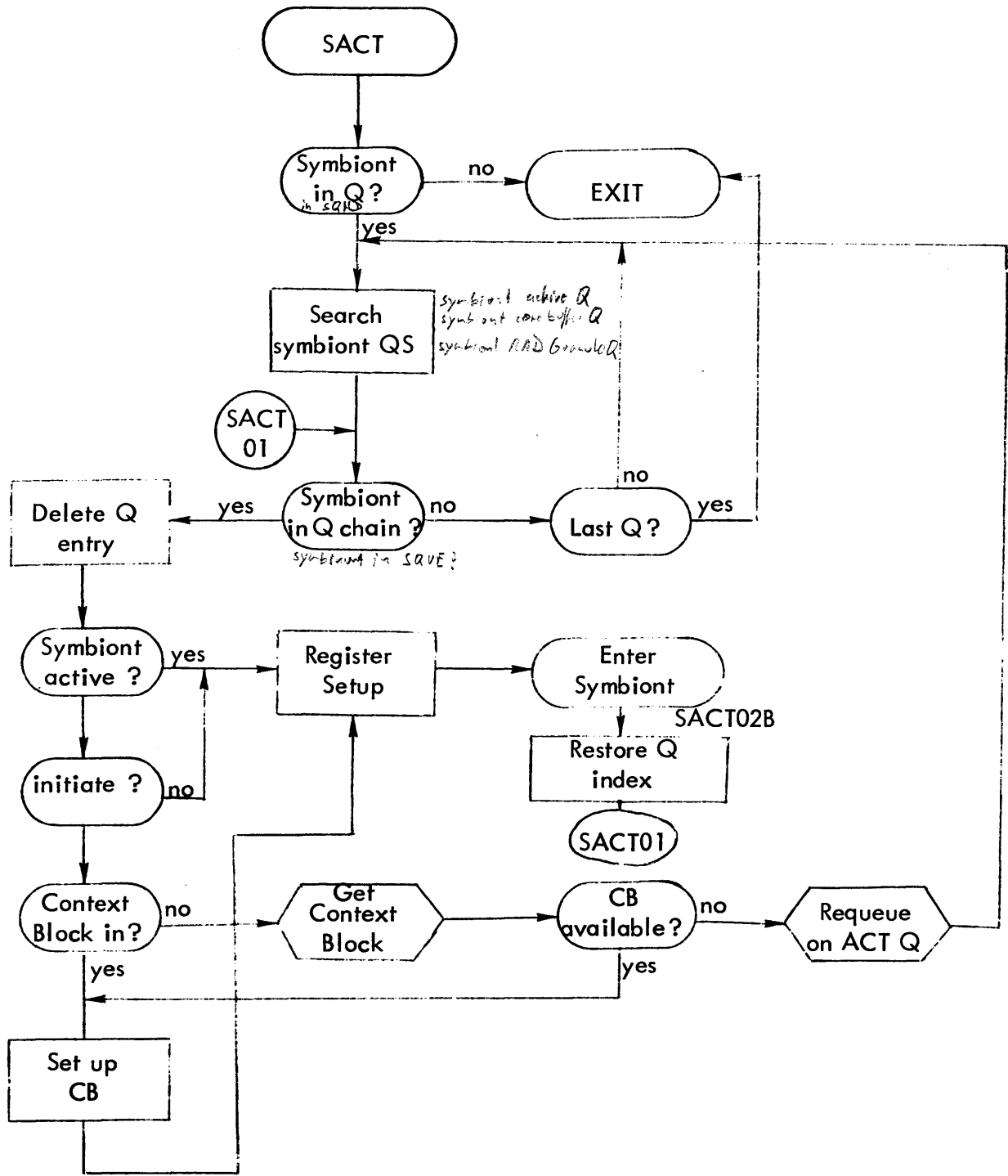
UTS-C01 TECHNICAL MANUAL

SYMTAB, is checked. If the device signal character is not "I" for initiate, the same conditions exist as described for the buffer and granule queues above. For a signal character of "I", SACT next determines whether a context block address exists in the proper entry of SCNTXT. If the context block address is found, then the symbiont relocation base is contained within the context block. The base address is placed in the proper entry in SRET, and other information is set in the context block as determined by the characteristics of the symbiont device. Now the same conditions exist as described for the buffer and granule queues above. When no context block address is found, SACT requests a context block via GSC. If the request cannot be satisfied, the symbiont is requeued and SACT moves on to the next queue. If the request is satisfied, SACT sets the same conditions as if the context block were found in the proper entry in SCNTXT.

SACT attempts to enter as many symbionts as possible. Each symbiont in turn returns control to SACT after initiating an appropriate I/O request with end action specified.

As SACT processes each queue it picks up the queue thread and marks the queue empty. This is consistent because when SACT is finished the queue will be empty logically for this time around. It is also very important because it allows a symbiont called by SACT to requeue itself for later recall by SACT and avoids the obscure race condition. A simple example of the race condition is that the symbiont needs a disc granule that will only be released when SACT gives up control to the batch user allowing him to read his cards and release the granules. This is by no means the only race condition avoided by this technique.

UTS-C01 TECHNICAL MANUAL





UTS-C01 TECHNICAL MANUALID

COOP - User Interface

PURPOSE

## Input cooperative:

To obtain card images written in the form of a linked disc granule file on the RAD. The file is created by INSYM (Section GA.01.03).

## Output cooperative:

To block card and print images in the form of a linked disc granule file on the RAD. The records of the file are transferred to the output device via OUTSYM (Section GA.01.04).

## Close cooperative file:

To terminate output cooperative files. A special end-of-file record control character is stored in the file and the files are added to the symbiont file directory via ADDF. Context blocks are released via RSC.

USAGE

## Input/Output cooperative:

Entry: BAL, SR4 COOP

Registers: R5 = JIT address

SR1 = function code/DCB address (8, 24)

Exit: B \*SR4

## Close cooperative file:

Entry: BAL, SR4 CCLOSE

Registers: R5 = JIT address

Exit: B \*SR4

DATA BASES

COOP:

SYMTAB - symbiont tables (Section VI.01)

Context block - (Section VI.03)

C:CSC - count of symbiont/cooperative RAD reads and writes

DCT - device control tables (Section VG.01)

RQLDGA - count of free symbiont RAD granules.

J:JIT - address of Job Information Table which contains the following items:  
(Section VA)

J:CBPOOL - head of free context block pool

J:DBPOOL - head of free data block pool

J:PUF - peripheral use flags for user

JCOVP - cooperative buffers first virtual page number

BL:OFS - Batch limit, number of output file slots

SYSID - JIT system I. D., used to identify file in RBBAT communications

JRBID - JIT remote batch I. D., used to identify remote output

PRT - JIT job priority, passed to RBBAT for output scheduling.

JITUSCDX - head of the chain of context blocks in use by the user (Section VA)

E:NSYMD - no symbiont RAD event code

RQLDGA - count of free symbiont RAD granules

SUBROUTINES

Internal

COPDCB is used to initialize the context block DCB for RAD read/write

Entry: BAL, SR4 COPDCB

Registers: R3 = Context block address

Exit: B \*SR4

COPDER1, also addressed as COPERLG, is used to record Blink/Flink errors in the error log via ERRLOG

Entry: BAL, SR4 COPDER1

Registers: SR3 = address context block

Exit: B \*SR4

UTS-C01 TECHNICAL MANUAL

COPEA is used to perform end-action type cleanup for input cooperative disc read and output cooperative write.

Entry: BAL, SR4 COPEA

Registers: R6 = Data buffer address  
SR1 = I/O opcode, DCB address (8, 24)  
SR3 = context block address

Exit: B \*SR4

COPGSB is used to get a data buffer via GSB or, if one was already acquired, to get the address of the buffer from the context block.

Entry: BAL, SR4 COPGSB

Registers: R3 = context block address

Exit: \*SR4

ALLOCATE is used to get TWO PAGES of core for cooperative buffers via T:GNVPI and allocate buffers as required. The TWO PAGES are acquired starting at JCOVP in the users context area

Entry: BAL, SR4 ALLOCATE

Exit: B \*SR4

External

BCDTOEBC - is used to convert a BCD record to EBCDIC (Section DC)  
CHKDA - is used to verify a disc address (Section DA)  
CNTXTSET - is used to structure a context block (Section GA.05.01)  
GCD - is used to get a context block (Section GA.03.02)  
RCD - is used to release a context block (Section GA.03.02)  
RCB - is used to release a data buffer (Section GA.03.02)  
SYMTABCK - is used to determine if a device is in the symbiont tables (Section GA.05.01)

GETI - is used to get input file  
RSG - is used to release a symbiont granule (Section GA.03.02)  
IOSPIN - is used to await completion of a previous COOP operation  
MSRLP34 - is used to expand user tables in an output record  
SGC:NCB - is used to park a user via E:NSYMF until either output file slots or RBBAT communication buffers are available (Section GA.01.01)  
SGCQ - is used to trigger RBBAT and pass information to him (Section GA.01.01)

UTS-C01 TECHNICAL MANUALERRORS

RECOVER - COOP can effect a system failure with code x'2D' when it determines that the users context has been clobbered such that there are more open output files than there are buffers for them.

INTERACTIONS

QUEUE is used to queue an I/O request (Section DA).

T:GNVPI is used to get 2 pages of core (Section FA).

DESCRIPTION

COOP determines whether the device that is being requested is symbiont operated by checking for its DCT index in SNDDX. If it is not in SNDDX, or if the DCB is open in diagnostic mode, or if the device is operator's console, COOP exits directly to QUEUE with the request.

If the I/O operation to be performed is input and this is the first operation (i. e. the input dcb is 'super' closed), COOP requests from GETI the information in the GI tables. The information in these tables was put there when MBS (Section GB/GC) selected the job to be started. GETI returns the disc address of the first block in the file in SRI (Section GA.02.01).

If there is already a context block assigned to the device specified in the calling DCB, the context block address is stored in the DCB. This funnels all DCB's opened to the same device through a single context block. If there is no context block and no pages have yet been allocated for cooperative buffers the subroutine ALLOCATE obtains the pages, links the context and data buffers and initializes the heads of the buffer pools, J:CBPOOL and J:DBPOOL. A context buffer and data buffer are requested via GCD and GSB, respectively, on the user's initial request for a record from a symbiont input device.

COOP then reads in a sector size data buffer. When that I/O is complete, a record is transferred from the data buffer to the user's buffer as specified in the DCB. Subsequent reads then involve only the transfer of a record from the data block to the users buffer until the data block is exhausted. At that time another RAD read request is issued. This process continues until an end-of-data record control code (X'40') is encountered in a block that has a forward link address of 0, i. e. end of input file whereupon the COOP returns a '!FIN' record to the user.

DESCRIPTION (Cont'd.)

The record passed to the user's buffer and the calling DCB are processed by COOP in the same manner as they would be processed by QUEUE. If the file conversion bit is set in the I/O function code, the record is converted from BCD to EBCDIC via BCDTOEBC. Secondary storage granules are released via RSG after they are used, to free secondary storage for other symbiont/cooperative activities.

If the operation to be performed is output, COOP, on the initial request to a symbiont-operated device, requests a context buffer via GCD and a data buffer via GSB. If there are no secondary storage granules associated with the file, COOP gets a granule via GSG. Next COOP sets up the control character and byte count for the output image, which is found via the calling DCB, and inserts them into the data buffer. If the data buffer is filled, it is linked to the next secondary storage sector (GSG, if necessary); written out to secondary storage via QUEUE: and the data buffer is released (RSB). COOP processes the calling DCB for each write in the same manner as QUEUE would process it.

Output files are terminated by CCLOSE (in module SUPCLS). This routine is invoked via a CAL1, 9 6. This CAL is executed when a user terminates (LOGOFF) or when an on-line user issues a PRINT command to TEL. CCLOSE runs the chain of in-use context blocks (head is JITUSCDX in JIT) and for each output context block it:

1. Stores an end-of-data record control code (X'40') as the last code in the current data buffer.
2. Stores a 0 forward link RAD address in the buffer. (1 and 2 together define the termination of an output file to the output symbiont.)
3. Causes the current block to be written to RAD.
4. Sets a closing-file flag in the context block.
5. If the file was a batch punch file with 2 or less cards (separator cards) it is deleted.

After the block has been written out by COOP, the closing-file flag is checked. If set, the file is added to the symbiont file directory by constructing the appropriate RBBAT communication message and triggering RBBAT via SGCQ.

UTS-C01 TECHNICAL MANUAL

ID

INSYM - Input symbiont routine

PURPOSE

To read and block card images from a symbiont input device and write a linked symbiont file on RAD.

USAGE

Entry: B 0, R2 = from symbiont activate routine, SACT

Registers: R2 = Symbiont entry point address  
R3 = Symbiont SYMTAB index  
R4 = Symbiont relocation base address  
SR4 = Context buffer address

Exit: TSTACK contains the symbiont exit address to SACT

OPERATOR COMMUNICATION

See SYMCOM (Section GA.04.01).

