CRAY RESEARCH, INC.

CRAY-1® COMPUTER SYSTEMS

CRAY-OS MESSAGE MANUAL

SR-0039

Copyright© 1980, 1981 by CRAY RESEARCH, INC. This manual or parts thereof may not be reproduced in any form without permission of CRAY RESEARCH, INC.

RECORD OF REVISION

PUBLICATION NUMBER SR-0039

Each time this manual is revised and reprinted, all changes issued against the previous version in the form of change packets are incorporated into the new version and the new version is assigned an alphabetic level. Between reprints, changes may be issued against the current version in the form of change packets. Each change packet is assigned a numeric designator, starting with 01 for the first change packet of each revision level.

Every page changed by a reprint or by a change packet has the revision level and change packet number in the lower righthand corner. Changes to part of a page are noted by a change bar along the margin of the page. A change bar in the margin opposite the page number indicates that the entire page is new; a dot in the same place indicates that information has been moved from one page to another, but has not otherwise changed.

Requests for copies of Cray Research, Inc. publications and comments about these publications should be directed to:

CRAY RESEARCH, INC.,
1440 Northland Drive,
Mendota Heights, Minnesota 55120

Revision	Description
	October, 1980 - Original printing.
01	July, 1981 - This change packet brings the manual into agreement with Version 1.10 of the released software. Major changes include changing the SKOL messages from uncoded to coded messages and adding the Exchange Processor (EXP) messages, Control Statement Processor (CSP) messages, $SYSLIB messages, MODSEQ messages, MODSET messages, SYSREF messages, Station Call Processor (SCP) messages, and EXEC Message Processor (MEP) messages. Section 1 has been rewritten and technical changes are noted by change bars. Miscellaneous technical and editorial changes are also included.

PREFACE

The CRAY-OS Message Manual describes all user logfile, system logfile, and program output listing messages issued by CRAY-OS software. The manual is divided into three sections: coded messages, CFT compiler messages, and uncoded messages.

The coded messages are alphabetized by code. Each message is accompanied by the class (reprieve information) of the message, the cause of the message, the action to be taken, and the source of the message.

The CFT compiler messages are alphabetized by message. Each message is accompanied by the class (reprieve information) of the message, the cause of the message, and the action to be taken.

The uncoded messages that begin with the actual message are alphabetized by message. The messages that begin with a variable contain the word *variable* at the beginning of the message and are alphabetized by the actual message. The messages that begin with special characters are alphabetized by special character. Each uncoded message is accompanied by the class (reprieve information) of the message, the cause of the message, the action to be taken, and the source of the message.

The following Cray Research publications provide supplemental information.

SR-0000 CAL Assembler Version 1 Reference Manual
SR-0011 CRAY-OS Version 1 Reference Manual
SR-0014 Library Reference Manual
SR-0033 SKOL Reference Manual
SR-0009 CRAY-1 FORTRAN (CFT) Reference Manual

AB000 - JOB STEP ABORTED. P=*xxxxxxx*.

CLASS: Informative

CAUSE: The message preceding AB000 gives the cause of abort. AB000 supplies the user field address where the job step abort occurred.

ACTION: Use the P address supplied to debug the user program.

SOURCE: EXP

AB001 - END-OF-FILE ON READ

CLASS: 4

CAUSE: End of file was reached on the control statement file and CSP tried to read another record from $CS.

ACTION: Forward a system dump of the error to a Cray Research analyst.

SOURCE: EXP

AB002 - INVALID LOCK OR UNLOCK INDICATION

CLASS: 4; reprievable

CAUSE: An F$IOA request (lock or unlock) was issued and the value specified by S1 was not a valid lock or unlock request code. Probable user error.

ACTION: Review the description of the F$IOA call in the CRAY-OS Version 1 Reference Manual. If the program specified the call correctly, obtain a dump of the job (JTA, DSP) and forward it to a Cray Research analyst.

SOURCE: EXP

AB003 - DEVICE ALLOCATION TABLE SPACE EXHAUSTED

CLASS: 4; reprievable

CAUSE: DAT space is not available in the system DAT area.

ACTION: Determine why the DATs are full. If a front-end system is down, DAT space will be freed as soon as the front-end successfully logs on. If too many system resources, such as SDR datasets, are using DAT space, delete the unused datasets. If the error occurs frequently, reassemble the system with a larger DAT area specified in the STP constant.

SOURCE: EXP

AB004 - DATASET NOT OPEN

CLASS: 4; reprievable

CAUSE: A buffered I/O request was issued for a dataset that was not opened. Probable user error.

ACTION: Ensure that an F$OPN request is being issued before the first I/O request. If the F$OPN is being issued, obtain a dump of the job (JTA, DSP) and forward the dump to a Cray Research analyst.

SOURCE: EXP

AB005 - INVALID OPEN

CLASS: 4; reprievable

CAUSE: One of the following conditions has occurred:

1. The type of open request specified was invalid
2. The dataset is already opened. (See the COS Reference Manual, publication SR-0011.)

ACTION: Check the F$OPN request type for validity. If the dataset is opened more than once, a CLOSE request must be issued after each OPEN. If the request is being issued correctly, obtain a dump of the job (JTA, DSP) and forward it to a Cray Research analyst.

SOURCE: EXP

AB006 - NO READ PERMISSION

CLASS: 4; reprievable

CAUSE: An attempt was made to read a dataset and the dataset was not accessed with read permission. Probable user error.

ACTION: Specify the read permission control word when the dataset is accessed. If the read permission control word was specified correctly, obtain a dump of the job (JTA) and forward it to a Cray Research analyst.

SOURCE: EXP

AB007 - NO WRITE PERMISSION

CLASS: 4; reprievable

CAUSE: An attempt was made to write a dataset that had not been accessed with write permission and/or did not have unique access.

ACTION: Specify the write permission control word and/or unique access when the dataset is accessed.

SOURCE: EXP

AB008 - ILLEGAL BITS SET IN RFL REQUEST WORD

CLASS: 4; reprievable

CAUSE: The system received a user request for a system-privileged memory operation.

ACTION: If there is an error in the memory request being issued by the program, correct the error and resubmit the request. If there is no apparent error, obtain a dump of the registers at the time of the request and forward the dump to a Cray Research analyst.

SOURCE: EXP

AB009 - ATTEMPT TO DELETE MEMORY OUTSIDE PROGRAM AREA

CLASS: 4; reprievable

CAUSE: The system received a user request to delete memory. The specified first word address plus the specified length define an area beyond the user field.

ACTION: If the memory delete request is incorrect, modify the request and resubmit the job. Otherwise, obtain a dump of the JTA at the time of the request and forward the dump to a Cray Research analyst.

SOURCE: EXP

AB010 - NO AVAILABLE DISK SPACE

CLASS: 400; reprievable

CAUSE: The system has no more disk space to allocate to the dataset being written.

ACTION: If there is a shortage of disk space, delete any datasets that are no longer required; then rerun the job. If there is not a shortage of disk space, obtain a dump of the job (JTA) and forward the dump to a Cray Research analyst.

SOURCE: EXP

AB011 - SYSTEM DIRECTORY IS FULL

CLASS: 4000; reprievable

CAUSE: An ACCESS ENTER request was being processed for a dataset and the resident system directory table area is full.

ACTION: If there are entries in the system directory that are no longer needed, clear the system directory by performing a restart of COS and by specifying *SDR in the parameter file. Then run a JSYSDIR job, specifying only the datasets to be entered into the SDR. If all entries in the SDR are to be maintained, the value of NE@SDR in the STP system tables must be increased, COS must be reassembled, and COS must be restarted to increase the number of resident SDR entries. If these two conditions do not exist, obtain a system dump and forward it to a Cray Research analyst.

SOURCE: EXP

AB012 - JTA OVERFLOW

CLASS: 4

CAUSE: The job has used too much space in the pool area of the Job Table Area (JTA).

ACTION: Restructure the job to use fewer conditional, iterative, and LIBRARY statements.

SOURCE: EXP

AB013 - MORE MEMORY REQUESTED THAN AVAILABLE

CLASS: 4

CAUSE: A memory allocation request was issued for more memory than available in the machine.

ACTION: Determine if the request is valid. If not, correct the program and resubmit the job. If it is valid, it might be necessary to adjust I@MEM.

SOURCE: EXP

AB014 - MORE MEMORY REQUESTED THAN ALLOWED

CLASS: 4

CAUSE: A memory allocation request was issued for more memory than allowed by the maximum field length value in the JTA.

ACTION: Correct the memory request and resubmit the job.

SOURCE: EXP

AB015 - ACCESS BEFORE ACQUIRE FAILED

CLASS: 2000; reprievable

CAUSE: During ACQUIRE processing, an ACCESS request failed. The previous message (a PDM message) should explain the failure.

ACTION: Obtain a dump of the PDD used to make the request and a dump of the job (JTA) and forward these to a Cray Research analyst.

SOURCE: EXP

AB016 - SUBDATASET, $IN, CANNOT BE DISPOSED

CLASS: 2000; reprievable

CAUSE: The system detected a request to DISPOSE a file named $IN. This request is invalid.

ACTION: If a job control statement is in error, correct the statement and resubmit the job. If there is no error in the control statement, forward the job output to a Cray Research analyst.

SOURCE: EXP

AB017 - INVALID CLOSE REQUEST

CLASS: 4; reprievable

CAUSE: The ODN specified by the CLOSE request pointed to a DSP that should be in the user field, but the address is not below the value in JCHLM.

ACTION: Determine if the address in the ODN is valid. If not a user error, obtain a dump of the job (JTA, ODN, DSPs) and forward these to a Cray Research analyst.

SOURCE: EXP

AB018 - DATASET ALREADY OPENED

CLASS: 4; reprievable

CAUSE: An F$DNT request was issued for a dataset that was already opened. Probable user error.

ACTION: Issue a CLOSE request before a second F$DNT request for a dataset is issued. If CLOSE is being issued, obtain a dump of the job (JTA, DSP) and forward it to a Cray Research analyst.

SOURCE: EXP

AB019 - JOB COMMUNICATION BLOCK DESTROYED

CLASS: 0; unreprievable

CAUSE: One of the following conditions was detected:

1. One of three words in the JCB has been destroyed - the word containing HLM, the word containing the base address of I/O buffers, or the word containing the base address of DSP or base address of LFTs.
2. HLM is less than the length of the JCB.
3. HLM is greater than the base of the LFTs.
4. LFT area overlays DSP area.
5. Calculated LFT area + DSP area overlays buffer area.
6. Base address of I/O buffers is greater than the field length.

ACTION: If an error in the program is causing the JCB to be overwritten, correct the program and resubmit the job. If there is no error in the program, obtain a dump of the JCB and forward the dump to a Cray Research analyst.

SOURCE: EXP

AB020 - INVALID SYSTEM REQUEST PARAMETER

CLASS: 4; reprievable

CAUSE: An EXP system call routine received an invalid parameter, generally an address outside the user field. Probable user error.

ACTION: Obtain a dump of the JTA and verify the addresses or values passed to the system. If there is no apparent user error, forward the dump to a Cray Research analyst.

SOURCE: EXP

AB021 - DATASET NOT FOUND

CLASS: 4; reprievable

CAUSE: A system routine required an existing DNT for processing the specified dataset but was unable to locate the DNT or a dataset not local to the job or known by the system was loaded.

ACTION: Obtain a dump of the JTA and verify that the dataset name being passed to the system is valid. Next, follow the chain of DNTs to locate a DNT with the user-specified name. If a DNT containing the user-specified name exists and if the user-specified name is correct, forward the dump to a Cray Research analyst.

SOURCE: EXP

AB022 - INVALID PROGRAM LOAD DATASET

CLASS: 4; reprievable

CAUSE: The system task attempted to load a program into memory and encountered an unexpected EOR, or a PDT was missing, or the module is not absolute.

ACTION: Verify that the specified program is in fact a loadable program dataset. If not, correct the control statement and resubmit the job. If the dataset is valid, recreate the load dataset.

SOURCE: EXP

AB023 - JOB TIME LIMIT EXCEEDED

CLASS: 200; reprievable

CAUSE: The job has exceeded the specified or default time limit.

ACTION: First, determine that the program was not looping. If execution was proper, raise the time limit value on the job statement and resubmit the job. If the operating system is interacting incorrectly with the job, obtain a system dump and forward the dump to a Cray Research analyst.

SOURCE: EXP

AB024 - OPERATOR DROPPED THE JOB

CLASS: 10; reprievable

CAUSE: The system operator issued a 'DROP' command for the specified job.

ACTION: Investigate the cause of the operator action. If the operating system is interacting incorrectly with the job, obtain a system dump and forward the dump to a Cray Research analyst.

SOURCE: EXP

AB025 - USER PROGRAM REQUESTED ABORT

CLASS: 2; reprievable

CAUSE: The user program issued a request to terminate the job step.

ACTION: If the request was issued by a library routine, obtain a dump of the JTA and DSP and forward the dump to a Cray Research analyst.

SOURCE: EXP

AB026 - INVALID (UNDEFINED) USER REQUEST

CLASS: 4; reprievable

CAUSE: An invalid system call request was issued. Probable user error.

ACTION: Obtain a dump of the JTA and validate the registers used for the system call. If the request was valid, forward the dump to a Cray Research analyst.

SOURCE: EXP

AB027 - CALL NOT BETWEEN USER BA AND LA

CLASS: 4; reprievable

CAUSE: The system received a request for which the required address parameters were not valid. Probable user errors.

ACTION: Obtain a dump of the JTA and examine the registers used for the system call. If the addresses are correct, forward the dump to a Cray Research analyst.

SOURCE: EXP

AB029 - LOGICAL DEVICE NAME NOT FOUND

CLASS: 4; reprievable

CAUSE: The specified logical device name could not be found in an existing equipment table or the device is inoperative.

ACTION: Use the STORAGE command at the MCU console to verify that the specified device name exists and is operational. If the device name was specified incorrectly, correct the control statement and resubmit the job.

SOURCE: EXP

AB030 - BLOCK NUMBER ERROR

CLASS: 4; reprievable

CAUSE: The system I/O routines detected that the block control word of the record that was just read is not the expected block number.

ACTION: Obtain a dump of the DSP and buffers for the dataset in error and forward them to a Cray Research analyst.

SOURCE: EXP

AB031 - UNRECOVERABLE DATA ERROR

CLASS: 4; reprievable

CAUSE: The disk queue manager detected a discrepancy in the checksum during an I/O operation.

ACTION: Dump the system log and determine if other datasets on the same device are receiving similar error messages. If so, notify the Cray engineer of a possible failing device. If it is not a common error and if the problem persists for the dataset, the dataset must be recreated.

SOURCE: EXP

AB032 - UNRECOVERABLE HARDWARE ERROR

CLASS: 4; reprievable

CAUSE: The disk queue manager detected a permanent hardware error and was unable to complete the I/O operation.

ACTION: Check the status of the device. If the device was not ready, ready it and resubmit the job. If the device appears to be operational, notify a Cray Research analyst.

SOURCE: EXP

AB033 - READ AFTER WRITE OR PAST END-OF-DATA

CLASS: 4; reprievable

CAUSE: An invalid I/O operation was attempted; either a read operation was issued after a write operation or a read request was issued after end of data was detected by the system.

ACTION: In the first case, determine if the correct sequence of I/O operations is being issued. A READ operation can follow a WRITE operation without an intervening REWIND only when the access mode is random. In the second case, determine if end-of-data is being tested according to the documentation for the request being issued. If the message is being produced without apparent cause, obtain a dump of the DSP and buffers at the time of failure and forward them to a Cray Research analyst.

SOURCE: EXP

AB034 - UNKNOWN ERROR

CLASS: 4; reprievable

CAUSE: The disk queue manager returned an unrecognizable error.

ACTION: Obtain a dump of the job including JTA, DSP, and buffers and forward the dump to a Cray Research analyst.

SOURCE: EXP

AB035 - INVALID PROCESSING DIRECTION

CLASS: 4; reprievable

CAUSE: An I/O request was issued but the request was not for a valid operation.

ACTION: Obtain a dump of the JTA and DSP. If the requested processing direction was valid, forward the dump to a Cray Research analyst.

SOURCE: EXP

AB036 - DATASET PREMATURELY TERMINATED

CLASS: 4; reprievable

CAUSE: A blocked read request was issued for a dataset. There were no more blocks to be read but an end-of-data control word was not detected.

ACTION: Obtain a dump of the DSP and buffers at the time of the failure and forward them to a Cray Research analyst.

SOURCE: EXP

AB037 - DATASET PARAMETER TABLE INVALID

CLASS: 4

CAUSE: This message is issued for one or more of the following reasons:

1. An I/O request was received by the system and the DSP was busy.
2. One of the buffer pointers in the DSP was invalid.
3. The record control word address in the DSP was invalid.
4. A dataset was being closed and the buffer address in the DSP was in the user field.
5. The dataset name in the DSP has been cleared; the user has overlayed the DSP.
6. The I/O operation to be performed on the dataset is not valid for the status of the dataset.

ACTION: If the user program is checking active I/O operations correctly and the DSP pointers look valid, forward a dump of the JTA, DSP, and buffers to a Cray Research analyst. Otherwise, correct the program and resubmit the job.

SOURCE: EXP

AB038 - OPERATOR KILLED THE JOB

CLASS: 0; not reprievable

CAUSE: Execution of the job was terminated by the operator.

ACTION: If the reason for the operator action was not caused by the job in question, resubmit the job. Otherwise, use available tools to determine the cause of the problem.

SOURCE: EXP

AB039 - OPERATOR RERAN THE JOB

CLASS: 20; reprievable

CAUSE: The job was requested for execution by the operator.

ACTION: Not applicable

SOURCE: EXP

AB040 - INVALID DISPOSITION CODE

CLASS: 4; reprievable

CAUSE: The disposition code (DC) of the JCL statement is not specified as documented in the COS Reference Manual.

ACTION: If the disposition code is incorrect, correct it and resubmit the job. If the disposition code is correct, forward the job output to a Cray Research analyst.

SOURCE: EXP

AB041 - 'ENTER' ALLOWED ON ACCESS ONLY

CLASS: 4000; reprievable

CAUSE: A permanent dataset control statement was specified with ENTER as a selected parameter. ENTER is valid only for the ACCESS control statement.

ACTION: If the control statement is in error, remove the ENTER operand and resubmit the job. Otherwise, forward the job output to a Cray Research analyst.

SOURCE: EXP

AB043 - ALLOWABLE USER LOG SIZE EXCEEDED

CLASS: 400; reprievable

CAUSE: The size of the logfile for the job has exceeded the system generated size (I@LGUSZ). The job is writing log records in a loop or the job is issuing more requests than are allowed.

ACTION: In the first case, correct the error causing the loop and resubmit the job. In the second case, increase the value of I@LGUSZ. If there is no apparent cause of the failure, forward the job output to a Cray Research analyst.

SOURCE: EXP

AB044 - INVALID DATASET NAME

CLASS: 4; reprievable

CAUSE: The dataset name in the control statement contains an invalid character or is longer than 7 characters.

ACTION: Correct the dataset name and resubmit the job. If the name is valid, forward the job output to a Cray Research analyst.

SOURCE: EXP

AB045 - SPECIFIED LM IS TOO BIG

CLASS: 400; reprievable

CAUSE: An attempt was made to open a dataset by specifying a size limit that is greater than the allowable size for the dataset.

ACTION: Adjust the dataset size limit to be within the system specification. If a larger system specification is required, change the system parameter I@MAXLM or I@OMLM in COSTXT and regenerate the system.

SOURCE: EXP

AB046 - DATASET SIZE LIMIT EXCEEDED

CLASS: 400; reprievable

CAUSE: The last write operation caused the file to exceed the size specified in the Dataset Name Table or to exceed the maximum size allowed by the site.

ACTION: Verify that the program was not in a loop. If the dataset size needs to be larger, use an ASSIGN control statement with the specified LM parameter.

SOURCE: EXP

AB047 - DATASET NOT AVAILABLE FROM STATION

CLASS: 2000

CAUSE: An ACQUIRE control statement was issued for a dataset that could not be staged or the front end to which the ACQUIRE was issued does not support that control statement.

ACTION: If ACQUIRE is supported by the front end, ensure that the dataset exists. Resubmit the job.

SOURCE: EXP

AB048 - DATASET CAN'T BE SAVED ON FRONT-END

CLASS: 2000

CAUSE: A DISPOSE request was issued but the dataset could not be saved by the front end.

ACTION: Determine why the dataset could not be saved (e.g., lack of space). Request assistance from a front-end system programmer if necessary. Resubmit the job.

SOURCE: EXP

AB049 - INVALID LFT'S IN USER AREA

CLASS: 4; reprievable

CAUSE: The system detected that the Logical File Tables in the user area did not match the LFTs in the Job Table Area.

ACTION: Obtain a dump of the JTA, Dataset Parameter Tables, and LFTs and forward the dump to a Cray Research analyst.

SOURCE: EXP

AB051 - INVALID POINTER TO FIRST JTA LFT

CLASS: 4; reprievable

CAUSE: The system detected that the Job Table Area pointer to the first Logical File Table was 0.

ACTION: Obtain a dump of the JTA, Dataset Parameter Tables, and LFT and forward the dump to a Cray Research analyst.

SOURCE: EXP

AB052 - NO USER LFT DN MATCH IN JTA LFT'S

CLASS: 4; reprievable

CAUSE: The system was searching the Job Table Area Logical File Table change and could not locate the dataset name specified in the search argument.

ACTION: Obtain a dump of the JTA, Dataset Parameter Tables, and LFTs. A system dump may also be necessary. Forward all documentation to a Cray Research analyst.

SOURCE: EXP

AB053 - FLOATING POINT ERROR

CLASS: 100; reprievable

CAUSE: The CPU detected a floating-point underflow or overflow while executing an instruction.

ACTION: Correct the program error and resubmit the job.

SOURCE: EXP

AB054 - OPERAND RANGE ERROR

CLASS: 4; reprievable

CAUSE: An attempt was made to load an A or S register from outside the user field.

ACTION: Correct the program and resubmit the job.

SOURCE: EXP

AB055 - PROGRAM RANGE ERROR

CLASS: 4; reprievable

CAUSE: An attempt was made to load a P register from outside the user field.

ACTION: Correct the program and resubmit the job.

SOURCE: EXP

AB056 - UNCORRECTED MEMORY ERROR

CLASS: 40

CAUSE: A double-bit memory error occurred while the job was executing.

ACTION: Resubmit the job. If the error continues to occur, notify the operator.

SOURCE: EXP

AB057 - CONSOLE INTERRUPT

CLASS: Informative; not reprievable

CAUSE: An abort command was entered at an interactive terminal, which caused the executing program to stop running.

ACTION: Not applicable

SOURCE: EXP

AB058 - ERROR EXIT

CLASS: 4; reprievable

CAUSE: A zero (0) instruction was executed in the user area.

ACTION: Correct the program and resubmit the job.

SOURCE: EXP

AB061 - NO INVOKE REQUEST WAS PROVIDED

CLASS: 4; reprievable

CAUSE: A call was made to F$INV without the required information.

ACTION: Send the required information.

SOURCE: EXP

AB062 - INVOKE REQUEST ALREADY PENDING

CLASS: 4; reprievable

CAUSE: An invoke request has already been sent and another one is waiting. The one waiting will not be processed.

ACTION: Send the request after the other one has been completed.

SOURCE: EXP

AB063 - INVOKE LEN NOT A MULTIPLE OF O'1000

CLASS: 4; reprievable

CAUSE: The job class structure length is not a multiple of 1000 octal.

ACTION: Recreate the job class structure.

SOURCE: EXP

AB064 - INVOKE LEN GREATER THAN MAX ALLOWED

CLASS: 4

CAUSE: The length of the job class structure is greater than the maximum length allowed.

ACTION: Recreate the job class structure or change the installation parameter.

SOURCE: EXP

AB066 - DATASET HAS RELATED DISPOSES ACTIVE

CLASS: 4

CAUSE: The job tried to open a dataset which had disposes still outstanding against it.

ACTION: Restructure the job so it does not reuse the same local dataset name, or delay the job with DISPOSE, WAIT.

SOURCE: EXP

AB067 - INVALID PROCEDURE DATASET

CLASS: 4

CAUSE: Procedure datasets cannot be memory-resident or unblocked.

ACTION: Correct the ASSIGN statement for the procedure dataset and resubmit the job.

SOURCE: EXP

AB068 - PROCEDURE NEST LEVEL EXCEEDED

CLASS: 4

CAUSE: Procedures that execute other procedures have exceeded the limit of I@PRIVL - 1.

ACTION: Investigate procedure nesting structure or increase the value of I@PRIVL.

SOURCE: EXP

AB070 - CONSOLE ATTENTION REQUEST

CLASS: 10000; reprievable

CAUSE: An ATTENTION command was entered at an interactive terminal.

ACTION: Not applicable

SOURCE: EXP

AB071 - BAD CLASS STRUCTURE

CLASS: 4; reprievable

CAUSE: An attempt was made to invoke an invalid class structure.

ACTION: Redefine the class structure and rerun JCSDEF.

SOURCE: EXP

AB072 - DSP DESTROYED BY USER

CLASS: 4

CAUSE: This message is issued for one or more of the following reasons:

1. An I/O request was received by the system and the DSP was busy.
2. One of the buffer pointers in the DSP was invalid.
3. The record control word address in the DSP was invalid.
4. A dataset was being closed and the buffer address in the DSP was in the user field.
5. The dataset name in the DSP has been cleared; the user has overlayed the DSP.
6. The I/O operation to be performed on the dataset is not valid for the status of the dataset.

ACTION: If the user program is checking active I/O operations correctly and the DSP pointers look valid, forward a dump of the JTA, DSP, and buffers to a Cray Research analyst. Otherwise, correct the program and resubmit the job.

SOURCE: EXP

AB073 - UNDEFINED FUNCTION CODE IN F$INS

CLASS: 4; reprievable

CAUSE: The system function F$INS was called with a parameter (=subfunction code) outside the limits of the defined subfunction table. F$INS provides for installation-defined subfunctions. The table is initially empty. Thus, a call to F$INS made with an empty subfunction table results in this error regardless of the subfunction code.

ACTION: Check the call of the F$INS function (or the INSFUN macro) to insure that the subfunction code is being passed properly. If it is being passed properly, check the subfunction table INSTAB to insure that it is defined properly.

SOURCE: EXP

AB074 - DUMPJOB PROCESSING INHIBITED

CLASS: 4000; reprievable

CAUSE: A DUMPJOB or F$DJA was attempted while an execute-only dataset was loaded in the user field.

ACTION: Remove the call to DUMPJOB when an execute-only dataset might be loaded.

SOURCE: EXP

AB075 - NO PERMISSIONS GRANTED FOR EXECUTE-ONLY DATASET

CLASS: 4000; reprievable

CAUSE: An attempt was made to directly open, dump, dispose, or otherwise modify an execute-only dataset.

ACTION: Change the job so all references and modifications to the dataset are made through the Control Statement Processor (CSP).

SOURCE: EXP

AB076 - DATASET IS ALREADY ACCESSED BY THE JOB

CLASS: 4; reprievable

CAUSE: An attempt was made to access a dataset in the System Directory Table (SDR) with a dataset name which is already local to the job.

ACTION: Change the job so the dataset is not local at the time of the F$ASD call.

SOURCE: EXP

AB077 - INTERNAL CONTROL STATEMENT PROCESSOR (CSP) ERROR

CLASS: 0; not reprievable

CAUSE: COS detected an error in a request and/or stored information maintained by CSP.

ACTION: Obtain a dump of the job (JTA and JCB) and forward it to a Cray Research analyst.

SOURCE: EXP

AB078 - PRIVILEGED SYSTEM REQUEST

CLASS: 4000; reprievable

CAUSE: User program made a system request (F$xxx) which is restricted to privileged processes.

ACTION: Ensure that the request is the one desired with the correct format.

SOURCE: EXP

AB079 - REFERENCE TO UNASSIGNED JCL SYMBOL

CLASS: 4; reprievable

CAUSE: An attempt has been made to use the value of a JCL symbol which has not yet been given a value.

ACTION: Ensure that the assigned symbol is given a value prior to its usage within a JCL expression.

SOURCE: EXP

AB080 - RECEIVE BUFFER TOO SMALL

CLASS: 4; reprievable

CAUSE: The user supplied table for a system request is too small to receive the requested data. This can occur with F$LIB and F$SYM requests.

ACTION: Ensure that enough area is allocated for the requested data.

SOURCE: EXP

AB081 - UNDEFINED JCL SYMBOL REFERENCED

CLASS: 4; reprievable

CAUSE: An attempt has been made to get or set the value for an unknown JCL symbol.

ACTION: Ensure that the symbol name is correct.

SOURCE: EXP

AB082 - JCL SYMBOL CANNOT BE MODIFIED BY THE JOB

CLASS: 4; reprievable

CAUSE: An attempt has been made to modify the value of a symbol which is either a system constant symbol or a system set by COS.

ACTION: Check the usage of the offending symbol.

SOURCE: EXP

AB083 - INVALID MESSAGE CLASS

CLASS: 4; reprievable

CAUSE: A F$MSG call was made with the message class supplied out of the valid range.

ACTION: Correct the message class field in the F$MSG call.

SOURCE: EXP

AC001 - NO ACCOUNT NUMBER SPECIFIED

CLASS: Fatal

CAUSE: In the utility ACCTDEF, one of the directives in the input file did not contain a value for the required AC parameter.

ACTION: Either remove the erroneous directive or correct the error and rerun the program (ACCTDEF). The accounting dataset will not be created or modified until the error no longer exists.

SOURCE: ACCOUNTING

AC002 - 16 CHARACTER ACCOUNT NUMBER ILLEGAL

CLASS: Fatal

CAUSE: In the utility ACCTDEF, one of the directives in the input file contained an AC parameter value that exceeded the 15 characters allowable for an account number.

ACTION: Either remove the erroneous directive or correct the error and rerun the program (ACCTDEF). The accounting dataset is not created or modified until the error no longer exists.

SOURCE: ACCOUNTING

AC003 - 16 CHARACTER PASSWORD ILLEGAL

CLASS: Fatal

CAUSE: In the utility ACCTDEF, one of the directives in the input file contained a PW parameter value that exceeded the 15 characters allowable for a password.

ACTION: Either remove the erroneous directive or correct the error and rerun the program (ACCTDEF). The accounting dataset is not created or modified until the error no longer exists.

SOURCE: ACCOUNTING

AC004 - ENTRY NOT IN $ACCT FILE

CLASS: Fatal

CAUSE: In the utility ACCTDEF, one of the directives (CHANGE, ACTIVATE, INACTIVE or DELETE) in the input file contained values for the AC (account number) and PW (password) parameters that did not match an AC/PW entry in the accounting dataset.

ACTION: Either remove the directive from the input file or use a CREATE directive to place the specified account number/password in the accounting dataset. The accounting dataset is not created or modified until the error no longer exists.

SOURCE: ACCOUNTING

AC005 - ENTRY TO BE CHANGED IS INACTIVE

CLASS: Fatal

CAUSE: In the utility ACCTDEF, one of the CHANGE directives in the input file referenced an inactive AC/PW entry in the accounting dataset.

ACTION Remove the directive from the input file or use an ACTIVATE directive to activate the AC/PW to be changed, then use the CHANGE directive; or use a DELETE directive to delete the AC/PW referenced; then use the CREATE directive to create a new AC/PW entry. The accounting dataset is not created or modified until the error no longer exists.