DATA BASES

SYMTAB - symbiont tables (Section VI.01)  
DCT - device control tables (Section VG.01)  
Context Block (Section VI.03)  
C:CSC - count of symbiont/cooperative disc reads and writes  
RQLDGA - symbiont input threshold.  
BL:IFS - count of input file slots available in RBBAT  
AIF - code for add input file (RBBAT)  
RBLIMS - remote batch terminal DCT index limits  
SSIG - symbiont signal character

UTS-C01 TECHNICAL MANUAL

SUBROUTINES

Internal

SCRIOB is used to initialize the symbiont dcb contained in the context block.

Entry: BAL, SR2 SCRIOB

Input: R1 = context block address  
symbiont exit address contained in TSTACK

Output: SR4 = symbiont exit address pulled from TSTACK  
SR1 = DCB address  
DCB = initialized

External

QFP11NCB is used to pull SR4 to queue for communication (RBBAT) buffer. Return is made to R4 and R11 is pushed. D1, D2 and D3 are saved in context block and restored in return.

SGCQ is used to add an input file to RBBAT file tables by means of a communication buffer.

REG is used to load R4 (Symbiont base address), R3 (Symbiont table index), D2 (Symbiont signal) and R1 (Symbiont context).

QFACTP11 is used to queue the symbiont for re-activation. PULL's from TSTACK and returns with TSTACK reloaded.

SUSP is used to suspend symbiont activity.

UTS- C01 TECHNICAL MANUAL

INTERACTIONS

QUEUE1 is used to read input records from the input symbiont device and write blocked records to RAD.

DESCRIPTION

The input symbiont is initiated from the console by an unsolicited keyin. After requesting a core buffer via REQCB, the symbiont begins reading records and building the appropriate RAD files, via REQDB and QUEUE1. A JOB control command causes initiation of a new file and entering of the previous file into the symbiont file directory via SGCQ. A FIN control command causes the closing of the current file and the symbiont terminates activity via SUSPTERM. A FIN control command is never included in the file. A control character and byte count is included with each card image. There are five different control characters which indicate BCD record, EOD record, binary record, end of buffer, and error encountered by the input device (Section VI.04).

The input symbiont recognizes the "L", "S", and "X" character. The "L" signal causes the symbiont to suspend operation after the completion of this file. It will not be activated until an unsolicited keyin signal of "I" or "C" referencing it is received. The "S" signal causes the symbiont to suspend operation via SUSPTERM. The proper signal for continuing is an unsolicited keyin signal of "C" referencing the appropriate input device. The "X" signal causes the symbiont to simulate reading a "!FIN" card. The truncated file is entered into the symbiont file directory via ADDF.

For a description of the flow of control in the input symbiont, see Symbiont Control in the Symbiont/Cooperative Overview (Section GA).

The input symbiont passes three types of files to RBBAT:

1. A job file (a !JOB followed by all its control commands and data)
2. A file containing multiple !RB's and a job file.
3. A file containing only !RB's



UTS-C01 TECHNICAL MANUAL

ID

OUTSYM - Output symbiont routine

PURPOSE

To output individual records from a blocked cooperative RAD file to a symbiont output device.

USAGE

Entry: B 0, R2 from symbiont activate routine, SACT

Registers: R2 = symbiont entry point address  
R3 = device SYMTAB index  
R4 = symbiont relocat on base address  
SR3 = operating symbiont context buffer address

Exit: To SACT at the address contained in the last word in TSTACK.

OPERATOR COMMUNICATION

See SYMCOM (Section GA. 04. 01).

DATA BASES

SYMTAB - symbiont tables (Section VI. 01).

Context block (Section VI. 03).

C:CSC - count of symbiont/cooperative RAD reads and writes

\*DCT7 - used to obtain lines per page for print type devices

SGCBUF - RBBAT communication (see Section GA. 01. 01), used to obtain PRIORITY, RBID, SYSID and SDA of output file.

RBLIMS - Used to differentiate local and remote devices.

BL:OFS - Used on partial file add to determine output file slot availability.

RB:FLAG - for remote devices, used in special handler communication.

SUBROUTINES

External

CHKDA is used to verify that the forward link in a buffer read from RAD is a valid RAD address (Section DA).

COPERLG is used to insert an entry in the error log (Section GA. 01. 02).

RSG is used to release a symbiont or file granule (Section GA. 03. 02).

UTS-C01 TECHNICAL MANUALSUBROUTINES (cont'd.)

SAQNSERT is used to queue the symbiont for re-activation when a block read from RAD does not contain a valid record control code (Section GA. 05. 01).

SUSPTERM is used to suspend or terminate the symbiont (Section GA. 04. 02).

RELCB is used to release a symbiont data buffer (Section GA. 03. 01).

REQCB is used to request a symbiont data buffer (Section GA. 03. 01).

QFACT is used to effect a processing delay via re-activation to retry a failure such as failure to release a symbiont granule due to ALLOCAT stack full.

SRELBUF is used to effect a no-pause buffer release (Section GA. 03. 01).

SGCQ is used to trigger RBBAT and pass or request symbiont file information (Section GA. 01. 01).

SGCRA is used to release an RBBAT communication buffer after receiving information (Section GA. 01. 01).

QFP11NCB is used to pull SR4 and Queue on activate queue for a no-  
communication (RBBAT) buffer delay.

INTERACTIONS

QUEUE1 is used to read sectors of blocked records from RAD and to write individual records to the output symbiont device.

DESCRIPTION

Operator intervention or the insertion of an output file causes the initiation of the output symbiont. Processing of output files continues until operator intervention or the exhaustion of complete files for the device. Normal processing involves:

- a) maintenance of a recovery stack for print devices to allow print page backup in the event of a paper jam, or suspension of partial file.
- b) acquiring via a communication with RBBAT the output file specific information.
- c) deblocking the cooperative output file and executing the following record control codes:
  - 1) 4, 5 - print, type, plot or punch BC bytes
  - 2) 6, 7 - process with format BC bytes
  - 3) x'40' - end of block - fetch another
  - 4) x'84' - suspend for forms change and signal remote or local operator with BC bytes of forms message.

UTS-C01 TECHNICAL MANUALDESCRIPTION (cont'd.)

- 5) x'80' - produce on-line banner paper separator for bursting.
  - 6) x'88' - internal code which flags disc error messaging in process.
- d) Releasing of old disc granules as recovery backup point passes over them.

Several notable exceptions to normal processing are:

- a) recognition of inconsistent blocks by control code and disc address checks and producing ERRLOG entry and informing operator.
- b) recognition of the following system operator or handler inserted signals:
  - 1) 'Q' - pass the current partial output file back to RBBAT at the print recovery point for rescheduling or deletion.
  - 2) 'R' - recover from paper jam by restarting at the print recovery point.
  - 3) 'X' - abort printing of current file and delete remaining granules.
  - 4) 'L' - lock device at termination of current file
  - 5) 'S' - suspend operation until signaled to continue 'C' or recover 'R'.
  - 6) 'C' - continue after suspension from current point.

The output symbiont performs double duty by serving as the delete symbiont file operative. When entered with a SYMTAB index which indicates a 0 DCT index (no device) the symbiont reads the linkage of a symbiont file and releases the granules without regard to record control format. The efficient multi-task performs this service with negligible overhead.

UTS-C01 TECHNICAL MANUAL

ID

ADD - get a file

PURPOSE

To get for the executing job, at time of first card read, resource and limit information passed down from RBBAT.

USAGE

ENTRY: BAL, SR4 GETI  
REGISTERS: R1, R5, R14, R15 VOLATILE  
SR1 (R8) = input file starting disc address on return  
D4 (R15) = SYSID on  
R1 = partition number on return.  
EXIT: B \*SR4

INPUT

S:CUN - identifies the entry by position in GIB:UN  
PLH:SID - User SYSID by PARTITION

OUTPUT

GI - The GETI tables  
GIB:UN - User number  
GIH:TIM - maximum run time (stored in J:MRT)  
GI:ASPN - shareable Drives/Spindles Bit Map (stored in J:ASPIN)  
GI:RES - max core, 9T, 7T, SPINDLES (stored in JB:MNPA, JB:M9T,  
JB:M7T, JB:MSP)  
GIB:SLN, GIB:XLN - shared and exclusive serial number table linkages  
(stored in JB:SLNK, JB:XLNK)  
GIB:RID - remote identification (stored in JRBID)  
GI:SDA - Starting disc address, input file  
GIB:PRT - partition number (stored in JB:PNR)

UTS-C01 TECHNICAL MANUALERRORS

RECOVER - GETI can effect a system recovery with code x'2C' when the current user number cannot be located in the GIB:UN table. This can happen in two cases, both indicating problems elsewhere in the system. First, if a job reads past the end of its file, GETI will be requested to get the file again. While exiting the job CCI has read the FIN card more than once. Second, if a job gets started without MBS selecting it, the entry will not be in the GETI tables. This indicates a problem in job initiation or MBS-GETI communication, or ghost reading C device.

RESTRICTIONS

There are only three slots in the GETI tables (this may be increased by modifying COMBAT, GETI and RBBAT). Because there are only 16 batch partitions and there exists small probability that more than a few of these need initiation at any time the number was assumed to be small. Because of the queueing nature of the implementation more than one entry is needed. The structure of the tables is such that if the number of entries is evenly dividible by three, then table space is most efficiently used.

DESCRIPTION

GETI is called by COOP at the time the symbiont input dcb is opened, to return the disc address of the block containing the first card (!JOB). GETI compares S:CUN with GIB:UN until a match indicates the correct entry. If not found, RECOVER with code x'2C'. The information in the various parallel GI tables is moved to the JIT (maximum run time, - J:ASPIN, maximum core pages allowed, maximum 9T, 7T and spindles, linkage to the serial numbers, remote batch identifier, JB:PNR. The GI table entry is released.

SUBROUTINES (in ADD)

PULL11X - Pulls SR4 from TSTACK and exits indirect SR4.  
 B11PIX - Branch indirect SR4 after incrementing.  
 XT0BCD, IDT0BCD - Convert contents of R6 or 4 bytes respectively from D2 (binary) to D3 (EBCIDIC) right justified. Link is SR4 uses D1.  
 JTYPED Type on OC, (D3) bytes, at word address in D2 and upon end action release the monitor buffer. Link is SR4, R6 to SR3 non-volatile.

UTS-C01 TECHNICAL MANUAL

- JTYPE - Entry to JTYPED with end-action prespecified.
- NXTSID - Using S:USID, assign the next incremental SYSID, between SMUIS+i and x'5B5A'. Link is SR4, uses D1 and ENABLES.

UTS-C01 TECHNICAL MANUAL

ID

REQDC - Symbiont core and RAD management

PURPOSE

To provide for get and release of symbiont data buffers and symbiont RAD granules. In addition this module enqueues the requesting symbiont when a resource is not available and it always enqueues the releasing symbiont.

DESCRIPTION

This module consist of the routines REQDB, REQCB and RELCB - get granule and get and release data buffer. Both the symbiont buffer pool and the symbiont granule pool are allocated at sysgen at user option. Since there may be many symbiont devices there may also be competition for symbiont core and RAD resources. This competition is managed by the queueing technique described in Section GA.01.01 (SACT). A queue is provided for each type of resource. When a symbiont requests a resource that is in use it is put into the appropriate queue. When it is reactivated, the entry point will have been adjusted to enter the symbiont at the point where the resource was requested. When the symbiont that is using the resource finally releases it, it too is put onto the tail of the queue for that resource. This gives the queued requesting symbiont a chance to get the resource before the releasing symbiont grabs it again.

UTS-C01 TECHNICAL MANUAL

ID

REQDB - Request symbiont granule

PURPOSE

To obtain a RAD granule for a requesting input symbiont.

USAGE

Entry: BAL, SR4 REQDB  
Registers: R3 = device SYMTAB index  
SR3 = device context block address  
Output: SR1 = granule address  
Exit: B \*SR4 (granule available exit)  
B QFGP11 (Exit if no granule available)

DATA BASES

SYMTAB - symbiont tables (Section VI.01)  
Context block (Section VI.03)

SUBROUTINES

External

SQINSERT is used to insert the SYMTAB index for a device into the symbiont queue for re-activation if a granule is not available (Section GA.05.01).

GSG - Get symbiont granule.  
REG - Used to establish symbiont registers  
QFGP11 - Used to queue symbiont for re-activation

DESCRIPTION

If there are granules available, a granule is obtained via GSG and returns to the input symbiont. If there are not enough granules, the symbiont entry point for activation is set to the BAL to REQDB and the symbiont is queued for reactivation via QFGP11. Exit will be to the symbiont exit point which is pulled from TSTACK and the calling symbiont will be placed in the RAD queue. When the symbiont is selected from the queue for execution, control will enter it at the call on REQDB.



UTS-C01 TECHNICAL MANUAL

ID

REQCB - Request symbiont core buffer

PURPOSE

To obtain a core data buffer for a requesting symbiont.