SOURCE: ACCOUNTING

AC006 - DUPLICATE AC/PW ENTRY

CLASS: Warning

CAUSE: In the utility ACCTDEF, one of the CREATE directives in the input file referenced an account number and password that already existed in the accounting dataset.

ACTION: Not applicable

SOURCE: ACCOUNTING

AC007 - ENTRY IS ALREADY ACTIVE

CLASS: Warning

CAUSE: In the utility ACCTDEF, one of the ACTIVATE directives in the input file attempted to activate an account number and password that were already active in the accounting dataset.

ACTION: Not applicable

SOURCE: ACCOUNTING

AC008 - ENTRY IS ALREADY INACTIVE

CLASS: Warning

CAUSE: In the utility ACCTDEF, one of the INACTIVE directives in the input file attempted to inactivate an account number and password that were already inactive in the accounting dataset.

ACTION: Not applicable

SOURCE: ACCOUNTING

AC009 - NO NEW AC NUMBER OR PASSWORD SPECIFIED

CLASS: Warning

CAUSE: In the utility ACCTDEF, one of the CHANGE directives in the input file did not contain values for the NAC and NPW parameters.

ACTION: Not applicable

SOURCE: ACCOUNTING

AC010 - NO $ACCT FILE-ONLY CREATE DIR. LEGAL

CLASS: Fatal

CAUSE: In the utility ACCTDEF, one of the directives in the input file was not a CREATE directive. CREATE directives are the only type allowed when the accounting dataset is being created.

ACTION: Remove the directive that is not a CREATE directive from the input file and rerun the program (ACCTDEF). The accounting dataset is not created or modified until the error no longer exists.

SOURCE: ACCOUNTING

AC011 - DIRECTIVE NOT RECOGNIZED BY ACCTDEF

CLASS: Fatal

CAUSE: In the utility ACCTDEF, one of the directives in the input file was not recognized by the utility.

ACTION: Remove from the input file the directive that is unrecognizable and rerun the program (ACCTDEF). The accounting dataset is not created or modified until the error no longer exists.

SOURCE: ACCOUNTING

AC020 - NO DIRECTIVES TO BE PROCESSED

CLASS: Fatal

CAUSE: The utility ACCTDEF was run without any directives in the input file.

ACTION: Put directives in the input file.

SOURCE: ACCOUNTING

AC021 - DIRECTIVE WITH FATAL ERROR EXISTS

CLASS: Fatal

CAUSE: When the utility ACCTDEF was run, a fatal error was detected in at least one of the directives in the input file.

ACTION: Correct the directives that contain errors.

SOURCE: ACCOUNTING

AC022 - INVALID CHARACTER FOUND ON DIRECTIVE

CLASS: Fatal

CAUSE: When a directive was cracked, an invalid character was found.

ACTION: Correct the directive that contains the error.

SOURCE: ACCOUNTING

AC023 - PREMATURE END OF INPUT LINE IN DIRECTIVE

CLASS: Fatal

CAUSE: When a directive was cracked, the input line was found to end prematurely.

ACTION: Correct the directive that contains the error.

SOURCE: ACCOUNTING

AC030 - REQUIRED PARAMETER NOT FOUND

CLASS: Fatal

CAUSE: An error was found after the call was made to process the parameters on the given directive.

ACTION: Correct the directive and rerun ACCTDEF.

SOURCE: ACCOUNTING

AC031 - OUTPUT PARAMETER OVERFLOW

CLASS: Fatal

CAUSE: An error occurred on a call to process the parameters on a directive.

ACTION: Correct the directive and rerun ACCTDEF.

SOURCE: ACCOUNTING

AC032 - SYNTAX ERROR ON DIRECTIVE

CLASS: Fatal

CAUSE: An error occurred on a call to process the parameters on a directive.

ACTION: Correct the directive and rerun ACCTDEF.

SOURCE: ACCOUNTING

AC033 - UNKNOWN OR DUPLICATE KEYWORD

CLASS: Fatal

CAUSE: An error occurred on a call to process the parameters on a directive.

ACTION: Correct the directive and rerun ACCTDEF.

SOURCE: ACCOUNTING

AC034 - UNEXPECTED SEPARATOR ON DIRECTIVE

CLASS: Fatal

CAUSE: An error occurred on a call to process the parameters on a directive.

ACTION: Correct the directive and rerun ACCTDEF.

SOURCE: ACCOUNTING

AC035 - PARAMETER CANNOT BE EQUATED

CLASS: Fatal

CAUSE: An error occurred on a call to process the parameters on a directive.

ACTION: Correct the directive and rerun ACCTDEF.

SOURCE: ACCOUNTING

AC036 - PARAMETER MUST HAVE A VALUE

CLASS: Fatal

CAUSE: An error occurred on a call to process the parameters on a directive.

ACTION: Correct the directive and rerun ACCTDEF.

SOURCE: ACCOUNTING

AC037 - TOO MANY PARAMETERS ON DIRECTIVE

CLASS: Fatal

CAUSE: An error occurred on a call to process the parameters on a directive.

ACTION: Correct the directive and rerun ACCTDEF.

SOURCE: ACCOUNTING

AC038 - INVALID RETURN STATUS, CANNOT RECOVER

CLASS: Fatal

CAUSE: An error occurred on a call to process the parameters on a directive.

ACTION: Correct the directive and rerun ACCTDEF.

SOURCE: ACCOUNTING

AC100 - 16 CHARACTER ACCOUNT NUMBER ILLEGAL

CLASS: Fatal

CAUSE: A 16-character account number was specified on the ACCOUNT statement. Only 15 characters are allowed.

ACTION: Correct the account number.

SOURCE: ACCOUNTING

AC101 - 16 CHARACTER PASSWORD ILLEGAL

CLASS: Fatal

CAUSE: A 16-character password was specified on the ACCOUNT statement. Only 15 characters are allowed.

ACTION: Correct the password.

SOURCE: ACCOUNTING

AC102 - ACCOUNTING DATASET DOES NOT EXIST

CLASS: Fatal

CAUSE: The accounting dataset does not exist.

ACTION: If accounting is not mandatory, remove the ACCOUNT control statement from the job deck. If accounting is mandatory, see a systems analyst.

SOURCE: ACCOUNTING

AC103 - ERROR ENCOUNTERED ON ACCESS OF ACCOUNTING DATASET

CLASS: Fatal

CAUSE: The accounting dataset was not accessed.

ACTION: The accounting dataset may be destroyed. See a systems analyst.

SOURCE: ACCOUNTING

AC104 - INVALID ACCOUNT NUMBER/PASSWORD

CLASS: Fatal

CAUSE: The account number and password specified on the ACCOUNT statement are not in the accounting dataset.

ACTION: Check account number and password. See the analyst in charge for a proper account number and password.

SOURCE: ACCOUNTING

AC105 - ACCOUNT NUMBER/PASSWORD IS INACTIVE

CLASS: Fatal

CAUSE: The account number and password specified on the account control statement is inactive and, therefore, invalid.

ACTION: Use an active account number and password.

SOURCE: ACCOUNTING

AC106 - PASSWORD IS REQUIRED ON ACCOUNT CARD

CLASS: Fatal

CAUSE: A password on the ACCOUNT control statement is mandatory. No password was specified on the ACCOUNT statement.

ACTION: Obtain the appropriate password to be used with the account number.

SOURCE: ACCOUNTING

AC107 - ACCOUNTING IS MANDATORY

CLASS: Informative

CAUSE: Errors have occurred with accounting. Those errors must be corrected because accounting is mandatory.

ACTION: This message appears after a fatal error notice. Correct the fatal error.

SOURCE: ACCOUNTING

AD001 - BINARY MUST BE ABSOLUTE

CLASS: Job abort

CAUSE: A relocatable binary or bad data is present on the $BLD input dataset.

ACTION: Generate absolute binary load modules by adding the ABS pseudo instruction to the relocatable APML modules.

SOURCE: ADSTAPE

AD002 - LENGTH OR ORIGIN TOO LARGE

CLASS: Job abort

CAUSE: The binary load module on $BLD contains a binary text length or origin that is too large and that results in binary text being loaded beyond an I/O Processor field length of 65K parcels (16K words).

ACTION: Reduce the binary size or origin; reassemble.

SOURCE: ADSTAPE

AD003 - PREMATURE END OF RECORD

CLASS: Fatal; job abort

CAUSE: End of record was encountered on the $BLD dataset before all of the expected binary text or program descriptor table was read. The $BLD file probably does not contain binary load modules.

ACTION: Examine the job control language for errors.

SOURCE: ADSTAPE

AD004 - COMPILATION ERRORS

CLASS: Job abort

CAUSE: The binary load modules on $BLD contain assembly or compilation errors.

ACTION: Add a DEBUG parameter to the associated APML statements to allow use of the binary modules with errors; reassemble.

SOURCE: ADSTAPE

AU001 - SIZE PARAMETER VALUE ILLEGAL

CLASS: AB015; user-requested abort.

CAUSE: An error status was returned from the library routine DTB, called by AUDIT, to convert the control statement parameter SZ value from decimal to binary. Possible DTB routine error.

ACTION: Check the SZ parameter on the AUDIT control statement for a legal numeric value. Report any illegal value to the site analyst.

SOURCE: AUDIT

AU002 - DSC READ ERROR

CLASS: AB025; user-requested abort.

CAUSE: AUDIT encountered an error while reading the Dataset Catalog.

ACTION: See site analyst.

SOURCE: AUDIT

AU003 - NO DATASETS SELECTED

CLASS: No abort

CAUSE: AUDIT found no datasets meeting the control statement selection requirements.

ACTION: Not applicable

SOURCE: AUDIT

AU004 - ILLEGAL DATE *date* OR TIME *time*

CLASS: Not reprievable

CAUSE: The decimal value to be converted to a binary value was invalid.

ACTION: Correct the data being converted and resubmit the request.

SOURCE: $SYSLIB

BD001 - BUILD VERSION *x.xx mm/dd/yy*

CLASS: Informative

CAUSE: The BUILD program has begun its processing using the *x.xx* version with date *mm/dd/yy*.

ACTION: Not applicable

SOURCE: BUILD

CA000 - SYSTEM ERROR AT P=*addr*

CLASS: Job abort

CAUSE: CAL has detected an internal error and is unable to proceed.

ACTION: Refer the problem to a Cray systems analyst.

SOURCE: CAL

CA002 - *n* WARNING ERRORS IN *identnam*

CLASS: Warning

CAUSE: CAL issues this message for each program module in which warning errors are detected.

ACTION: Correct errors that caused program execution error messages.

SOURCE: CAL

CA003 - *n* FATAL ERRORS IN *identnam*

CLASS: Job abort if DEBUG not specified on CAL statement.

CAUSE: CAL issues this message for each program module in which fatal assembly errors are detected.

ACTION: Correct all fatal errors. See the CRAY-1 CAL Assembler Version 1 Reference Manual for a list of fatal errors.

SOURCE: CAL

CA004 - MISSING IDENT STATEMENT

CLASS: Warning

CAUSE: An END pseudo or end-of-file on the source dataset occurred before an IDENT pseudo instruction was found.

ACTION: Check the source dataset for matching IDENT and END pseudo instructions.

SOURCE: CAL

CA005 - MISSING END STATEMENT IN *identnam*

CLASS: Warning

CAUSE: On the source dataset, an end-of-file occurred before an END pseudo instruction corresponding to the IDENT pseudo with the program name *identnam*.

ACTION: Check the source dataset for matching IDENT and END pseudo instructions.

SOURCE: CAL

CA006 - EMPTY SOURCE FILE, DN=*dn*

CLASS: Warning

CAUSE: An end-of-file or end-of-data was encountered on the source dataset before any source statements.

ACTION: Check the job control statements and the source dataset for a problem that causes a null file.

SOURCE: CAL

CA008 - *n* RECORDS EXCEED 90 CHARACTERS, DN = *dn*

CLASS: Informative

CAUSE: The given number of records in the named dataset contain more than 90 characters. The most typical cause is UPDATE sequence numbers that extend past column 90. (CAL truncates the long records to 90 characters). This message is issued also when a binary dataset is erroneously read.

ACTION: If the records exceed 90 characters, break up the long records with continuation lines.

SOURCE: CAL

CA012 - *n*K MEMORY + *m*K I/O BUFFERS USED

CLASS: Informative

CAUSE: All programs in the current file of the source dataset have been assembled. *n* is the octal number of 512-word memory blocks required in the user portion of the job field. *m* is the octal number of 512-word blocks required in the I/O table and buffer area of this job field.

ACTION: Not applicable

SOURCE: CAL

CA013 - BAD BINARY TEXT, DN = *dn* (ERROR CODE = *xn*)

CLASS: Fatal

CAUSE: An error was discovered in the binary system text *dn*. Following are the error codes and their meanings.

Error code	Meaning
P1	Prologue field BSTTT ≠ 1
P2	Prologue field BSTWC less than LE@BSTPR
P3	EOR encountered while prologue was being read
P4	EOF, EOD, or null record encountered while prologue was being read
H1	EOF, EOD, or null record encountered while subtable header was being read
H2	Header field BSTTT ≠ 1
H3	Header field BSTWC less than 1
H4	Header field BSTID not recognized

Error code	Meaning
M1	EOR encountered while TMDF was being read
M2	EOF, EOD, or null record encountered while TMDF was being read
M3	Length of TMDF entry less than 0
M4	Length of TMDF entry = 0
M5	Global word count exceeded during TMDF processing
S1	EOR encountered while TSYM entry was being read
S2	EOF, EOD, or null record encountered while TSYM entry was being read
S3	Global word count exceeded during TSYM processing
E1	Epilogue field BSTWC ≠ 1
E2	Global word count not equal to sum of subtable word counts

ACTION: Generate a new binary system text from the original source system text and rerun the job with the new binary system text; or rerun the job with the source system text in place of the binary system text; or show listing and DSDUMP output of offending binary system text to a Cray Research Analyst.

SOURCE: CAL

CA014 - *symbol* DOUBLY-DEFINED IN BINARY TEXT *text*

CLASS: Fatal

CAUSE: The named symbol is defined in the named binary system text but was defined differently in a previous system text.

ACTION: Remove one of the offending definitions from the source system texts, generate a new binary system text, and resubmit job.

SOURCE: CAL

CA015 - MACRO *opsyn* NOT FOUND; BINARY TEXT *text*

CLASS: Fatal

CAUSE: The named binary system text contains an OPSYN directive of the form *name* OPSYN *opsyn*, but no macro or pseudo-op with the name *opsyn* is known to the assembler.

ACTION: Correct the spelling of *opsyn*; or remove the OPSYN from the named system text; or define the offending macro in a previous system text or prior to the OPSYN directive in the named system text.

SOURCE: CAL

CA016 - MACRO *mmmmmmmm* redefined, DN=*xxxxxxxx*

CLASS: Informative

CAUSE: A definition for the named macro appears in the named dataset, but the macro has been previously defined.

ACTION: If the redefinition is intentional, the new definition will be used; otherwise, remove the unwanted macro definition(s).

SOURCE: CAL

CC001 - CONTROL STATEMENT TERMINATOR MISSING

CLASS: Not reprievable

CAUSE: A terminator character - either a period or a right parenthesis - was expected and not found in the control statement.

ACTION: Correct the control statement and resubmit the request.

SOURCE: $SYSLIB

CC001 - DIRECTIVE (*name*) TERMINATOR MISSING

CLASS: Fatal

CAUSE: See message.

ACTION: Correct directive.

SOURCE: LDR

CC002 - LITERAL SEPARATOR MISSING

CLASS: Not reprievable

CAUSE: The beginning of a literal string was found but no data followed the apostrophe.

ACTION: Correct the statement and reissue the request.

SOURCE: $SYSLIB

CC003 - CONTROL STATEMENT TABLE OVERFLOW - USE CONTINUATION CARDS

CLASS: Not reprievable

CAUSE: In processing the control statement, CCS exceeded the size of the expanded control statement area in the JCB.

ACTION: Separate the control statement parameters into additional statements and reissue the request.

SOURCE: $SYSLIB

CC003 - DIRECTIVE (*name*) TABLE OVERFLOW - USE CONTINUATION CARDS

CLASS: Fatal

CAUSE: See message.

ACTION: Correct directive.

SOURCE: LDR

CC004 - CONTROL STATEMENT CONTAINS ILLEGAL CHARACTER

CLASS: Not reprievable

CAUSE: A character in the control statement is not within the range of allowable characters.

ACTION: Correct the control statement and reissue the request.

SOURCE: $SYSLIB

CC004 - DIRECTIVE (*name*) CONTAINS ILLEGAL CHARACTER

CLASS: Fatal

CAUSE: See message.

ACTION: Correct directive.

SOURCE: LDR

CC005 - NO CONTROL STATEMENT IN JCB

CLASS: Not reprievable

CAUSE: The $CCS routine was called but the control statement field in the JCB was empty.

ACTION: Correct the program that called $CCS and resubmit the job.

SOURCE: $SYSLIB

CC005 - NO DIRECTIVE STATEMENT

CLASS: Fatal

CAUSE: See message.

ACTION: Correct directive.

SOURCE: LDR

CC006 - MISSING OR EXTRA OPEN PARENTHESES

CLASS: Not reprievable

CAUSE: The control statement does not contain balanced parentheses.

ACTION: Ensure that every open parenthesis is paired with a closed parenthesis.

SOURCE: $SYSLIB

CF000 - CFT VERSION - *mm/dd/yy x.xx*

CLASS: Informative

CAUSE: Initiation of CFT. *mm/dd/yy* is the date this version of CFT was assembled (e.g., 02/29/80). *x.xx* is the release level (e.g., 1.09).

ACTION: Not applicable

SOURCE: CFT compile time

CF001 - COMPILE TIME = *xxxxx.xxxx* SECONDS

CLASS: Informative

CAUSE: Termination of compilation

ACTION: Not applicable

SOURCE: CFT compile time

CF002 - *n* LINES, *m* STATEMENTS

CLASS: Informative

CAUSE: Termination of compilation. There were *n* lines compiled comprising *m* FORTRAN statements.

ACTION: Not applicable

SOURCE: CFT run-time

CF004 - 1 ERROR
CF004 - *n* ERRORS } Alternate forms

CLASS: Informative

CAUSE: An error count was encountered during compilation.

ACTION: Fix errors; recompile.

SOURCE: CFT run-time

CF005 - JOB ABORTED

CLASS: Informative

CAUSE: Fatal errors were encountered during compilation and ON=A is in effect.

ACTION: Fix errors; recompile.

SOURCE: CFT run-time

CH001 - ILLEGAL SR PARAMETER OPTION

CLASS: Fatal

CAUSE: The SR parameter on the CHARGES control statement was assigned an illegal or unavailable option.

ACTION: Correct the SR parameter option in error.

SOURCE: CHARGES

CL001 - JOB CLASS STRUCTURE INVOKED: *class*

CLASS: Informative

CAUSE: The job class structure named in the message has been invoked.

ACTION: Not applicable

SOURCE: JSH

CL003 - JOB--------ASSIGNED TO CLASS-------------P = *nn.nn**

CLASS: Informative

CAUSE: A job is assigned to a class.

ACTION: Not applicable

SOURCE: JSH

CJ001 - JOB DOESN'T FIT INTO ANY CLASS

CLASS: Fatal

CAUSE: Invalid class assignment

ACTION: Change the job card parameters so that the job can fit into a class.

SOURCE: CSP

CJ002 - CLASS SPECIFIED BY CL PARAMETER NOT FOUND

CLASS: Fatal

CAUSE: The class specified by the CL parameter cannot be found. The job aborts.

ACTION: Change the CL parameter to a recognizable class.

SOURCE: CSP

CJ003 - JOB DOESN'T FIT CLASS SPEC'D BY CL PARAMETER

CLASS: Fatal

CAUSE: Invalid class assignment by the CL parameter.

ACTION: Modify the CL parameter to fit the job into the class.

SOURCE: CSP

CM001 - *n* FILES READ - *m* RECORDS READ - *p* DIFFERENCES

CLASS: Informative

CAUSE: This message summarizes the results of COMPARE for one dataset. COMPARE compares complete datasets, giving the number of files, the number of records, and the number of differences in the datasets. If the two datasets have different record or file counts, or modified record synchronization (by content scanning), the results are approximate.

ACTION: Not applicable

SOURCE: COMPARE

CM002 - MORE THAN *n* DIFFERENCES. JOB TERMINATED

CLASS: AB025; user-requested abort

CAUSE: Number of differences exceeds value specified on ME parameter.

ACTION: Correct possible endless loop or increase ME parameter on the control statement if more differences are desired.

SOURCE: COMPARE

CM003 - A FILE MAY NOT BE COMPARED TO ITSELF

CLASS: AB025; user-requested abort

CAUSE: The value of the A and B parameters on the control statement are the same. This is not legal for COMPARE.

ACTION: Change the value of either A or B.

SOURCE: COMPARE

CM004 - OUTPUT MAY NOT BE WRITTEN TO AN INPUT FILE

CLASS: AB025; user-requested abort

CAUSE: The value of the L parameter is the same as the value of either the A or the B parameter.

ACTION: Change one of the values.

SOURCE: COMPARE

CM005 - CONTEXT MODE NOT IMPLEMENTED FOR BINARY MODE. COMPARE RUN WITHOUT CONTEXT PRINTING

CLASS: No abort

CAUSE: Context printing and binary mode options were both selected on the control statement. This combination is not handled by COMPARE. The comparison is made in binary mode without context printing.

ACTION: Do not select both options at once. If the binary mode is selected, the context can be obtained by using DSDUMP.

SOURCE: COMPARE

CM006 - CP < 0 NOT ALLOWED. COMPARE RUN WITHOUT CONTEXT PRINTING

CLASS: No abort

CAUSE: A negative value was illegally selected for the CP parameter. COMPARE does the comparison without context printing.

ACTION: Correct the value of the CP parameter.

SOURCE: COMPARE

CM007 - COMPARE WIDTH NOT IMPLEMENTED FOR BINARY MODE

CLASS: Fatal

CAUSE: Both the CW and the DF=B parameters were selected in a JOB statement.

ACTION: Change the CW parameter or the DF parameter.

SOURCE: COMPARE

CM008 - PROGRAM ERROR

CLASS: Fatal

CAUSE: A program logic error was encountered.

ACTION: See a Cray Research analyst.

SOURCE: COMPARE

CS000 - LINE *xxx* ==> *card image*

CLASS: Informative

CAUSE: An error has been detected during JCL block structure validation that concerns this line of the procedure dataset. The exact error message follows this message.

ACTION: Correct the problem as indicated by the error message that follows this message.

SOURCE: CSP

CS001 - FIRST STATEMENT NOT JOB STATEMENT

CLASS: Not reprievable; job terminated by CSP

CAUSE: First control statement is not a 'JOB' statement; syntax error occurred in JOB statement; job input dataset is not blocked; or required JOB parameters are missing or badly equated.

ACTION: See the CRAY-OS Version 1 Reference Manual for the format of the JOB statement; block the job if necessary.

SOURCE: CSP

CS002 - SYSTEM ERROR ON LOAD

CLASS: Not reprievable; job terminated by CSP

CAUSE: An internal COS error was detected by CSP.

ACTION: See the CRAY-OS Version 1 Reference Manual for the format of the JOB statement; block the job if necessary.

SOURCE: CSP

CS003 - SYSTEM ERROR ON RELEASE

CLASS: Not reprievable; job terminated by CSP

CAUSE: An internal COS error was detected by CSP.

ACTION: See the CRAY-OS Version 1 Reference Manual for the format of the JOB statement; block the job if necessary.

SOURCE: CSP

CS005 - ILLEGAL PARAMETER VALUE OPTION

CLASS: Not reprievable; CSP abort

CAUSE: IOAREA parameter is not LOCK or UNLOCK; NORERUN parameter is not ENABLE or DISABLE; RERUN parameter is not ENABLE or DISABLE; or SWITCH parameter value is not ON or OFF.

ACTION: Not applicable

SOURCE: CSP

CS011 - *n* PARAMETER IN ERROR

CLASS: Not reprievable; job terminated by CSP

CAUSE: The JOB statement contains an invalid parameter (for example, an illegal priority).

ACTION: See the CRAY-OS Version 1 Reference Manual for the format of the JOB statement; block the job if necessary.

SOURCE: CSP

CS015 - BAD LIBRARY DIRECTORY: *n*

CLASS: Not reprievable; CSP abort

CAUSE: A user-defined library dataset (from the LIBRARY control statement) has a bad directory table. Either the named dataset is not a library or the library directory for the named dataset has been destroyed.

ACTION: Not applicable

SOURCE: CSP

CS016 - PREMATURE END OF PROGRAM MODULE

CLASS: Not reprievable; CSP abort

CAUSE: While reading the dataset containing an absolute program module, CSP detected an abnormal end. The absolute module is probably bad.

ACTION: Reload the program.

SOURCE: CSP

CS017 - INVALID PROGRAM LOAD TABLE

CLASS: Not reprievable; CSP abort

CAUSE: An absolute program module contains invalid LDR table or the program is not an absoulte load module.

ACTION: Not applicable

SOURCE: CSP

CS018 - FATAL PROGRAM COMPILATION ERRORS

CLASS: Not reprievable; CSP abort

CAUSE: The program contains compilation or assembly errors.

ACTION: Correct the program and rerun.

SOURCE: CSP

CS019 - PROGRAM IS NOT AN ABSOLUTE MODULE

CLASS: Not reprievable; CSP abort

CAUSE: The program is not a valid LDR absolute module.

ACTION: Load the program with LDR.

SOURCE: CSP

CS020 - ILLEGAL PROGRAM LOAD ADDRESS

CLASS: Not reprievable; CSP abort

CAUSE: For CAL programs, the program origin of an absolute module is not O'200.

ACTION: Set the origin to O'200.

SOURCE: CSP

CS021 - NOT ENOUGH MEMORY TO LOAD PROGRAM

CLASS: Not reprievable; CSP abort

CAUSE: The CRAY-1 installation parameter for the maximum memory size a user job can request is not large enough to load the program.

ACTION: Restructure the program to use overlays; see a system analyst.

SOURCE: CSP

CS024 - ABSOLUTE MODULE READ ERROR

CLASS: Not reprievable; CSP abort

CAUSE: An I/O error occurred during a CSP read of an absolute module dataset.

ACTION: The disk copy of the dataset might be bad; try again.

SOURCE: CSP

CS034 - NO ACCOUNT STATEMENT - ACCOUNTING MANDATORY

CLASS: Fatal

CAUSE: Accounting is mandatory; the job did not contain an ACCOUNT statement.

ACTION: Put an ACCOUNT statement in the job deck.

SOURCE: ACCOUNTING

CS035 - INVALID LOCATION OF ACCOUNT STATEMENT

CLASS: Fatal

CAUSE: The ACCOUNT statement did not appear after the JOB card.

ACTION: Place the ACCOUNT card immediately following the JOB card.

SOURCE: ACCOUNTING

CS036 - SYNTAX ERROR ON ACCOUNT STATEMENT

CLASS: Fatal

CAUSE: A syntax error was encountered on the ACCOUNT statement.

ACTION: Correct the ACCOUNT statement.

SOURCE: ACCOUNTING

CS037 - CONTINUATION NOT ALLOWED WITH ACCOUNT CARD

CLASS: Fatal

CAUSE: The ACCOUNT statement was illegally continued.

ACTION: Remove the continuation.

SOURCE: ACCOUNTING

CS038 - *verb* DOES NOT APPEAR WITHIN A *blocktype*

CLASS: Not reprievable; CSP abort

CAUSE: A statement that has either ELSE, ELSEIF, or EXITLOOP as the verb does not appear within a conditional or iterative block.

ACTION: If the statement is intended to be a component of a block, then place it within the block.

SOURCE: CSP

CS039 - INVALID ELSE IN CONDITIONAL BLOCK

CLASS: Not reprievable; CSP abort

CAUSE: Either an ELSE has appeared before an ELSEIF or more than one ELSE is in the conditional block.

ACTION: Place all ELSEIF blocks prior to the ELSE block or remove the extra ELSE block.

SOURCE: CSP

CS040 - INCOMPLETE CONSTRUCTION OF *blocktype*

CLASS: Not reprievable; CSP abort

CAUSE: A block type of CONDITIONAL, ITERATIVE, or PROCEDURE DEFINITION did not have an ending block delimiter of ENDIF, ENDLOOP, or ENDPROC, respectively.

ACTION: Ensure the proper occurrence of the ending block delimiter.

SOURCE: CSP

CS041 - UNEXPECTED END OF CONTROL STATEMENT FILE

CLASS: Not reprievable

CAUSE: CSP has detected an end of control statement while it was skipping to the end of a JCL block.

ACTION: This condition should have been detected during block validation. Consult a Cray Research analyst.

SOURCE: CSP

CS042 - STRUCTURAL ERROR IN CONTROL STATEMENT FILE --- PROCEDURE NOT INVOKED

CLASS: Not reprievable; CSP abort

CAUSE: An error was detected within the block structure of a procedure which is to be invoked.

ACTION: Correct the reported errors.

SOURCE: CSP

CS043 - WARNING ITERATIVE BLOCK DOES NOT CONTAIN AN EXITLOOP STATEMENT

CLASS: Informative

CAUSE: A LOOP/ENDLOOP control statement sequence does not contain an EXITLOOP statement, causing the sequence to be iterated until a job step abort occurs.

ACTION: Add an EXITLOOP statement if desired.

SOURCE: CSP

CS044 - INVALID LIBRARY DATASET NAME: *dsname*

CLASS: Not reprievable; CSP abort

CAUSE: The name is not a valid dataset name and therefore cannot be located.

ACTION: Remove the name from the library searchlist and correct its entry on the library searchlist.

SOURCE: CSP

CS045 - NON-EXECUTABLE STATEMENT ENCOUNTERED

CLASS: Not reprievable; CSP abort

CAUSE: An interactive job has tried to enter, via the terminal, a control statement which corresponds to a JCL block structure.

ACTION: Do not use block structure statements as interactive statements.

SOURCE: CSP

CS046 - JCL EXPRESSION IS TOO COMPLICATED

CLASS: Not reprievable; CSP abort

CAUSE: The expression has overflowed into the internal tables of the expression evaluator.

ACTION: Use parentheses to simplify the statement.

SOURCE: CSP

CS047 - CSP INTERNAL PROCESSING ERROR

CLASS: Not reprievable; CSP abort

CAUSE: The Control Statement Processor has detected its own error.

ACTION: Consult a Cray Research analyst.

SOURCE: CSP

CS048 - LIBRARY SEARCHLIST OVERFLOW

CLASS: Not reprievable; CSP abort

CAUSE: An attempt was made to expand the number of names in the library searchlist beyond 64.

ACTION: Consolidate the libraries.

SOURCE: CSP

CS049 - ILLEGAL CONTINUED CONTROL STATEMENT

CLASS: Not reprievable; CSP abort

CAUSE: A control statement was encountered in which the verb was followed by a continuation character.

ACTION: Place the continuation character after the statement's initial separator.

SOURCE: CSP

CS050 - EXPRESSION ERROR
- *reason*
- NEAR *data*

CLASS: Not reprievable

CAUSE: *reason* defined as one of the following:
UNBALANCED PARENTHESES
MISSING OPERATOR
ILLEGAL OPERATOR
ILLEGAL NUMBER
ILLEGAL SYMBOL
ILLEGAL LITERAL STRING
SYNTAX ERROR
OPERATOR NOT UNARY
EXPRESSION TOO COMPLICATED
LITERAL TOO LONG
TABLE OVERFLOW: SIMPLIFY

ACTION: Correct the error indicated by *reason*.