USAGE

Entry: BAL, SR4 REQCB  
Register: R3 = device SYMTAB index  
SR3 = device context block address  
Exit: B \*SR4 (Buffer available exit)  
B GFGP11

DATA BASES

SYMTAB - symbiont tables (Section VI.01)  
Context block (Section VI.03)

DESCRIPTION

If a buffer is available, its address is returned in D3. If none is available, an attempt to obtain a free physical page is made. If available, the page allocated as two buffers. One is given to the requesting symbiont and the other is placed in the available buffer list.

When no physical pages are available, GFGP11 is used to queue the symbiont for re-activation.

UTS-C01 TECHNICAL MANUAL

ID

RELCB - Release symbiont core buffer

PURPOSE

To release the specified symbiont core buffer

USAGE

Entry:           B   RELCB  
Registers:       R3 = device SYMTAB index  
                  SR2 = symbiont return address  
                  SR3 = device context block address  
                  SR4 = symbiont end-action exit address  
                  D3 = address of core buffer to be released  
Exit:            B   \*SR4

DATA BASES

SYMTAB - symbiont tables (Section VI. 01)  
Context block (Section VI. 03)

SUBROUTINES

Internal

SRELBUF - is used to release a physical core buffer. If both halves of the physical page become available the page is released to the physical page pool.

Entry:           BAL, SR4   SRELBUF  
Registers:       D3 = buffer address  
Exit:            B   \*SR4

CHKFREE - is used to search for symbiont buffers which may be released to the monitors free physical page pool.

Entry:           BAL, 3   CHKFREE  
Registers:       D3 = buffer address  
Exit:            B   0, 3    NONE TO REL  
                  B   1, 3    PHYSICAL PAGE IS TO BE REL

UTS-C01 TECHNICAL MANUAL

DESCRIPTION

The core buffer indicated in D3 is released.

DATA BASE

SPOOL - Head of available symbiont buffers  
MB:PPUT - Physical page link table  
M:FPPH - Head of physical page chain  
M:FPPT - Tail of physical page chain  
M:FPPC - Count of available physical pages

UTS-C01 TECHNICAL MANUAL

ID

SGC - Symbiont Ghost Communication (in module SACT)

PURPOSE

To control all communication between the monitor and the symbiont ghost (RBBAT) and thus all symbiont file access.

USAGE

SGCQ - Queue a communication  
Entry: BAL, R4 SGCQ  
Registers: D1, D2, D3 (R12-R14) three word argument block  
all registers preserved  
Output: R1 communication buffer used word address, and  
message passed to RBBAT.  
Returns: B 1, R4 NORMAL  
B 0, R4 no communication buffer available

SGCQ2 - Queue a double communication  
Entry: BAL, R4 SGCQ2  
Registers: D1, D2, D3, D4 (R12-R15) four word argument block  
R0, R1, R12, R14 Volatile  
Action: Two communication buffers are passed to RBBAT  
D1, D2 to comm. buff 1, words 1, 2  
comm. buff 1 word 3 points to second comm. buff  
D3, D4 to comm. buff 2 words 2, 3  
Output: R1 is second comm. buff. word address  
Return: B 1, R4 NORMAL  
B 0, R4 no communications buffer available

SGCR - Release a communication buffer  
Entry: BAL, R4 SGCR  
Registers: R0, R1, D1 (R12) Volatile  
Action: Buffer at word address in R1 is released  
Return: B 0, R4

UTS-C01 TECHNICAL MANUAL

SGCRA - Alternate communication buffer release.  
Entry: BAL, R4 SGCRA  
Registers: R0 Volatile  
D1 (R12) word address of buffer to release  
R1 buffer displacement byte of that buffer  
Return: B 0, R4

SGC:NCB - No-communication buffer when required by user  
Entry: B SGC:NCB  
Registers: SAVES ALL REGISTERS  
Action: REPORT E:NSYMF for user via T:REG  
Return: B -1, R4 AT E:SYMF reactivation

DATA BASES

SGCHD - Word containing heads of free and active communication chains and tail of active chain (free, TAIL, 0, ACTIVE 8, 8, 8, 8, ).  
SGCBUF - Base address of contiguous linked area of three-word communication buffers.

DESCRIPTION

These are subroutines called by KEYIN, COOP, T:JOBENT, INSYM, OUTSYM, etc. performing one of two functions: pass information to RBBAT, and release the message area after receiving information from RBBAT. To pass to RBBAT, a check is made to determine if enough buffers are available, these buffers are acquired, the message is stored, and RBBAT is awakened. If buffers are not available, the symbiont must wait on the activate queue and try later, or the user must wait on the event E:NSYMF. When RBBAT makes a buffer available, E:SYMF is reported. Releasing buffers is simply management of the free buffer chain. See Section VI.01 for various message formats.

UTS-C01 TECHNICAL MANUAL

ID

SYMCOM      Operator - symbiont communication

PURPOSE

To allow the operator to communicate with any symbiont device.  
'KEYSUB' is the name of the source and ROM files for this module.  
The module is located in the KEYIN overlay.

USAGE

ENTRY:            BAL, SR4 SYMCOM  
REGISTERS:        R2 = DCT table index  
                  D1 = signal character  
EXIT:             B \*SR4

INPUT

An operator communication keyin of the form: !Syyndd, s  
where:

S      specifies a symbiont  
yy     specifies the device type  
n      specifies the device IOP designation  
dd     specifies the symbiont device number  
s      specifies the signal character (I, X, L, R, S, C)

DATA BASES

DCTX          device table index (Section VG.01)  
SYMTAB        symbiont tables (Section VI.01)

SUBROUTINES

IOCQUEUE is used to queue messages for the OC device (Section DA)  
SAQNSERT is used to queue the symbiont (Section GA.04.01)  
SYMTABCK is used to determine if a given device is in the symbiont  
table (Section GA.04.01)

OUTPUT

Messages that may be typed to the OC device:  
yyndd SYMB      NOT ACTIVE  
yyndd SYMB      ACTIVE  
yyndd SYMB      NOT SUSPENDED

UTS-C01 TECHNICAL MANUALERRORS

An undecipherable keyin results in the following error notification to the OC device  
 cr EH? cr

DESCRIPTION

The symbiont device designation is checked for validity. If not valid, the keyin error message is typed.

SYMCOM takes one of three possible actions depending on the input signal character and the current state of the symbiont:

1. Store input signal character in SSIG
2. Store input signal character in SSIG and queue the symbiont for activation via SAQNSERT
3. Output one of the informational error messages (see OUTPUT above)

The following table gives the action for each of the possible combinations:

SSTAT SSIG	Inactive 0	Suspended 0 S	Active 1	Locked 2
Input 1 SSIG	Set SSIG and queue for activation	Error 1	Error 1	Set SSIG and queue for activation
S	Error 0	Error 0	Set SSIG	Error 0
X	Error 0	Set SSIG and queue for activation	Set SSIG	Error 0
L	Set SSIG	Set SSIG and queue for activation	Set SSIG	Error 0
C	Error 0	Set SSIG and queue for activation	Error 2	Error 2
R	Error 0	Set SSIG and queue for activation	Error 2	Error 2

- Error 0 - yyndd symb not active  
 Error 1 - yyndd symb active  
 Error 2 - yyndd symb not suspended

UTS-C01 TECHNICAL MANUAL

The symbiont routines check at various points whether a relevant signal character has been input and takes the appropriate action.



UTS-C61 TECHNICAL MANUAL

ID

SYMSUBR - Symbiont Subroutines

PURPOSE

SYMSUBR contains subroutines used by other symbiont and cooperative modules. SYMSUBR also defines the value of the context block parameters SCJOBX, SCGBX, SCGINFOX, SCBINFOX, SCDCBX, and CDLINK (Section VI.03). ADDFEND and SAQNSERT are alternate entry points for SQINSERT.

DATA BASES

SYMTAB - symbiont tables (reference section VI.01)  
Context block (reference section VI.03)

SUBROUTINES

CNXTSET - Set up and clear context block. The 40-word context block is zeroed except words two and three which contain the Stack Pointer Doubleword for the printer recovery stack which is located at word four.

ENTRY: BAL = SR4 CNXTSET  
REGISTERS: D3 = context block address  
EXIT: B SR4

SYMTABCK - This routine checks if the device specified in R2 is a symbiont device.

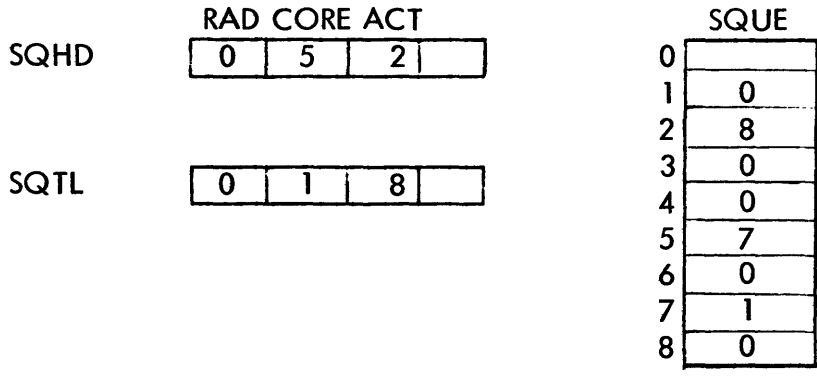
ENTRY: BAL, SR4 SYMTABCK  
+1 return if symbiont device  
+2 return if not a symbiont device  
REGISTERS: R2 = DCT index (DCTX)  
OUTPUT: R3 = SYMTAB index is symbiont device

SQINSERT - Insert Symtab index into specified symbiont queue.

ENTRY: BAL, SR4 SQINSERT  
REGISTERS: R2 = queue index  
0 for RAD request queue  
1 for core buffer queue  
2 for activate queue  
R3 = SYMTAB index  
EXIT: B \*SR4

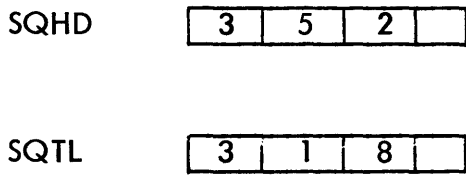
UTS-C01 TECHNICAL MANUAL

The SYMTAB index for a device may be placed in one of three queues to cause the device to wait for either RAD space, a core buffer, or activation. Each queue consists of a pointer to the (SYMTAB) entry at the head of the queue (SQHD), a pointer to the tail of the queue (SQTL), and the queue itself which consists of entries in SQUE that have the index to the next entry in the queue. An entry of zero indicates the end of the queue. SQHD and SQTL are byte-indexed by 0, 1, or 2 to obtain the head and tail of the RAD, core, or activate queues, respectively. For example:



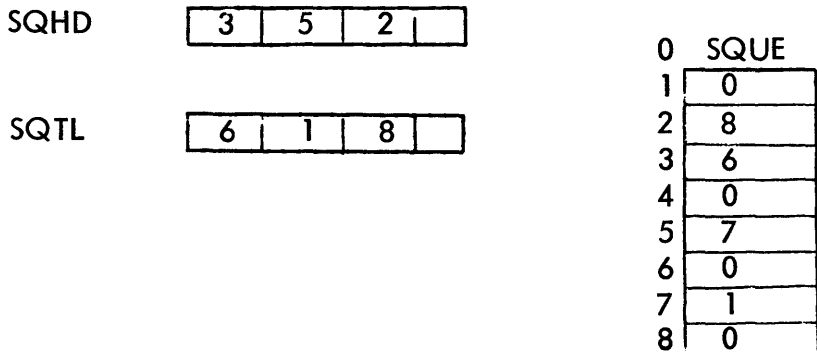
RAD queue: null  
 Core queue: 5 7 1      Head = 5 → 7 → 1 = Tail  
 Activate queue: 2 8      Head = 2 → 8 = Tail

Adding Symtab index 3 to the RAD queue would result in:



and no change to SQUE.

Adding Symtab index 6 to the RAD queue results in:



UTS-C01 TECHNICAL MANUAL

- ADDFEND - I/O end action address specified in ADDF (Section GA.02.01). When the I/O operation requested by ADDF is completed, end action control comes to ADDFEND which queues for activation the symbiont that made the call on ADDF.
- ENTRY: LI, SR2 ADDFEND  
B QUEUE1 I/O with end action
- REGISTERS: SR3 = context block address
- SAQNSERT - Queue symbiont for activation. This entry loads R2 with the activate queue index and falls into SQINSERT.
- ENTRY: BAL = SR4 SAQNSERT
- QFP11NCB - Queue for RBBAT communication buffer. This routine uses the context buffer area SCDCBX+10 to save registers SR2, D1, D2, D3 before queuing the symbiont.
- ENTRY: BAL, R4 QFP11NCB
- REGISTERS: SR2, D1, D2 and D3 preserved; SR3 = context block address
- OUTPUT: SR2, D1, D2 and D3 restored from context block and control returned to BAL+1  
R1 contains context block address
- EXIT: BAL+1
- QFACTP11 - Is used to pull callers return address from TSTACK and queue symbiont for re-activation.
- QFACT - Assumes that callers return address is in R11. It then queues the symbiont for re-activation
- QFACTP1 - Assumes that callers return address is in R11 and that symbiont resource queue index is in R2. It then queues the symbiont for re-activation.
- SYMCO }  
QFCBP11 } - Is used to pull callers return address from TSTACK and queue symbiont for core buffer
- SYMDQ }  
QFGP11 } - Is used to pull callers return address from TSTACK and queue symbiont in disc queue

UTS-C01 TECHNICAL MANUALID

SUB - Other miscellaneous

PURPOSE

SUB contains subroutines used by the monitor, MBS and the symbiont

DATA BASES

SSIG - Symbiont signal

SUBROUTINES

## Internal

REG - using register SR3 load symtab index into R3; Symbiont base address into R4; Symbiont signal character into D2; and context block address into R1.

ENTRY: BAL, SR2 REG

OUTPUT: R1 - Symbiont base address; R3 Symtab index; R4 Symbiont base address; D2 Symbiont Signal.

EXIT: B \*SR2

UTS-C01 TECHNICAL MANUALID

SUSPTERM - SUSP suspend symbiont, TERM terminate symbiont

SUSP:PURPOSE

To cause a symbiont to suspend operation on the specified device and type the suspend message on the OC device.

USAGE

ENTRY: B SUSP  
REGISTERS: R3 = SYMTAB device index  
SR2 = symbiont entry address  
EXIT: Symbiont exit address = last word pushed in TSTACK

DATA BASES

SYMTAB symbiont tables (Section VI. 01)

SUBROUTINES

RSC is used to release a core buffer (Section GA. 02. 01)

INTERACTIONS

OCQUEUE is used to queue messages for the OC device (Section DA)

OUTPUT

The following message on the OC device

Syyndd SUSPENDED

DESCRIPTION

On being entered by the symbiont, SUSP marks the symbiont as suspended on the specified device by setting the appropriate entry in SSTAT, a sub-table of SYMTAB, inactive. SUSP sets the symbiont entry address into the appropriate entry in SRET. When the operator issues a 'continue' keyin on this symbiont, execution will resume as the specified symbiont entry point.

UTS-C01 TECHNICAL MANUAL

TERM:

PURPOSE

To terminate symbiont operation of the specified device.

USAGE

ENTRY: B TERM

REGISTERS: R3, SYMTAB  
SR3 = context block address

EXIT: Symbiont exit address = last word pushed in TSTACK

DATA BASES

SYMTAB symbiont tables (Section VI.01)

SUBROUTINES

RSC is used to release a context block (Section GA.02.01)

DESCRIPTION

The symbiont context block address is cleared from the appropriate entry in SCNTXT, a sub-table of SYMTAB. The symbiont context block is released via RSC.

Symbiont status is set to inactive, SSTAT (x) = 0 and the signal character table entry is cleared, SSIG (X) = 0.

ID

Symbiont File Handling - RBBAT

OVERVIEW

RBBAT (say Rabbit) is a system initiated ghost job which manages symbiont input and output files and multi batch scheduling. Remote batch control is also a function of RBBAT but this function is discussed in another section.

When it is first called, RBBAT takes master mode which it retains throughout its operation. Once it is logged on it never exits but instead reports a sleep event to the scheduler when it has nothing left to do. New tasks activate RBBAT with a wakeup event via T:GJOBSTRT. Since the Rabbit is master mode it retains control when it is processing and, in general, will not be swapped when awake. The great majority of its I/O is to the operators console, and this is done from monitor buffers via NEWQNW no wait. RBBAT does occasionally read and write disc, and these I/O's are done via NEWQ with wait so it is possible the Rabbit will be swapped there.

All of the symbiont file and multi batch scheduling tables are contained within RBBAT and these do not require resident core. This allows the input and output queues to be larger than was possible when the tables were resident. The tables cannot be accessed from the monitor, however, and RBBAT must be called for all operations on them. The argument that accesses to these tables are relatively few and far between is the motivation for RBBAT.

During the course of a normal batch job the Rabbit is involved four times:

1. The input symbiont or JOBENT calls RBBAT to check the input symbiont file and insert it in the input queue.
2. Sometime later the multi-batch scheduler in RBBAT selects the job for execution and moves its information to core.
3. When the job completes the output COOP calls RBBAT to add the output file(s) to its tables.
4. The output symbiont calls RBBAT to request an output file and the job is output.

UTS-C01 TECHNICAL MANUAL

On-line jobs may require the Rabbit for several reasons:

1. The output COOP calls RBBAT to add on-line output.
2. The TEL !JOB command calls JOBENT which calls the Rabbit for the status of a batch job.
3. The TEL !CANCEL command calls JOBENT which calls RBBAT to delete an input file from its tables.

RBBAT is also called by KEYIN to handle the PRIO, DELETE, and DISPLAY keyins.

In operation RBBAT consists of several special purpose routines accessing a common data base, and this document will take the same approach to describe it.

DATA BASES

SYMBIONT FILE and MULTI-BATCH TABLES.

The symbiont file and multi-batch scheduling tables have names of the form BX: WOW where X is the size (Byte, H.W., Word) and WOW is the description. They will be referred to here as the BX tables.



These tables are chained by priority so that searches need not include every entry. Some of the tables exist for both input and output and others for input only.

BH:HPRI and BH:TPRI

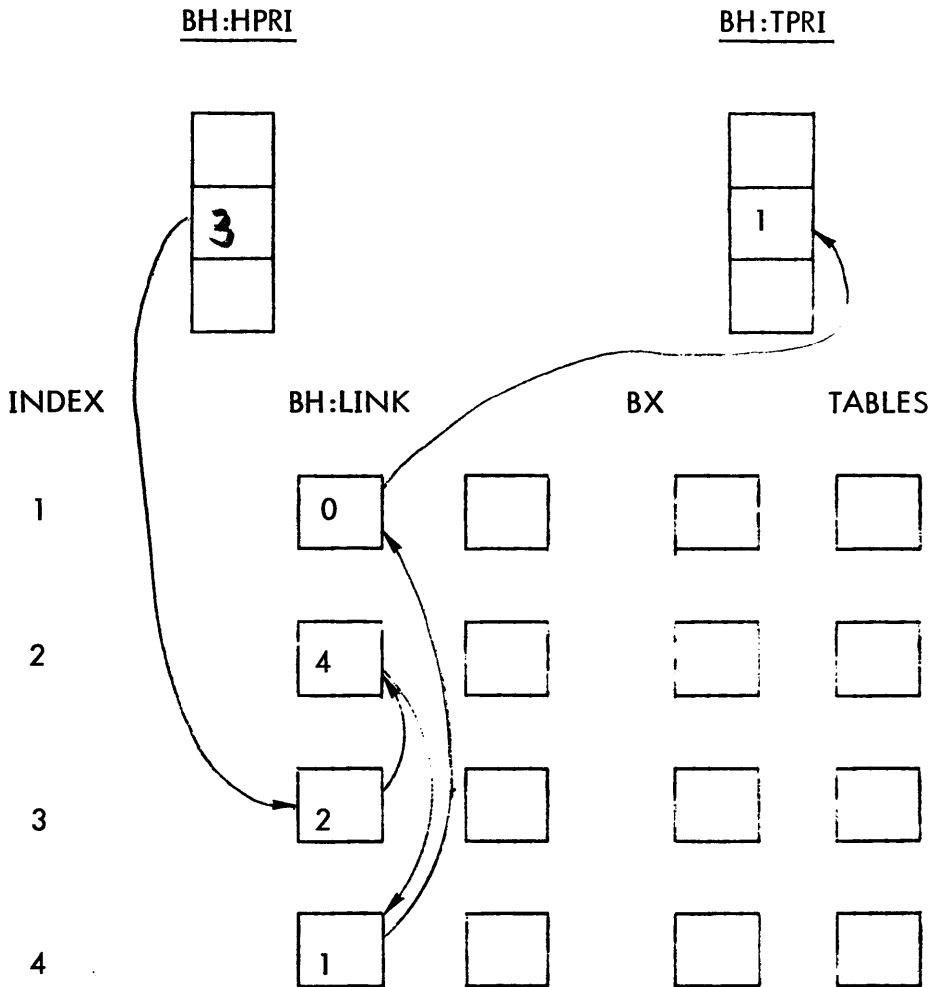
These are parallel tables indexed by priority which contain the head and tail respectively of the chain for the priority used as an index. The priorities are:

<u>PRIORITY (HEX)</u>	<u>NAME</u>	<u>DESCRIPTION</u>
0-F	FIPRI (F)	Input files of priority 0-F
10	RUNPRI	Jobs currently running
11-1F	FOPRI (1F)	Output files of priority 1 (11) through F (1F)
20	MFPRI	Special output files called "message files" created by RBBAT
21	DELPRI	Symbiont files queued to be deleted by the delete symbiont.
22	FREI	Free input file slots
23	FREO	Free output file slots

The heads and tails in BH:HPRI and BH:TPRI are the indices into the BX tables. A head of zero means there are no entries for this priority.

BH:LINK

BH:LINK exits for both input and output files and is the link to the next entry for this priority. If BH:LINK is zero for an entry, it is the tail of a priority chain.



While input and output slots share some tables and may both be found on the delete chain they are actually separate, with output slots always having higher indices than any input slot.

BH:SID

This table contains the sysid for batch input and output files.

BW:SDA

This table contains the starting disc address for both input and output files. Byte zero is used to contain several flags used to process the file. They are:

<u>VALUE</u>	<u>NAME</u>	<u>DESCRIPTION</u>
Y8	ORDBIT	ORDER was specified on the !LIMIT command
Y4	ACCTBIT	ACCOUNT was specified on the !LIMIT command
Y2	INBIT	This is an input entry on the delete chain.
Y1	XHOLDBIT	Applies only to Remote batch
Y08	MFBIT	This is a message file.

BB:RID

This table contains the RBID of the input or output file. For non Remote Batch systems it is always zero.

BB:DEV, BB:PI

These are actually two names for the same tables. For input files it is BB:PI which is a counter of the partial priority acquired by a job when it cannot be scheduled. (See Multi Batch Scheduler). For output files it is BB:DEV and contains the device type (5 = CP, 6 = LP, etc.) of the file.

The rest of the BX tables exist for input files only.

BD:ACCT

BD:ACCT contains the account from the job card (left adjusted, blank filled) which is used for the ACCOUNT option and JOBENT deletes.

BH:TIME

This table contains the TIME from the !LIMIT command.

BW:RES

This table contains 7 bits each of SPindles, 7T rack tapes, 9T rack tapes and K Core from the !LIMIT command in the form:

GEN, 1, 7, 1, 7, 1, 7, 1, 7    D, SP, D, 7T, D, 9T, D, C

D if set means this is a default.

BH:PART

This table contains one bit for each of the 16 possible partitions in the system. If the bit is set the resources required for this job are satisfied by the partition.

BH:SLNK and BH:XLNK

These tables contain the heads of the shared and exclusive private pack serial number chains respectively for this job. The heads are indices into the serial number tables S#H:LNK and S#W:SER.

S#H:LNK

This table contains the links for the serial number chains for the batch jobs. Entry zero is the head of the free chain. S#H:LNK is to BH:XLNK and BH:SLNK as BH:LINK is to BH:HPRI.

S#W:SER

This table is parallel to S#H:LNK and contains the actual serial numbers of private packs.

COUNTERS

BL:IFS, BL:OFS

These are counters in the monitor of the number of input and output free slots respectively remaining in the ghosts tables.

S:BFIS

S:BFIS is the monitor count of the number of input files in the system. RBBAT increments it when it adds an input file and decrements it when it schedules a batch job or deletes an input file.

GETI TABLES

The GETI tables have names of the form GIX: WOW will be referred to as the GI tables. They are tables three entries long used to pass data on scheduled input files to the input COOP. For a description see section on GETI.

UTS-C01 TECHNICAL MANUAL

GHOST COMMUNICATION

Requests are communicated to the ghost in three word buffers chained through byte zero of their first word. The heads of the chains are in the word SGCHD which appears as follows.



HOF = head of the free chain  
 HOU = head of buffers waiting for RBBAT  
 TOU = tail of buffers waiting for RBBAT

The links are displacements to the buffer from SGCBUF which is currently EQUed to SGCHD. The heads and buffers are, of course, resident.

OPERATION

CONTROL

CONTROL is the section of RBBAT which is the first entered when the Rabbit is awakened. Using SGCHD it obtains and frees a combuf queued for the ghost and obtains byte 2 of word 0 into R5 (usually a DCTx) and the address of the communication buffer into R7. It then obtains the ghost function code (GFC) from byte 3 of word 0 and enters a processing routine (listed below) based on the GFC. The processing routine may return to CTR1 to release the combuf to the free chain or CTR4 to keep it loose. In either case CONTROL then gets the next combuf and repeats the above process until all combufs have been exhausted. When this occurs it disables and reports E:SL to the scheduler via T:REG.