SOURCE: $SYSLIB

CS101 - BAD CHARACTER IN PROTOTYPE STATEMENT

CLASS: Not reprievable; CSP abort

CAUSE: A character in the prototype statement is not in the ASCII graphic set.

ACTION: Repair the procedure prototype statement.

SOURCE: CSP

CS102 - BAD LITERAL CHARACTER IN PROTOTYPE STATEMENT

CLASS: Not reprievable; CSP abort

CAUSE: A character in the prototype statement is not in the ASCII graphic set.

ACTION: Repair the procedure prototype statement.

SOURCE: CSP

CS103 - UNTERMINATED LINE IN PROTOTYPE STATEMENT

CLASS: Not reprievable; CSP abort

CAUSE: The prototype statement does not end with a terminator (period or right parenthesis).

ACTION: Repair the procedure prototype statement.

SOURCE: CSP

CS104 - UNTERMINATED LITERAL IN PROTOTYPE STATEMENT

CLASS: Not reprievable; CSP abort

CAUSE: A closing apostrophe is missing from a literal string specification.

ACTION: Repair the procedure prototype statement.

SOURCE: CSP

CS105 - PROTOTYPE SEPARATOR OUT OF SEQUENCE AFTER *n*

CLASS: Not reprievable; CSP abort

CAUSE: The prototype statement contains bad delimiter usage such as *key* = = ... or *key* = A/B/C. *n* is near the offending sequence.

ACTION: Repair the procedure prototype statement.

SOURCE: CSP

CS106 - RESERVED NAME DATA USED IN PROTOTYPE

CLASS: Not reprievable; CSP abort

CAUSE: The name DATA is reserved. It cannot be used in the prototype as a positional parameter or as a keyword.

ACTION: Repair the procedure prototype statement.

SOURCE: CSP

CS107 - DUPLICATED PARAMETER NAME IN PROTOTYPE: *n*

CLASS: Not reprievable; CSP abort

CAUSE: A keyword or a positional parameter specified by the prototype was not unique.

ACTION: Repair the procedure prototype statement.

SOURCE: CSP

CS108 - PARAMETER TOO LONG IN PROTOTYPE: *n*

CLASS: Not reprievable; CSP abort

CAUSE: A keyword or a positional parameter specified by the prototype was longer than eight characters.

ACTION: Repair the procedure prototype statement.

SOURCE: CSP

CS109 - POSITIONAL PARAMETER AFTER KEYWORDS IN PROTOTYPE: *n*

CLASS: Not reprievable; CSP abort

CAUSE: A positional parameter did not precede a keyword parameter in the prototype statement.

ACTION: Repair the procedure prototype statement.

SOURCE: CSP

CS110 - NULL PARAMETER IN PROTOTYPE STATEMENT

CLASS: Not reprievable; CSP abort

CAUSE: A keyword or a positional parameter specified by the prototype was less than one character long.

ACTION: Repair the procedure prototype statement.

SOURCE: CSP

CS111 - BAD CHARACTER IN PROCEDURE INVOCATION

CLASS: Not reprievable; CSP abort

CAUSE: See message

ACTION: Repair the procedure invocation statement.

SOURCE: CSP

CS112 - BAD LITERAL CHARACTER IN PROCEDURE INVOCATION

CLASS: Not reprievable; CSP abort

CAUSE: See message

ACTION: Repair the procedure invocation statement.

SOURCE: CSP

CS113 - UNTERMINATED LINE IN PROCEDURE INVOCATION

CLASS: Not reprievable; CSP abort

CAUSE: See message

ACTION: Repair the procedure invocation statement.

SOURCE: CSP

CS114 - UNTERMINATED LITERAL IN PROCEDURE INVOCATION

CLASS: Not reprievable; CSP abort

CAUSE: See message

ACTION: Repair the procedure invocation statement.

SOURCE: CSP

CS115 - SEPARATOR OUT OF SEQUENCE AFTER: *n*

CLASS: Not reprievable; CSP abort

CAUSE: See message

ACTION: Repair the procedure invocation statement.

SOURCE: CSP

CS116 - UNRECOGNIZED KEYWORD: *n*

CLASS: Not reprievable; CSP abort

CAUSE: A keyword specified by the invocation has no matching keyword in the prototype.

ACTION: Repair the procedure invocation statement.

SOURCE: CSP

CS117 - KEYWORD TOO LONG: *n*

CLASS: Not reprievable; CSP abort

CAUSE: A keyword is more than eight characters long.

ACTION: Repair the procedure invocation statement.

SOURCE: CSP

CS118 - POSITIONAL PARAMETER AFTER KEYWORDS: *n*

CLASS: Not reprievable; CSP abort

CAUSE: A positional parameter followed the introduction of keyword parameters in the invocation.

ACTION: Repair the procedure invocation statement.

SOURCE: CSP

CS119 - EXTRA POSITIONAL PARAMETER: *n*

CLASS: Not reprievable; CSP abort

CAUSE: Too many positional parameters were specified by the invocation statement.

ACTION: Repair the procedure invocation statement.

SOURCE: CSP

CS120 - DUPLICATED KEYWORD: *n*

CLASS: Not reprievable; CSP abort

CAUSE: The indicated keyword appears two or more times in the invocation

ACTION: Repair the procedure invocation statement.

SOURCE: CSP

CS121 - KEYWORD USED WITHOUT ASSIGNING IT A VALUE: *n*

CLASS: Not reprievable; CSP abort

CAUSE: A non-null value was not provided for the indicated keyword.

ACTION: Repair the procedure invocation statement.

SOURCE: CSP

CS122 - NO VALUE WAS ASSIGNED TO *n*

CLASS: Not reprievable; CSP abort

CAUSE: A non-null value was not provided for the indicated keyword.

ACTION: Repair the procedure invocation statement.

SOURCE: CSP

CS123 - &DATA NOT DELIMITED PROPERLY

CLASS: Not reprievable; CSP abort

CAUSE: The &DATA statement was used incorrectly in the procedure body.

ACTION: Repair the procedure body statement.

SOURCE: CSP

CS125 - NO SUCH FORMAL PARAMETER: *n*

CLASS: Not reprievable; CSP abort

CAUSE: The indicated string was preceded by an ampersand, signaling a substitutable string. However, the string was not specified by the prototype statement.

ACTION: Repair the procedure body statement.

SOURCE: CSP

CS126 - ZERO LENGTH PARAMETER NAME

CLASS: Not reprievable; CSP abort

CAUSE: A substitutable string is less than one character long.

ACTION: Repair the procedure body statement.

SOURCE: CSP

CS127 - PARAMETER NAME TOO LONG

CLASS: Not reprievable; CSP abort

CAUSE: A substitutable string is more than eight characters long.

ACTION: Repair the procedure body statement.

SOURCE: CSP

CS128 - UNTERMINATED LITERAL STRING

CLASS: Not reprievable; CSP abort

CAUSE: Apostrophes delimiting a literal string are not balanced.

ACTION: Repair the procedure body statement.

SOURCE: CSP

CS129 - REPLACEMENT STRING TOO LONG; PUT QUOTES AROUND &*n*

CLASS: Not reprievable; CSP abort

CAUSE: A replacement string was non-literal resulting in substitution overflow.

ACTION: Enclose the replacement string in string quotes within the procedure body.

SOURCE: CSP

CS130 - ILLEGAL TERMINATOR FOR &DATA FILENAME: *n*

CLASS: Not reprievable; CSP abort

CAUSE: The &DATA argument was improperly specified.

ACTION: Repair the procedure body statement.

SOURCE: CSP

CS131 - ZERO-LENGTH &DATA FILENAME

CLASS: Not reprievable; CSP abort

CAUSE: The &DATA argument was improperly specified.

ACTION: Repair the procedure body statement.

SOURCE: CSP

CS132 - &DATA FILENAME TOO LONG: *n*

CLASS: Not reprievable; CSP abort

CAUSE: The &DATA argument was improperly specified.

ACTION: Repair the procedure body statement.

SOURCE: CSP

CS133 - UNEXPECTED END OF CONTROL STATEMENT FILE

CLASS: Not reprievable; CSP abort

CAUSE: Premature end of control statement occurred while the procedure invocation was being interpreted.

ACTION:

SOURCE: CSP

CS134 - UNEXPECTED END OF PROCEDURE FILE

CLASS: Not reprievable; CSP abort

CAUSE: Premature end of procedure prototype and procedure body file occurred.

ACTION: Check the procedure file for completeness.

SOURCE: CSP

CT000 - JCB DESTROYED, JCSTN RESET

CLASS: Informative

CAUSE: The loader overwrote at least one word in the JCB that stores the step failed information.

ACTION: Reorganize the job stream or remove the offending step; resubmit the job.

SOURCE: STEP

CT000 - PASSED

CLASS: Informative

CAUSE: All job steps completed normally.

ACTION: Not applicable

SOURCE: STEP

CT000 - TEST FAILED

CLASS: Informative

CAUSE: The LAST= parameter was specified as FAIL. The test failed.

ACTION: Not applicable

SOURCE: STEP

CT000 - TEST INDETERMINATE

CLASS: Informative

CAUSE: The LAST= parameter was specified as INDET. Therefore, it is impossible to determine whether the test passed.

ACTION: Not applicable

SOURCE: STEP

DP001 - BAD DELIMITER FOLLOWING KEYWORD

CLASS: Fatal

CAUSE: Syntax error in directive

ACTION: Correct directive.

SOURCE: LDR

DP002 - KEYWORD (*name*) UNKNOWN OR DUPLICATED

CLASS: Fatal

CAUSE: See message.

ACTION: Correct directive.

SOURCE: LDR

DP003 - KEYWORD (*name*) MUST BE SPECIFIED

CLASS: Fatal

CAUSE: See message.

ACTION: Correct directive.

SOURCE: LDR

DP004 - KEYWORD (*name*) MUST BE EQUATED

CLASS: Fatal

CAUSE: See message.

ACTION: Correct directive.

SOURCE: LDR

DP005 - KEYWORD (*name*) CANNOT BE EQUATED

CLASS: Fatal

CAUSE: See message.

ACTION: Correct directive.

SOURCE: LDR

DP007 - CONTINUATION CARD NOT FOUND

CLASS: Fatal

CAUSE: See message.

ACTION: Correct directive.

SOURCE: LDR

FT000 - BAD CALL TO LIBRARY ERROR PROCESSOR

CLASS: 2; user abort; reprievable

CAUSE: The FTLIB error message processor was entered with an illegal error message code.

ACTION: See systems analyst.

SOURCE: $FTLIB

FT001 - BACKSPACE ON NULL DATASET

CLASS: 2; user abort; reprievable

CAUSE: Attempt was made to backspace on a dataset that does not exist.

ACTION: Adjust program logic to create the dataset before backspacing, perhaps by rewinding datasets.

SOURCE: $FTLIB

FT002 - REPETITION COUNT < 1

CLASS: 2; user abort; reprievable

CAUSE: The size of an array named without subscripts in an I/O statement is 0 or negative.

ACTION: Adjustable dimensional arrays must be correctly dimensioned.

SOURCE: $FTLIB

FT003 - CALL OUT OF SEQUENCE

CLASS: 2; user abort; reprievable

CAUSE: A routine was called out of sequence. The FTLIB I/O routines are called in the following sequence.

1. Initialize
2. Process list items
3. Terminate

ACTION: Check for the following errors:

Compiler error
Error in CAL routine coding
Illegal arithmetic on variables assigned a label

SOURCE: $FTLIB

FT004 - READ PAST EOD

CLASS: 2; user abort; reprievable

CAUSE: Attempt was made to read a file after the end of data had been read.

ACTION: Test for EOF and EOI WITH END= or UNIT or EOF functions.

SOURCE: $FTLIB

FT005 - DECODE RECORD LENGTH < 1 OR > 152

CLASS: 2; user abort; reprievable

CAUSE: The record length (n parameter) in a DECODE statement is less than 1 or greater than 152.

ACTION: Correct n parameter.

SOURCE: $FTLIB

FT006 - NUMERIC : ILLEGAL CHARACTER

CLASS: 2; user abort; reprievable

CAUSE: An illegal character appears in an input field being converted by a format (e.g., a 9 in an octal number or a Q in a decimal number).

ACTION: Determine if the correct data line is being processed by the correct format statement. Count columns. Check for errors in data fields.

SOURCE: $FTLIB

FT007 - NUMERIC : OVERFLOW

CLASS: 2; user abort; reprievable

CAUSE: A number being converted by a format on input exceeds the maximum allowable value for that type of number.

ACTION: Check format and data record for correct match. Count columns. Embedded and trailing blanks are interpreted as 0.

SOURCE: $FTLIB

FT008 - EXPONENT OVERFLOW

CLASS: 2; user abort; reprievable

CAUSE: A floating-point number being converted by a format on input has an exponent larger than 2466.

ACTION: Check format and data line. Count columns. Trailing and embedded blanks are interpreted as zero.

SOURCE: $FTLIB

FT009 - EXPONENT UNDERFLOW

CLASS: 2; user abort; reprievable

CAUSE: A floating-point number being converted by a format on input has an exponent less than -2466.

ACTION: Check format and data line.

SOURCE: $FTLIB

FT010 - NULL FIELD

CLASS: 2; user abort; reprievable

CAUSE: No characters appear in a field being converted on input by a format.

ACTION: Check format and data lines.

SOURCE: $FTLIB

FT011 - ILLEGAL LOGICAL VALUE

CLASS: 2; user abort; reprievable

CAUSE: Input format conversion fields for logical values must begin with 'T', '.T', 'F', or '.F'.

ACTION: Check format and data lines. Count columns. Check for errors.

SOURCE: $FTLIB

FT012 - POSITION < 1 or > 152

CLASS: 2; user abort; reprievable

CAUSE: An attempt was made to position before column 1 or after column 152 during formatted I/O.

ACTION: Check format. Look for fields that are too large and for repeated parenthesis groups without an embedded slash. Tab or X count might be too large.

SOURCE: $FTLIB

FT013 - FORMAT MUST BEGIN WITH (

CLASS: 2; user abort; reprievable

CAUSE: Formats must begin with a left parenthesis in either character position 1 or 9.

ACTION: Check variable formats (explicit formats are checked at compile time).

SOURCE: $FTLIB

FT014 - FIELD SPECIFICATION MISSING

CLASS: 2; user abort; reprievable

CAUSE: No width was specified after a format character (e.g., I rather than I5).

ACTION: Check variable formats (explicit formats are checked at compile time).

SOURCE: $FTLIB

FT015 - ILLEGAL SEQUENCE OF CHARACTERS

CLASS: 2; user abort; reprievable

CAUSE: A field is missing in a format (e.g., -7 must be followed by X or P).

ACTION: Check variable formats (explicit formats are checked at compile time).

SOURCE: $FTLIB

FT016 - UNMATCHED OR TOO MANY PARENTHESES

CLASS: 2; user abort; reprievable

CAUSE: Parentheses were nested more than nine deep in a format or parentheses are out of order.

ACTION: Check variable formats (explicit formats are checked at compile time).

SOURCE: $FTLIB

FT017 - ILLEGAL FORMAT CHARACTER

CLASS: 2; user abort; reprievable

CAUSE: An unrecognized character appeared in a format.

ACTION: Check variable formats (explicit formats are checked at compile time).

SOURCE: $FTLIB

FT018 - ILLEGAL FIELD WIDTH

CLASS: 2; user abort; reprievable

CAUSE: Improper field width (e.g., the d width is greater than the w width as in F10.20).

ACTION: Check variable formats (explicit formats are checked at compile time).

SOURCE: $FTLIB

FT019 - VALUE AND SPECIFICATION TYPE DIFFER

CLASS: 2; user abort; reprievable

CAUSE: A variable is being read or written under format control and the format conversion specification is inappropriate for the variable type (e.g., an integer being printed with an F20.10 specification).

ACTION: Check format and I/O list. Determine if variables are properly typed (and spelled). If necessary, a typeless function can be used. For example, PRINT 10, OR(X,0) instead of PRINT 10, X.

SOURCE: $FTLIB

FT020 - RECORD LENGTH EXCEEDED

CLASS: 2; user abort; reprievable

CAUSE: One of the following errors exists:

1. An attempt was made to read more columns than were supplied on an input record.
2. An attempt was made to ENCODE or DECODE more columns than the *n* parameter specified.
3. An unformatted read called for more values than the record contained.

ACTION: Check format. Check encode/decode record length. Check for proper array dimensions. Determine if the expected record is being read.

SOURCE: $FTLIB

FT021 - WRITE PAST END OF ALLOCATED AREA

CLASS: 2; user abort; reprievable

CAUSE: An attempt was made to write a larger dataset than was allocated.

ACTION: Check allocation size. Check for program in an output loop.

SOURCE: $FTLIB

FT022 - ENCODE RECORD LENGTH < 1 OR > 152

CLASS: 2; user abort; reprievable

CAUSE: The record length (the *n* parameter) in a ENCODE statement is less than 1 or greater than 152.

ACTION: Correct *n* parameter.

SOURCE: $FTLIB

FT023 - READ AFTER UNCLEARED ERROR

CLASS: 2; user abort; reprievable

CAUSE: An error occurred on a buffer in/out operation and a subsequent buffer operation was attempted on the same file without an intervening UNIT function.

ACTION: Use the UNIT function to detect I/O errors.

SOURCE: $FTLIB

FT024 - BUFFER WORD COUNT < 1

CLASS: 2; user abort; reprievable

CAUSE: Input buffer word count must be 1 or more. Output buffer word count can be 0 to write a null read or to terminate a series of partial buffers.

ACTION: Check subscripts in arrays in buffer in/out statements. Determine if both list items are in either the same array or the same common block and if the second follows the first in storage order.

SOURCE: $FTLIB

FT025 - RECORD '*name*' ON '*unit*' SKIPPED

CLASS: Informative logfile message

CAUSE: During search for a namelist input record, a record with name *name* was encountered and skipped.

ACTION: Determine if NAMELIST input record and read statements are in the correct order. Disable the message by calling RNLSKIP(0).

SOURCE: $FTLIB

FT026 - ILLEGAL TYPE CONVERSION

CLASS: 2; user abort; reprievable

CAUSE: A type conversion that is not defined (e.g.,. LOGICAL=1) was encountered while NAMELIST input was being processed.

ACTION: Check item types and spelling. Check for error in data. Determine if RNLTYPE was called to disable all type conversion.

SOURCE: $FTLIB

FT027 - UNRECOGNISED DATA TYPE

CLASS: 2; user abort; reprievable

CAUSE: System error in namelist input conversion

ACTION: See systems analyst.

SOURCE: $FTLIB

FT028 - ERROR IN CONSTANT FIELD

CLASS: 2; user abort; reprievable

CAUSE: Illegal or inappropriate character appeared in a constant being processed by namelist input.

ACTION: Examine data line for erroneous character.

SOURCE: $FTLIB

FT029 - BAD SUBSCRIPT

CLASS: 2; user abort; reprievable

CAUSE: Illegal subscript appeared in a namelist input record.

ACTION: Examine data line for spelling errors.

SOURCE: $FTLIB

FT030 - TOO MANY SUBSCRIPTS

CLASS: 2; user abort; reprievable

CAUSE: More subscripts were specified on a namelist data line for an array than were declared in a DIMENSION statement.

ACTION: Examine data line for spelling errors.

SOURCE: $FTLIB

FT031 - NOT FOLLOWED BY REPLACEMENT CHARACTER

CLASS: 2; user abort; reprievable

CAUSE: The namelist input item was not followed by a legal replacement separator.

ACTION: Examine data line. Call RNLREP to add a legal separator; check calls to other RNL routines for consistency.

SOURCE: $FTLIB

FT032 - NAME NOT IN GROUP LIST

CLASS: 2; user abort; reprievable

CAUSE: A namelist input line contains a variable name that is not in the group namelist for this record.

ACTION: Examine data line for spelling errors. Check namelist definition.

SOURCE: $FTLIB

FT033 - CONSTANT NOT PROPERLY TERMINATED

CLASS: 2; user abort; reprievable

CAUSE: Improper or missing terminator on a namelist input record.

ACTION: Examine data line. Call RNLSEP to add separation character. Check other calls to RNL routines for consistency.

SOURCE: $FTLIB

FT034 - ILLEGAL CHARACTER IN CONSTANT

CLASS: 2; user abort; reprievable

CAUSE: A namelist input record contains a character that is illegal (e.g., 9B).

ACTION: Examine data line for erroneous character.

SOURCE: $FTLIB

FT035 - NAME TOO LONG

CLASS: 2; user abort; reprievable

CAUSE: A namelist input line contains a variable name with more than 8 characters.

ACTION: Examine data line for spelling errors. Examine calls to RNLREP for consistency. An alphanumeric as a replacement separator is illegal without a separating space code.

SOURCE: $FTLIB

FT036 - NAME MUST BEGIN WITH ALPHABETIC CHARACTER

CLASS: 2; user abort; reprievable

CAUSE: The namelist data record contains a variable name that does not begin with an alphabetic character.

ACTION: Examine data line. Look for missing comment separator or spelling error. Add call to RNLCOMM. Check calls to other RNL routines for consistency.

SOURCE: $FTLIB

FT037 - MISSING OR ILLEGAL CONSTANT

CLASS: 2; user abort; reprievable

CAUSE: A namelist input record has a missing or unrecognizable constant.

ACTION: Examine data line for spelling errors. Check for inconsistent calls to RNL routines.

SOURCE: $FTLIB

FT038 - CONSTANT LIST EXCEEDS VARIABLE LIST

CLASS: 2; user abort; reprievable

CAUSE: More constants were specified for a namelist input item than could be used to fill the item.

ACTION: Examine data line. Look for extra delimiters. Check array dimensions and spelling.

SOURCE: $FTLIB

FT039 - NAMELIST RECORD OUT OF ORDER

CLASS: 2; user abort; reprievable

CAUSE: A namelist record with a name different from the one being searched for was encountered. RNLSKIP had been previously called to make this a fatal error.

ACTION: Check order of reads and records. Call RNLSKIP to ignore this condition.

SOURCE: $FTLIB

FT040 - READ PAST EOF OR EOD

CLASS: 2; user abort; reprievable

CAUSE: An end of file or end of data was encountered while a search was being made for a namelist record. No END= branch was supplied.

ACTION: Check spelling of data line record names. Check order of reads and records. Use END= specifier.

SOURCE: $FTLIB

FT041 - ILLEGAL CHARACTER ARGUMENT

CLASS: 2; user abort; reprievable

CAUSE: Character arguments to the RNL or WNL routines must be in the range 1 to 127 inclusive.

ACTION: Check call.

SOURCE: $FTLIB

FT042 - LENGTH OUT OF RANGE

CLASS: 2; user abort; reprievable

CAUSE: The length parameter for WNLLONG must be either -1 or in the range 9 to 160 inclusive.

ACTION: Check call.

SOURCE: $FTLIB

FT043 - LINE LENGTH TOO SHORT FOR NAMELIST OUTPUT

CLASS: 2; user abort; reprievable

CAUSE: The length defined by a call to WNLLONG is not long enough to write the particular list item.

ACTION: Use a longer line (or use smaller names and values).

SOURCE: $FTLIB

FT044 - UNRECOGNIZED VARIABLE TYPE

CLASS: 2; user abort; reprievable

CAUSE: The variable type defined in the namelist description table is not recognized by the namelist routines. Possible compiler or storage error by user program.

ACTION: Check array subscripts. Try ON=0 (subscript bounds checking). See a Cray Research systems analyst.

SOURCE: $FTLIB

FT045 - READ AFTER WRITE OR EOD

CLASS: 2; user abort; reprievable

CAUSE: An attempt was made to read a dataset when the previous operation was a write or the previous operation read the end of data.

ACTION: Check program flow. Use END=. Check for missing REWIND.

SOURCE: $FTLIB

FT046 - DIVIDE BY ZERO

CLASS: 2; user abort; reprievable

CAUSE: An attempt was made to perform an integer divide by 0 or an attempt was made to use the integer MOD function with 0 for the second argument.

ACTION: Check for program error.

SOURCE: $FTLIB

FT047 - COPY OF *n* RECORDS COMPLETED

CLASS: Informative

CAUSE: The copy routine processed *n* records.

ACTION: Not applicable

SOURCE: $FTLIB

FT048 - COPY OF *n* RECORDS *m* FILES COMPLETED

CLASS: Informative

CAUSE: The copy routine copied *m* files that contained *n* records.

ACTION: Not applicable

SOURCE: $FTLIB

FT049 - ILLEGAL CHARACTER IN DECIMAL CONVERSION

CLASS: 2; user abort; reprievable

CAUSE: The DTB routine encountered a non-numeric character.

ACTION: Check character string input to DTB.

SOURCE: $FTLIB

FT050 - ILLEGAL CHARACTER IN OCTAL CONVERSION

CLASS: 2; user abort; reprievable

CAUSE: The octal character to binary converter (OTB) encountered a non-octal character.

ACTION: Check character string input to OTB.

SOURCE: $FTLIB

FT051 - BAD SCALAR ARGUMENT TO FTLIB MATH ROUTINE

CLASS: 2; user abort; reprievable

CAUSE: A scalar FTLIB math routine was called with one or more arguments that were out of range for the particular function (e.g., SQRT(-5.), ATAN(0,0.)).

ACTION: Consult traceback listing to determine function. Check for proper use of function. Note that some pseudo vector library routines call a vector routine to process a vector one element at a time.

SOURCE: $FTLIB

FT052 - BAD VECTOR ARGUMENT TO FTLIB MATH ROUTINE

CLASS: 2; user abort; reprievable

CAUSE: A vector FTLIB math routine was called with one or more elements of its first argument out of range for the particular function (e.g., SQRTV(-5.), ATANV(0,0.)). This message does not imply that the second argument is not in error. No further check is made.

ACTION: Consult traceback listing to determine function. Check for proper use of function. Note that some pseudo vector library routines call a vector routine to process a vector one element at a time.

SOURCE: $FTLIB

FT053 - BAD SECOND VECTOR ARGUMENT TO FTLIB MATH ROUTINE

CLASS: 2; user abort; reprievable

CAUSE: A vector FTLIB math routine was called with one or more elements of its second argument out of range for the particular function. This messaage does not imply that the first argument is not in error. No further check is made.

ACTION: Consult traceback listing to determine function. Check for proper use of function. Note that some pseudo vector library routines call a vector routine to process a vector one element at a time.

SOURCE: $FTLIB

FT054 - EXIT CALLED BY *routine*

CLASS: 1; normal termination; reprievable

CAUSE: Named routine performed a CALL EXIT.

ACTION: Not applicable

SOURCE: $FTLIB

FT055 - PDUMP ABORT AFTER COMPLETION

CLASS: 2; user abort; reprievable

CAUSE: The DUMP entry in PDUMP was called to request an abort after memory was dumped.

ACTION: Not applicable

SOURCE: $FTLIB

FT056 - PDUMP ERROR: LWA LESS THAN FWA, DUMP IGNORED

CLASS: Informative

CAUSE: The range specified to PDUMP is improper. The last word address must not be less than the first word address. The dump request is ignored.

ACTION: Correct call.

SOURCE: $FTLIB

FT057 - PDUMP ERROR: F LT 0 OR GT 7

CLASS: Informative

CAUSE: The third parameter to PDUMP, F, must be in the range 0 to 7 inclusive. The dump request is ignored.

ACTION: Correct call.

SOURCE: $FTLIB

FT058 - PDUMP ERROR: INCORRECT NUMBER OF ARGUMENTS

CLASS: 2; user abort; reprievable

CAUSE: PDUMP must be called with at least 3 arguments.

ACTION: Check call.

SOURCE: $FTLIB

FT059 - USER REMARK

CLASS: Informative

CAUSE: User call was issued to REMARK subroutine to write a message to the logfile.

ACTION: Not applicable

SOURCE: $FTLIB

FT060 - USER REMARK

CLASS: Informative

CAUSE: User call was issued to REMARKF to provide a message to the logfile under format control. REMARKF allows the FT060 - to be replaced if the first eight characters of the message are XXXXX - .

ACTION: Not applicable

SOURCE: $FTLIB

FT061 - INVALID BUFFER ADDRESS FOR UNBLOCKED DATASET

CLASS: 2; user abort; reprievable

CAUSE: The starting or ending address of an unblocked dataset operation is out of range.

ACTION: Check subscripts.

SOURCE: $FTLIB

FT062 - INVALID WORD COUNT FOR UNBLOCKED DATASET

CLASS: 2; user abort; reprievable

CAUSE: The array size on an unblocked I/O operation must be a multiple of 512 words.

ACTION: Correct dimension or subscripts.

SOURCE: $FTLIB

FT063 - STOP *n* IN *m*

CLASS: 1; normal termination; reprievable

CAUSE: A STOP *n* was executed in routine *m*.

ACTION: Not applicable

SOURCE: $FTLIB

FT064 - PAUSE NOT SUPPORTED, STOP SUBSTITUTED

CLASS: Informative

CAUSE: A PAUSE was executed. This version of FTLIB does not support PAUSE; substitute with STOP (see FT063).

ACTION: Not applicable

SOURCE: $FTLIB

FT065 - PAUSE *n* IN *m*

CLASS: Informative

CAUSE: A PAUSE *n* was executed in routine *m*. Operator intervention is requested to either resume execution or drop the job.

ACTION: Inform operator of required action.

SOURCE: $FTLIB

FT066 - DIRECT I/O PROHIBITED UNDER SEQUENTIAL ACCESS

CLASS: 2; user abort; reprievable

CAUSE: Attempt was made to read/write using the REC= control list option on a dataset that was accessed as sequential.

ACTION: Remove REC= from the control list of the I/O statement or close dataset and open it with direct access.

SOURCE: $FTLIB

FT067 - SEQUENTIAL I/O PROHIBITED UNDER DIRECT ACCESS

CLASS: 2; user abort; reprievable

CAUSE: An attempt was made to perform I/O without the REC= option while the unit was connected for direct access.

ACTION: Use REC= option in read/write control list or close dataset under direct access and open it under sequential access.

SOURCE: $FTLIB

FT068 - RECORD NUMBER MUST BE > ZERO FOR DIRECT ACCESS

CLASS: 2; user abort; reprievable

CAUSE: REC= control list option on read/write statement supplied a value less than or equal to 0. The parameter specifies a record number. A record number must be a positive integer.

ACTION: Check syntax of offending FORTRAN I/O statement. Change the REC= specifier to a positive integer.

SOURCE: $FTLIB

FT069 - RECORD LENGTH MUST BE > ZERO FOR DIRECT ACCESS

CLASS: 2; user abort; reprievable

CAUSE: FORTRAN OPEN statement specifies fixed record length for direct access connection. This record length must be a positive integer value.

ACTION: Check OPEN statement. Change RECL= control list specifier to a positive integer.

SOURCE: $FTLIB

FT070 - RECORD ADDRESS TOO LARGE FOR DIRECT ACCESS

CLASS: 2; user abort; reprievable

CAUSE: The address of the record number specified on the FORTRAN direct access read/write statement exceeds 24 bits.

ACTION: Verify the record number specified in the I/O statement. If the record number is valid, check the OPEN statement that connected the dataset to a unit. Verify the record length provided in the OPEN control list. If this is also correct, the maximum size under direct access has been reached.