RBBAT PROCESSING ROUTINES

<u>GFC</u>	<u>ROUTINE</u>	<u>DESCRIPTION</u>
0	CTR1	NOP - RELEASE BUFFER ONLY
1	AIF	ADD INPUT FILE
2	AIFJE	ADD INPUT FILE (JOBENT)
3	AOF	ADD OUTPUT FILE (BATCH)
4	AOFNB	ADD OUTPUT FILE (NON-BATCH)
5	AOFP	ADD OUTPUT FILE (PARTIAL)
6	GOF	GET OUTPUT FILE
7	MBSCALL	MULTI-BATCH SCHEDULE
8	KPRIO	PRIORITY KEYIN
9	KDEL	DELETE KEYIN

UTS-C01 TECHNICAL MANUAL

10	KDISP	DISPLAY KEYINS
11	JESTAT	JOBENT STATUS
12	JEDEL	JOBENT DELETE
13 - 18	REMOTE BATCH ONLY	
19	RCVRY	RECOVER
20	RCVRY 1	POINT OF RECOVERY

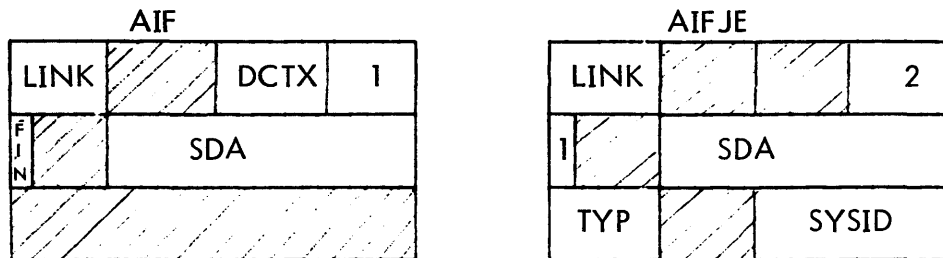
ROUTINE

AIF, AIFJE

Add input file

CALLED BY

INSYM, JOBENT

COMMUNICATION BUFFER

DCTX = DCT index of input device

SDA = Starting disc address

FIN = 1 if last job of stream

TYP = 0 = BATCH, 1 = TERMINAL, 2 = PROCESSOR

DESCRIPTION

If the entry is made to AIFJE the DCT index is set to LCLX a special local DCT index and the SYSID and TYP is obtained from the combuf. If entry is to AIF the DCTX is obtained from the communication buffer (we assume it is not remote) the TYP is zero and the SYSID is obtained with a BAL to NXTSID. Otherwise these routines are identical.

A free input file entry is obtained and the SDA is saved in it. Then a 256 word buffer is obtained, and, using the SDA, the first sector of the symbiont file is read into it. The first command in that sector must be !JOB or the operator is informed that the JOB command is missing, the entry is moved to the delete chain and the delete symbiont started to delete it.

UTS-C01 TECHNICAL MANUAL

When a !JOB command is found a 256 word message file buffer is obtained and initialized to be a symbiont output file block. If the job should be aborted because of !JOB or !LIMIT errors, this message file will be written to disc and added as an output file, and the input file will be deleted. The user is thus informed of the error without the job ever having been run. A burst of the job card surrounded by asterisks is put into this message file as well as a line giving the SYSID. As the !JOB and !LIMIT cards are processed they are listed in this file and flagged if errors occur. If the job is accepted the message file buffer is simply released.

After the message file has been started the BX tables are initialized with the system defaults and the SYSID, and the RBID which is obtained from the DCTX (in this case 0). Then the !JOB card is scanned and checked for errors. During this process the account is put in BD:ACCT, and the priority is saved to link the entry later. The job message is typed to the operator using TYP to determine which message. Each !LIMIT command is scanned and error checked, and the resource values replace the defaults in the BX tables as follows:

<u>OPTION</u>	<u>TABLE</u>	<u>BYTE/BIT</u>
7T	BW:RES	1
9T	BW:RES	2
TIME, TI	BH:TIME	-
CORE, CO	BW:RES	3
SP	BW:RES	0
ORDEr, OR	BW:SDA	Y8
NORDEr, NO	DEFAULT	
ACCOunt, AC	BW:SDA	Y4
MOUNt, MO		
9T, 7T	IGNORED	
X, S	LINKED INTO CHAIN FROM BH:XLNK, BH:SLNK to S#W:SER	

If the !JOB and !LIMIT commands are scanned without errors the entry in the BX tables is attached to the tail of the chain of the saved priority, S:BFIS is incremented, and MBSOP (see MBS) is called to set the appropriate bits in BH:PART. A call is made to MBS to see if this job can be run immediately, the input buffer is released, and AIF (AIFJE) exits to CTRL.

ROUTINE

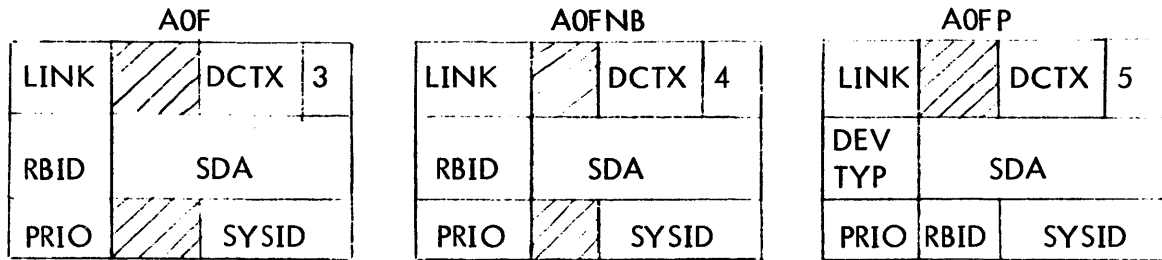
AOF, AOFNB, AAFP

Add output file.

CALLED BY

COOP, JOBENT, OUTSYM

COMMUNICATION BUFFER



PRIO = Priority  
DEV TYP = Device type - 5 = CP 6 = LP, etc.

DESCRIPTION

All of these routines add the specified output file to the BX output tables in RBBAT. The simplest case is AOFNB (add output file - not from batch - e.g. online, jobent.) In this case the buffer is parsed and the information placed in a BX table entry which is chained to the tail of the appropriate priority. The device type is obtained from DCT4 and all devices of this type are checked for activity. If one is found inactive its symbiont is started via SAQNSERT.

AOFNB then exits to CTR1. AOF P (add output file - partial) is the same as AOFNB except that the device type is obtained from the combuf (the reason applies only to Remote Batch). AOF P is used by OUTSYM to replace a partially output file in the tables as a result of the 'Q' signal. AOF (add output file from batch job) differs from the other two in that the running chain is searched for a file of the same SYSID, and if one is found it is removed from the chain and its serial number entries released. Once this has occurred the processing is identical to AOFNB. The special processing of AOF is explained by Remote Batch which requires flags to be passed from input to output files and recovery which demands that the serial number entries be kept intact until the job has completed.



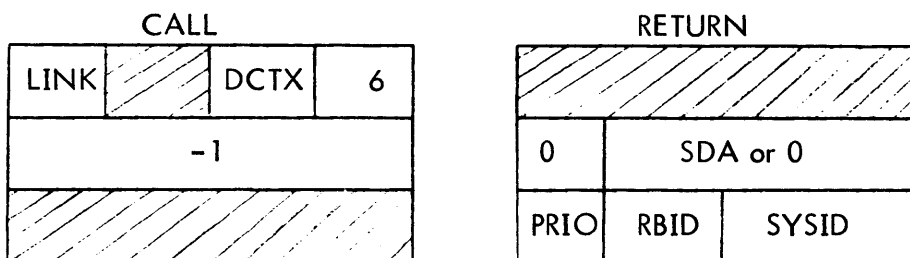
ROUTINE

GOF  
Get output file

CALLED BY

OUTSYM

COMMUNICATION BUFFER



DESCRIPTION

When an OUTSYM is initially entered by SACT it discovers it has no file to output and makes a GOF call to RBBAT to obtain one. This is done by making the GOF call through SGCQ then putting the combuf number in the top byte of SRET and the address to continue when the file is obtained in the rest of SRET. OUTSYM then exits to SACT with SSIG intact. When the Rabbit has found a file it puts the parameters back into the caller's combuf, unchains it completely and restarts the symbiont via SAQNSERT. The return is to the address in SRET where the buffer is emptied and released.

GOF in RBBAT checks first to see if the DCT index is zero and if so selects the head of the delete chain and passes the disc address from BW:SDA to the delete symbiont. If the DCTX is non-zero, the output priority chains are searched from highest to lowest priority for a file of the right device type (and in the local case RBID = 0). When one is found the file information is moved to the combuf (note the similarity between GOF return and A0FP), and the entry is released. SAQNSERT is then used to restart the symbiont. If no file can be found for a symbiont, zero is returned as the SDA, causing OUTSYM to terminate. The -1 in the GOF call is for recovery purposes. GOF always exits to CTR4 to avoid releasing the combuf.

UTS-C01 TECHNICAL MANUAL

ROUTINE

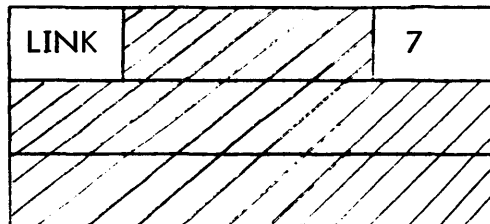
MBSCALL

Request a multi-batch schedule

CALLED BY

CLOCKP (T:BTSCHE)

COMMUNICATION BUFFER



DESCRIPTION

When T:BTSCHE determines it is possible to schedule a new batch job it calls RBBAT to do an MBS with this call. MBSCALL simply BALs to MBS and exits to CTRL.

ROUTINE

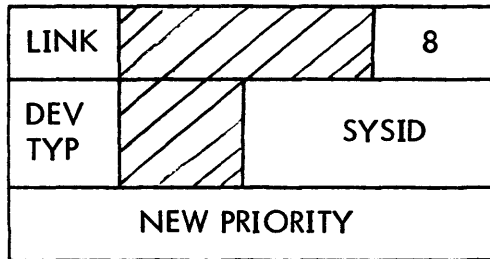
KPRIO

Process PRIO Keyins

CALLED BY

KEYIN

COMMUNICATION BUFFER



DEV TYP - 0 = ALL, 4 = CR, 5 = CP, 6 = LP

DESCRIPTION

KPRIO searches the appropriate chains based on the DEVTYP (DEVTYP 4 is made 1 for convenience). When an entry is found with the correct SYSID and DEVTYP (input files are by definition CR), it is unlinked and chained back in at the priority specified in the combuf. If the priority is changed on an input file its BB:PI is set to zero and S:MBSF is set to cause an MBS. When its search is completed KPRIO returns to CTRL.

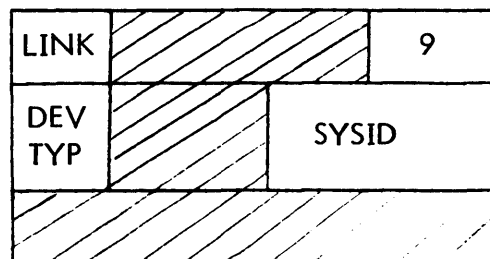
ROUTINE

KDEL  
Process DELETE keyins

CALLED BY

KEYIN

COMMUNICATION BUFFER



DESCRIPTION

KDEL searches the chains as in KPRIO above. When an appropriate entry is found it is unlinked and chained to the delete chain and the delete symbiont is initiated.

UTS-C01 TECHNICAL MANUAL

ROUTINE

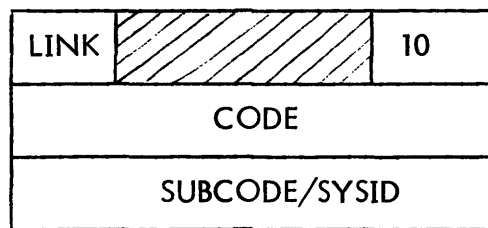
KDISP

Process DISPLAY keyins

CALLED BY

KEYIN (DISPLAY)

COMMUNICATION BUFFER



<u>DISP OPTION</u>	<u>CODE</u>	<u>SUBCODE</u>
(NONE)	0	0
OC	0	-5
NORUN	-1	-5
(SYSID)	1	SYSID

DESCRIPTION

Only the display keyins listed above are processed by RBBAT. All others are handled by DISPLAY internally. Using the CODE and SUBCODE, KDISP determines what to display and where. All displays go to the OC except DISP (NO-OPTION) which is sent to the line printer. Headings are printed as they are needed. If the option was (SYSID) and the sysid is running or non-existent the operator is told; for the other options he is told if there is nothing to display.