SOURCE: $FTLIB

FT071 - UNFORMATTED RECORD LENGTH MUST BE A WORD BOUNDARY

CLASS: 2; user abort; reprievable

CAUSE: OPEN statement specified direct access and unformatted data. Therefore, the record length must be a multiple of eight characters. (Unformatted I/O is done on a word basis with eight characters per word.)

ACTION: Change the record length defined within the FORTRAN OPEN statement to a positive multiple of eight or change the I/O statement from unformatted to formatted I/O.

SOURCE: $FTLIB

FT072 - DIRECT READ ATTEMPTED BEYOND LAST RECORD

CLASS: 2; user abort

CAUSE: An attempt was made to read an unwritten record.

ACTION: Verify the record number specified in the READ statement. If the record number is valid, the requested record must be written before it can be read.

SOURCE: $FTLIB

FT075 - ILLEGAL UNIT NUMBER

CLASS: 2; user abort

CAUSE: Unit number specified was not an integer value in the range [0,102].

ACTION: Change the value of the unit number.

SOURCE: $FTLIB

FT076 - OLD STATUS ILLEGAL WITHOUT FILE SPECIFIER

CLASS: 1; user abort

CAUSE: Status was specified OLD but no file name was specified on the OPEN statement.

ACTION: Provide a file specifier.

SOURCE: $FTLIB

FT077 - NEW STATUS ILLEGAL WITHOUT FILE SPECIFIER

CLASS: 1; reprievable

CAUSE: Status was specified NEW but no file name was specified on the OPEN statement.

ACTION: Provide a file specifier or change the status specifier.

SOURCE: $FTLIB

FT078 - FILE ALREADY CONNECTED TO ANOTHER UNIT

CLASS: 1; reprievable

CAUSE: Execution of an OPEN statement was attempted on a file that was already connected to another unit.

ACTION: Close the file. Open the file, using the desired unit specifier.

SOURCE: $FTLIB

FT079 - SCRATCH STATUS ILLEGAL WITH A FILE SPECIFIER

CLASS: 1; reprievable

CAUSE: Status was specified SCRATCH with a named file.

ACTION: Change the status specifier or omit the file specifier.

SOURCE: $FTLIB

FT080 - RECL SPECIFIER ILLEGAL IF SEQUENTIAL ACCESS

CLASS: 1; reprievable

CAUSE: RECL was specified illegally when a dataset was being connected for sequential access. It can be specified only for direct access.

ACTION: Change the access method to direct or omit the RECL specifier.

SOURCE: $FTLIB

FT081 - ACCESS SPECIFIER MUST BE SEQUENTIAL OR DIRECT

CLASS: 1; reprievable

CAUSE: The access method specified in an OPEN statement is not SEQUENTIAL or DIRECT.

ACTION: Change the access specifier to SEQUENTIAL or DIRECT.

SOURCE: $FTLIB

FT082 - FORM SPECIFIER NOT RECOGNIZED

CLASS: 1; reprievable

CAUSE: The form specifier in an OPEN statement is neither FORMATTED nor UNFORMATTED.

ACTION: Change the form specifier to a legal value.

SOURCE: $FTLIB

FT083 - BLANK SPECIFIER NOT RECOGNIZED

CLASS: 1; reprievable

CAUSE: The blank specifier in an OPEN statement is neither NULL nor ZERO.

ACTION: Change the blank specifier to a legal value.

SOURCE: $FTLIB

FT084 - STATUS SPECIFIER NOT RECOGNIZED

CLASS: 1; reprievable

CAUSE: OPEN statement - status specified was not OLD, NEW SCRATCH or UNKNOWN.
CLOSE statement - status specified was not KEEP or DELETE.

ACTION: Change the status specifier to a legal value.

SOURCE: $FTLIB

FT085 - REQUESTED ACCESS CONFLICTS WITH EXISTING ACCESS

CLASS: 1; reprievable

CAUSE: The current access method conflicts with the method requested in the OPEN statement. Dataset is already connected to a unit.

ACTION: Close the dataset and reconnect the unit and dataset, using the requested access specifier; or change the access specifier to match the access method of the existing connection.

SOURCE: $FTLIB

FT086 - REQUESTED STATUS CONFLICTS WITH THE EXISTING STATUS

CLASS: 1; reprievable

CAUSE: Dataset is already connected to the unit. The current status conflicts with the status requested in the OPEN statement.

ACTION: Close the dataset and reconnect the unit and dataset, using the requested status specifier; or change the status specifier to match the status of the existing connection.

SOURCE: $FTLIB

FT087 - REQUESTED FORM CONFLICTS WITH THE EXISTING FORM

CLASS: 1; reprievable

CAUSE: The current form conflicts with the form requested in the OPEN statement. Dataset is already connected to the unit.

ACTION: Close the dataset and reconnect the unit and dataset, using the requested form specifier; or change the form specifier to match the form of the existing connection.

SOURCE: $FTLIB

FT088 - REQUESTED RECL CONFLICTS WITH EXISTING RECL

CLASS: 1; reprievable

CAUSE: The RECL requested differs from the RECL established with this connection. Unit and dataset are already connected for direct access.

ACTION: Change the RECL specifier to match that of the existing connection.

SOURCE: $FTLIB

FT089 - MISSING UNIT SPECIFIER

CLASS: 1; reprievable

CAUSE: An OPEN or a CLOSE statement is missing a unit specifier; or an INQUIRE statement has neither a unit nor file specifier.

ACTION: Supply a unit specifier.

SOURCE: $FTLIB

FT090 - NEW STATUS ILLEGAL IF DATASET ALREADY EXISTS

CLASS: 1; reprievable

CAUSE: Status in an OPEN statement is specified NEW for a dataset that already exists.

ACTION: Change status specifier to OLD or UNKNOWN or omit the status specifier.

SOURCE: $FTLIB

FT091 - OLD STATUS ILLEGAL IF DATASET DOES NOT EXIST

CLASS: 1; reprievable

CAUSE: Status in an OPEN statement is specified OLD for a dataset that does not exist.

ACTION: Change status specifier to NEW or UNKNOWN or omit the status specifier.

SOURCE: $FTLIB

FT998 - DUPLICATE CALL TO **ERP
FT999 - $OUT DSP POSSIBLY DESTROYED } Issued as a pair

CLASS: 2; user abort; reprievable

CAUSE: The $FTLIB error processor is unable to write a message on file $OUT. The $OUT DSP might have been destroyed; a catastrophic error might have occurred in the I/O library; or the user might have made a library call error after being reprieved from a library error.

ACTION: Library errors are not reprievable. If $OUT file is not printable, examine program for subscript out of range. Try the ON=0, bounds checking, CFT option, or the IOAREA,LOCK control card to protect the DSPs.

SOURCE: $FTLIB

GP001 - BAD DELIMITER FOLLOWING KEYWORD

CLASS: Not reprievable

CAUSE: In processing a control statement, $GPARAM located a keyword and could not find a delimiter that would indicate the end of the operand.

ACTION: Correct the control statement and resubmit the request.

SOURCE: $SYSLIB

GP002 - KEYWORD *keyword* UNKNOWN OR DUPLICATED

CLASS:

CAUSE: A keyword in the control statement could not be identified or was identical to one that had already been processed.

ACTION: Correct the control statement and reissue the request.

SOURCE: $SYSLIB

GP003 - KEYWORD *keyword* MUST BE SPECIFIED

CLASS:

CAUSE: A keyword that is required on the control statement is missing.

ACTION: Correct the control statement and reissue the request.

SOURCE: $SYSLIB

GP004 - KEYWORD *keyword* MUST BE EQUATED

CLASS: Not reprievable

CAUSE: An equated value was expected for the keyword.

ACTION: Correct the control statement and reissue the request.

SOURCE: $SYSLIB

GP005 - KEYWORD *keyword* CANNOT BE EQUATED

CLASS: Not reprievable

CAUSE: An equated keyword was detected but invalid.

ACTION: Correct the control statement and reissue the request.

SOURCE: $SYSLIB

GP006 - KEYWORD *keyword* VALUE OVERFLOWED WITH *data*

CLASS:

CAUSE: The value of the specified keyword overflowed the size of the parameter array.

ACTION: Correct the data and reissue the request.

SOURCE: $SYSLIB

GP007 - CONTINUATION CARD NOT FOUND

CLASS: Not reprievable

CAUSE: A continuation character was detected but a subsequent continuation statement was not found.

ACTION: Correct the control statement and resubmit the request.

SOURCE: $SYSLIB

GP008 - SYNTAX ERROR ENCOUNTERED

CLASS: Not reprievable

CAUSE: GETPARAM has detected a violation of the control statement syntax.

ACTION: Ensure that the separator combinations are proper; for example an equals sign does not follow a colon.

SOURCE: $SYSLIB

GP009 - POSITIONAL PARAMETER *rmn* OVERFLOWED on '*data*'

CLASS: Not reprievable

CAUSE: The *n*th positional parameter has been given too large a value.

ACTION: Correct the positional value.

SOURCE: $SYSLIB

GP010 - INTERNAL ERROR WITH $CCS DETECTED

CLASS: Not reprievable

CAUSE: GETPARAM has detected an error with the cracked table.

ACTION: See a Cray Research analyst.

SOURCE: $SYSLIB

GP011 - INTERNAL ERROR WITH GETPARAM DETECTED

CLASS: Not reprievable

CAUSE: GETPARAM has detected an error which has occurred during its processing.

ACTION: See a Cray Research analyst.

SOURCE: $SYSLIB

GP012 - INVALID CONTROL TABLE STRUCTURE

CLASS: Not reprievable

CAUSE: The call to GETPARAM specifies a parameter control table that is the wrong format.

ACTION: Ensure that the control table follows the rules set forth by GETPARAM. See the Library Reference Manual, publication SR-0014.

SOURCE: $SYSLIB

GP013 - POSSIBLE KEYWORD *keyword* TREATED AS A POSITIONAL PARAMETER

CLASS: Informative

CAUSE: The parameter *keyword* has been placed in a positional slot and *keyword* is a valid keyword.

ACTION: If *keyword* is intended to be the positional value ignore the message. If *keyword* is to be the keyword-only selection of *keyword*, use commas to skip over the positional parameters.

SOURCE: $SYSLIB

GP014 - POSITIONAL PARAMETER *nnn* MUST BE SPECIFIED

CLASS: Not reprievable

CAUSE: The *n*th positional parameter has been omitted when its appearance is required.

ACTION: Specify the parameter on the statement. If a null value is desired use commas.

SOURCE: $SYSLIB

JS001 - CHANGED STATE

CLASS: Informative

CAUSE: This message shows the changes in job status in JXSTAT.

ACTION: Not applicable

SOURCE: JSH

JS001 - JOB----------. P=*nn.nn* MP=*nn.nn* SIZE=*nnnn* FREE MEMORY=*nnnn*

CLASS: Informative

CAUSE: A job's memory is changing according to the specifications in the message.

ACTION: Not applicable

SOURCE: JSH

JS002 - JOB RELOCATED FROM *addr1* TO *addr2* (ABSOLUTE, * 1000) BY A CALL AT *n**

CLASS: Informative

CAUSE: The job is being relocated in memory according to the specifications in the message.

ACTION: Not applicable

SOURCE: JSH

JS003 - JOB WAS ROLLED OUT TO WAIT FOR MEMORY

CLASS: Informative

CAUSE: The job needs more memory than is currently available; therefore, the job is rolled out.

ACTION: Not applicable

SOURCE: JSH

JS004 - FIELD LENGTH CHANGED TO n WORDS

CLASS: Informative

CAUSE: The field length is changed to indicate the new field length.

ACTION: Not applicable

SOURCE: JSH

JS005 - ROLL IMAGE WAS UNREADABLE; JOB RERUN

CLASS: Informative

CAUSE: The roll image is unreadable and the job is rerun.

ACTION: Not applicable

SOURCE: JSH

JS006 - JOB RECOVERED

CLASS: Informative

CAUSE: The job is recovered.

ACTION: Not applicable

SOURCE: JSH

JS007 - ROLL IMAGE WAS UNREADABLE; JOB WAS NOT RERUNNABLE.

CLASS: Informative

CAUSE: The roll image was unreadable but the job was not rerunnable.

ACTION: Not applicable

SOURCE: JSH

LD001 - DUPLICATE PROGRAM BLOCK NAME ENCOUNTERED AND SKIPPED (*dn*).

CLASS: Not reprievable

CAUSE: See message.

ACTION: Check program for the block duplication.

SOURCE: LDR

LD002 - DATASET (*dn*) NULL FILE OR BAD TABLE

CLASS: Fatal

CAUSE: Input to LDR is in error.

ACTION: See system analyst.

SOURCE: LDR

LD003 - DATASET (*dn*) INITIAL TABLE NOT PDT

CLASS: Fatal

CAUSE: Input to LDR is in error.

ACTION: See system analyst.

SOURCE: LDR

LD004 - BLANK COMMON REDEFINED (PREVIOUS OR LARGER) DEFINITION USED

CLASS: Informative

CAUSE: See message.

ACTION: Not applicable

SOURCE: LDR

LD005 - SHORTER COMMON (*name*) REFERENCED IN OR (*program name*) 1ST DEFINITION USED

CLASS: Informative

CAUSE: See message.

ACTION: Not applicable

SOURCE: LDR

LD006 - DUPLICATE ENTRY LOADED AND IGNORED (*entry name*)

CLASS: Informative

CAUSE: A duplicate entry was encountered. Error obstructs program execution.

ACTION: Check program for duplicate entry.

SOURCE: LDR

LD007 - RELOCATABLE LOAD MODULE IN ABSOLUTE LOAD OR ABSOLUTE LOAD MODULE IN RELOCATABALE LOAD

CLASS: Caution; job aborts when load completes.

CAUSE: See message.

ACTION: Check load modules.

SOURCE: LDR

LD008 - DATASET (*name*) MULTIPLE INTER RELOC BLOCK

CLASS: Fatal

CAUSE: Possible problem in CAL or CFT

ACTION: See a system analyst.

SOURCE: LDR

LD009 - UNSATISFIED EXTERNAL (*name*) CALLED BY (*routine*)

CLASS: Warning

CAUSE: External called is not in one of user libraries.

ACTION: Correct external.

SOURCE: LDR

LD010 - BEGIN EXECUTION

CLASS: Informative

CAUSE: Program being loaded has started to execute.

ACTION: Not applicable

SOURCE: LDR

LD010 - UNSATISFIED EXTERNALS FOUND ABORT OPTION

CLASS: Fatal

CAUSE: Unsatisfied externals

ACTION: Correct problem in user libraries.

SOURCE: LDR

LD011 - LDR INITIATED

CLASS: Informative

CAUSE: Use of CNS parameter in LDR control statement

ACTION: Not applicable

SOURCE: LDR

LD012 - DATASET (*name*) INVALID BLOCK NAME

CLASS: Fatal

CAUSE: Invalid block name. Possible problem in CAL or CFT

ACTION: Correct dataset name.

SOURCE: LDR

LD013 - START ENTRY NOT VALID

CLASS: Warning

CAUSE: Start entry is missing.

ACTION: Include start entry.

SOURCE: LDR

LD014 - LONGER COMMON (*block name*) REFERENCED IN (*program name*)

CLASS: Note

CAUSE: Previously defined common block is longer in a new routine. Error obstructs program execution.

ACTION: Not applicable

SOURCE: LDR

LD015 - NO START ADDRESS FOUND, FIRST ENTRY USED

CLASS: Note

CAUSE: Start address missing. Program execution begins at first entry point.

ACTION: Include start address.

SOURCE: LDR

LD016 - MEMBER ERROR (*name*)

CLASS: Caution

CAUSE: A member was not found in the file directive.

ACTION: Correct overlay directive or change file name.

SOURCE: LDR

LD017 - DIRECTIVE ERROR (*erroneous directive*)

CLASS: Caution; job aborts when load completes.

CAUSE: Overlay directive syntax error.

ACTION: Correct directive.

SOURCE: LDR

LD019 - COMPILATION ERRORS IN (*program name*) OR (*dn*)

CLASS: Caution; job aborts when load completes.

CAUSE: See message.

ACTION: Correct errors.

SOURCE: LDR

LD020 - DATASET NAME TOO LONG (*name*)

CLASS: Caution

CAUSE: See message.

ACTION: Change dataset name.

SOURCE: LDR

LD021 - MULTIPLE LOAD DATASETS IGNORED IN OVL MODE

CLASS: Caution

CAUSE: User has more than one dataset in the LDR statement in OVL mode.

ACTION: Not applicable

SOURCE: LDR

LD022 - ILLEGAL CHARACTER IN DIRECTIVE (*directive*)

CLASS: Caution

CAUSE: See message.

ACTION: Correct directive.

SOURCE: LDR

LD023 - ILLEGAL MAP VALUE, CHANGING TO MAP=PART

CLASS: Note

CAUSE: Illegal map value

ACTION: Correct map value.

SOURCE: LDR

LD024 - TYPE = (*table type*) UNRECOGNIZABLE TABLE

CLASS: Fatal

CAUSE: Table in error. Possible problem in CAL or CFT.

ACTION: See system analyst.

SOURCE: LDR

LD025 - INVALID TABLE STRUCTURE (BI) FOR MODULE (*table number*)

CLASS: Fatal

CAUSE: Table in error. Possible problem in CAL or CFT.

ACTION: See a Cray Research analyst.

SOURCE: LDR

LD027 - DATASET (*name*) BAD LIBRARY FORMAT

CLASS: Fatal

CAUSE: Library format in error.

ACTION: Check libraries for the bad format. Generate a new library.

SOURCE: LDR

LD028 - SPEC BLANK COMMON ADDR NOT LARGE ENOUGH

CLASS: Caution

CAUSE: Starting blank common address not large enough.

ACTION: Modify SBCA directive.

SOURCE: LDR

LD029 - DATASET (*dn*) UNABLE TO OPEN

CLASS: Fatal

CAUSE: Dataset could not be opened and may be non-existant.

ACTION: See a Cray Research analyst.

SOURCE: LDR

LD031 - DATA FOR COMMON BLOCK (*program name*) IGNORED

CLASS: Caution

CAUSE: Text not in LDR tables.

ACTION: See a system analyst.

SOURCE: LDR

LD034 - ILLEGAL REFERENCE TO RELOCOVL ENTRY POINT *entry*

CLASS: Fatal

CAUSE: A module has declared an external that is defined to be an entry point in a relocatable overlay.

ACTION: Correct the reference.

SOURCE: LDR

LD035 - ILLEGAL REFERENCE TO RELOCOVL PROGRAM BLOCK *module* BLOCK *block*

CLASS: Fatal

CAUSE: A Block Relocation Table that defines an address not within the module itself (such as labeled COMMON) refers to an address within the overlay module.

ACTION: Correct the program.

SOURCE: LDR

LD100 - DATASET (*name*) ALL FILES SEARCH

CLASS: Informative

CAUSE: All files were searched in selective load.

ACTION: Not applicable

SOURCE: LDR

LD200 - DATASET (*name*) INVALID READ TRY AGAIN

CLASS: Informative

CAUSE: No dataset name to load selectively.

ACTION: Check directives.

SOURCE: LDR

LD300 - DATASET (*dn*) NO SELECTIVE MODULES

CLASS: Informative

CAUSE: No selective modules

ACTION: Correct directive.

SOURCE: LDR

LD400 - NAME (*name*) INCLUDED BEFORE

CLASS: Informative

CAUSE: The INCLUDE module was previously included.

ACTION: Remove duplicate

SOURCE: LDR

LD500 - NAME (*module name*) EXCLUDED BEFORE

CLASS: Informative

CAUSE: Module name was previously excluded.

ACTION: Remove duplicate

SOURCE: LDR

LD600 - DATASET (*name*) INVALID SELECTIVE FILE

CLASS: Informative

CAUSE: Invalid selective file

ACTION: Check directives.

SOURCE: LDR

MQ010 - MODSEQ, VERSION *x.xx* (*mm/dd/yy*) - TABLE CREATION RUN

CLASS: Informative

CAUSE: The presence of the CREATE parameter on the MODSEQ control statement.

ACTION: Not applicable

SOURCE: MODSEQ

MQ011 - *nnn* TRANSLATION-TABLE ENTRIES WRITTEN (CHECKSUM = *xxxxxxxxxxx*)

CLASS: Informative

CAUSE: The *nnn* translation-table entries are written with checksum = *xxxxxxxxxxx*.

ACTION: Save the checksum and compare it to the corresponding value in the MQ021 message written during a resequencing run.

SOURCE: MODSEQ

MQ012 - *nnn* SOURCE LINES, *nn* DECKS COPIED FROM SR TO NSR

CLASS: Informative

CAUSE: End of a MODSEQ creation run.

ACTION: Not applicable

SOURCE: MODSEQ

MQ013 - *nnn* GHOST NAMES STRIPPED FROM NSR

CLASS: Informative

CAUSE: Ghost deck names were found in columns 73-80 of some of the cards in the new source file (NSR) dataset.

ACTION: Not applicable

SOURCE; MODSEQ

MQ020 - MODSEQ, VERSION *x.xx* (*mm/dd/yy*) - RESEQUENCING RUN

CLASS: Informative

CAUSE: The absence of both the CREATE and REVERSE parameters on the MODSEQ control statement.

ACTION: Not applicable

SOURCE: MODSEQ

MQ021 - *nnn* TRANSLATION-TABLE ENTRIES READ (CHECKSUM = *xxxxxxxxxx*)

CLASS: Informative

CAUSE: End of the input phase of a MODSEQ resequencing run.

ACTION: Compare the checksum to the corresponding value in the MQ011 message written during the creation run.

SOURCE: MODSEQ

MQ022 - *nnn* LINES, *nn* DIRECTIVES TRANSLATED

CLASS: Informative

CAUSE: End of a MODSEQ resequencing run.

ACTION: Not applicable

SOURCE: MODSEQ

MQ030 - MODSEQ, VERSION *x.xx* (*mm/dd/yy*) - REVERSE RESEQUENCING RUN

CLASS: Informative

CAUSE: The CREATE parameter is missing from the MODSEQ control statement and the REVERSE parameter is present.

ACTION: Not applicable

SOURCE: MODSEQ

MQ090 - *nnn* SOURCE LINES CONTAINED BAD SEQUENCE NUMBERS

CLASS: Fatal

CAUSE: One or more records in the old source file (SR) failed a test for a period in column 89. These records and possibly others contain missing or malformed UPDATE sequence numbers.

ACTION: If the number of source lines that have bad sequence numbers is equal to the number of source lines read (see message MQ012), the SR dataset is bad. Ensure the SR dataset is generated by UPDATE with the selected options S=*sr*, DW=80, and SQ. If the numbers reported by MQ012 and MQ090 are different, the SR dataset has been modified so some but not all sequence numbers have been disturbed.

SOURCE: MODSEQ

MQ100 - *nnn* RESEQUENCING ERROR(S) DETECTED

CLASS: Warning

CAUSE: One or more resequencing errors (MQ101, MQ102, MQ103, MQ104, or MQ105) were detected during a MODSEQ resequencing run.

ACTION: See the entries for the individual error messages.

SOURCE: MODSEQ

MQ101 - DIRECTIVE *nn* ON LINE *nnn*. ID NOT REPRESENTED IN PL: *xxxxx*

CLASS: Warning

CAUSE: A reference is made by an old mod to an ID that is not present in the original PL.

ACTION: Repair the incorrect directive, either before or after the MODSEQ resequencing run.

SOURCE: MODSEQ

MQ102 - DIRECTIVE *nn* ON LINE *nnn*. NUMBER MISSING FROM PL: *xxxxx.nnn*

CLASS: Warning

CAUSE: A reference is made by an old mod to a line number absent in the original PL (although the ID is present).

ACTION: Repair the incorrect directive, either before or after the MODSEQ resequencing run.

SOURCE: MODSEQ

MQ103 - DIRECTIVE *nn* ON LINE *nnn*. RANGE SPANS DECKS: *xxxxx.mmm,yyyyy.nnn*

CLASS: Fatal

CAUSE: A deletion range begins and ends in two different decks, or the range is contained within one deck but the beginning follows the end. The test that produces this message is suppressed if an MQ101 or an MQ102 error has been detected in the same UPDATE directive.

ACTION: Repair the incorrect directive, either before or after the MODSEQ resequencing run. Repair may involve rewriting the mod.

SOURCE: MODSEQ

MQ104 - IN THE COMMENT ON LINE *nnn*. ID NOT REPRESENTED IN PL: *xxxxx*.

CLASS: Warning

CAUSE: A character string in the comment field of a translated directive looks like an UPDATE sequence number but cannot be translated.

ACTION: None

SOURCE: MODSEQ

MQ105 - IN THE COMMENT ON LINE *nnn*. NUMBER MISSING FROM PL: *xxxxx.nnn*

CLASS: Warning

CAUSE: A character string in the comment field of a translated directive looks like an UPDATE sequence number but cannot be translated.

ACTION: None

SOURCE: MODSEQ

MQ106 - DIRECTIVE *nn* ON LINE *nnn*. RANGE INVALID: *xxxxx.mmm,yyyyy.nnn*

CLASS: Fatal

CAUSE: The beginning of a deletion range follows the end on a resequencing run.

ACTION: The mod may need to be rewritten.

SOURCE: MODSEQ

MQ111 - THE SR DATASET CANNOT BE $IN OR $OUT

CLASS: Fatal

CAUSE: The old source file (SR) was equated to an improper dataset name.

ACTION: Omit the parameter or equate it to a proper dataset name.

SOURCE: MODSEQ

MQ112 - THE NSR DATASET CANNOT BE $IN OR $OUT

CLASS: Fatal

CAUSE: The new source file (NSR) was equated to an improper dataset name.

ACTION: Omit the parameter or equate it to a proper dataset name.

SOURCE: MODSEQ

MQ113 - THE TT DATASET CANNOT BE $IN OR $OUT

CLASS: Fatal

CAUSE: The Translation Table (TT) parameter was equated to an improper dataset name.

ACTION: Omit the parameter or equate it to a proper dataset name.

SOURCE: MODSEQ

MQ114 - THE OM DATASET CANNOT BE $OUT

CLASS: Fatal

CAUSE: The old mod (OM) parameter was equated to an improper dataset name.

ACTION: Omit the parameter or equate it to a proper dataset name.

SOURCE: MODSEQ

MQ115 - THE NM DATASET CANNOT BE $IN OR $OUT

CLASS: Fatal

CAUSE: The new mod (NM) parameter was equated to an improper dataset name.

ACTION: Omit the parameter or equate it to a proper dataset name.

SOURCE: MODSEQ

MQ116 - THE L DATASET CANNOT BE $IN

CLASS: Fatal

CAUSE: The L parameter was equated to an improper dataset name.

ACTION: Omit the parameter or equate it to a proper dataset name.

SOURCE: MODSEQ

MQ117 - THE E DATASET CANNOT BE $IN OR $OUT

CLASS: Fatal

CAUSE: The E parameter was equated to an improper dataset name.

ACTION: Omit the parameter or equate it to a proper dataset name.

SOURCE: MODSEQ

MQ121 - WARNING: NO SR FILE IS READ IN A RESEQUENCING RUN

CLASS: Warning

CAUSE: The old source file (SR) parameter should have been omitted.

ACTION: Omit the parameter.

SOURCE: MODSEQ

MQ122 - WARNING: NO NSR FILE IS WRITTEN IN A RESEQUENCING RUN

CLASS: Warning

CAUSE: The new source file (NSR) parameter specified should have been omitted.

ACTION: Omit the parameter.

SOURCE: MODSEQ

MQ123 - WARNING: NO OM FILE IS READ IN A CREATION RUN

CLASS: Warning

CAUSE: The old mod (OM) parameter should have been omitted.

ACTION: Omit the parameter.

SOURCE: MODSEQ

MQ124 - WARNING: NO NM FILE IS WRITTEN IN A CREATION RUN

CLASS: Warning

CAUSE: The new mod (NM) parameter should have been omitted.

ACTION: Omit the parameter.

SOURCE: MODSEQ

MQ125 - WARNING: THE L FILE IS NOT USED IN A CREATION RUN

CLASS: Warning

CAUSE: The L parameter should have been omitted.

ACTION: Omit the parameter.

SOURCE: MODSEQ

MQ126 - WARNING: THE E FILE IS NOT USED IN A CREATION RUN

CLASS: Warning

CAUSE: The E parameter should have been omitted.

ACTION: Omit the parameter.

SOURCE: MODSEQ

MQ127 - WARNING: THE REVERSE KEYWORD IS IGNORED IN A CREATION RUN

CLASS: Warning

CAUSE: The REVERSE parameter should have been omitted.

ACTION: Omit the parameter.

SOURCE: MODSEQ

MQ131 - THE SR DATASET IS NOT LOCAL TO THE JOB

CLASS: Fatal

CAUSE: No Dataset Name Table exists for the old source file (SR) dataset. This dataset is required input for a creation run.

ACTION: Examine the job control language for an error in misnaming or prematurely releasing the SR dataset. If necessary, add an ACCESS or ACQUIRE statement.

SOURCE: MODSEQ

MQ132 - THE OM DATASET IS NOT LOCAL TO THE JOB

CLASS: Fatal

CAUSE: No Dataset Name Table exists for the old mod (OM) dataset. This dataset is required input for a resequencing run.

ACTION: Examine the job control language for an error in misnaming or prematurely releasing the OM dataset. If necessary, add an ACCESS or ACQUIRE statement.

SOURCE: MODSEQ

MQ133 - THE TT DATASET IS NOT LOCAL TO THE JOB

CLASS: Fatal

CAUSE: No Dataset Name Table exists for the Translation Table (TT) dataset. This dataset is required input for a resequencing run.

ACTION: Examine the job control language for an error in misnaming or prematurely releasing the TT dataset. If necessary, add an ACCESS or ACQUIRE statement.

SOURCE: MODSEQ

MQ141 - THE SR FILE DOES NOT BEGIN WITH THE MASTER CHARACTER, '*x*'

CLASS: Fatal

CAUSE: The master character specified by the master character (MC) parameter (an asterisk by default) is incorrect, or the old source file specified by the old source file (SR) parameter contains incorrect data.

ACTION: Verify that the master character displayed in the message is correct. Examine the SR file to determine if it contains the correct UPDATE source cards.

SOURCE: MODSEQ

MQ142 - THE OM FILE DOES NOT BEGIN WITH THE MASTER CHARACTER, '*x*'

CLASS: Fatal

CAUSE: The master character specified by the master character (MC) parameter (an asterisk by default) is incorrect, or the file of old mods specified by the old mods file (OM) parameter contains incorrect data.

ACTION: Verify that the master character displayed in the message is correct. Examine the OM file to determine if it contains the correct UPDATE mod decks.

SOURCE: MODSEQ

MS001 - DATASET *dsname* NOT FOUND

CLASS: Fatal

CAUSE: Named input dataset to MODSET was not found.

ACTION: Correct specified dataset name or verify existence of dataset, and rerun MODSET.

SOURCE: MODSET

MS100 - NO OUTPUT DATASET SPECIFIED

CLASS: Fatal

CAUSE: No specification made for the C parameter.

ACTION: Specify dataset to receive combination of selected mods and rerun MODSET.

SOURCE: MODSET

MS130 - DIRECTIVE RECORD TOO LONG

CLASS: Fatal

CAUSE: A directive record longer than 200 words was encountered in the directive dataset.

ACTION: Rebuild or reformat the directive dataset and rerun MODSET.

SOURCE: MODSET

MS140 - VERIFIED MOD RECORD TOO LONG

CLASS: Fatal

CAUSE: A record in the verified mods file is longer than 200 words.