For DISP (NO OPTION) and DISP OC, all the chains except free and delete are searched and each entry is displayed. If there was no option the display is in the form of a message file output to the printer by the symbiont; otherwise all output is to the OC via NEWQNW.

For DISP NO RUN only input chains are search and all entries with BH:PART = 0 (no valid partitions) are displayed on the OC. For DISP (SYSID) all the chains below delete are searched and all entries with the appropriate SYSID are displayed on the OC. When KDISP completes it exits to CTRL.

ROUTINE

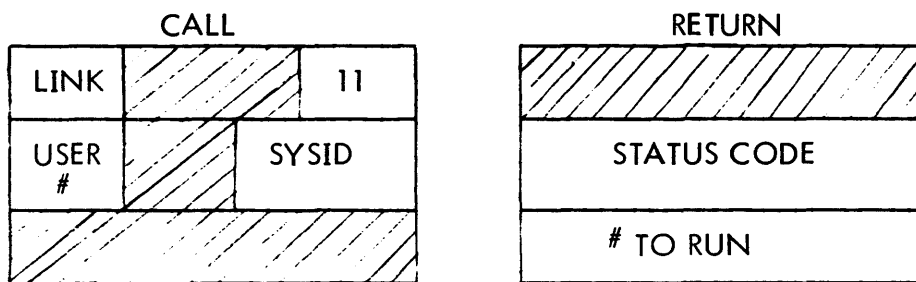
JESTAT

Process JOBENT status request

CALLED BY

JOBENT

COMMUNICATION BUFFER



USER # = USER NUMBER OF REQUESTING USER  
 STATUS CODE - 0 = COMPLETED, 1 = RUNNING, 2 = WAITING TO RUN,  
 4 = WAITING TO OUTPUT  
 Code 3 (Doesn't exist) is checked by JOBENT

DESCRIPTION

When JOBENT receives a status request CAL it first checks to see if the SYSID specified exceeds S:USID. If this is true it returns code 3 to the user indicating that the SYSID doesn't exist. If the SYSID exists he places a status call to the Rabbit in a combuf via SGCQ and REGs the user to sleep with E:QA.

When RBBAT receives the request it searches the chains for the SYSID and places the appropriate code back into the callers combuf. If the SYSID is online the status applies to any output he may have created. If the code is 2 the number ahead is determined by counting entries in a search from priority F to the requested SYSID and this count is placed in the combuf. RABBIT then reports E:UQA on the user via T:RUE and exits via CTR4 leaving the combuf unchained.

When the user is wakened JOBENT removes the information from the combuf, releases it, and passes the information to the user.

ROUTINE

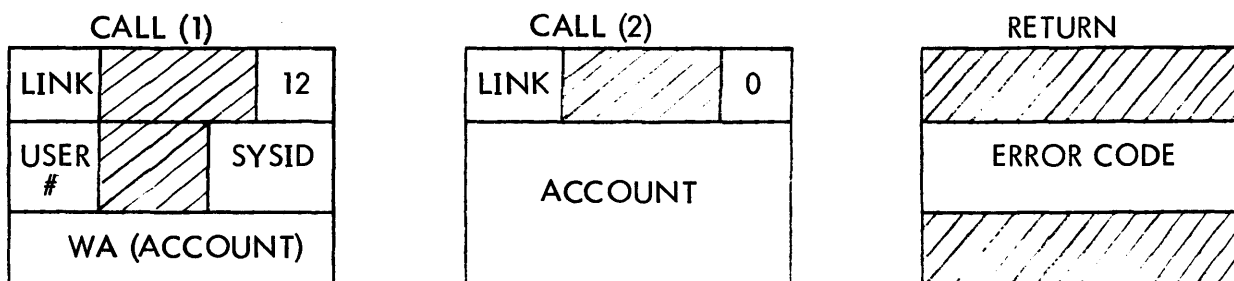
JEDEL

Process JOBENT delete request

CALLED BY

JOBENT

COMMUNICATION BUFFER



ACCOUNT = USERS ACCOUNT FROM JIT

ERROR CODE - 0 =DELETE DONE

X'3A' = SYSID NOT THAT OF INPUT FILE

X'39' = WRONG ACCOUNT ON FILE

DESCRIPTION

The communication between JOBENT and RBBAT is performed using E:QA-E:UQA as in JESTAT above. Two combufs are obtained by JOBENT using SGCQ2 in order to have room for the account. The second buffer has GFC = 0 and is released by the ghost when it is encountered.

JEDEL searches the input chains for the given SYSID, and if it is not found returns code X'3A'. If the SYSID is found the account in BD:ACCT is checked against that in the combuf, and if they do not compare code X'39' is returned. If the accounts are identical the file is deleted and code 0 is returned.

All of the other calls to RBBAT are involved either in Remote Batch processing or Rabbit recovery, which are described elsewhere.

UTS-C01 TECHNICAL MANUAL

ID

MBS - Multi-Batch Scheduler

ENTRY

BAL, 11 MBS  
for Multi Batch Schedule  
BAL, 11 MBSOP  
to set BH:PART bits based on resource values

MBSOP is called by AIF in RBBAT when a new job has just been added. MBS is called by AIF in RBBAT when a new job is added and by MBS CALL when T:BTSCHEd issues an MBS call to the ghost. MBS is located in RBBAT.

DATA BASES

BX tables - see SYMBIONT FILE HANDLING  
GI tables - see ADD  
PARTITION tables - see section (VI.02)  
RESOURCE tables - see section (VI.02)  
AVR tables - see section (VI.02)

DESCRIPTION

The first action taken by MBS is to check two cells in core - PL:LK and PL:CHG. If the first is set, CONTROL is in the process of modifying the partition limits and MBS exits. If bits are set in PL:CHG MBS calls its subroutine REPART to modify the bits in each file's BH:PART that control has marked in PL:CHG as having their partition attributes changed.

When this step is completed MBS does the following calculation to determine if there may be a batch job available to schedule:

$$\left( \left[ S:BUAIS \right] - \left[ S:BUIS \right] \right) * S:BFIS$$

If the result is not positive MBS exits. The next step is to check GI:FRE. If it is zero there is no space in the GETI tables and MBS sets S:MBSF to assure another try in 5 seconds and exits.

When all of these tests are successful MBS will try to schedule a batch job. A user number is obtained and restored in PL:CHG. This accomplishes the dual purpose of saving this number and informing CONTROL that an MBS is in progress. (MBS will be used here both as the routine name and as a mnemonic for multi-batch-schedule). From this point on, any unsuccessful exit from MBS will clear PL:CHG and release the user number.

MBS builds a bit map in R15 with a bit on for each partition which is neither running nor locked and whose minimum resources are still available in the system. This amounts to a list of partitions ready to accept a job. At the same time a dummy BW:RES is built containing the maximum of each resource available. If there are no partitions which can accept a job MBS exits.

The highest priority job is selected as the "candidate" job (CJ) first. Whenever a CJ cannot be run and has priority X'F' MBS checks if it is possible to ever run him and if so exits.

All currently running jobs and the CJ are checked to see if ACCOUNT restrictions prevent running the CJ, and all queued jobs to see if ORDER restrictions do so. If either prevent running him a new CJ is selected. The next check made is to see if the CJ requires more resources than are available on the system. The CJ's BW:RES values are compared with the maxima saved earlier and, if they are greater, a new CJ is selected. (It should be mentioned that when a CJ is rejected his BB:PI is incremented by a system manager set value and when BB:PI reaches 256 it is zeroed and the job is advanced one priority).

The shared serial numbers specified in the job are searched for in AVRTBL and if not found the SP resource value is incremented. If a serial number is found already in exclusive use the CJ is rejected. A bit map is constructed with ones corresponding to the expected shared spindles in AVRTBL which will later be stored in GI:ASPN. At the end of this process the SP resource is checked against the maxima calculated earlier and the CJ is rejected if too many are required. The serial numbers of shared packs which are already mounted are not carried into core when the job is run. The final check made in the CJ is whether there is room in LSERIAL, the in-core serial number table to hold his serial numbers. If there is not, the CJ is rejected.

If all of the above tests are passed the CJ will be run. The entry in the BX tables is moved to the running chain. A slot in the GI tables is obtained and the information from the BX tables moved into it. The serial numbers are moved from S#H:LNK and S#W:SER in the ghost to LSERIAL and TSERIAL in core, and the CJ's resource requirements are added to the currently in use counts. A partition is picked by anding his BH:PART with the bits for available partitions determined earlier and



selecting the highest numbered partition whose bit remains set. A DISABLE is executed and the user number previously obtained is returned to the head of the queue so that it will be picked for this job, S:BUI5 is incremented, S:BFIS is decremented, and the batch bit is set in his UH:FLG. MBS then BALs to ADD 1 still disabled to start up the job and returns to the beginning of MBS to try to start up another user.

ID

REMOTE BATCH PROCESSING

OVERVIEW

Remote Batch Processing (RBP) in UTS is accomplished by three separate processes running asynchronously. They are:

DSCIO - is the handler for the 7601(-4)-7670 device combination. It is a normal (if especially complex) device handler request driven by IOQ-SERDEV.

RBSS - is, in effect, the line manager for RBP. It is entered every 5 seconds from CLOCKP and at that time checks the status of each line. New connections and disconnections are handled and appropriate symbionts are started to keep an active line busy.

RBBAT - The symbiont ghost (see SYMBIONT FILE HANDLING) does some special handling of remote files and contains routines to process Remote control commands (RBCC).

These three routines communicate through and are controlled by the bits in a monitor word table RB:FLAG. This table is indexed by DCT index but exists only for those DCT indices which represent RB devices. The limits on these DCT indices are contained in RBLIMS a double word in the monitor constructed such that the following determines whether a DCTX is remote.

	CLM, DCTX	RBLIMS
	BCR, 9	ITS\$REMOTE
or	BCS, 9	ITS\$LOCAL

An understanding of RB:FLAG is critical to the understanding of Remote Batch and therefore it will be discussed first.

RB:FLAG

RB:FLAG contains the following bits:

1. DUPBIT EQU X'8000'

DUPBIT specifies the duplex (1=FULL) of the 7601(-4)(DSC). It is set by sysgen and is always preserved across hang ups and dial ups of RBTS. Full duplex DSCs have two I/O addresses and DSCIO uses DUPBIT to determine whether input and output operations occur on different addresses.

UTS-C01 TECHNICAL MANUAL

## 2. CTRBIT EQU X'100'

CTRBIT is used to disable RBSS when RBBAT is processing information that may change the optimal use of the line. One example of its use is during the input of a job stream. It is set to prevent the scheduling of output before RBBAT can produce a message file which should be the next use of the line. When a DSC has CTRBIT set in its RB:FLAG it is ignored by RBSS entirely.

## 3. ACTBIT EQU X'200'

When ACTBIT is set for an RB device it implies that an RBT is connected and has been logged on. Normal operations may proceed on this line.

## 4. LIPBIT EQU X'8'

If LIPBIT is set for a DSC an RBT is connected and is logging on. Symbiont input and output are illegal for this line, and special actions are taken to log the line on. Any line with an RBT connected will have either ACTBIT or LIPBIT set.

## 5. LOFBIT EQU X'2000'

LOFBIT is used by RBBAT to record the fact that on !RBDISC RBCC has been received from the terminal. LOFBIT is changed to DISCBIT when the job stream that contained the !RBDISC has been completely read in.

## 6. DISCBIT EQU X'4000'

DISCBIT directs DSCIO to translate any read for input order into a disconnect order. When an !RBDISC is received all output is sent before the RBT is disconnected. When RBSS attempts to use the line for input (the lowest priority use) it is disconnected instead.

## RBXBIT EQU X'10000'

RBXBIT directs DSCIO to translate any order into a disconnect order. This bit is used to implement the !RBX and !RBDISC keyins.

## OFFBIT EQU X'20000'

OFFBIT disables RBSS entirely on a line preventing connection of new RBTS. It is used in the !RBX keyin and is never set until the line is disconnected.

## SYSBIT EQU X'1000

SYSBIT is used to record the fact that the work station logged onto a line may use the :SYS account. It is set from the :RBLOG record at logon time.

UTS-C01 TECHNICAL MANUAL

HALBIT EQU X'800'

HALBIT specifies that an !RBHOLD ALL has been received from the RBT and all output files directed to the terminal other than by default are to be held.

MFPBIT EQU X'400'

MFPBIT informs RBSS that message files are pending output to the RBT on this line.

PRPBIT EQU X'40'

PRPBIT informs RBSS that print files are waiting to be output on this line.

PUPBIT EQU X'80'

PUPBIT informs RBSS that punch files are waiting to be output on this line.

PUNBIT EQU Y08

PUNBIT informs RBSS that a BEL signal has been received from the RBT on this line, specifying readiness to receive punched output.

FIABIT EQU X'20

FIABIT is used by RBBAT to record the fact that the stream pending bits (PRP, PUP, PUN) may have been affected by RBCCS in the job stream. These bits will be re-established at the end of the job stream.