ACTION: Change the invocation to point to a different dataset, or rebuild or reformat the verified mods dataset and rerun MODSET.

SOURCE: MODSET

MS150 - TOO MANY MODS IN DIRECTIVE FILE

CLASS: Fatal

CAUSE: More than 400 mods specified in directive dataset.

ACTION: Reorganize the job to use smaller directive files and rerun MODSET.

SOURCE: MODSET

MS160 - TABLE OVERFLOW

CLASS: Fatal

CAUSE: There are more than 400 mods in the verified mods, unverified mods, or additional mods dataset.

ACTION: Restructure the job to use smaller mod files and rerun MODSET.

SOURCE: MODSET

PD000 - *n* ADDITIONAL BLOCKS FOR EXPANDING DATASET

CLASS: Informative

CAUSE: A permanent dataset being replaced has been expanded by *n* blocks.

ACTION: None; PDM has automatically adjusted the dataset.

SOURCE: PDM

PD000 - PDN = *pdn* ID = *id* ED = *ed* US = *us*

CLASS: Informative

CAUSE: This is a header message for PDM messages that follow.

ACTION: Not applicable

SOURCE: PDM

PD000 - *n* UNUSED BLOCKS FOR CONTRACTING DATASET

CLASS: Informative

CAUSE: A permanent dataset that contracted in size has been adjusted by ADJUST or RELEASE. The size of an adjusted dataset has been contracted by *n* blocks.

ACTION: None; PDM is informing the user about unused disk space.

SOURCE: PDM

PD001 - DATASET NOT LOCAL TO JOB

CLASS: Informative

CAUSE: A SAVE, DELETE, ADJUST, MODIFY, or DUMPTM request was issued to PDM that could not find a local dataset with the dataset name specified in the DN parameter.

ACTION: Create the dataset, or recheck the dataset name.

SOURCE: PDM

PD002 - MAINTENANCE PERMISSION NOT GRANTED

CLASS: Informative

CAUSE:

SAVE statement:	The user attempted to save a dataset with control words that did not match those of the previous edition.
DELETE statement:	The requester did not supply the proper maintenance control word in order to delete the dataset; or the requester did not supply the UQ parameter on the ACCESS statement.

ACTION: Specify the correct maintenance control word on the request, or gain unique access.

SOURCE: PDM

PD003 - EDITION ALREADY EXISTS

CLASS:

CAUSE: A request to SAVE a specific edition of a dataset was issued and a dataset with the specified edition already exists.

ACTION: Delete the saved edition and reissue the request or change the edition parameter.

SOURCE: PDM

PD004 - DSC FULL

CLASS:

CAUSE: PDM has no room in the Dataset Catalog to save or update the new information.

ACTION: Consult a systems analyst.

SOURCE: PDM

PD005 - INVALID FUNCTION CODE

CLASS:

CAUSE: The function code given in the PDD is out of range or a queue that was neither input nor output was given for an EQSDT request.

ACTION: Correct the function code.

SOURCE: PDM

PD006 - DATASET ALREADY EXISTS

CLASS:

CAUSE: An attempt was made to save or modify a dataset that already exists.

ACTION: Change either PDN or ID values for the SAVE or MODIFY request.

SOURCE: PDM

PD007 - NO PERMISSION GRANTED

CLASS:

CAUSE: ACCESS statement: The requested permission did not match the SAVE permission.
MODIFY statement: One of the three permission control words did not match.

ACTION: Specify the correct value for the requested permission.

SOURCE: PDM

PD009 - DATASET NOT FOUND

CLASS:

CAUSE: An attempt was made to access a dataset that does not exist in the permanant dataset base.

ACTION:

SOURCE: PDM

PD010 - EDITION NOT FOUND

CLASS:

CAUSE: An attempt was made to access a non-existent edition of a permanent dataset.

ACTION:

SOURCE: PDM

PD011 - DATASET NOT PERMANENT

CLASS:

CAUSE: A request was made to either DELETE, ADJUST, or MODIFY a local dataset that is not permanent.

ACTION:

SOURCE: PDM

PD014 - CONTINUATION ERROR

CLASS:

CAUSE: The Dataset Catalog contains erroneous data.

ACTION: Consult a site analyst.

SOURCE: PDM

PD015 - DAT FULL

CLASS:

CAUSE: The system pool does not have room to move the dataset's disk allocation information.

ACTION: If recurring, consult the systems analyst.

SOURCE: PDM

PD018 - DATASET ALREADY ACCESSED BY JOB

CLASS:

CAUSE: An attempt was made to access a dataset that is already accessed.

ACTION:

SOURCE: PDM

PD021 - DATASET NOT REWOUND

CLASS:

CAUSE: An attempt was made to change or create a permanent dataset that has no disk allocation assigned to it.

ACTION: This will only occur for datasets that do not meet the close conditions described in the CRAY-OS Version 1 Reference Manual. For such a dataset, the user must manually force a flushing of the buffer.

SOURCE: PDM

PD025 - NO DATASET NAME IN PDD

CLASS:

CAUSE: A request was made in which neither the DN nor the PDN parameter was selected.

ACTION: Correct the parameters.

SOURCE: PDM

PD027 - MULTIPLE EDITIONS EXIST

CLASS:

CAUSE: An attempt was made to change one of the permission control words for a permanent dataset that has more than one edition.

ACTION: All corresponding permission control words must match for all editions.

SOURCE: PDM

PD028 - NO UNIQUE ACCESS ON SYSDIR

CLASS:

CAUSE: An attempt was made to uniquely access a dataset that is also entered into the system directory.

ACTION: Not applicable

SOURCE: PDM

PD029 - TEXT INCONSISTENCY

CLASS:

CAUSE: PDD indicates the user has text associated with a DISPOSE, but the length of the text is 0.

ACTION: Clear the field PMTXT or set PMTXL.

SOURCE: PDM

PD030 - MAX TEXT LENGTH EXCEEDED

CLASS:

CAUSE: An attempt was made to dispose a dataset with text associated in a version of COS that does not support that length of text.

ACTION:

SOURCE: PDM

PD031 - DEVICE DOWN

CLASS:

CAUSE: The dataset resides wholly or partially on a device currently not available.

ACTION:

SOURCE: PDM

PD034 - ACCESS DENIED DUE TO ALLOCATION CONFLICT

CLASS:

CAUSE: The dataset resides on a known flawed area of the disk or on an area that has been reserved for some other dataset(s).

ACTION: Consult the the systems analyst.

SOURCE: PDM

PD035 - DSC ENTRY INVALID - ACCESS IMPOSSIBLE

CLASS:

CAUSE: The Dataset Catalog entry for the dataset contains erroneous data.

ACTION: Consult a systems analyst.

SOURCE: PDM

PD038 - MULTI-TYPE DISK ALLOCATION INCONSISTENCY - ACCESS IMPOSSIBLE

CLASS:

CAUSE: A discrepancy exists in at least one DSC entry for a dataset that has multiple DSC entries.

ACTION: Consult a systems analyst.

SOURCE: PDM

PD039 - MULTI-TYPE DATASET REFERENCES INVALID QDT ENTRY - ACCESS DENIED

CLASS:

CAUSE: The Queued Dataset table size has become smaller.

ACTION: Consult a systems analyst.

SOURCE: PDM

RI001 - JOB FAILED TO ROLL IN-RERUN

CLASS: Informative

CAUSE: The job failed to roll in and was rerun.

ACTION: Not applicable

SOURCE: CSP

RI002 - JOB FAILED TO ROLL IN AND NOT RERUNNABALE

CLASS: Informative

CAUSE: The job failed to roll in and was not rerunnable.

ACTION: Resubmit the job.

SOURCE: CSP

RJ001 - JOB RERUN BY SYSTEM RECOVERY

CLASS: Informative

CAUSE: During restart a nonrecoverable job was rerun.

ACTION: Not applicable

SOURCE: CSP

RJ002 - SYSTEM RECOVERY - JOB NOT RERUNNABLE OR RECOVERABLE

CLASS: Informative

CAUSE: During restart a nonrecoverable job was rerun.

ACTION: Resubmit the job.

SOURCE: CSP

SC001 - CFFT2 CALLED WITH PARAMETER N NOT OF THE FORM N=2**I WHERE $3 \leq I \leq 16$

CLASS: 2; user abort; reprievable

CAUSE: CFFT2 was called with an input vector that was not of length 2^I.

ACTION: See the Library Reference Manual for instructions on using the routine CFFT2.

SOURCE: $SCILIB

SC002 - CRFFT2 CALLED WITH PARAMETER N NOT OF THE FORM N=2**I WHERE $3 < I < 17$

CLASS: 2; user abort; reprievable

CAUSE: CRFFT2 was called with N out of range or not of length 2^I.

ACTION: See the Library Reference Manual for correct usage of CRFFT2.

SOURCE: $SCILIB

SC003 - FILTERG CALL WITH N LESS THAN M

CLASS: 2; user abort; reprievable

CAUSE: In a call to FILTERG, two input vectors of length N and M, respectively, were encountered with M larger than N.

ACTION: See the Library Reference Manual for correct usage of FILTERG.

SOURCE: $SCILIB

SC004 - FILTERS CALLED WITH N LESS THAN M

CLASS: 2; user abort; reprievable

CAUSE: In a call to FILTERS, two input vectors of length M and N, respectively, were encountered with M larger than N.

ACTION: See the Library Reference Manual for correct usage of FILTERS.

SOURCE: $SCILIB

SC005 - ICAMAX CALLED WITH N > 4096

CLASS: 2; user abort; reprievable

CAUSE: ICAMAX has an input vector of length N. The algorithm assumes that N is always less than or equal to 4096.

ACTION: Compile the original BLAS FORTRAN version of ICAMAX, which makes no assumptions about N.

SOURCE: $SCILIB

SC006 - ISAMAX CALLED WITH N > 4096.

CLASS: 2; user abort; reprievable

CAUSE: ISAMAX was called with the length of the input vector greater than 4096.

ACTION: Compile the original BLAS FORTRAN version of ISAMAX, which makes no assumptions about N.

SOURCE: $SCILIB

SC007 - ISMAX CALLED WITH N > 4096

CLASS: 2; user abort; reprievable

CAUSE: The ISMAX algorithm assumes that the input vector will have a length not exceeding 4096.

ACTION: Write a routine to partition the input into segments of length 4096 or less.

SOURCE: $SCILIB

SC008 - ISMIN CALLED WITH N > 4096

CLASS: 2; user abort; reprievable

CAUSE: The ISMIN algorithm assumes that the input vector is of length 4096 or less.

ACTION: Partition the input vector into segments of length 4096 or less.

SOURCE: $SCILIB

SC009 - MINV CALLED WITH MATRIX REDUCED TO A SINGULAR MATRIX

CLASS: 2; user abort; reprievable

CAUSE: A system of linear equations cannot be meaningfully solved if the matrix describing the system is singular. When the algorithm encounters a singular matrix, it stops and prints an error message.

ACTION: Analyze the data or method closely.

SOURCE: $SCILIB

SC010 - MINV CALLED WITH A PARTIAL PRODUCT OF PIVOT ELEMENTS LESS THAN EPS

CLASS: 2; user abort; reprievable

CAUSE: During the computation of MINV, the product of the pivot elements became less than the EPS specified in the call.

ACTION: Redefine EPS to be smaller or analyze data and method more carefully.

SOURCE: $SCILIB

SC011 - MXM CALLED WITH NAR NONPOSITIVE

CLASS: 2; user abort; reprievable

CAUSE: NAR specifies one dimension of a matrix product. This dimension must be positive.

ACTION: Determine if the parameter NAR is nonpositive.

SOURCE: $SCILIB

SC012 - MXM CALLED WITH NBC NONPOSITIVE

CLASS: 2; user abort; reprievable

CAUSE: NBC specifies one of the dimensions of a matrix multiply. This dimension must be positive.

ACTION: Determine if the parameter NBC is positive.

SOURCE: $SCILIB

SC013 - MXM CALLED WITH NAC NONPOSITIVE

CLASS: 2; user abort; reprievable

CAUSE: NAC specifies one of the dimensions of a matrix multiply. This dimension must be positive.

ACTION: Determine if the parameter NAC is positive.

SOURCE: $SCILIB

SC014 - MXMA CALLED WITH NAR NONPOSITIVE

CLASS: 2; user abort; reprievable

CAUSE: NAR specifies one dimension of a matrix product. This dimension must be positive.

ACTION: Determine if the parameter NAR is nonpositive.

SOURCE: $SCILIB

SC015 - MXMA CALLED WITH NBC NONPOSITIVE

CLASS: 2; user abort; reprievable

CAUSE: NBC specifies one of the dimensions of a matrix multiply. This dimension must be positive.

ACTION: Determine if the parameter NBC is positive.

SOURCE: $SCILIB

SC016 - MXMA CALLED WITH NAC NONPOSITIVE

CLASS: 2; user abort; reprievable

CAUSE: NAC specifies one of the dimensions of a matrix multiply. This dimension must be positive.

ACTION: Determine if the parameter NAC is positive.

SOURCE: $SCILIB

SC017 - PACK CALLED WITH NBITS NOT EQUAL TO 1, 2, 4, 8, 16, or 32

CLASS: 2; user abort; reprievable

CAUSE: The PACK subroutine assumes that the length of the partial words to be packed will be a small power of 2.

ACTION: Assign the partial words a length that is a small power of 2.

SOURCE: $SCILIB

SC018 - RCFFT2 CALLED WITH PARAMETER N NOT EQUAL TO FORM N=2**I WHERE 3 < I < 17

CLASS: 2; user abort; reprievable

CAUSE: RCFFT2 is written to operate on an input vector of length N, where N is a power of 2 and is within a certain range.

ACTION: See the Library Reference Manual for correct usage of the parameter N in a call to RCFFT2.

SOURCE: $SCILIB

SC019 - UNPACK CALLED WITH NBITS NOT EQUAL TO 1, 2, 4, 8, 16 or 32

CLASS: 2; user abort; reprievable

CAUSE: The parameter NBITS is assumed to be a small power of 2.

ACTION: NBITS must be equal to a small power of 2.

SOURCE: $SCILIB

SC020 - SROTM - CALLED WITH INCORRECT PARAMETER KEY

CLASS: 2; user abort; reprievable

CAUSE: The last parameter of a call to SROTM is incorrect.

ACTION: See the Basic Linear Algebra Subprograms for CFT Usage, CRI publication 2240208.

SOURCE: $SCILIB

SC021 - MXMA CALLED WITH TOO MANY PARAMETERS

CLASS: 2; user abort; reprievable

CAUSE: A MXMA call exceeded the limit of 12 parameters.

ACTION: See the Library Reference Manual.

SOURCE: $SCILIB

SC022 - SMACH CALLED WITH PARAMETER JJ NOT EQUAL TO 1, 2 or 3

CLASS: 2; user abort; reprievable

CAUSE: The JJ parameter contains an illegal value.

ACTION: Check the parameter JJ in the call to SMACH. See the Library Reference Manual.

SOURCE: $SCILIB

SF001 - *n* IDENTIFIERS, *m* REFERENCES FOR *program*

CLASS: Informative

CAUSE: End of SKOL cross reference generation. *n* includes all programmer-invented names except those that occur only inside literal strings. *m* excludes those that occur inside a literal string and therefore do not appear in the cross reference listing. *program* is the name of the MAIN segment, or the name of the first major segment if there is no MAIN segment.

ACTION: Not applicable

SOURCE: SKOLREF

SF002 - MORE THAN 5197 TABLE ENTRIES; REMAINDER IGNORED

CLASS: Warning

CAUSE: The SKOL program contains too many programmer-invented names and therefore cannot be fully cross referenced.

ACTION: Break the program into separate compilations.

SOURCE: SKOLREF

SF003 - PROGRAM LONGER THAN 9999 LINES; REMAINDER IGNORED

CLASS: Warning

CAUSE: The SKOL program is too long to be fully cross referenced.

ACTION: Break the program into separate compilations.

SOURCE: SKOLREF

SK001 - SKOL PREPROCESSOR, REVISION LEVEL *x.n*

CLASS: Informative

CAUSE: Start of macro translation

ACTION: Not applicable

SOURCE: SKOL

SK002 - TRANSLATION TIME = *minutes.seconds*

CLASS: Informative

CAUSE: End of macro translation

ACTION: Not applicable

SOURCE: SKOL

SK003 - *n* MACRO TRANSLATION ERRORS

CLASS: Fatal

CAUSE: End of a macro translation in which fatal errors occurred

ACTION: Examine the error listing, correct the errors, and resubmit the job. If it is desired to proceed with the next job step following the macro translation without first correcting all the errors, place a %E*n* directive in the source text. The value of *n* must be greater than or equal to the number of fatal errors in order to change the class of this message from fatal to warning.

SOURCE: SKOL

SK003 - *n* MACRO TRANSLATION WARNINGS

CLASS: Warning

CAUSE: End of a macro translation in which warnings were issued

ACTION: Examine the error listing and correct the causes of the warning messages if a clean listing is desired.

SOURCE: SKOL

SK004 - AN %E DIRECTIVE WAS PROCESSED THAT ALLOWS UP TO *n* ERRORS

CLASS: Fatal

CAUSE: End of a macro translation in which the source text contained a %E*n* directive and in which more than *n* fatal errors were detected.

ACTION: Increase the value of *n* or reduce the number of fatal errors.

SOURCE: SKOL

SK006 - COULDN'T ASSIGN 'O' DATASET

CLASS: Fatal

CAUSE: The dataset that is to receive the generated FORTRAN code is already in use for some other purpose.

ACTION: Change the O parameter on the SKOL control statement, or omit the O parameter so that O=$SKF by default.

SOURCE: SKOL

SK007 - COULDN'T ACCESS *dataset*

CLASS: Fatal

CAUSE: The named dataset, which can be the source input (I) dataset, a supplementary input (M) dataset, or DKOLTXT, cannot be accessed.

ACTION: For SKOL's implicit dataset access method to work, an edition of the dataset must exist with no ID and no read permission word. If an ID or R parameter is required, add an ACCESS statement to the control statement file.

SOURCE: SKOL

SK008 - OUTPUT BUFFER OVERFLOW

CLASS: Fatal

CAUSE: A user-defined macro expansion is too large for the buffer.

ACTION: Redefine the macro in error, using inner macros, where appropriate, to reduce the amount of expansion done at a single time. If necessary, turn on SKOL's macro trace feature.

SOURCE: SKOL

SK009 - STORAGE OVERFLOW
*------------ ERROR - EXPANSION BUFFER OVERFLOW

CLASS: Fatal

CAUSE: Either a used-defined macro or constant caused a recursive match or a user-defined macro expansion was too large for the buffer.

ACTION: If the error is not obvious, turn on SKOL's macro trace feature. If no accidental recursion occurred, redefine the macro that is overflowing the buffer.

SOURCE: SKOL

SK009 - STORAGE OVERFLOW
*------------ - ERROR - INPUT BUFFER OVERFLOW

CLASS: Fatal

CAUSE: A statement or a literal string contains more than 25 source lines.

ACTION: Check the column between the sequence number and the nesting level for an apostrophe that indicates a literal string continuation.

SOURCE: SKOL

SK009 - STORAGE OVERFLOW
*------------ - ERROR - LABEL STACK OVERFLOW

CLASS: Fatal

CAUSE: An error occurred in a user-defined structural macro (one that manipulates the label stack) or the nesting level exceeded 46.

ACTION: Correct the user-defined macro or reduce the nesting level by breaking the major segment into separate routines, subroutines, or functions.

SOURCE: SKOL

SK009 - STORAGE OVERFLOW
*------------ - ERROR - LABEL STACK UNDERFLOW

CLASS: Fatal

CAUSE: An error occurred in a user-defined structural macro (one that manipulates the label stack) or the standard macros that protect against stack underflow have been redefined.

ACTION: Correct the user-defined macro or change the patterns of any user-defined macros that conflict with the standard macros.

SOURCE: SKOL

SK009 - STORAGE OVERFLOW
*------------ ERROR - MACRO BUFFER OVERFLOW

CLASS: Fatal

CAUSE: Too many user-defined macros for the available space

ACTION: Make some of the macros local by placing their definitions inside a major segment, break up the program in separate compilations, or recompile the SKOL macro translator with a larger dimension for the array MM.

SOURCE: SKOL

SK009 - STORAGE OVERFLOW
*------------ - ERROR - MACRO STACK OVERFLOW

CLASS: Fatal

CAUSE: An error occurred in a user-defined structural macro (one that manipulates the macro symbol stack), or the nesting level exceeded 46.

ACTION: Correct the user-defined macro, or reduce the nesting level by breaking the major segment into separate routines, subroutines, or functions.

SOURCE: SKOL

SK009 - STORAGE OVERFLOW
*---------- - ERROR - MACRO STACK UNDERFLOW

CLASS: Fatal

CAUSE: An error occurred in a user-defined structural macro (one that manipulates the macro symbol stack) or the standard macros that protect against stack underflow were redefined.

ACTION: Correct the user-defined macro or change the patterns of any user-defined macros that conflict with the standard macros.

SOURCE: SKOL

SK009 - STORAGE OVERFLOW
*---------- - ERROR - OUTPUT BUFFER OVERFLOW

CLASS: Fatal

CAUSE: A user-defined macro expansion is too large for the buffer.

ACTION: Redefine the macro in error, using inner macros, where appropriate, to reduce the amount of expansion done at a single time. If necessary, turn on SKOL's macro trace feature.

SOURCE: SKOL

SK009 - STORAGE OVERFLOW
*---------- - ERROR - PARAMETER BUFFER OVERFLOW

CLASS: Fatal

CAUSE: The sum of the lengths of the actual parameters exceeds the space availabale for them.

ACTION: Check the column between the sequence number and the nesting level for an apostrophe that indicates a literal string continuation. Redefine the macro in error, using inner macros, where appropriate, to reduce the amount of expansion done at a single time. If necessary, turn on SKOL's macro trace feature.

SOURCE: SKOL

SL000 - BAD CALL TO SYSTEM ERROR PROCESSOR

CLASS: 2; user abort; reprievable

CAUSE: Illegal error message number

ACTION: See a Cray Research analyst.

SOURCE: $SYSLIB

SL001 - *action*, *dn* I/O SYSTEM ERROR

CLASS: 2; user abort; reprievable

CAUSE: The system returned an undefined error after an I/O operation.

ACTION: Forward a dump of the JTA, DSP, and buffers to a Cray Research analyst.

SOURCE: $SYSLIB

SL002 - *action*, *dn* UNRECOVERED HARDWARE ERROR

CLASS: 2; user abort; reprievable

CAUSE: The system detected a permanent hardware error and was unable to complete the I/O operation.

ACTION: Check the status of the device. If the device was not ready, make it ready and resubmit the job. If the device appears to be operational, notify a Cray Research analyst.

SOURCE: $SYSLIB

SL003 - *action*, *dn* DISK DATA ERROR

CLASS: 2; user abort; reprievable

CAUSE: During an I/O operation, the system detected a discrepancy in the checksum.

ACTION DUMP the system log to determine if other datasets on the same device are receiving similar error messages. If so, notify a Cray Research engineer of a possible failing device. If it is not a common error and if the problem persists for the dataset, the dataset must be recreated.

SOURCE: $SYSLIB

SL004 - *action*, *dn* BLOCK NUMBER ERROR

CLASS: 2; user abort; reprievable

CAUSE: An I/O request completed but the block number that was read or written was not the expected block number. Probable system error or dataset being read is unblocked.

ACTION: Forward a dump of the JTA, DSP and buffer to a Cray Research analyst.

SOURCE: $SYSLIB

SL005 - *action*, *dn* READ ON WRITE ONLY DATASET

CLASS: 2; user abort; reprievable

CAUSE: An input request was issued on a dataset that was not accessed with read permission.

ACTION: If the dataset has write only permission, correct the program and rerun the job. Otherwise, forward a dump of the JTA, DSP, and buffers to a Cray Research analyst.

SOURCE: $SYSLIB

SL006 - *action*, *dn* DATASET PREMATURELY TERMINATED

CLASS: 2; user abort; reprievable

CAUSE: No end-of-data control word is present.

ACTION: If the end of data is not being tested correctly, correct the program and rerun the job. Otherwise, forward a dump of the JTA, DSP, and buffers to a Cray Research analyst.

SOURCE: $SYSLIB

SL007 - *action*, *dn* DATASET NOT OPEN

CLASS: 2; user abort; reprievable

CAUSE: A request to perform I/O was issued to a dataset that was not open.

ACTION: Verify that an OPEN request was issued and successfully completed before the I/O request was made. If not, correct the program and resubmit the job. Otherwise, forward a dump of the JTA and the DSPs to a Cray Research analyst.

SOURCE: $SYSLIB

SL008 - *action*, *dn* DATASET DOES NOT EXIST

CLASS: 2; user abort; reprievable

CAUSE: An I/O request was issued but the dataset contained no data.

ACTION: Determine if the dataset was created prior to an attempt to use it. It might be necessary to recreate the dataset.

SOURCE: $SYSLIB

SL009 - *action*, *dn* READ AFTER WRITE

CLASS: 2; user abort; reprievable

CAUSE: A read operation was issued after a write operation without an intervening rewind.

ACTION: Determine if the correct sequence of I/O operations is being issued. If not, correct the program and resubmit the job. Otherwise, forward a dump of the JTA, DSP, and buffers to a Cray Research analyst.

SOURCE: $SYSLIB

SL010 - *action*, *dn* READ PAST END OF DATA

CLASS: 2; user abort; reprievable

CAUSE: A read request was issued after end-of-data was detected.

ACTION Determine if end-of-data is being tested correctly. If so, correct the program and submit the job. Otherwise, forward a dump of the JTA, DSP, and buffers to a Cray Research analyst.

SOURCE: $SYSLIB

SL011 - *action*, *dn* WRITE ON READ ONLY DATASET

CLASS: 2; user abort; reprievable

CAUSE: The process direction passed to the system was invalid.

ACTION: Forward a dump of the JTA and DSPs to a Cray Research analyst.

SOURCE: $SYSLIB

SL012 - *action*, *dn* WRITE PAST END OF ALLOCATED AREA

CLASS: 2; user abort; reprievable

CAUSE: A sequential write request was issued for a block number beyond the expected block number for the next write request.

ACTION: Determine if the block number is beyond the expected block number. If so, correct the program and reissue the request. If not, forward a DUMP of the JTA, DSP, and buffers to a Cray Research analyst.

SOURCE: $SYSLIB

SL013 - *action*, *dn* RANDOM WRITE RECORD MUST END ON RECORD BOUNDARY

CLASS: 2; user abort; reprievable

CAUSE: Only complete records are processed by random I/O.

ACTION: Determine if the program is correctly issuing write requests. If not, correct the program and reissue the request. Otherwise, forward a dump of the JTA, DSP, and buffers to a Cray Research analyst.

SOURCE: $SYSLIB

SL014 - *action*, *dn* RANDOM FILE BUFFER MUST BE AT LEAST TWO BLOCKS LONG

CLASS: 2; user abort; reprievable

CAUSE: The minimum buffer size for a random dataset is two blocks or 1024 words.

ACTION: Correct the specified buffer size and resubmit the request.

SOURCE: $SYSLIB

SL015 - *action*, *dn* CHARACTER WRITE ILLEGAL ON RANDOM FILE

CLASS: 2; user abort; reprievable

CAUSE: A write characters request for a random access dataset was detected.

ACTION: Correct the write request or the access mode.

SOURCE: $SYSLIB

SL016 - *action*, *dn* WRITE PAST END OF DATA

CLASS: 2; user abort; reprievable

CAUSE: A write request was issued after the end of data was detected.

ACTION: If the end of data is not being tested correctly, correct the program and rerun the job. Otherwise, forward a dump of the JTA, DSP, and buffers to a Cray Research analyst.

SOURCE: $SYSLIB

SL017 - *action*, *dn* UNCLEARED END OF FILE

CLASS: 2; user abort; reprievable

CAUSE: An input operation attempted to read beyond the end of file or a read was issued without an intervening test for the end of file.

ACTION: If the end of file was not tested correctly, correct the program and rerun the job. Otherwise, forward a dump of the JTA, DSP, and buffers to a Cray Research analyst.

SOURCE: $SYSLIB

SL018 - *action*, *dn* INVALID PROCESSING DIRECTION

CLASS: 2; user abort; reprievable

CAUSE: An I/O request was issued but the processing direction did not match the processing direction specified when the dataset was opened.

ACTION: Correct the I/O request and resubmit the job.

SOURCE: $SYSLIB

SL019 - *action*, *dn* UNDEFINED I/O ERROR

CLASS: 2; user abort; reprievable

CAUSE: I/O system error

ACTION: See a Cray systems analyst.

SOURCE: $SYSLIB

SL020 - *dn* UNIT NUMBER < 0 or < 102

CLASS: 2; user abort; reprievable

CAUSE: The dataset to be used by DUMPJOB contains an invalid unit number.

ACTION: Correct the request.

SOURCE: $SYSLIB

SL021 - *dn* ERROR ON OPEN

CLASS: 2; user abort; reprievable

CAUSE: The dataset cannot be opened for input or output.

ACTION: If proper write or read permission was not used, correct the program and rerun the job. Otherwise, forward a dump of the JTA, DSP, and buffers to a Cray systems analyst.

SOURCE: $SYSLIB

SL022 - *action*, *dn* REQUEST INVALID FOR UNBLOCKED DATASET

CLASS: 2; user abort; reprievable

CAUSE: An I/O request that is not supported (e.g., BACKSPACE or formatted I/O) was issued while an unblocked dataset was being processed.

ACTION: Consult the CRAY-OS Version 1 Reference Manual for restrictions on unblocked datasets. Correct the program and resubmit the request.

SOURCE: $SYSLIB

SL023 - *action*, *dn* INVALID WORD ADDRESS

CLASS: 2; user abort; reprievable

CAUSE: A position request was issued for an unblocked dataset but the word address for the request was not a multiple of 512.

ACTION: Correct the word address of the request and resubmit the job.

SOURCE: $SYSLIB

SL024 - *action*, *dn* INVALID BIO FUNCTION FOR UNBLOCKED DATASET

CLASS: 2; user abort; reprievable

CAUSE: A buffered I/O request that is not supported was issued while an unblocked dataset was being processed.

ACTION: Consult the CRAY-OS Version 1 Reference Manual for restrictions on unblocked datasets. Correct the program and resubmit the job.

SOURCE: $SYSLIB

SL024 - *action*, *dn* INVALID BIO FUNCTION FOR UNBLOCKED DATASET

CLASS: Reprievable

CAUSE: A CAL program issued an illegal buffer I/O macro.

ACTION: If the buffer I/O macro was not issued according to the specifications in the CRAY-OS Version 1 Reference Manual, correct the program and rerun the job. Otherwise, forward a dump of the JTA, DSP, and buffers to a Cray Research analyst.

SOURCE: $SYSLIB

SL027 - BUFFER SIZE INVALID FOR UNBLOCKED REQUEST

CLASS: Not reprievable; CSP abort

CAUSE: This message is issued by ASSIGN when the undefined dataset structure (U) parameter is specified in conjunction with the buffer size (BS) parameter.

Undefined dataset structure (e.g., unblocked) uses a user-defined buffer. This buffer is not maintained by COS, thus the BS parameter is illegal.

ACTION: Not applicable

SOURCE: $SYSLIB

SL028 - *n* MEMORY RESIDENT INVALID FOR UNBLOCKED DATASET

CLASS: Not reprievable; CSP abort

CAUSE: This message is issued by ASSIGN when the undefined dataset structure (U) parameter is specified in conjunction with the memory resident (MR) parameter.