HUBIT EQU Y1

HUBIT is used by DSCIO and RBSS to inform the other that a hang up has been detected on the line and reported to RBBAT.

SSSBIT EQU X'10'

SSSBIT is used to allow input with an output symbiont suspended. Each DSC has two slots in the symbiont device tables, the normal and the alternate. Most processing occurs on the normal "channel". When it is running output and is suspended, SSSBIT is set to switch both DSCIO and RBSS to the alternate channel to read input.

FRBIT EQU X1

FRBIT is set by RBSS when an input symbiont is initiated on a line to cause the translation of that symbiont's first read into a polling sequence. DSCIO resets FRBIT when a poll is successful. When a record is received and FRBIT is set DSCIO initializes BPBIT.

BPBIT EQU Y8

BPBIT is the block protect bit toggle. Messages to and from the RBT must have an alternating block protect bit, and BPBIT is used to produce that bit on output and check it on input. BPBIT is toggled each time a record is sent or received.

FINBIT EQU X'4'

FINBIT is set by DSCIO when it encounters a !FIN Command in an input stream. DSCIO continues to read the RBT until its hopper is empty and then gives the FIN to the input symbiont.

IGBIT EQU Y4

If cards are read after FINBIT is set, DSCIO sets IGBIT and ignores them. IGBIT tells RBBAT to inform the RBT that cards were ignored.

MORBIT EQU Y2

If the RBT hopper goes empty before FINBIT is set, DSCIO assumes that it is reading a continued job stream and sets MORBIT. This causes the handler to continue to read the RBT until the rest of the job stream is sent.

EMBIT EQU X'2'

EMBIT is set by the NOEM option on the !RBSIZE command. When it is set DSCIO always sends the full 80 characters of card punch records.

### RBSS

RBSS is entered every 4.8 seconds by CLOCKP and does the following for each RB device:

1. If any of HUBIT, OFFBIT and CTRBIT are set RBSS goes to the next device since it is disabled for this line.
2. The symbiont index (SYMX) is obtained and SSIG is tested. If it is non-zero the line is busy and RBSS goes to the next device.
3. A TDV is done on the device and if it is found disconnected RBSS goes to step 7.
4. If neither ACTBIT nor LIPBIT is set this is a new connection. RBSS sets LIPBIT and CTRBIT and reports remote batch dial up (RBDU) to RBBAT. It then goes to the next device.

UTS-C01 TECHNICAL MANUAL

5. If LIPBIT is set RBSS starts up an input symbiont to read the !RBID command.
6. If ACTBIT is set RBSS will start up a symbiont on the line. If SSSBIT is set RBSS is operating on the alternate symbiont and will always start up input. Otherwise a symbiont is selected by the following priority scheme (a symbiont is started only if its stream pending bit is set):

<u>STREAM (PRIORITY ORDER)</u>	<u>PENDING BIT(S)</u>
MESSAGE FILES	MFPBIT
PUNCH	PUPBIT and PUNBIT
PRINT	PRPBIT
INPUT	default

When RBSS starts a symbiont on a line it sets SYMX to input or output as required and DCT 4 to the appropriate device type. If the symbiont is input FRBIT is set. After the symbiont is initiated via SAQNSERT, RBSS goes to the next device.

7. When a device is found disconnected by RBSS, it checks ACTBIT and LIPBIT and if neither are set goes to the next device. If either are set, RBSS reports remote batch hang up (RBHU) to RBBAT and goes to the next device.

DSCIO

DSCIO is the UTS DSC-RBT handler. It operates from function codes (FC) passed through the I/O system by IOQ. DSCIO is capable of executing a great number of function codes but those currently received from IOQ (not counting followon) are:

<u>FC</u>	<u>MEANING</u>
0	PUNCH RECORD
1	PRINT RECORD (NOVFC)
2	READ RECORD
3	PRINT RECORD (VFC)

These operations will be discussed separately.

FC = 2 READ

All read operations performed by DSCIO in UTS RBP are requested by the input symbiont. When RBSS initiates an input symbiont that symbiont immediately issues a read to DSCIO. Since FRBIT is set, the handler interprets this as a

poll-for-input request. The input poll consists of transmitting an EOT followed by a DC1 and then reading the line. The EOT puts the RBT in transmit mode and the DC1 is transmit start. If the terminal has nothing to transmit and is in unattended mode it will reply with DC1 (no data), in attended mode it will not reply and the read operation will time out. In both cases the handler assumes no data is ready to be transmitted from the RBT and cancels the input symbiont with an 'X' signal. The symbiont terminates and will be restarted 5 seconds later by RBSS to poll again.

Eventually the poll succeeds and a record is received instead of a time out (T. O) or DC1. Since FRBIT is set the BLOCK PROTECT BIT of this record is used to initialize BPBIT, and FRBIT is reset. The record is translated from ASCII to EBCDIC, error checked, and if OK passed to the symbiont who issues another read.

In this case FRBIT is no longer set and DSCIO performs the read operation by issuing an ACK (to acknowledge the last record) followed by a read. The ACK is sent on the succeeding read operation to avoid 7670 timing peculiarities. This record is again translated and checked and passed to INSYM. Whenever cards are being read by DSCIO, CTRBIT is set to avoid scheduling conflicts between RBSS and RBBAT at job stream end.

The process described in the last paragraphs continues until either a !FIN Command or an EOT control character is received from the RBT. When a !FIN is received, DSCIO sets FINBIT and follows on directly to another ACK and read. If the result is a record, IGBIT is set and another ACK-READ is performed. This is continued until either an EOT is received or a read times out. When either of these occurs the remote card reader hopper is empty and DSCIO gives a FIN ('X' signal) to INSYM terminating input.

If EOT is received and FINBIT is not set, the job stream being received is assumed to be continued. (The RBT sends EOT automatically when the hopper empties in unattended mode, and its operator does so in attended.) In this case MORBIT is set and DSCIO begins to repeatedly execute DC1-READ-READ until another record is received. If the first read times out the second is not done. In this manner DSCIO uses one I/O request to poll the terminal using the 5 seconds T. O. to separate the polls. The second read is necessary since, if the RBT is in unattended mode, the first read will return DC1. When a record is finally received it goes to the symbiont, MORBIT is cleared and the normal reading process is restarted.

The reading of input is special cased when LIPBIT is set. In this case the record input is placed in a monitor buffer, and its address is passed to RBBAT in a remote logon record received (RBLRR) call. FINBIT is set and the read for hopper end is performed. When INSYM gets the FIN it has read no records and terminates as though a poll had returned no data.

UTS-C01 TECHNICAL MANUAL

If a record is found in error in any read process, a NAK (negative acknowledgement) is sent and the read is repeated. The ERROR HANDLING section describes this further.

FC = 1    PRINT    RECORD    (NOVFC)

This type of print operation is seldom used. The print select character is obtained and will be passed to P3 the common output routine described below. Starting at the greater of the byte count (IOQ9) and the contents of RBB:LPZ (set by !RBSIZE), blanks are stripped from the end of the record and the byte count appropriately adjusted. Processing is then passed to P3.

FC = 0    PUNCH    RECORD

The punch select character is obtained for P3. If EMBIT is reset the blank stripping described above occurs using RBB:CPZ rather than RBB:LPZ. If EMBIT is set, the byte count is set to 80 and P3 is entered without stripping.

FC = 3    PRINT    RECORD    (VFC)

This is by far the most common output request made to the handler. The format byte is obtained from the start of the message and the byte count and buffer address adjusted around it. If the format byte is the entire message, it is made a blank and the count and address are not adjusted. The top four bits of the format byte are used to determine the type of format operation (F=TOP OF FORM, C=SPACE) and the bottom four are stored in RBB:SPC as the count of formats. If the format type is unknown the adjusted message is output as though it were a NOVFC print. Special hard coded messages are used to do the formatting. A message is sent, and when it is acknowledged, RBB:SPC is decremented. Formats continue to be sent until RBB:SPC reaches zero at which time the original message is sent as though it had been a NOVFC print.

P3 - OUTPUT    RECORD

P3 is the common entry point for all output requests to DSCIO. It is entered by PRINT, PUNCH and PRINT via FORMAT PRINT.

The select character supplied by the caller is given the right block protect by BPBIT, BPBIT is toggled, and the select character is stored away into the message leading characters placed in the command list block by SYSGEN. The message is translated to ASCII, and as this is done, is shifted left three bytes. The three overflow bytes are placed in the CLIST following the leading characters (Ø. SOH, SEL, STX) and the three bytes left at the end of the message are filled with the trailing characters (EM. ETX. BCC). If the byte count is 128 the EM is



excluded. The message is output with the CLIST part and the buffer part data chained together. The acknowledgement is then read, and the request is complete. If NAK instead of ACK is received for any output request, format or message, that output is repeated until an ACK is received. See ERROR HANDLING.

### SPECIAL CASES

Three control messages can be received by DSCIO when another response is expected. They are DC1, BEL, and EOT. Another unexpected receipt is described with EOT.

#### DC1

The normal use of DC1 was described in READ. If it is received when reading for input with FRBIT reset, it is ignored and the read is repeated. If it is received when reading for ACK on output, it is treated as ACK. (Remote operators should note that this might be handy).

#### BEL

When BEL is received PUNBIT is set. On input BEL is otherwise treated like DC1; on output it is treated as NAK.

#### EOT

The use of EOT on input was described in READ. On output it is a request to suspend the current output and read input. If LIPBIT or DISCBIT is set it is ignored, otherwise SSSBIT is set, any format underway is completed, and the symbiont is suspended. RBSS will automatically start up input. If SOH is received when reading for ACK the same steps are taken. This is because in attended mode it is possible to transmit cards just as output is beginning.

Another special case is that when RBXBIT or DISCBIT is set. For READ operations and DISCBIT or any operation and RBXBIT, the current function is translated to a disconnect order. When this order completes, RBHU is reported to RBBAT.

### ERROR HANDLING

Three basic types of errors are detected by DSCIO:

1. Input record errors. (parity, block protect, block check, leading characters, trailing characters.)
2. NAK or unknown response to output records.
3. Time outs not expected by DSCIO.

Each error is logged in the system errorlog as it is detected, and an appropriate retry is done by the handler. For case 1) NAK is sent, and for 2) and 3) the operation is repeated. Retrys are attempted forever. Each time the caller-requested retrys are exhausted the current FC is saved in RBB:SFC and BEL is sent to the terminal. When the BEL has been sent the FC is retrieved from RBB:SFC, the retrys are replenished, and retrys continue.

Whenever unexpected time outs occur a TDV is performed, and if the terminal has disconnected processing ceases, and RBHU is reported to RBBAT. Whenever a hang-up is detected or disconnect is done the current symbiont is 'X'ed (INPUT) or 'S'ed (OUTPUT) after RBHU has been reported, and the current request is complete.

UTS TECHNICAL MANUAL

RBBAT

DATA BASES

RBBAT uses several RBP specific tables in the remote batch side of its processing. Some of these are in core while others are contained in the ghost itself. These tables are indexed by DCTX and exist only for remote DCT indices. In addition, some of them have one entry for local devices indexed by LCLX a special local DCT index generated by SYSGEN.

RBB:ID is a core byte table containing the RBID of the terminal logged on to this DSC. The RBID is obtained from the :RBLOG record at logon time. The byte at LCLX contains zero.

RBD:WSN is a core table containing the WSN that the RBT connected to this DSC used to log on.

RBB:MFC is a ghost table containing the total number of message files pending for this RBT.

RBB:MXP is a ghost table containing the maximum priority for job submission, obtained from the :RBLOG record at logon time.

RB:MFAD is a table in the ghost which contains the address of the message file buffer for this RBT, and RB:SPMF is a ghost table containing a count of the space used up in the current message file buffer. The entries in these tables at LCLX are used for the message files written locally for !JOB and !LIMIT aborts.

GENERAL INFORMATION

MESSAGE FILES are one sector symbiont output files written by RBBAT containing workstation specific information. A message file is built for each job stream from the RBT, and in other special cases. These files are added as normal output files of a special high priority (MFPRI) and SYSID=0. Message files are the primary form of system - workstation communication.

BLOCKING OF RB INPUT FILES

When the input symbiont is packaging input from remote batch terminals into job files it attaches !RB commands to the front of the jobs they precede. Thus the input stream on the left produces the files on the right.

UTS TECHNICAL MANUAL

!RB	FILE 1	!RB !RB !JOB 1 X X X
!RB		
!JOB 1		
X	FILE 2	!RB !JOB 2 X X X
X		
X		
!RB		
!JOB 2		
X	FILE 3	!JOB 3 X X X
X		
X		
!JOB 3		
X		
X	FILE 4	!RB  !RB
!RB		
!RB		
!FIN		

File 4 has no job following the !RB commands, so this is not a true job file and will not be run. CCI ignores any command preceeding !JOB.