Undefined dataset structure (e.g., unblocked) uses a user-defined buffer. This buffer is not maintained by COS, thus the MR parameter is illegal.

ACTION: Not applicable

SOURCE: $SYSLIB

SL029 - INVALID DEVICE TYPE

CLASS: Not reprievable; CSP abort

CAUSE: Message issued by ACCESS if the device type is not mass storage or tape, and by ASSIGN if the device type is not interactive or mass storage.

ACTION: Correct the value of the DT parameter on the ACCESS or ASSIGN call.

SOURCE: $SYSLIB

SL031 - ILLEGAL BFI CHARACTER

CLASS: Not reprievable; CSP abort

CAUSE: This message is issued by ASSIGN when the BFI character is not in the range 000_8 to 177_8, non-numeric, and is not OFF.

ACTION: Correct the BFI character.

SOURCE: CSP

SL041 - *xxxxxxxx* ALREADY OPEN. CANNOT REDEFINE DATASET

CLASS: Informative

CAUSE: This message is issued by ASSIGN when the buffer size (BS), dataset size (S), logical device (DV), random (RDM) or undefined dataset structure (U) parameters are specified and the dataset is already open.

ACTION: Dataset characteristics listed above are not changed if the dataset is already open. The dataset must be closed before these characteristics can be modified.

SOURCE: $SYSLIB

SL042 - *xxxxxx* IS A DSN ALREADY IN USE

CLASS: Not reprievable; CSP abort

CAUSE: This message is issued by ASSIGN when the alias name (A parameter) is the same as a previously assigned dataset name (DN parameter).

ACTION: An alias name must be unique among previously assigned datasets. For example,

```
ASSIGN, DN = ABC
ASSIGN, DN = DEF, A = ABC
```

is a misuse of the alias parameter. To reuse the name ABC, it must first be closed (or released to the system).

SOURCE: $SYSLIB

SP001 - INPUT DATASET NOT SPECIFIED

CLASS: Fatal

CAUSE: The I parameter on the SPAWN control statement was illegally equated to zero.

ACTION: Resubmit the job with I set to a valid dataset name or defaulted to $CPL.

SOURCE: SPAWN

SP002 - INPUT DATASET DOES NOT EXIST

CLASS: Fatal

CAUSE: The dataset specified by the I parameter on the SPAWN control statement could not be found.

ACTION: Change the specification of the I parameter on the SPAWN control statement and resubmit the job.

SOURCE: SPAWN

SP003 - BAD SL PARAMETER VALUE

CLASS: Fatal

CAUSE: The specification of the SL (sleep time) parameter on the SPAWN control statement contains non-numeric characters.

ACTION: Correct the SL parameter and resubmit the job.

SOURCE: SPAWN

SP004 - BAD DW PARAMETER VALUE

CLASS: Fatal

CAUSE: The data width (DW) parameter is calling for less than one character. The DW parameter might contain a non-numeric character.

ACTION: Correct the DW parameter and resubmit the job.

SOURCE: SPAWN

SP005 - BAD GL PARAMETER VALUE

CLASS: Fatal

CAUSE: The GL (group size) parameter on the SPAWN control statement contains a non-numeric character.

ACTION: Correct the GL parameter and resubmit the job.

SOURCE: SPAWN

SP006 - JOB DELIMITER NOT SPECIFIED

CLASS: Fatal

CAUSE: The SP parameter on the SPAWN control statement was illegally set to 0.

ACTION: Respecify the SP parameter to match the text on job separator statements on the input dataset. A minimum of eight characters of text are significant and required for this separator.

SOURCE: SPAWN

SP200 - INPUT DATASET EMPTY OR MISPOSITIONED

CLASS: Fatal

CAUSE: The first attempt to read from the input dataset resulted in EOF or EOD status.

ACTION: Rebuild or rewind the input dataset as appropriate and resubmit the job.

SOURCE: SPAWN

SP201 - INPUT DATASET MISPOSITIONED OR STRUCTURE ERROR

CLASS: Fatal

CAUSE: The first card image read from the input dataset was not a job separator card as was expected.

ACTION: Verify that the first card image in the dataset is a job separator card. Reposition or rewind the input dataset and resubmit the job.

SOURCE: SPAWN

SP202 - INPUT DATASET STRUCTURE ERROR

CLASS: Fatal

CAUSE: Read status of EOF or EOD was returned on the next read after encountering the job separator card in the input dataset.

ACTION: Rebuild the dataset, ensuring that a job separator card is not the last card image in it. Resubmit the job.

SOURCE: SPAWN

SP203 - INPUT DATASET STRUCTURE ERROR

CLASS: Fatal

CAUSE: Two successive job separator cards were encountered in the input dataset.

ACTION: Correct the input dataset, either by inserting job control text between the adjacent separator cards or by removing one of the cards. Resubmit the job.

SOURCE: SPAWN

SP800 - *n* JOBS DISPOSED TO THE *dn* QUEUE

CLASS: Informative

CAUSE: This message indicates completion of the SPAWN execution.

ACTION: Not appropriate

SOURCE: SPAWN

SP801 - SPAWN COMPLETE

CLASS: Informative

CAUSE: This message indicates completion of the SPAWN execution.

ACTION: Not applicable

SOURCE: SPAWN

SP901 - DISPOSE TO SPECIFIED QUEUE UNSUCCESSFUL

CLASS: Fatal

CAUSE: Internal error occurred in processing the DISPOSE request.

ACTION: Dumps should be taken and analyzed. Report the problem to a Cray Research analyst.

SOURCE: SPAWN

SR001 - SYSREF, VERSION *x.xx*

CLASS: Informative

CAUSE: Start of job step

ACTION: Not applicable

SOURCE: SYSREF

SR002 - *nn* MODULES, *nnn* SYMBOLS, *nnnn* REFERENCES

CLASS: Informative

CAUSE End of job step

ACTION: Not applicable

SOURCE: SYSREF

SR010 - BINARY SYMBOL FILE MISSING OR MISNAMED

CLASS: Fatal

CAUSE: No DNT exists for the dataset specified by the X parameter ($XRF by default).

ACTION: Check the SYSREF control statement for coding errors. If necessary, add an X parameter to one or more preceding CAL and/or APML control statements.

SOURCE: SYSREF

SR100 - READ_STATUS = *n*; RECP = *nn*; ^ RECP.WORD_COUNT = *n*; TOTAL_LENGTH = *nnn*; ACTUAL_LENGTH - *nnn*;

CLASS: Fatal

CAUSE: One of the four fatal messages SR101, SR102, SR103, or SR104 has just been issued. If any of these errors occurs repeatedly, there is a program logic error in SYSREF, CAL, or APML.

ACTION: Use DSDUMP to dump the X dataset that was input to SYSREF. Give the listing and the job's log file to a Cray Research analyst.

SOURCE: SYSREF

SR101 - PREMATURE END-OF-RECORD WHILE READING HEADER

CLASS: Fatal

CAUSE: A program logic error occurred while SYSREF was reading the X dataset.

ACTION: Use DSDUMP to dump the X dataset that was input to SYSREF. Give the listing and the job's log file to a Cray Research analyst.

SOURCE: SYSREF

SR102 - PREMATURE END-OF-RECORD WHILE READING SYMBOL ENTRIES

CLASS: Fatal

CAUSE: A program logic error occurred while SYSREF was reading the X dataset.

ACTION: Use DSDUMP to dump the X dataset that was input to SYSREF. Give the listing and the job's log file to a Cray Research analyst.

SOURCE: SYSREF

SR103 - RECORD LONGER THAN INDICATED BY HEADER

CLASS: Fatal

CAUSE: A program logic error occurred while SYSREF was reading the X dataset.

ACTION: Use DSDUMP to dump the X dataset that was input to SYSREF. Give the listing and the job's log file to a Cray Research analyst.

SOURCE: SYSREF

SR104 - SYMBOL COUNT WRONG OR SYMBOL ENTRIES MALFORMED

CLASS: Fatal

CAUSE: A program logic error occurred while SYSREF was reading the X dataset.

ACTION: Use DSDUMP to dump the X dataset that was input to SYSREF. Give the listing and the job's logfile to a Cray Research analyst.

SOURCE: SYSREF

SR150 - *nnn* SYMBOL ENTRIES WERE READ BUT *nnn* WERE OUTPUT

CLASS: Warning

CAUSE: The number of symbols output was different from the number read from the X dataset. If too few symbols were output, SYSREF's sort routine is not maintaining the linked list correctly.

ACTION: Save a copy of the X dataset on tape. Give the tape and listing of the job's logfile to a Cray Research analyst.

SOURCE: SYSREF

SR200 - DUPLICATE PROGRAM NAME *xxxxxxx* IN RECORDS *m* AND *n*

CLASS: Warning

CAUSE: A binary symbol record from the X dataset has the same module name as an earlier binary symbol record on the same dataset. Record numbers such as *m* and *n* start at 1.

ACTION: Check to see if the routine in question is being assembled twice or if there is a name conflict between two different source files being assembled.

SOURCE: SYSREF

SX001 - ZERO INCREMENT AT LINE *m*

CLASS: Fatal

CAUSE: An attempt was made to execute a FOR statement whose increment value in the BY clause resolved to 0. *m* indicates the location of the FOR statement in the SKOL source listing.

ACTION: Correct the SKOL program's logic.

SOURCE: SKOL run-time support routines ($FTLIB)

SX002 - FINAL VALUE NOT EXACT AT LINE *m*

CLASS: Fatal

CAUSE: An attempt was made to execute a FOR statement in which the difference between the final value in the TO clause and the initial value was not an exact multiple of the increment value in the BY clause. *m* indicates the location of the FOR statement in the SKOL source listing.

ACTION: Correct the SKOL program's logic.

SOURCE: SKOL run-time support routines ($FTLIB)

SX003 - NO SITUATION WAS SIGNALLED AT LINE *m*

CLASS: Fatal

CAUSE: The end of an UNTIL block was reached before a situation statement was executed. *m* refers to either the THEN statement in a multiple situation case structure or the ENDUNTIL statement in a simple situation case structure.

ACTION: Add either a situation statement or a looping structure inside the UNTIL block.

SOURCE: SKOL run-time support routines ($FTLIB)

SX004 - VALUE OUT OF RANGE AT LINE *m*

CLASS: Fatal

CAUSE: An attempt was made to execute a WHEN statement whose selector expression resolved to a value outside the range of the selector type. *m* indicates the location of the WHEN statement in the SKOL source listing.

ACTION: Change the selector expression or the selector type or define a new type, perhaps a numeric one.

SOURCE: SKOL run-time support routines ($FTLIB)

SX005 - RECORD SPACE EXHAUSTED AT LINE *m*

CLASS: Fatal

CAUSE: The value of rtype-AVAILP was 0 when the NEW statement attempted to allocate a record. *m* indicates the location of the NEW statement in the SKOL source listing.

ACTION: Execute MAKEAVAIL for the appropriate record type before executing the first NEW statement. Provide enough FREE statements to ensure that all unneeded records are released promptly. Enlarge the maximum number of records by changing the RECORD declaration. As a last resort, output the error message and append (NOVALIDATE) to the NEW statement. See the SKOL Reference Manual for an example.

SOURCE: SKOL

SX006 - ILLEGAL COROUTINE FINISH AT LINE *m*

CLASS: Fatal

CAUSE: The end of a coroutine was reached before an ACTIVATE or SUSPEND statement was executed. *m* indicates the location of the ENDCOROUTINE statement in the SKOL source listing.

ACTION: See the SKOL Reference Manual for correct usage.

SOURCE: SKOL

SX007 - RECURSION UNDERFLOW AT LINE *m*

CLASS: Fatal

CAUSE: An attempt was made to access the recursion stack with a zero subscript. This indicates illegal nesting of recursive routines. *m* indicates the SKOL source line containing the declaration of the size of the recursion stack.

ACTION: Define all recursive routines at level 1.

SOURCE: SKOL

SX008 - RECURSION OVERFLOW AT LINE *m*

CLASS: Fatal

CAUSE: The recursion stack does not have enough room to hold the parameters, local variable, and two control variables needed for the current recursive routine invocation. If *m* refers to an EXECUTE statement, overflow occurred while trying to add parameters to the stack. If *m* refers to a ROUTINE statement, overflow occurred while trying to add LOCAL variables to the stack.

ACTION: Enlarge the recursion stack by modifying the first value specified in parentheses in the RECURSIVE declaration. For each potential level of recursion, allow 2+P+L elements in the stack, where P is the number of parameters and L is the number of local variables. If two or more recursive routines are active simultaneously, their effect on the stack size is additive.

SOURCE: SKOL

SX009 - STRING OVERFLOW AT LINE *m*

CLASS: Fatal

CAUSE: The length of the result string exceeds the maximum length specified in the STRING declaration. *m* indicates the location of the offending statement in the SKOL source listing.

ACTION: Increase the dimension specified in the STRING declaration.

SOURCE: SKOL

SX010 - COLLISION IN CHARACTER MAPPING AT *n*; DISCOVERED AT LINE *m*

CLASS: Fatal

CAUSE: A CHAR_SETUP statement detected a duplicated character in the program's TYPE CHAR statement. *n* is one greater than the ASCII value of the character. *m* indicates the location of the CHAR_SETUP statement in the SKOL source listing.

ACTION: If the TYPE CHAR statement specifies the same character twice, remove one of the instances. One of the duplicate specifications might be in a default subtype list; see the SKOL Reference Manual for a complete list of all the default members of the CHAR type.

SOURCE: SKOL

SX011 - HOLE IN OUTPUT CHARACTER TABLE AT *n*; DISCOVERED AT LINE *m*

CLASS: Fatal

CAUSE: A WRITESTRING statement's string argument contains a character whose internal (ordinal) value was defined with an identifier rather than with a single-character literal. *n* is the internal value of the character, which corresponds to the sequential position of that character in the TYPE CHAR declaration. *m* indicates the location of the offending statement in the SKOL source listing.

ACTION: Correct the SKOL program's logic so that any string to be output by WRITESTRING consists entirely of the ordinal values of characters that have corresponding external values.

SOURCE: SKOL

SX012 - UNDEFINED CHARACTER VALUE *n* OUTPUT AT LINE *m*

CLASS: Fatal

CAUSE: A WRITESTRING statement's string argument contains a value less than 1 or greater than the number of characters declared in the TYPE CHAR statement. *n* is the offending value from the string. *m* indicates the location of the WRITESTRING statements in the SKOL source listing.

ACTION: Any string to be output by WRITESTRING must consist entirely of ordinal values in the range [1...N], where N is the number of declared characters.

SOURCE: SKOL

SX013 - UNDEFINED CHARACTER '*c*' READ AT LINE *m*

CLASS: Fatal

CAUSE: A READSTRING statement encountered an undeclared character in the input. *m* indicates the location of the READSTRING statement in the SKOL source listing.

ACTION: Add all expected characters (including the character '*c*' to the TYPE CHAR declaration or delete the TYPE CHAR, CHAR COMMON, and CHAR_SETUP statements; change all READSTRING and WRITESTRING statements to READ, WRITE, INPUT, and OUTPUT statements; and avoid all reference to the ORD, CHR, and VALUE functions and to the built-in CHAR subtypes.

SOURCE: SKOL

SY001 - RLS COULD NOT FIND A DNT FOR *dataset*

CLASS: Informative

CAUSE: Attempt to release a dataset for which no DNT existed

ACTION: Not applicable

SOURCE: EXP

SY002 - ATTEMPT TO DISPOSE AN EMPTY DATASET *dataset*

CLASS: Informative

CAUSE: Attempt to dispose a dataset which was empty. No actual dispose was done.

ACTION: If dataset was not supposed to be empty, check the job to ensure the dataset is being created properly.

SOURCE: EXP

SY006 - BINARY INSTRUCTIONS SYSTEM-HANG MACRO

CLASS: Informative

CAUSE: Message is formatted in low memory and will never be put in any log. Its purpose is to signal the beginning of the hang message which STP creates when it detects an internal error and intends to crash. The binary instructions will be the last few parcels before the hang macro, and the hang macro will be of the form ERR*xx* where *xx* is AN, SN, AZ, SZ, etc. This pattern and the accompanying hang address can be used to find the cause of the crash.

ACTION: Not applicable

SOURCE: COMMON SUBROUTINE

SY013 - SYSTEM STARTUP LEVEL *starttype*

CLASS: Informative

CAUSE: This message appears every time the system starts. The *starttype* can be INSTALL, DEADSTART, or RESTART, depending on which Startup option was performed.

ACTION: Not applicable

SOURCE: STARTUP

SY014 - $SYSLOG RECOVERY SUCCESS CODE *x*

CLASS: Informative

CAUSE: This recovery message appears after every Restart, Deadstart, and Install. Either the systemlog is recovered at the proper point in the existing edition or a new edition is initialized. *x* can be one of the following:

OK The existing edition of the systemlog has been recovered successfully.

1 The systemlog does not exist, is on a down device, or has an AI conflict. A new edition is initialized.

2 The verify word is bad. The Dataset Parameter Table is rebuilt, and the system log is read to *eof*.

3 The edition number does not match. A new edition is begun.

4 DSP failed validation. Rebuild and read to *eof*.

5 DPEOR is not set. Rebuild and read to *eof*.

6 Bad Record Control Word in log. Rebuild and read to *eof*.

7 Bad Forward Word Index in RCW. Rebuild and read to *eof*.

8 Bad block number in RCW. Rebuild and read to *eof*.

9 The $SYSTEMLOG buffer memory was not recovered. The DSP is rebuilt, and the systemlog is read to *eof*.

ACTION: Not applicable

SOURCE: MSG

SY015 - $SYSTEMLOG SAVED AS EDITION *x*

CLASS: Informative

CAUSE: This message appears whenever a new edition of $SYSTEMLOG is saved.

ACTION: Not applicable

SOURCE: MSG

SY031 - BLOCKED FOR MEMORY { MS / MU }

CLASS: Informative

CAUSE: Space is not currently available in the memory pool for building a record for the systemlog (MS) or for a user's $LOG (MU). The MSG task will continue its processing of messages, however, first transferring earlier records from the memory pool to the appropriate logs and then attempting to rebuild the current blocked message again and to build any subsequent message requests.

ACTION: Not applicable

SOURCE: MSG

SYDPS1 - WAIT AND NOWAIT ILLEGAL TOGETHER - DEFAULT USED

CLASS: Not reprievable

CAUSE: A DISPOSE request specified both WAIT and NOWAIT.

ACTION: The system issues the warning message and then uses the installation default.

SOURCE: $SYSLIB

SYMFY1 -INVALID EQUATE GIVEN FOR THE EXO PARAMETER

CLASS: Not reprievable

CAUSE: The EXO parameter in a MODIFY request contains a value that is neither ON nor OFF. The EXO parameter is ignored.

ACTION: Correct the value.

SOURCE: $SYSLIB

SYSAV1 - INVALID VALUE GIVEN FOR THE EXO PARAMETER

CLASS: Not reprievable

CAUSE: The EXO parameter in a SAVE request contains a value that is neither ON nor OFF. The EXO parameter is ignored.

ACTION: Correct the value.

SOURCE: $SYSLIB

UD001 - PL DATE *mm/dd/yy* LEVEL: *n*

CLASS: Informative

CAUSE: UPDATE initialization reports the date of PL generation and the last mod ident or deck name added.

ACTION: Not applicable

SOURCE: UPDATE

UD002 - *n* WARNING ERRORS

CLASS: Informative

CAUSE: UPDATE directive references to deleted cards.

ACTION: Determine if all card references are as expected. See the UPDATE error listing for the suspect directives.

SOURCE: UPDATE

UD003 - EMPTY INPUT FILE, DN = *dn*

CLASS: Informative

CAUSE: The named input dataset is empty.

ACTION: Determine if the primary input file or READ datasets are non-null.

SOURCE: UPDATE

UD004 - DATASET NOT LOCAL, DN = *dn*

CLASS: Not reprievable; UPDATE abort

CAUSE: The PL has not been accessed or the primary input dataset has not been accessed.

ACTION: Access the dataset prior to UPDATE execution.

SOURCE: UPDATE

UD005 - RECURSIVE READ, DN = *dn*

CLASS: Fatal

CAUSE: A recursive input situation caused by READ directives was detectd by UPDATE. UPDATE will abort at end of execution.

ACTION: Check all READ directives.

SOURCE: UPDATE

UD006 - ILLEGAL READ DSNAME: *n*

CLASS: Fatal

CAUSE: An illegal dataset name was specified by a READ directive. UPDATE will abort at end of execution.

ACTION: Check READ directives.

SOURCE: UPDATE

UD007 - CONTROL STATEMENT ERROR

CLASS: Not reprievable; immediate UPDATE abort

CAUSE: The cause is one or more of the following:

1. PL name = new PL name
2. F and Q both specified
3. P=0 and I=0
4. Master character or comment character illegal
5. Illegal data width

ACTION: Fix the UPDATE control statement.

SOURCE: UPDATE

UD008 - *ERROR* MODS W/OUT ID - N=0

CLASS: Fatal

CAUSE: A new PL has been requested, but no mod ID was specified for modifications. UPDATE has suppressed new PL creation and will abort at the end of execution.

ACTION: A mod ID must be supplied to create a new PL.

SOURCE: UPDATE

UD009 - MASTER CHARACTER MISMATCH

CLASS: Not reprievable; immediate UPDATE abort

CAUSE: The master character supplied by the control statement parameter * does not match the master character recorded by UPDATE on the PL.

ACTION: Change the control statement * parameter.

SOURCE: UPDATE

UD010 - P IS NOT A PROGRAM LIBRARY

CLASS: Not reprievable; immediate UPDATE abort

CAUSE: The PL contains erroneous data.

ACTION: Determine if the P parameter specifies a program library.

SOURCE: UPDATE

UD011 - *n* UPDATE ERRORS

CLASS: Informative

CAUSE: Directive errors were encountered.

ACTION: Correct the directive.

SOURCE: UPDATE

UD012 - PL FORMAT CONVERSION COMPLETE

CLASS: Informative

CAUSE: An old-format PL was converted by UPDATE in a preliminary pass. UPDATE execution is proceeding.

ACTION: Not applicable

SOURCE: UPDATE

UD013 - SEQ. NO. > 131071 DIVIDE TEXT

CLASS: Fatal

CAUSE: A deck or common deck or modification set has reached the card limit for a single identifier name. UPDATE will abort at end of execution.

ACTION: Split the text into 2 or more decks or insertions.

SOURCE: UPDATE

UD014 - NUMBER OF IDENTS > 16383

CLASS: Fatal

CAUSE: A PL can have a maximum of 16383 names (decks, common decks and mod ids).

ACTION: Resequence the PL by converting it to source and making a new PL, thus removing all mod ids or combine several decks into one deck.

SOURCE: UPDATE

UD015 - UNKNOWN "Q" DK: *n* *ERROR*

CLASS: Fatal

CAUSE: A deck or common deck specified by the Q control statement parameter is unknown to UPDATE. UPDATE will abort at end of execution.

ACTION: Change the control statement or determine if all decks specified are known to UPDATE.

SOURCE: UPDATE

UD016 - PL MASTER CHARACTER IS: *m*

CLASS: Informative

CAUSE: The PL master character is not the default (*).

ACTION: Not applicable

SOURCE: UPDATE

UD017 - MOD ID SKIPPED: *name*

CLASS: Informative

CAUSE: A modification group was skipped by UPDATE. Execution is proceeding.

ACTION: Not applicable

SOURCE: UPDATE

UD018 - *n* UNPROCESSED MODS

CLASS: Informative

CAUSE: Modifications were not processed for one of the following reasons:

1. UPDATE was unable to locate a specific card of a known identifier.
2. Quick mode operation with modification applied to decks not selected.

ACTION: Not applicable

SOURCE: UPDATE

UD019 - *n* LONG INPUT RECORDS

CLASS: Informative

CAUSE: Input records greater than 80 columns were truncated at column 80.

ACTION: Not applicable

SOURCE: UPDATE

UD020 - EXCESSIVE INPUT ERRORS - ABORT

CLASS: Fatal

CAUSE: More than 100 input errors were detected. UPDATE aborted. A DECK or COMDECK directive may be missing.

ACTION: Correct the errors.

SOURCE: UPDATE

UD021 - *n* OVERLAPPING MODS

CLASS: Informative

CAUSE: Inserted cards were referenced by subsequent mods.

ACTION: Check input to determine if the overlaps were proper and expected.

SOURCE: UPDATE

UT004 - ILLEGAL CHARACTER IN DECIMAL CONVERSION

CLASS: Fatal; job is terminated

CAUSE: One of the numeric parameters given on the control statement (IW, IR, IF, NW, NR, NF, IS, or NS) contains a non-numeric character.

ACTION: Correct the parameter with the non-numeric value.

SOURCE: DSDUMP

UT005 - INITIAL SECTOR BEYOND END OF DATASET

CLASS: Fatal; job is terminated

CAUSE: The IS parameter value given on the control statement is greater than the size of the input dataset.

ACTION: Correct the IS value.

SOURCE: DSDUMP

UT006 - BAD PARAMETER IN DSDUMP CALL

CLASS: Fatal; job is terminated

CAUSE: One of the numeric parameters given on the control statement (IW, IR, IF, NW, NR, NF, IS, or NS) contains a non-numeric character.

ACTION: Correct the parameter with the non-numeric value.

SOURCE: DSDUMP

UT101 - SYSTEM ERROR ON READING *dn*

CLASS: Fatal; user-requested abort

CAUSE: DUMP asked the operating system to move part of $DUMP from disk to the I/O buffer and to suspend DUMP until the move was completed. When DUMP was resumed, the dataset parameter table (DSP) indicated that no data was put into the buffer.

ACTION: Determine that the DUMPJOB statement precedes the DUMP statement or that the input dataset has been properly created and accessed. Otherwise, report this to the site analyst since the system might be doing the I/O incorrectly.

SOURCE: DUMP

UT102 - JCB POINTERS IN *dn* ARE INVALID SO DUMP CANNOT LOCATE THE LFT AND DSP'S

CLASS: Informative; processing continues

CAUSE: When DSP is specified on the DUMP statement, DUMP uses the information contained in the Job Communication Block (JCB) to locate the LFT and DSPs in $DUMP. Before using this information, DUMP validates it so that it is not pointing to the wrong area.

ACTION: Use the FW and LW parameters to dump at least words 0 through 200 to see what information has overwritten the JCB. This should give a clue as to what part of the job is storing information in the wrong area.

SOURCE: DUMP

UT104 - END OF DUMP

CLASS: Informative

CAUSE: DUMP has completed printing all requested information.

ACTION: Not applicable

SOURCE: DUMP

UT110 - FORMAT EXCHANGE PACKAGE CALLED WITH DSP ADDRESS = 0

CLASS: Fatal; user-requested abort

CAUSE: The dataset parameter table address for the listing dataset is a required parameter for the subprogram that formats the exchange package. A value of 0 is not legal.

ACTION: If the user program calls subprogram FXP, check the code to determine if the DSP address is passed to FXP. If it is a standard CRI product, such as DUMP, that has called FXP, report this to the site analyst.

SOURCE: DUMP

UT111 - FORMAT B REGISTERS CALLED WITH DSP ADDRESS = 0

CLASS: Fatal; user-requested abort

CAUSE: See message UT110; substitute FBR for EXP.

ACTION: See message UT110; substitute FBR for EXP.

SOURCE: DUMP

UT112 - FORMAT T REGISTERS CALLED WITH DSP ADDRESS = 0

CLASS: Fatal; user-requested abort

CAUSE: See message UT110; substitute FTR for EXP.

ACTION: See message UT110; substitute FTR for EXP.

SOURCE: DUMP

UT113 - FORMAT V REGISTERS CALLED WITH DSP ADDRESS = 0

CLASS Fatal; user-requested abort

CAUSE: See message UT110; substitute FVR for EXP.

ACTION: See message UT110; substitute FVR for EXP.

SOURCE: DUMP

UT115 - END OF INPUT REACHED BEFORE LW

CLASS: Fatal; user-requested abort

CAUSE: DUMP uses the LW parameter value from the DUMP control statement and the BA and LA values from the exchange package to determine when to stop reading $DUMP for more memory image. An error occurred when DUMP calculated that there was more to read and the system calculated that there was no more to read.

ACTION: Determine if $DUMP was properly created and accessed by the DUMPJOB statement. If it was, report this to the site analyst.

SOURCE: DUMP

UT120 - ILLEGAL FORMAT TYPE :*f* * ,ASSUMING O

CLASS: Informative; processing continues

CAUSE: DUMP gives the option of printing the memory in several different formats. The type specified *f* was a type not allowed. DUMP used the O or octal format.

ACTION: Check the DUMP control statement for the specified format type and compare with the CRAY-OS Version 1 Reference Manual for the allowed types.

SOURCE: DUMP

UT203 - *dn* DOES NOT EXIST

CLASS: Fatal; user-requested abort

CAUSE: When DUMP began reading the specified input dataset, it found the dataset to be empty.

ACTION: The DUMPJOB statement must precede the DUMP statement and the input dataset must be properly created and accessed.

SOURCE: DUMP

UT400 - $\left\{\begin{matrix}\text{EOD}\\\text{EOF}\\\text{BOD}\end{matrix}\right\}$ENCOUNTERED ON *dn*

CLASS: Informative

CAUSE: This message indicates the type of block control word that was encountered on dataset *dn* just before terminating. If it was EOF or EOD, the dataset is backspaced, positioning it before the EOF or EOD.

ACTION: Not applicable

SOURCE: SKIP

UT403 *x* RECORDS *y* FILES SKIPPED ON *dn*

CLASS: Informative

CAUSE: This message indicates the actual number of records (*x*) and files (*y*) that were bypassed on dataset *dn*. It appears at the end of every SKIP job, whether the termination is normal or abort.

ACTION: Not applicable

SOURCE: SKIP

UT430 - NF or NR PARAMETER NON-NUMERIC

CLASS: Fatal; job aborts

CAUSE: The NF or NR parameter on the control statement contains a non-numeric character.

ACTION: Correct the NF or NR value.

SOURCE: SKIP

UT431 - ERROR OPENING FILE

CLASS: Fatal; job aborts

CAUSE: An undefined error occurred during an attempt to open the input dataset.

ACTION: See system analyst.

SOURCE: SKIP

UT432 - ERROR - INPUT FILE POSITIONED AFTER EOD

CLASS: Fatal; job terminates

CAUSE: An attempt was made to skip forward, but the dataset was already positioned after the end of data (EOD). The dataset was then backspaced before the EOD, and the job was terminated.

ACTION: Reposition the dataset before executing a SKIP statement.

SOURCE: SKIP

UT433 - ILLEGAL CHARACTER IN DECIMAL CONVERSION

CLASS: Fatal; job aborts

CAUSE: The NF or NR parameter on the control statement contains a non-numeric character.

ACTION: Correct the NF or NR value.

SOURCE: SKIP

UT434 - ILLEGAL SKIP AFTER WRITE

CLASS: Fatal; job terminates

CAUSE: The DSP flag indicates that the previous operation on this dataset was a write, and a SKIP now is illegal.

ACTION: Correct error

SOURCE: SKIP

UT435 - DATASET ALREADY AT EOF

CLASS: Fatal; job terminates

CAUSE: An attempt was made to skip forward on a SKIPR call but the dataset is already positioned after an EOF.

ACTION: Reposition the dataset before executing a SKIPR statement.