DESCRIPTION

RBBAT for non-remote systems is described in SYMBIONT FILE HANDLING, so this will be a description of the effects of RBP on RBBAT. First RBP specific ghost function codes (GFC) will be described, then the effects of RBP on other calls to RBBAT.

REMOTE BATCH SPECIFIC ROUTINES

ROUTINE

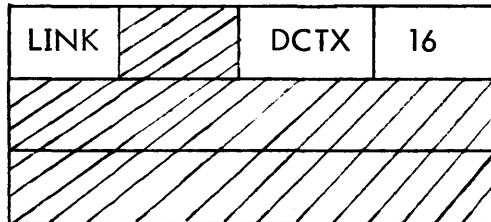
RB\$DU  
 Remote Batch Dial-up

CALLED BY

RBSS

UTS TECHNICAL MANUAL

COMMUNICATION BUFFER



DESCRIPTION

RB\$DU tells the operator that the line is connected, then sends the salutation to the RBT using NEWQWW no-wait with an FC of format print. It then clears CTRBIT and exits to CTR1.

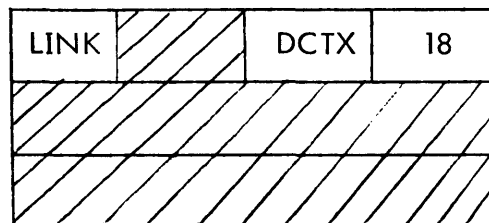
ROUTINE

RB\$HU  
 Remote Batch hang-up

CALLED BY

DSCIO, RBSS

COMMUNICATION BUFFER



DESCRIPTION

RB\$HU first checks LIPBIT and ACTBIT and if both are reset exits to CTR1. In obscure cases RBHU could be reported by both DSCIO and RBSS, and this check prevents duplication of effort. If SSSBIT is not set the normal symbiont channel's SSIG is checked for 'S'. If the 'S' is present, an output symbiont was running at the time of the hangup and was suspended by DSCIO. The signal is changed to 'Q' and the symbiont restarted to save the untransmitted portion of the file. If SSSBIT is set, SSIG is checked, and if it is not 'X' the above process is executed. If the signal is 'X' a !RBABORT was received before hang-up, and the symbiont is started with the 'X' in place to release the file.

UTS TECHNICAL MANUAL

When this symbiont management is complete RB\$HU sets XHOLDBIT in the BW:SDA of all files for this RBT and deletes any message files. The RB tables are set to their initial values, and all bits except DUPBIT are cleared in RB:FLAG. The contents of RB:XFLG (a cell in the monitor normally zero, contains OFFBIT if an RBX keyin has been done) are added in, the operator is told of the disconnection, and RB\$HU exits to CTR1.

ROUTINE

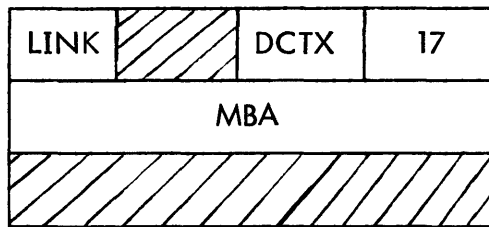
RB\$LRR

Remote Batch logon record received.

CALLED BY

DSCIO

COMMUNICATION BUFFER



MBA = Address of monitor buffer containing logon record.

DESCRIPTION

The first action taken by RB\$LRR is to move the logon record to its own buffer and release the monitor buffer. The image is checked to see that the command is !RBID and that the WSN is present, not in use, and has legal syntax. If these are in error, an appropriate error message is printed, the salutation is repeated, CTRBIT is cleared and RB\$LRR exits to CTR1 with LIPBIT still set to repeat the logon process. If the WSN can be obtained it is used as a key to read the :RBLOG file. Should an abnormal be returned on this read an error message is sent as described above. All error/abnormal codes other than X'43' (no such key) are reported to the operator. If the keyed read is successful, the information in the :RBLOG record is moved to tables, and the operator is told of the logon. A message file is built for the RBT containing logon confirmation and the status of all files for the WSN currently in the system. If there is a message on the !RBID command it is sent to the operator as though it had been an !RBMSG command. LIPBIT and CTRBIT are cleared, ACTBIT and MFPBIT are set, and RB\$LRR exits to CTR1.

UTS TECHNICAL MANUAL

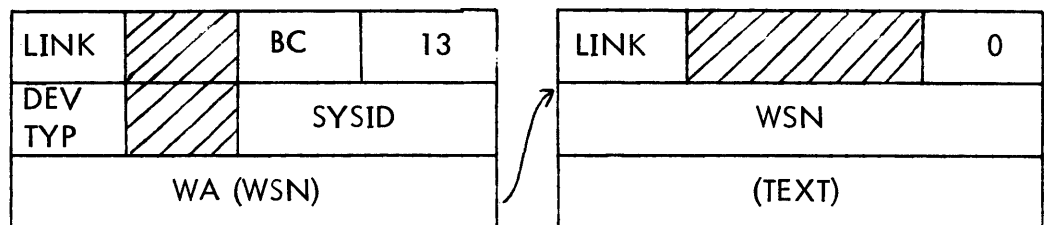
ROUTINE

KSWIT  
 Process RBSWITCH keyin

CALLED BY

KEYIN

COMMUNICATION BUFFER



DEV TYP - as in PRIO, DELETE  
 BC = byte count of WSN  
 WSN is left justified

DESCRIPTION

KSWIT first checks the WSN by doing a keyed read on :RBLOG which also obtains the RBID. If the read fails the operator is told that the WSN was bad. Appropriate files are found using SYSID and DEV TYP and their BB:RID is replaced with the RBID obtained from the WSN. Stream pending and XHOLDBIT bits are set when necessary, and for LOCAL switches symbionts are started. When all files are completed KSWIT exits to CTRL.

ROUTINE

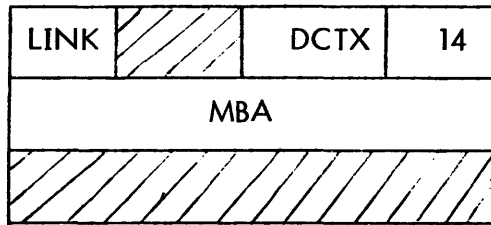
KSEND  
 Process RBSEND keyin

CALLED BY

KEYIN

UTS TECHNICAL MANUAL

COMMUNICATION BUFFER



MBA = address of monitor buffer containing message in TEXTC

DESCRIPTION

The message is moved to an internal buffer which already contains the identifier (\*\*OPERATOR MESSAGE\*\*) and the MBUF is released. If a message file is already being prepared for this RBT, KSEND adds this message and exits to CTRL. Otherwise a new message file is written containing only the heading and this message, and KSEND exits to CTRL.

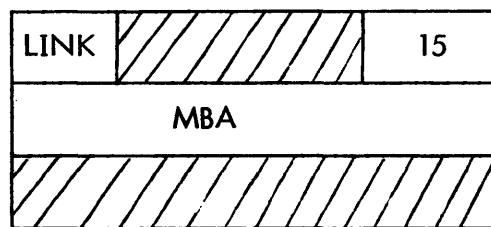
ROUTINE

KBCST  
 Process RBBDCST keyin

CALLED BY

KEYIN

COMMUNICATION BUFFER



MBA = address of MBUF containing message in TEXTC

0 = turn off message

DESCRIPTION

If MBA is non-zero the message is moved into the message file heading sent to all RBTs, the MBUF is released and KBCST exits to CTRL. If MBA = 0 KBCST moves blanks into the heading message area and exits to CTRL.



UTS TECHNICAL MANUAL

RBP EFFECTS ON OTHER ROUTINES

Each of the routines discussed here is described for non-remote systems in SYMBIONT FILE HANDLING. These discussions describe only the special handling carried out on remote files.

AIF GFC=1

When RBBAT reads the first sector of the input file from disc it enters a special routine to process any RBCC's which may be appended to the front of the job. If there is no message file in progress for this RBT, one is started with the heading message. As RBCC's are processed they are listed in the message file, and for some of them FIABIT is set. When the !JOB command is encountered control passes back to the normal job processing routines. If end of file is reached before !JOB the file contained only !RB commands and is put on the delete chain. If a command is encountered which does not begin !RB and is not !JOB the file is treated as having a missing !JOB command.

RBCC's are processed as follows:

!RBMSG

The message is moved to a buffer containing the identifier and the whole message is typed to the operator.

!RBDISC

LOFBIT is set

!RBSIZE

The options on the command are scanned and processed, LP goes to RBB:LPZ, CP to RBB:CPZ, and EM and NOEM toggle EMBIT. Errors on the card are flagged and an error message added to the message file.

!RBCONT, !RBREPRINT, !RBABORT, !RBSAVE

If SSSBIT is not set, the command is unknown. Otherwise the appropriate signal ('C', 'R', 'X', 'Q') is put in SSIG for the suspended normal symbiont. This will be acted upon at job stream end.

The rest of these descriptions apply to each SYSID specified or implied (ALL) on the command. ALL refers to all files with the RBT's RBID. SYSID's in error are flagged and skipped over, and an error message is put in the message file. FIABIT is set for all but the !RBSTATUS command.

UTS TECHNICAL MANUAL

!RBSTATUS

If all is specified or implied an appropriate status message is selected for each file and moved to the message file. Individual SYSIDS are saved in the stack until the command is processed and then used to select messages. This is to allow for the printing of error messages. When status messages are moved to the message file a header and footer are provided. For ALL, if no files exist, a message informing the user of this fact is placed in the message file.

!RBHOLD

XHOLDBIT is set in the BW:SDA of each file with the given SYSID. If ALL is stated or implied HALBIT is set in RB:FLAG.

!RBRETRIEVE

XHOLDBIT is cleared in BW:SDA for each file. For ALL, HALBIT is reset.

!RBDELETE

If the file(s) is input or output it is moved to the delete chain. If it is running, E:ABRT is reported on the user and the running chain entry is released. This results in the output being deleted later. (NOTE: whenever anything is moved to the delete chain the delete symbiont is started.)

!RBSWITCH

The WSN is checked for validity and the RBID obtained from :RBLOG. The type is also checked and saved. If either of these are in error a message is placed in the message file and the command is aborted. For each SYSID:

The device type of each file is checked and if correct the RBID is changed to that of the WSN specified on the command. XHOLDBIT, the stream pending bits, and the local symbionts are handled appropriately.

Processing of remote jobs parallels that of local jobs except that:

1. The burst is not put in the message file and only !JOB and !LIMIT commands in error are listed.
2. The device and WSN prefix the job message to the operator.
3. The missing job command message is put in the message file rather than typed to the operator.

UTS TECHNICAL MANUAL

4. If the job's account is :SYS, SYSBIT is checked, and if off, the job is aborted and notice is placed in the message file.
5. If the job's priority exceeds RBB:MXP it is reduced to that value, and notice is placed in the message file.
6. The RBID is placed in BB:RID
7. The message file is not released if the job is accepted.

JOB STREAM END

If the bit specifying !FIN is set in the AIF call for a remote job, job stream end has been reached, and several special actions take place when the job processing is complete.

1. If SSSBIT is set, SSIG for the normal Channel is checked and if it is 'S' it is replaced with 'R'. Otherwise a restart command has already inserted the correct signal. The symbiont is started to take notice of the new signal and SSSBIT is cleared
2. If FIABIT is set it is cleared and all files for this RBT are checked for non-held print and punch PUPBIT and PRPBIT are set appropriately.
3. If PUPBIT is not set, PUNBIT is cleared. If PUPBIT is set and PUNBIT is not, a message is moved to the message file informing the RBT that punch is pending but not asked for.
4. If LOFBIT is set it is changed to DISCBIT, and the local and remote (via the message file) operators are told of the log-off.
5. If IGBIT is set a message is moved to the message file telling the RBT that cards were ignored.
6. CTRBIT is cleared, MFPBIT is set. A final top-of-form is placed in the message file; it is written, and its buffer is released.

AOF, AOFNB, AAFP GFC = 3, 4, 5

There are two basic differences in AOF's for remote output files.

1. If an AOF is done and no entry in the running chain is found for this SYSID the file is moved to the delete chain. This implements !RBDEL for

UTS TECHNICAL MANUAL

running files. Remember that XHOLDBIT is passed from input to output files by the AOF process.

2. When a file is added for an RBID and that RBID is not logged on XHOLDBIT is set in its BW:SDA. If a file is added by AOFNB or AOFB for a logged-on RBID and HALBIT is set in its RB:FLAG, XHOLDBIT is set for the file. If XHOLDBIT is not set the appropriate stream pending bit is set in this RBID's RB:FLAG.

GOF GFC = 6

1. Files with XHOLDBIT SET cannot be given as output to RBTs.
2. If a file given to an RBT is a message file, RBB:MFC is decremented, and if it reaches zero, MFPBIT is cleared. When no file can be found for a remote GOF call, the stream pending bit appropriate to the device type in DCT4 is cleared.

There is no change to other calls due to remote batch.