SOURCE: SKIP

UT436 - ATTEMPT TO BACKUP FROM BOD

CLASS: Fatal; job terminates

CAUSE: The dataset is already positioned at the beginning of data and the skip direction is reverse.

ACTION: Correct error

SOURCE: SKIP

UT501 - 16-CHARACTER PDS NAME ILLEGAL

CLASS: Fatal; job aborts

CAUSE: The PDS parameter value given on the control statement is too long.

ACTION: Correct the PDS name to 15 characters or less.

SOURCE: PDSDUMP

UT502 - 16-CHARACTER USER NUMBER ILLEGAL

CLASS: Fatal; job aborts

CAUSE: The US parameter value given on the control statement is too long.

ACTION: Correct the US number to 15 characters or less.

SOURCE: PDSDUMP

UT503 - ED PARAMETER VALUE NON-NUMERIC

CLASS: Fatal; job aborts

CAUSE: The ED parameter value given on the control statement contains a non-numeric character.

ACTION: Correct the ED value.

SOURCE: PDSDUMP

UT504 - INVALID CONTROL WORD VALUE

CLASS: Fatal; job aborts

CAUSE: The CW parameter value given on the control statement does not match the installation-defined control word.

ACTION: Correct the CW value.

SOURCE: PDSDUMP

UT505 - US PARAMETER EXCLUDES USE OF I/O PARAMETERS

CLASS: Fatal; job aborts

CAUSE: An ID parameter was specified on the control statement as well as on the I or the O keyword. It is illegal to request spooled datasets when the ID is specified.

ACTION: Eliminate either the ID parameter or the I (or O) parameter.

SOURCE: PDSDUMP

UT506 - ED PARAMETER REQUIRES PDS PARAMETER AND EXCLUDES I/O PARAMETERS

CLASS: Fatal; job aborts

CAUSE: An ED parameter was specified on the control statement and either a PDS parameter was not specified or an I or O keyword was specified.

ACTION: Correct the combination of ED, PDS, I, and O parameters.

SOURCE: PDSDUMP

UT507 - ERROR ON SYSTEM DATASET CATALOG PAGE REQUEST

CLASS: Fatal; job terminates

CAUSE: Undefined error returned after PDM page request. The DSC might be bad.

ACTION: See system analyst

SOURCE: PDSDUMP

UT508 - DELAY LIST FULL

CLASS: Fatal; job is terminated

CAUSE: The permanent dataset cannot be accessed now, but the PDSDUMP access delay list is already full.

ACTION: Rerun PDSDUMP job at a time when the requested datasets are less likely to be in use.

SOURCE: PDSDUMP

UT509 - UNABLE TO ACCESS ALL DATASETS. $OUT CONTAINS DATASETS NOT DUMPED

CLASS: Fatal; job terminates

CAUSE: PDSDUMP was unable to access and dump all the datasets after three passes through the access delay list.

ACTION: Rerun PDSDUMP job at a less busy time.

SOURCE: PDSDUMP

UT510 - SDT NOT FOUND *dn* ED = *xxxx*

CLASS: Not reprievable; job terminates

CAUSE: A bad status was returned by PDM during attempt to dequeue the SDT of spooled dataset *dn*. The SDT table might be bad.

ACTION: See a systems analyst.

SOURCE: PDSDUMP

UT511 - UNRECOVERABLE ACCESS ERROR *dn* ED = *xxxx*

CLASS: Not reprievable; job terminates

CAUSE: An undefined bad status was returned by PDM when trying to access dataset *dn*.

ACTION: See a systems analyst.

SOURCE: PDSDUMP

UT512 - DATASET BY-PASSED *dn* ED = *xxxx*

CLASS: Informative; job continues processing

CAUSE: Dataset *dn* has already been accessed by this job.

ACTION: Not applicable

SOURCE: PDSDUMP

UT513 - DELETED WHILE WAITING *dn* ED = *xxxx*

CLASS: Informative; job continues processing

CAUSE: The DSC does not contain the requested dataset *dn*.

ACTION: Not applicable

SOURCE: PDSDUMP

UT514 - DUMP TIME ERROR *dn* ED = *xxxx*

CLASS: Not reprievable; job terminates

CAUSE: A bad status was returned by PDM when trying to put the current time into the DSC entry of the specified dataset *dn*.

ACTION: See system analyst.

SOURCE: PDSDUMP

UT515 - DELETE ERROR *dn* ED = *xxxx*

CLASS: Not reprievable; job terminates

CAUSE: A bad status was returned by PDM when dumped dataset *dn* was being deleted.

ACTION: See a systems analyst.

SOURCE: PDSDUMP

UT516 - QUEUE SDT ERROR *dn* ED = *xxxx*

CLASS: Not reprievable; job terminates

CAUSE: A bad status was returned by PDM when trying to requeue the SDT of spooled dataset *dn*.

ACTION: See a systems analyst.

SOURCE: PDSDUMP

UT517 - DAT CONTINUATION ERROR *dn* ED = *xxxx*

CLASS: Not reprievable; job terminates

CAUSE: An undefined error occurred during reading of a DSC continuation page or the page read did not have its continuation flag bit set.

ACTION: See a systems analyst.

SOURCE: PDSDUMP

UT518 - I/O ERROR WHILE READING *dn*

CLASS: Fatal; job aborts

CAUSE: An error occurred during reading of dataset *dn*.

ACTION: See system analyst.

SOURCE: PDSDUMP

UT519 - US PARAMETER DOES NOT MATCH JOB US

CLASS: Fatal; job aborts

CAUSE: No CW parameter has been specified and the US parameter value given on the PDSDUMP control statement does not match the US parameter value from the JOB statement.

ACTION: Change the US parameter on either the JOB statement or the PDSDUMP statement.

SOURCE: PDSDUMP

UT520 - IF CW NOT SPECIFIED, JOB MUST HAVE US

CLASS: Fatal; job aborts

CAUSE: No CW parameter was specified on the PDSDUMP control statement, and no US parameter was specified on the JOB statement.

ACTION: Include a US parameter on the JOB card if CW is not specified on the PDSDUMP statement.

SOURCE: PDSDUMP

UT601 - EXTRACT LINE LIMIT EXCEEDED

CLASS: Informative

CAUSE: The maximum line count was reached before the entire input dataset was searched.

ACTION: Increase the line limit in the directive set if all entries that match the other directives criteria are to be output.

SOURCE: EXTRACT

UT601 - 16-CHARACTER PDS NAME ILLEGAL

CLASS: Fatal; job aborts

CAUSE: The PDS parameter value given on the control statement is too long.

ACTION: Correct the PDS name to 15 characters or less.

SOURCE: PDSLOAD

UT602 - *nn* RECORDS READ FROM *dataset*

CLASS: Informative

CAUSE: The end of an EXTRACT report has been reached; *nn* is the number of records that had been read from the input dataset when the maximum line count was satisfied or when the end-of-file was found.

ACTION: Not applicable

SOURCE: EXTRACT

UT602 - 16-CHARACTER USER NUMBER ILLEGAL

CLASS: Fatal; job aborts

CAUSE: The US parameter value given on the control statement is too long.

ACTION: Correct the US length to 15 characters or less.

SOURCE: PDSLOAD

UT603 - *nn* RECORDS WRITTEN ON *dataset*

CLASS: Informative

CAUSE: *nn* is the actual number of records that comprise the EXTRACT report and are written on the output dataset.

ACTION: Not applicable

SOURCE: EXTRACT

UT604 - ERROR READING SYSLOG

CLASS: User-requested abort

CAUSE: An unidentified error occurred when a record was being read from the input dataset.

ACTION: If further attempts to run EXTRACT jobs result in this same message, consult the site analyst.

SOURCE: EXTRACT

UT604 - INVALID CONTROL WORD VALUE

CLASS: Fatal; job aborts

CAUSE: The CW parameter value given on the control statement does not match the installation-defined control word.

ACTION: Correct the CW value.

SOURCE: PDSLOAD

UT605 - US PARAMETER EXCLUDES USE OF I/O PARAMETERS

CLASS: Fatal; job aborts

CAUSE: An ID parameter was specified on the control statement as well as on the I or the O keyword. It is illegal to request spooled datasets when the ID is specified.

ACTION: Eliminate either the ID parameter or the I (or O).

SOURCE: PDSLOAD

UT606 - ED PARAMETER REQUIRES PDS PARAMETER AND EXCLUDES USE OF I/O PARAMETERS

CLASS: Fatal; job aborts

CAUSE: An ED parameter was specified on the control statement and either a PSD parameter was not specified or an I or O keyword was specified.

ACTION: Correct the combination of ED, PDS, I, and O parameters.

SOURCE: PDSLOAD

UT607 - ED PARAMETER VALUE NON-NUMERIC

CLASS: Fatal; job aborts

CAUSE: The ED parameter value given on the control statement contains a non-numeric character.

ACTION: Correct the ED value.

SOURCE: PDSLOAD

UT608 - DATASET NOT CREATED BY PDSDUMP

CLASS: Fatal; job aborts

CAUSE: The dataset specified by the PDS parameter does not have the correct header block to indicate a PDSDUMP creation.

ACTION: Verify the PDS parameter values on this PDSLOAD job and the previous PDSDUMP job.

SOURCE: PDSLOAD

UT609 - DATASET BY-PASSED *dn* ED = *xxxx*

CLASS: Informative; job continues processing

CAUSE: An attempt was made to load edition *xxxx* of dataset *dn*. PDM already exists in the dataset catalog.

ACTION: Not applicable

SOURCE: PDSLOAD

UT610 - DATASET LOAN ERROR *dn* ED = *xxxx*

CLASS: Not reprievable; job aborts

CAUSE: An undefined error occurred while dataset *dn* was being loaded.

ACTION: See a systems analyst.

SOURCE: PDSLOAD

UT611 - I/O ERROR WHILE LOADING *dn*

CLASS: Fatal; job aborts

CAUSE: An I/O error occurred during loading of dataset *dn*, or the number of blocks written does not match the number read.

ACTION: See a systems analyst.

SOURCE: PDSLOAD

UT612 - US PARAMETER DOES NOT MATCH JOB US

CLASS: Fatal; job aborts

CAUSE: No CW parameter has been specified and the US parameter value given on the PDSLOAD control statement does not match the US parameter value from the JOB statement.

ACTION: Change the US parameter on either the JOB statement or the PDSLOAD statement.

SOURCE: PDSLOAD

UT613 - IF CW NOT SPECIFIED, JOB MUST HAVE US

CLASS: Fatal; job aborts

CAUSE: No CW parameter was specified on the PDSDUMP control statement, and no US parameter was specified on the JOB statement.

ACTION: Include a US parameter on the JOB statement if CW is not specified on the PDSLOAD statement.

SOURCE: PDSLOAD

UT650 - INSUFFICIENT MEMORY FOR RECORD AREA

CLASS: Fatal; job is aborted

CAUSE: Memory is not large enough to contain the number of records specified.

ACTION: Increase the job's memory size.

SOURCE: WRITEDS

UT901 - NOT ENOUGH MEMORY TO CONTAIN THE BINARY

CLASS: Fatal; job aborts

CAUSE: The input buffer is not large enough to contain the input binary dataset.

ACTION: Increase the job's memory size.

SOURCE: UNBLOCK

UT902 - FATAL ASSEMBLY ERROR IN INPUT BINARY

CLASS: Fatal; job aborts

CAUSE: The PDT table of the binary module indicates a fatal error at assembly time.

ACTION: The module must be re-assembled, either with the fatal error corrected or with DEBUG specified on the CAL statement.

SOURCE: UNBLOCK

UT903 - USE DEBUG PARAMETER ON CAL STATEMENT IF YOU STILL WANT THE BINARY

CLASS: Informative

CAUSE: This is a continuation of message UT902, which was caused by detecting a fatal assembly error. It always follows UT902 and precedes the job step abort.

ACTION: The module must be re-assembled, either with the fatal error corrected or with DEBUG specified on the CAL statement.

SOURCE: UNBLOCK

BAD SUBSCRIPT IN DATA STATEMENT

CLASS: Fatal

CAUSE: A subscript must be an integer constant or constant name in a DATA statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

BAD TRIP COUNT IN IMPLIED DO

CLASS: Fatal

CAUSE: Incrementation parameter m_3 has been assigned a value of 0 or $(m_2-m_1+m_3)/m_3$ is negative or 0.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

CALL OF NON EXTERNAL FUNCTION "*name*"

CLASS: Fatal

CAUSE: Called external procedure does not exist or a name has been used for both a variable and an external procedure.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

CAUTION - TYPE OF "*name*" NOT DECLARED

CLASS: Caution

CAUSE: *name* was declared EXTERNAL but did not appear in an explicit type statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

CHARACTER COUNT TOO LARGE

CLASS: Fatal

CAUSE: An R-form Hollerith constant is specified with more than eight characters or an H- or L-form Hollerith constant is specified with more than eight characters in other than a DATA statement or an actual argument list.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

CHARACTER LENGTH MUST BE >0 AND <505

CLASS: Fatal

CAUSE: A character entity must be assigned a length which is greater than 0 and less than 505.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

COMPILER ERROR

CLASS: Fatal

CAUSE: The CFT compiler detected an error in its internal tables.

ACTION: See a Cray Research analyst.

CONFLICTING TYPE FOR INTRINSIC FUNCTION "*name*" IGNORED

CLASS: Fatal

CAUSE: A type statement cannot change the type of an intrinsic function. The function must be declared EXTERNAL before its type can be changed.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

CONSTANT DIMENSION TOO LARGE

CLASS: Fatal

CAUSE: All dimensions must be less than 2^{22}.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

DEPENDENCY INVOLVING ARRAY *"name"* IN SEQUENCE NUMBER *n*

CLASS: Informative

CAUSE: A dependency exists, involving an array reference in sequence number *n* and an array reference in the AT SEQUENCE NO *m* line that precedes this message. This dependency inhibits DO-loop vectorization.

ACTION: Change program to eliminate dependency.

DIMENSION COUNT > SEVEN

CLASS: Fatal

CAUSE: More than seven dimensions appear in an array declarator.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

DIMENSION EXCEEDED

CLASS: Fatal

CAUSE: A subscript in a DATA statement element exceeds the corresponding array declaration.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

DIVIDE BY ZERO

CLASS: Fatal

CAUSE: Dividing by the constant 0 is illegal.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

DO ILLEGAL ON CONDITIONAL STATEMENT

CLASS: Fatal

CAUSE: This type of statement is not allowed as the conditional statement of a direct logical IF statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

DO INDEX ACTIVE

CLASS: Fatal

CAUSE: The loop control variable is already active from a previous loop.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

DO INDEX IN INPUT LIST

CLASS: Fatal

CAUSE: An attempt was made to read data into a DO variable.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

DO LOOP MAY NOT CROSS BLOCK BOUNDARY

CLASS: Fatal

CAUSE: A DO-loop that begins within an IF-block, ELSE-block or ELSE IF-block must be totally contained within that block. A block that begins within a DO-loop must be totally contained within the loop.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

DO TERMINATOR ILLEGAL IN CONDITIONAL BLOCK STATEMENT

CLASS: Fatal

CAUSE: DO-loop must not terminate on an IF(*e*)THEN, ELSE, ELSE IF, or ENDIF statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

DYNAMIC BLOCK "*name*" NOT IN PREVIOUS COMMON

CLASS: Fatal

CAUSE: Dynamic name must be declared as a common block previous to its appearance in a DYNAMIC compiler directive.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

EBCDIC NOT IMPLEMENTED

CLASS: Fatal

CAUSE: The current version of CFT allows only ASCII characters.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ENTIRE COMMON BLOCK OR EQUIVALENCE CONTAINING "*name*" MUST BE SAME TYPE

CLASS: Fatal

CAUSE: It is illegal to mix character and noncharacter entities in the same common block or equivalence.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ENTRY NAME ILLEGAL

CLASS: Fatal

CAUSE: ENTRY name not a function or subroutine name.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ENTRY STATEMENT ILLEGAL IN DO LOOP

CLASS: Fatal

CAUSE: The ENTRY statement must not be used in a DO-loop.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ENTRY STATEMENT ILLEGAL IN MAIN PROGRAM

CLASS: Fatal

CAUSE: The ENTRY statement must not be used in a main program. It is used only in a subroutine or function.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

EQUIVALENCE EXTENDS COMMON BLOCK BASE

CLASS: Fatal

CAUSE: Common block storage was illegally extended by adding storage units preceding the first storage unit specified in the COMMON statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

EQUIVALENCE OF "*name*" IN DIFFERENT COMMON BLOCKS

CLASS: Fatal

CAUSE: An EQUIVALENCE statement must not associate the storage sequences of two different common blocks in the same program unit.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ERROR IN CONSTANT

CLASS: Fatal

CAUSE: Illegal characters in constant, or constant out of range.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

EXPRESSION ILLEGAL IN INPUT LIST

CLASS: Fatal

CAUSE: Input list item is not a variable name, array element name, or array name.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL POINTEE "*name*"

CLASS: Fatal

CAUSE: A pointee cannot be a dummy argument or a pointer. It cannot be equivalenced or be specified in a common block statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL POINTER VARIABLE "*name*"

CLASS: Fatal

CAUSE: A pointer must be a simple variable. It cannot appear in an EQUIVALENCE statement. If defined in a PARAMETER or DATA statement, the definition must not precede its definition as a pointer.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL SPECIFIER FOR I/O CONTROL LIST OPTION

CLASS: Fatal

CAUSE: The I/O control list option selected does not support this specifier.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL STATEMENT LABEL IN IO CONTROL LIST

CLASS: Fatal

CAUSE: A 1- to 5-digit statement number is missing after END= or ERR=.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL STATEMENT SEQUENCE

CLASS: Fatal

CAUSE: An improper sequence of statement types has been encountered (e.g., a GO TO statement followed by a DIMENSION statement).

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL STATEMENT TYPE

CLASS: Fatal

CAUSE: A statement keyword is misspelled (e.g., DIMENSOIN) or is otherwise unidentifiable.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL SUBSCRIPT TYPE "*name*"

CLASS: Fatal

CAUSE: A subscript expression is not of type integer or contains a constant that exceeds $2^{24}-1$.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL SUBSTRING CONSTANT

CLASS: Fatal

CAUSE: An EQUIVALENCED substring must consist only of constants. The constants must be of type integer.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL SYNTAX IN NAMELIST

CLASS: Fatal

CAUSE: Illegal element found in NAMELIST statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL TYPE CONVERSION IN DATA STATEMENT

CLASS: Fatal

CAUSE: The types of a variable and an associated constant in a DATA statement differ. The type conversion required is illegal or undefined.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL TYPE FOR ASSIGNED VARIABLE

CLASS: Fatal

CAUSE: A variable referenced in an ASSIGN statement is not of type integer.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL TYPE LENGTH

CLASS: Fatal

CAUSE: Length specified is not allowed for this data type.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL UNIT NAME

CLASS: Fatal

CAUSE: An input/output unit identifier cannot be recognized as an integer value, a symbolic name, or a Hollerith character string.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL USE OF "*name*" IN I/O LIST

CLASS: Fatal

CAUSE: External, function, or program name not permitted in an I/O list.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL USE OF "*name*"

CLASS: Fatal

CAUSE: Group name referenced previous to its definition in a NAMELIST statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL USE OF ** IN CONSTANT EXPRESSION

CLASS: Fatal

CAUSE: A constant expression specifies exponentiation to a non-integer power.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL USE OF ASSUMED CHARACTER LENGTH

CLASS: Fatal

CAUSE: Character entity with a length of * must be a dummy argument, the symbolic name of a constant or an external function, whose name appears in a FUNCTION or ENTRY statement within the same program unit.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ILLEGAL USE OF ASSUMED SIZE ARRAY "*name*"

CLASS: Fatal

CAUSE: An array with an asterisk for the last dimension cannot be used without subscripts in an I/O statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

MISSING ARGUMENT ON FILE CONTROL STATEMENT

CLASS: Fatal

CAUSE: Unit identifier omitted on BACKSPACE, ENDFILE, or REWIND statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

MISSING COLON

CLASS: Fatal

CAUSE: A required colon has been omitted (for example, in a character substring).

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

MISSING COMMA OR RIGHT PARENTHESIS IN EQUIVALENCE

CLASS: Fatal

CAUSE: A required comma or right parenthesis has been omitted.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

MISSING COMMA

CLASS: Fatal

CAUSE: A required comma has not been specified.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

MISSING CONSTANT LIST

CLASS: Fatal

CAUSE: PARAMETER or DATA statement has not specified a constant list.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

MISSING DSP FOR DATASET "*name*"

CLASS: Fatal

CAUSE: Dataset has not been created or its name has not been assigned as an alias.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

MISSING END STATEMENT

CLASS: Warning

CAUSE: The last or only program unit being compiled lacks an END statement in its last line.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

MISSING LEFT PARENTHESIS IN EQUIVALENCE

CLASS: Fatal

CAUSE: Opening and closing parentheses do not match or required parenthesis is not present in an EQUIVALENCE statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

MISSING LEFT PARENTHESIS

CLASS: Fatal

CAUSE: Opening and closing parentheses do not match or required parenthesis is not present.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

MISSING OR ILLEGAL STATEMENT NUMBER LIST

CLASS: Fatal

CAUSE: Statement labels must be labels of executable statements that appear in the same program unit as the 2- or 3-branch arithmetic IF.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

UNDEFINED ITEM IN CONSTANT EXPRESSION

CLASS: Fatal

CAUSE: A constant expression in a PARAMETER or DATA statement is not specified with constants, the symbolic names of constants, or, in a DATA statement, with the names of implied-DO variables.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

UNDEFINED STATEMENT NUMBER *"number"*

CLASS: Fatal

CAUSE: A referenced statement label is not defined.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

UNDEFINED SUBSCRIPT *"name"*

CLASS: Fatal

CAUSE: The subscript that dimensions an array in a declarative statement is not a constant, a dummy argument, or a previously defined PARAMETER statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

UNDEFINED SUBSCRIPT *"name"* IN EQUIVALENCE

CLASS: Fatal

CAUSE: The subscript that dimensions an array in a declarative statement is not a constant, a dummy argument, or a previously defined PARAMETER statement in an EQUIVALENCE statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

UNEXPECTED END OF STATEMENT

CLASS: Fatal

CAUSE: A statement encountered is syntactically incomplete.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

UNIT=* ILLEGAL FOR DIRECT ACCESS

CLASS: Fatal

CAUSE: [UNIT=]* appeared in a direct access READ or WRITE statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

UNIT=* ILLEGAL FOR UNFORMATTED IO

CLASS: Fatal

CAUSE: [UNIT=]* appeared without a format identifier in a READ or WRITE statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

UNIT=* LEGAL ONLY IF IN READ OR WRITE

CLASS: Fatal

CAUSE: [UNIT=]* appeared in an auxiliary I/O statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

UNKNOWN LOGICAL OPERATOR

CLASS: Fatal

CAUSE: The characters following a period do not represent a logical operator.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

UPPER DIMENSION < LOWER DIMENSION

CLASS: Fatal

CAUSE: The lower dimension is not less than or equal to the upper dimension.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

USE OF END ILLEGAL IN WRITE CONTROL LIST

CLASS: Fatal

CAUSE: END= was illegally specified in a WRITE statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

VALUE NOT ASSIGNED TO FUNCTION NAME

CLASS: Fatal

CAUSE: A value statement for a function is missing in a function subprogram.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

VARIABLE "*name*" USED AS FUNCTION OR ARRAY

CLASS: Fatal

CAUSE: A simple variable is referenced with either subscripts or an argument list.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

VARIABLE DIMENSION ARRAY *"name"* MUST BE DUMMY ARGUMENT

CLASS: Fatal

CAUSE: A variably dimensioned array did not appear as a dummy argument at an entry point.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

VARIABLE DIMENSION ILLEGAL FOR ARRAY IN COMMON

CLASS: Fatal

CAUSE: An array named in a COMMON statement has a subscript that is not a constant.

ACTION: Check for a missing PARAMETER statement.

VARIABLE LIST LONGER THAN CONSTANT LIST

CLASS: Fatal

CAUSE: Constants and variables must correspond one-to-one in a DATA statement.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ZERO TO NEGATIVE POWER

CLASS: Fatal

CAUSE: Raising zero to a zero or negative power produces unpredictable results in an executable program.

ACTION: See the CRAY-1 FORTRAN (CFT) Reference Manual for correct usage.

ACCOUNT.

CLASS: Informative

CAUSE: An ACCOUNT statement was processed.

ACTION: Not applicable

SOURCE: CSP

ARGUMENT *n* IS *x*

CLASS: Informative

CAUSE: A macro operation containing one or more # operators was matched to the source text while a %X+ or %T+ directive was in effect. (This message does not appear for any argument strings that are null.)

ACTION: Not applicable

SOURCE: SKOL macro trace

BAD ADDRESS

CLASS: Non-fatal

CAUSE: A parameter file command contains an out-of-range address, or a required address is missing. The *EBP command requires the second address if the first address is followed by a comma.

ACTION: Correct the parameter file.

SOURCE: Startup

BAD BREAKPOINT NUMBER

CLASS: Non-fatal

CAUSE: The breakpoint number on a *EBP command is missing or not in the range 0-7.

ACTION: Correct the parameter file.

SOURCE: Startup

BAD DATA SPECIFICATION

CLASS: Non-fatal

CAUSE: The value specified on an enter-memory command is invalid or missing.

ACTION: Correct the parameter file.

SOURCE: Startup

BAD END OF FILE

CLASS: Non-fatal

CAUSE: No *END command was found in the parameter file.

ACTION: Correct the parameter file.

SOURCE: Startup

BAD END OF LINE

CLASS: Non-fatal

CAUSE: An unexpected character was found on a parameter file entry when only a carriage return or a blank was legal.

ACTION: Correct the parameter file.

SOURCE: Startup

BAD FORMAT ON FLAW CARD

CLASS: Non-fatal

CAUSE: A card within a *FLAW/*ENDFLW sequence contains a non-octal digit or the C or T indicator was not present where expected.

ACTION: Correct the parameter file.

SOURCE: Startup

BAD MEMORY SIZE

CLASS: Non-fatal

CAUSE: A *MEMSIZ command has a missing or bad memory size value. *MEMSIZ cannot be used to increase the configured memory size.

ACTION: Correct the parameter file.

SOURCE: Startup

BAD PARAMETER VALUE

CLASS: Non-fatal

CAUSE: An invalid option was specified or a keyword parameter was followed by an out-of-range value.

ACTION: Correct the parameter file.

SOURCE: Startup

BLOCKS RECEIVED FROM FRONT END - *n*

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

BLOCKS SENT TO THE FRONT END - *n*

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

CALL ERROR *overlay*

CLASS: Not reprievable

CAUSE: $OVERLAY encountered an error while processing a call.

ACTION: Correct the request and resubmit the job.

SOURCE: OVERLAY

CANNOT READ OVERLAY *overlay name*

CLASS: Not reprievable

CAUSE: The read issued to obtain an overlay failed.

ACTION: Retry the request. If the problem persists, recreate the overlay.

SOURCE: OVERLAY

CATENATION STRING IS *x*

CLASS: Informative

CAUSE: The macro pattern %'#'='#' was encountered in the source text (invoking the macro operator @MG) while a %D directive was in effect.

ACTION: Not applicable

SOURCE: SKOL macro trace

CHANNEL, *channel*, {ON / OFF} §

CLASS: Informative

CAUSE: Operator command issued to Station Call Processor (SCP) from Operator Station.

ACTION: Not applicable

SOURCE: SCP

§ Deferred implementation

CHANNEL SENT

CLASS: Informative

CAUSE: The station CHANNEL command was processed.

ACTION: Not applicable

SOURCE: CSIM

CLASS, $\left\{\begin{matrix}\text{ALL}\\ jcl\end{matrix}\right\}\left\{\begin{matrix}\text{ON}\\ \text{OFF}\end{matrix}\right\}$ §

CLASS: Informative

CAUSE: The command was issued to SCP from the master operator station.

ACTION: Not applicable

SOURCE: SCP

CLASS STRUCTURE LOADED FROM PDN=*pdn*, ID=*id*, ED=*ed*

CLASS: Informative

CAUSE: The job class structure was successfully loaded from the specified permanent dataset.

ACTION: Not applicable

SOURCE: Job Class Recovery

CLASS STRUCTURE ROLL DATASET NOT FOUND - NO RECOVERY POSSIBLE

CLASS: Non-fatal

CAUSE: Permanent Dataset Manager did not find the class structure roll dataset in the DSC.

ACTION: Run a job to establish the desired structure. Default structure is in effect.

SOURCE: Job Class Recovery

CLOSE CALLS - *n*

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

§ Deferred implementation

CPU PRIORITIES AFTER *x.x* CYCLES:

JOB	RHO	OLD CP	NEW CP	NAME
0	.0	.0	.0	*n*

CLASS: Informative

CAUSE: CPU priority calculations are traced in the system log.

ACTION: Not applicable

SOURCE: JSH

DATASET *dn* RECEIVED FROM STATION FRONT END

CLASS: Informative

CAUSE: The dataset was successfully transmitted from the front end.

ACTION: Not applicable

SOURCE: SCP

DATASET CATALOG ALLOCATES RESERVED TRACK *trnum* ON *ldv*

CLASS: Informative

CAUSE: The DSC as defined in the master device label is reserving a track that is already reserved. A flaw might have been added since the previous install.

ACTION: Not applicable

SOURCE: Startup

DATASET LOST FOR JOB *jobname* NUMBER *jsq*

CLASS: Not reprievable

CAUSE: Disk error occurred on the CRAY-1 when dataset was sent to front end.

ACTION: Not applicable

SOURCE: SCP

DATASET SENT: *dn*

CLASS: Informative

CAUSE: The dataset *dn* was sent to the CRAY-1 by a station SAVE or SUBMIT command.

ACTION: Not applicable

SOURCE: CSIM

DATE/TIME MISMATCH ON CLASS STRUCTURE ROLL DATASET

CLASS: Informative

CAUSE: The class structure was being written to disk but had not yet been completely written when the system was interrupted.

ACTION: Not applicable

SOURCE: Job Class Recovery

DATE/TIME MISMATCH PDN=*pdn* ID=*id* ED=*ed*. DEFAULT STRUCTURE USED

CLASS: Non-fatal

CAUSE: The dataset specified on the *JCLASS directive does not appear to be a valid class structure.

ACTION: Correct the dataset or change the *JCLASS directive.

SOURCE: Job Class Recovery

DEADSTART SELECTED - RECOVERY OF ROLLED JOBS DISABLED

CLASS: Informative

CAUSE: Recovery of rolled jobs was requested but the startup option is not a restart.

ACTION: None possible

SOURCE: Recovery of rolled jobs

DEFAULT JOB CLASS STRUCTURE USED

CLASS: Informative

CAUSE: The default job class structure was put into effect.

ACTION: Not applicable

SOURCE: Job Class Recovery

DEVICE, *device*, {ON / OFF} §

CLASS: Informative

CAUSE: The command was issued to SCP from the master operator station.

ACTION: Not applicable

SOURCE SCP

DEVICE SENT

CLASS: Informative

CAUSE: The station DEVICE comand was processed.

ACTION: Not applicable

SOURCE: CSIM

DISCONNECT, *id* §

CLASS: Informative

CAUSE: The command was issued to SCP from the master operator station.

ACTION: Not applicable

SOURCE: SCP

§ Deferred implementation

DISK BLOCKS MOVED - *n*

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

DN=SYSLOG FAILED ACCESS OF ED # *n*

CLASS: Informative

CAUSE: An attempt was made by STATS to access an edition beyond the current one.

ACTION: Check the edition numbers available for retrieval and the correlation between the SDATE and ED parameters or the EDATE and LASTED parameters.

SOURCE: Stats

DROP, *jsq* JOBNAME=*jobname* §

CLASS: Informative

CAUSE: The command was issued to SCP from the operator station.

ACTION: Not applicable

SOURCE: SCP

DROP SENT

CLASS: Informative

CAUSE: The station DROP command was processed.

ACTION: Not applicable

SOURCE: CSIM

---

§ Deferred implementation

DSDUMP IO ERROR

CLASS: Fatal; system abort

CAUSE: An undefined error occurred during the reading of the input dataset in unblocked format.

ACTION: Rerun the job. If the error occurs again, see an analyst.

SOURCE: DSDUMP

ENTER, { CLASS,*job class name* / ID,*id*,*tid* / PRIORITY,*priority* / TIME,*time* } JOBNAME=*jobname* §

CLASS: Informative

CAUSE: The command was issued to the Station Call Processor from the operator station.

ACTION: Not applicable

SOURCE: SCP

§ Deferred implementation

EXPANSION IS *x*

CLASS: Informative

CAUSE: A macro pattern match resulted in the insertion of new text into the expansion buffer while a %X, %X+, %T, or %T+ directive was in effect. (This message does not appear if the replacement string is null).

ACTION: Not applicable

SOURCE: SKOL macro trace

FL EXCEEDED *overlay name*

CLASS: Not reprievable

CAUSE: The specified overlay will not fit into the user's field length.

ACTION: Increase the memory for the job and resubmit.

SOURCE: OVERLAY

FOLLOWED BY *x*

CLASS: Informative

CAUSE: A macro pattern match resulted in the insertion of new text into the expansion buffer while a %X, %X+, or %Y directive was in effect. The text displayed by this message represents the contents of the expansion buffer immediately following the inserted text. Only the first 20 characters are displayed; excess characters are represented by '...'.

ACTION: Not applicable

SOURCE: SKOL macro trace

FORTRAN -- *x*

CLASS: Informative

CAUSE: A SKOL statement was completely processed and the resulting FORTRAN code was being output while a %X, %X+, or %O directive was in effect. The text displayed by this message represents a single FORTRAN statement including any necessary continuation cards. More than one FORTRAN statement, and therefore more than one of these messages, can be generated by a single SKOL statement.

ACTION: Not applicable

SOURCE: SKOL macro trace

HARDWARE ERROR ON UNIT; *dn*

CLASS: Fatal; job aborts

CAUSE: A disk error occurred during an attempt to read from or write to dataset *dn*.

ACTION: Not applicable

I/O ERROR ON $ROLL - DQM REPLY WAS *status*

CLASS: Informative

CAUSE: The disk queue manager returned an error status while reading the roll index.

ACTION: None. A new edition is created. Recovery is not possible.

SOURCE: Recovery of rolled jobs

ILLEGAL DATASET *dataset name*

CLASS: Fatal; job aborts

CAUSE: $OVERLAY detected that an illegal dataset was called.

ACTION: Correct the request and resubmit the job.

SOURCE: OVERLAY

ILLEGAL SINCE MASTER DEVICE

CLASS: Non-fatal

CAUSE: A parameter attempted to make the system master device unavailable or read-only.

ACTION: Correct the parameter file.

SOURCE: Startup

INDEX AI *alloc* MISMATCH ENTRY ZERO AI *alloc*

CLASS: Informative

CAUSE: The allocation unit in the DAT for $ROLL does not match the allocation unit in entry 0. This could result from a user save of a new edition during system operation.

ACTION: None. The dataset is cleared.

SOURCE: Recovery of rolled jobs

INDEX DEVICE *ldv* MISMATCH ENTRY ZERO DEVICE *ldv*

CLASS: Informative

CAUSE: The device named in the DAT for $ROLL does not match the device named in entry 0. This could result from a user save of a new edition during system operation.

ACTION: None. The dataset is cleared.

SOURCE: Recovery of rolled jobs

INTERNAL COUNTER = *n*

CLASS: Informative

CAUSE: An *n*@I= macro operation was executed while a %X, %X+, or %H directive was in effect.

ACTION: Not applicable

SOURCE: SKOL macro trace

IOP SUBSYSTEM IS DOWN

CLASS: Informative

CAUSE: Central Processing Unit has detected that communication with the I/O Subsystem (IOS) has terminated.

ACTION: Not applicable

SOURCE: MEP

IOP SUBSYSTEM IS UP

CLASS: Informative

CAUSE: The I/O Subsystem (IOS) has completed communication connection to Central Processing Unit.

ACTION: Not applicable

SOURCE: MEP

IOP SUBSYSTEM LOGGING IS DISABLED

CLASS: Informative

CAUSE: The I/O Subsystem (IOS) sent message disabling hardware error detection.

ACTION: Not applicable

SOURCE: MEP

IOP SUBSYSTEM LOGGING IS ENABLED

CLASS: Informative

CAUSE: The I/O Subsystem (IOS) sent message enabling hardware error detection.

ACTION: Not applicable

SOURCE: MEP

JOB *jobname* - DAT SPACE FULL (JSQ=*jsq*)

CLASS: Informative

CAUSE: The STP table area does not have sufficient room to read in the roll image DAT. Too many datasets are in the queues or the system had too little DAT space at startup.

ACTION: Verify the size of the DAT area in STP. A deadstart might be necessary.

SOURCE: Recovery of rolled jobs

JOB *jobname* - DATASET *dsname* DAT POINTS TO STP TABLES (JSQ=*jsq*)

CLASS: Informative

CAUSE: A dataset other than $CS or $IN has a DAT pointer that points to the STP table area.

ACTION: Not applicable

SOURCE: Recovery of rolled jobs

JOB *jobname* - DATASET *dsname* DAT VALIDATION ERROR (JSQ=*jsq*)

CLASS: Informative

CAUSE: One of the following errors was found while the DAT was being verified for the named dataset.

1. DAT page numbers are not consecutive from page 1.
2. Continuation page and current page are not both in the STP tables or both in the JTA.
3. JXT ordinal is nonzero for an STP DAT, or it does not match the ordinal in the DNT.
4. A DAT page pointer is out of range.
5. A device named in a DAT is non-existent or down.
6. An AI listed in a DAT is invalid for the device, or is already reserved.
7. DANPA does not point correctly to the next partition header.

ACTION: Not applicable

SOURCE: Recovery of rolled jobs

JOB *jobname* - DATASET *dsname* PDM STATUS *status* ON PSEUDO-ACCESS (JSQ=*jsq*)

CLASS: Informative

CAUSE: Permanent Dataset Manager returned an error status when Recovery attempted to access a dataset for the job being recovered. The dataset resides on a down or release device, or the Dataset Allocation Table in the Job Table Area fails to match the DAT in the DSC.

ACTION: Determine the cause of the error and correct it if possible. Rerun the job.

SOURCE: Recovery of rolled jobs

JOB *jobname* - DATASET *dsname* - QDT INDEX EXCEEDS VALID RANGE *n* (JSQ=*jsq*)

CLASS: Informative

CAUSE: A Dataset Name Table was found in the Job Table Area for the specified job that had a Queued Dataset Table index greater than the current size of the QDT. This can occur if the number of SDT entries is reduced in the new system.

ACTION: None; the job is not recoverable.

SOURCE: Recovery of rolled jobs

JOB *jobname* - DNT CHAIN BAD (JSQ=*jsq*)

CLASS: Informative

CAUSE: Recovery detected an error in the DNT chain. The reason is one or more of the following;

1. A DNT entry points to an inactive memory pool address.
2. The DNT entry points outside the JTA.
3. The dataset name in an active DNT is 0.
4. Permanent dataset DAT pointer points to STP tables.

ACTION: None

SOURCE: Recovery of rolled jobs

JOB *jobname* - I/O ERROR *errnum* ON ROLL DATASET (JSQ=*jsq*)

CLASS: Informative

CAUSE: The Disk Queue Manager returned an error when Recovery of Rolled Jobs attempted to read the roll image for the specified job.

ACTION: Determine the cause of the error and add a flaw card if necessary.

SOURCE: Recovery of rolled jobs

JOB *jobname* - JOB IS NOT RERUNNABLE (JSQ=*jsq*)

CLASS: Informative

CAUSE: A job that is not recoverable is also not rerunnable.

ACTION: None

SOURCE: Recovery of rolled jobs

JOB *jobname* - JOB SUCCESSFULLY RECOVERED (JSQ=*jsq*)

CLASS: Informative

CAUSE: Recovery of a rolled out job has been successfully completed.

ACTION: Not applicable

SOURCE: Recovery of rolled jobs

JOB *jobname* - JOB WILL BE RERUN (JSQ=*jsq*)

CLASS: Informative

CAUSE: A job that cannot be recovered can be rerun.

ACTION: None

SOURCE: Recovery of rolled jobs

JOB *jobname* - JTA IMAGE ROLL DAT BAD ORDINAL (JSQ=*jsq*)

CLASS: Informative

CAUSE: The Job Execution Table ordinal from one or more Dataset Allocation Table pages does not match the job's JXT ordinal.

ACTION: None

SOURCE: Recovery of rolled jobs

JOB *jobname* - JTA IMAGE ROLL DAT NEXT PAGE POINTER BAD (JSQ=*jsq*)

CLASS: Informative

CAUSE: The pointer to the next page of the Dataset Allocation Table for the rollout dataset does not point to the System Task Processor table area.

ACTION: None

SOURCE: Recovery of rolled jobs

JOB *jobname* - JTA IMAGE ROLL DAT PAGE NUMBER ERROR (JSQ=*jsq*)

CLASS: Informative

CAUSE: The image of the rollout dataset Dataset Allocation Table from the Job Table Area does not have consecutive page numbers beginning with page 1.

ACTION: Not applicable

SOURCE: Recovery of rolled jobs

JOB *jobname* - JTA LENGTH (JTL) *jtl* JXT LENGTH (JXJTL) *jxjtl* MISMATCH (JSQ=*jsq*)

CLASS: Informative

CAUSE: The Job Table Area length from field JTL of the roll image JTA does not match field JXJTL in the roll image copy of the Job Execution Table.

ACTION: None

SOURCE: Recovery of rolled jobs

JOB *jobname* - JXT NAME *jobname 1* MISMATCH JTA NAME *jobname 2* (JSQ=*jsq*)

CLASS: Informative

CAUSE: The job name from the Job Execution Table does not match the job name from the Job Table Area. The job name given is from the roll index.

ACTION: None

SOURCE: Recovery of rolled jobs

JOB *jobname* - NO INPUT SDT (JSQ=*jsq*)

CLASS: Informative

CAUSE: No input queue entry corresponding to the specified job could be found. Possible cause is an error such as a dataset residing on a down device.

ACTION: Determine the cause of the error and correct it if possible. Rerun the job.

SOURCE: Recovery of rolled jobs

JOB *jobname* - REQ SIZE *size* GREATER THAN AVAIL. LWA=*addr* FWA=*addr* (JSQ=*jsq*)

CLASS: Informative

CAUSE: A job has a Job Table Area that is too long to be read into available memory. The required amount of space and the first/last available addresses are given.

ACTION: Not applicable

SOURCE: Recovery of rolled jobs

JOB *jobname* - ROLL DAT POINTS TO MISSING DEVICE *ldv* (JSQ=*jsq*)

CLASS: Informative

CAUSE: The roll image for the dataset is on a non-existent device.

ACTION: Not applicable

SOURCE: Recovery of rolled jobs

JOB *jobname* - ROLL DAT POINTS TO DOWN DEVICE *ldv* (JSQ=*jsq*)

CLASS: Informative

CAUSE: The roll image for the dataset is on an unavailable device.

ACTION: Not applicable

SOURCE: Recovery of rolled jobs

JOB *jobname* - ROLLOUT APPARENTLY INCOMPLETE (JSQ=*jsq*)

CLASS: Informative

CAUSE: The system was interrupted when a rollout was initiated but not completed. The date/time of rollout in the first sector of the roll image does not match the date/time in the last word of the field length.

ACTION: None

SOURCE: Recovery of rolled jobs

JOB *jobname* INDEX DECLARES NOT RECOVERABLE (JSQ=*jsq*)

CLASS: Informative

CAUSE: The system roll index indicates that the specified job was in an irrecoverable state when the system was interrupted. The job might never have been rolled out, or it might have performed one of the following non-recoverable functions since it was last rolled:

1. DELETE, ADJUST, or MODIFY of a permanent dataset
2. Random write to a dataset
3. Write following any positioning operation on a dataset
4. Release of a local dataset

ACTION: None; the job is placed in the input queue.

SOURCE: Recovery of rolled jobs

JOB CLASS STRUCTURE RECOVERY COMPLETE

CLASS: Informative

CAUSE: The rolled class structure has been successfully recovered.

ACTION: Not applicable

SOURCE: Job Class Recovery

KILL, *jsq* JOBNAME=*jobname*§

CLASS: Informative

CAUSE: Command issued to Station Call Processor from operator station.

ACTION: Not applicable

SOURCE: SCP

KILL SENT

CLASS: Informative

CAUSE: The station KILL command was processed.

ACTION: Not applicable

SOURCE: CSIM

LIMIT, *v*§

CLASS: Informative

CAUSE: This command was issued to Station Call Processor from the master operator station.

ACTION: Not applicable

SOURCE: SCP

§ Deferred implementation

LIMIT SENT

CLASS: Informative

CAUSE: The station LIMIT command was processed.

ACTION: Not applicable

SOURCE: CSIM

LINK STATUS REQUEST SENT

CLASS: Informative

CAUSE: The station LINK command was processed.

ACTION: Not applicable

SOURCE: CSIM

LIST CANNOT HAVE AN ARGUMENT

CLASS: Fatal; job aborts

CAUSE: A dataset or module name was encountered in a LIST directive. Keyword is the only parameter permitted.

ACTION: Correct the LIST directive.

SOURCE: BUILD

LOGOFF SENT

CLASS: Informative

CAUSE: The station LOGOFF command was processed.

ACTION: Not applicable

SOURCE: CSIM

LOGON SENT

CLASS: Informative

CAUSE: The station LOGON command was processed.

ACTION: Not applicable

SOURCE: CSIM

MACRO DEFINITION; PATTERN IS *x*

CLASS: Informative

CAUSE: The macro pattern %'#'='#' (macro generation, invoking the operator @MG) or %'#'=' '#' (macro catenation, invoking the operator @MC) was encountered in the source text while a %D directive was in effect

ACTION: Not applicable

SOURCE: SKOL macro trace

MASS STORAGE STATUS REQUEST SENT

CLASS: Informative

CAUSE: The STORAGE station command was processed.

ACTION: Not applicable

SOURCE: CSIM

MAXIMUM EDITIONS REACHED

CLASS:

CAUSE: A dataset already exists with the same PDN and ID and the edition number is 4095.

ACTION: Delete or modify the 4095th edition.

SOURCE: PDM

MAXIMUM MEMORY USED - *n* WORDS

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

MEMORY INTEGRAL (EXECUTION TIME) *n* MWDS-SEC

CLASS: Informative

CAUSE: When the CHARGES program is run, this message issues if requested.

ACTION: Not applicable

SOURCE: CHARGES

MEMORY INTEGRAL (I/O WAIT TIME) - *n* MWDS-SEC

CLASS: Informative

CAUSE: When the CHARGES program is run, this message issues if requested.

ACTION: Not applicable

SOURCE: CHARGES

MEMORY PRIORITIES AFTER *x.x* CYCLES:

JOB	DELTA TIT	OLD MP	NEW MP	NAME	STATUS
0	.0	.0	.0	*n*	*ε*

CLASS: Informative

CAUSE: Memory priority calulations are traced in the system log.

ACTION: Not applicable

SOURCE: JSH

MEMORY RESIDENT DATASETS - *n*

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

MINIMUM MEMORY USED - *n* WORDS

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

MISSING COMMA

CLASS: Non-fatal

CAUSE: An unexpected character was found when the syntax of a parameter file entry required a comma.

ACTION: Correct the parameter file.

SOURCE: Startup

MISSING STARTUP OPTION

CLASS: Non-fatal

CAUSE: If a parameter file exists, it must contain at least an *INSTALL, *DEADSTART, or *RESTART directive. None were found.

ACTION: Correct the parameter file.

SOURCE: Startup

MISSING/BAD RRJ CODE

CLASS: Non-fatal

CAUSE: An *RRJ command was present but end-of-line was encountered too soon, or the parameter value was out-of-range.

ACTION: Correct the parameter file.

NUMBER OF RESIDENT SDR ENTRIES DECREASED

CLASS: Fatal

CAUSE: The system being started has room for fewer SDR entries in the STP table area than the entries on the roll dataset.

ACTION: Regenerate the system with more SDR entries or restart the system with a parameter file specifying *SDR to allocate a new roll dataset. Then run JSYSDIR to initialize the file entries.

SOURCE: SDR

OLD SYSTEM JXT COUNT *count* GREATER THAN CURRENT SYSTEM COUNT *count*

CLASS: Informative

CAUSE: Recovery is not possible because the previously running system was assembled with more JXT entries than the current system.

ACTION: None. The roll index is cleared.

SOURCE: Recovery of rolled jobs

ONLY FLAW CARD ALLOWED

CLASS: Non-fatal

CAUSE: A non-flaw card was found within a *FLAW/*ENDFLW sequence.

ACTION: Correct the parameter file.

SOURCE: Startup

OPEN CALLS - *n*

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

OPERATOR,*nid*,*tid*,*pw*,*npw*§

CLASS: Informative

CAUSE: Command issued to Station Call Processor from operator station.

ACTION: Not applicable

SOURCE: SCP

OPERATOR MESSAGE (*station operator text*)§

CLASS: Informative

CAUSE: Operator message issued to Station Call Processor from front-end station.

ACTION: Not applicable

SOURCE: SCP

§ Deferred implementation

OPERATOR SENT

CLASS: Informative

CAUSE: The station operator command was processed.

ACTION: Not applicable

SOURCE: CSIM

OVERLAY NOT FOUND *overlay name*

CLASS:

CAUSE: $OVERLAY was unable to find the specified overlay.

ACTION: Correct the request and resubmit the job.

SOURCE: OVERLAY

PARAMETER ERROR COUNT *n*

CLASS: Informative

CAUSE: Parameter checking is complete and ZY gives the total number of errors encountered.

ACTION: Not applicable

SOURCE: Startup

PATTERN MATCHED IS *x*

CLASS: Informative

CAUSE: A macro operation was matched to the source text while a %X, %X+, or %T %T+ directive was in effect.

ACTION: Not applicable

SOURCE: SKOL macro trace

PERMANENT FILE SPACE ACCESSED - *n* BLOCKS

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

PERMANENT FILE SPACE SAVED - *n* BLOCKS

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

PHYSICAL I/O REQUESTS - *n*

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

RECEIVED *jsq pdn* SZ = *size in words* SID = *fe* DID = *fe* TID = *tid* USERID = *account #*

CLASS: Informative

CAUSE: Dataset was received from front end.

ACTION: Not applicable

SOURCE: SCP

RECOVER§

CLASS: Informative

CAUSE: This command was issued to the Station Call Processor from the master operator station.

ACTION: Not applicable

SOURCE: SCP

§ Deferred implementation

RECOVERY OF ROLLED JOBS ABORTED

CLASS: Informative

CAUSE: Recovery is not possible due to a previously described error.

ACTION: Not applicable

SOURCE: Recovery of rolled jobs

REPLACEMENT STRING IS *x*

CLASS: Informative

CAUSE: The macro pattern %'#'='#' was encountered in the source text (invoking the macro operator @MG) while a %D directive was in effect.

ACTION: Not applicable

SOURCE: SKOL macro trace

RERUN, *jsq* JOBNAME=*jobname*§

CLASS: Informative

CAUSE: Command issued to Station Call Processor from operator station.

ACTION: Not applicable

SOURCE: SCP

RERUN SENT

CLASS: Informative

CAUSE: The station RERUN command was processed.

ACTION: Not applicable

SOURCE: CSIM

§ Deferred implementation

RESUME,{*jsq* / ALL} JOBNAME=*jobname*§

CLASS: Informative

CAUSE: Command issued to the Station Call Processor from the master operator station.

ACTION: Not applicable

SOURCE: SCP

RESUME SENT

CLASS: Informative

CAUSE: The station RESUME command was processed.

ACTION: Not applicable

SOURCE: CSIM

ROUTE,*oid*,*nid*§

CLASS: Informative

CAUSE: Command issued to the Station Call Processor from the master operator station.

ACTION: Not applicable

SOURCE: SCP

ROUTE SENT

CLASS: Informative

CAUSE: The station ROUTE command was processed.

ACTION: Not applicable

SOURCE: CSIM

§ Deferred implementation

$SDR, FILE ACCESS FAILED *status*

CLASS: Fatal

CAUSE: Permanent Dataset Manager returned the specified error status when STARTUP attempted to access the System Directory rollout dataset.

ACTION: Determine the cause of the error and correct it. The JSYSDIR job can be run to initialize the System Directory.

SOURCE: SDR

$SDR, FILE ADJUST FAILED *status*

CLASS: Fatal

CAUSE: Permanent Dataset Manager returned the specified error status when STARTUP attempted to increase the size of the System Directory roll dataset to allow for an increase in the number of SDR entries in the system being started.

ACTION: If the $SDR roll dataset is unusable, restart the system, specifying *SDR in the parameter file to force a new edition to be created. Then run a JSYSDIR job to initialize the $SDR entries.

SOURCE: SDR

SHUTDOWN§

CLASS: Informative

CAUSE: The command was issued to the Station Call Processor from the master operator station.

ACTION: Not applicable

SOURCE: SCP

§ Deferred implementation

STATUS *status* ON ACCESS OF *JCLASS DATASET PDN=*pdn*, ID=*id*, ED=*ed* DEFAULT STRUCTURE USED

CLASS: Non-fatal

CAUSE: STARTUP was unable to access the specified dataset; the default structure was put into effect.

ACTION: Determine the cause of the error and correct it. Run a job to establish the desired structure.

SOURCE: Job Class Recovery

STREAM,*id*,*ni*,*no*,*na*§

CLASS: Informative

CAUSE: Command issued to the Station Call Processor from the operator station.

ACTION: Not applicable

SOURCE: SCP

STREAM SENT

CLASS: Informative

CAUSE: The station STREAM command was processed.

ACTION: Not applicable

SOURCE: CSIM

SUSPEND, $\left\{ \begin{matrix} jsq \\ \text{ALL} \end{matrix} \right\}$ JOBNAME=*jobname*§

CLASS: Informative

CAUSE: The command was issued to the Station Call Processor from the master operator station.

ACTION: Not applicable

SOURCE: SCP

§ Deferred implementation

SUSPEND SENT

CLASS: Informative

CAUSE: The station SUSPEND command was processed.

ACTION: Not applicable

SOURCE: CSIM

SWITCH,*jsq*,*ssw*, { ON / OFF } §

CLASS: Informative

CAUSE: Command issued to the Station Call Processor from the operator station.

ACTION: Not applicable

SOURCE: SCP

§ Deferred implementation

SYSTEM DUMP AREA ALLOCATES RESERVED TRACK *trnum* ON *ldv*

CLASS: Informative

CAUSE: The area of the master device that has been reserved to contain the system dump produced by DDC contains a track that is already reserved. Usually caused by addition of a flaw.

ACTION: Not applicable

SOURCE: System Dump

SYSTEM DUMP NOT SAVED - PDM RETURNED STATUS *status*

CLASS: Informative

CAUSE: STARTUP was unable to save the system dump as a permanent dataset. The *status* value is the error code returned by PDM. The dump remains available in the reserved area as long as no new dump is taken.

ACTION: Determine the reason for failure to save the dump and correct it. Start the system again.

SOURCE: System Dump

SYSTEM DUMP SAVED - STATUS = *status* PDN = *pdn* UID = *id* ED = *ed*

CLASS: Informative

CAUSE: The system dump was successfully saved with the permanent dataset characteristics given.

ACTION: Not applicable

SOURCE: System Dump

SYSTEM STATUS REQUEST SENT

CLASS: Informative

CAUSE: The station SYSTEM command was processed.

ACTION: Not applicable

SOURCE: CSIM

TEMPORARY FILE SPACE USED - *n* BLOCKS

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

TIME EXECUTING IN CPU - *hh:mm:ss.ssss*

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

TIME WAITING FOR I/O - *hh:mm:ss.ssss*

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

TIME WAITING FOR JXT - *hh:mm:ss.ssss*

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

TIME WAITING TO EXECUTE - *hh:mm:ss.ssss*

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if requested.

ACTION: Not applicable

SOURCE: CHARGES

TOTAL SBUS USED - *n*

CLASS: Informative

CAUSE: When the CHARGES program is run, this message is issued if an installation parameter is set.

ACTION: Not applicable

SOURCE: CHARGES

TRANSMIT *jsq pdn* SZ = *size in words* SID = *fe* DID = *fe* TID = *tid* USERID = *account no*.

CLASS: Informative

CAUSE: Dataset was sent to front end.

ACTION: Not applicable

SOURCE: SCP

UNEXPECTED STATUS *status* ON ACCESS OF CLASS STRUCTURE ROLL DATASET

CLASS: Non-fatal

CAUSE: Permanent Dataset Manager returned the specified error status when STARTUP attempted to access the class structure roll dataset.

ACTION: Run a job to re-establish the desired structure. Default structure is in effect.

SOURCE: Job Class Recovery

UNEXPECTED STATUS ON ACCESS OF $ROLL WAS *status* - NEW EDITION CREATED - RRJ NOT POSSIBLE

CLASS: Informative

CAUSE: PDM returned an unexpected status when RRJ attempted to access the roll index dataset.

ACTION: None possible. Recovery of rolled jobs will not occur.

SOURCE: Recovery of rolled jobs

UNKNOWN LOGICAL DEVICE *ldv*

CLASS: Non-fatal

CAUSE: A parameter file entry specified a device name that has no matching EQT entry.

ACTION: Correct the parameter file.

SOURCE: Startup

UNKNOWN PARAMETER TYPE

CLASS: Non-fatal

CAUSE: ZY was unable to recognize a parameter file entry.

ACTION: Correct the parameter file.

SOURCE: Startup

Z HALTED AT BREAKPOINT DUE TO *DEBUG CARD

CLASS: Informative

CAUSE: A *DEBUG card was encountered during parameter file processing, specifying that STARTUP was to wait for operator action before continuing.

ACTION: Enter "RUN TASK 0" at operator station console to cause STARTUP to begin processing a Restart, Deadstart, or Install.

SOURCE: Startup

Z HALTED AT BREAKPOINT DUE TO ERRORS IN ZY

CLASS: Non-fatal

CAUSE: ZY detected at least one error in the parameter file.

ACTION: Examine ZLOG from the operator station console. If the errors flagged are acceptable, enter "RUN TASK 0" to cause STARTUP to continue. If the errors are not acceptable, correct the parameter file and begin the startup again.

SOURCE: Startup

variable ACCESS ERROR: *xx*

CLASS: Fatal; job aborts

CAUSE: Error number *xx* occurred during an attempt to access dataset *variable*.

ACTION: Not applicable

SOURCE: BUILD

variable ADDED TO TOP OF LABEL STACK AT LEVEL *n-d*

CLASS: Informative

CAUSE: An @LS*d* macro operation was executed while a %X, %X+, or %H directive was in effect.

ACTION: Not applicable

SOURCE: SKOL macro trace

variable ADDED TO TOP OF LABEL STACK (LEVEL *n*)

CLASS: Informative

CAUSE: An @*n*@LS0 macro operation was executed while a %X, %X+, or %H directive was in effect.

ACTION: Not applicable

SOURCE: SKOL macro trace

variable CALCULATED POSITION DISAGREES WITH W@DPRCW

CLASS:

CAUSE: A request was issued to get the currect position in the dataset. The calculated position did not agree with the position recorded in the DSP. Probable system error.

ACTION: Forward a dump of the JTA, DSP, and buffers to the Cray analyst.

SOURCE: POS

variable CHARACTERS IN LONGEST MACRO TRACE STRING

CLASS: Informative

CAUSE: End of a macro translation in which the macro trace feature was activated

ACTION: Not applicable

SOURCE: SKOL

variable DATASET CONTAINS COMPRESSED DUMP - MUST USE XCOMP

CLASS: Fatal

CAUSE: See message.

ACTION: Use XCOMP if decompression is desired.

SOURCE: FDUMP

variable DATASET DAT CONTAINS AI CONFLICT AT TRACK *trnum* ON *ldv*

CLASS: Informative

CAUSE: The specified dataset is reserving a previously reserved track on the specified device. A new flaw has been added or a system error has occurred.

ACTION: None possible. The dataset cannot be accessed.

SOURCE: Permanent dataset recovery

variable DATASET RESIDES ON DOWN DEVICE *ldv*

CLASS: Informative

CAUSE: The specified dataset resides wholly or partially on a device whose EQT entry shows the device is not available, and for which the release flag is not set, or the DAT contains a reference to a device for which no matching EQT entry could be found.

ACTION: Determine why the device is unavailable, correct the EQT, and re-deadstart.

SOURCE: Permanent dataset recovery

variable DATASET RESIDES ON MISSING/RELEASED DEVICE *ldv*

CLASS: Informative

CAUSE: The specified dataset resides wholly or partially on a device whose EQT entry specifies that the device is to be released or on a device for which no matching EQT entry was found.

ACTION: None possible. The dataset has been deleted from the DSC.

SOURCE: Permanent dataset recovery

variable DEVICE LABEL NOT FOUND

CLASS: Fatal

CAUSE: STARTUP was unable to locate a device label on the specified device (*variable*), and the UP flag was not set in the EQT entry for the device.

ACTION: Determine whether the device is functioning properly. If not, correct the hardware problem and attempt to start the system again. If the device is functioning properly, the device label has been destroyed or the track containing it has become unreadable. A new label must be written on the device using the *UP parameter file command. If the device was not properly released before the label was destroyed, one or more permanent datasets might be lost. If the device is the system master device, the *UP command cannot be used, and an install type of startup is necessary.

SOURCE: Startup

variable DSC ENTRY CONTAINS CATASTROPHIC ERROR - ENTRY *status*

CLASS: Informative

CAUSE: The Dataset Catalog entry for the specified device contains at least one error condition from which STARTUP is unable to recover. *status* is deleted if the DSC entry was cleared; *status* is retained if the DSC entry was flagged as an error but left intact. The following error conditions are considered catastrophic:

1. DAT contains an AI that is out of range for the device.
2. A continuation DSC entry does not exist when one is required (current AI count not exhausted, first DAT body is not on first DSC page, etc.).

3. The DSC entry is a second or subsequent entry for a multitype dataset, and a previously encountered entry for that dataset was flagged as having a catastrophic error.
4. The DSC entry contains no DAT bodies at all.
5. The continuation entry pointed to by the DSC entry is not a continuation entry, or points back to a different DSC entry.
6. The DSC entry contains a larger text block than can be handled by the system being started.
7. DAT contains an invalid allocation style.

If the status is retained, the dataset cannot be accessed.

ACTION: If *status* is deleted, none possible. If *status* is retained, examine the DSC entry to determine the reason for the error. Clear the entry with the *DSCERR parameter file card.

SOURCE: Permanent dataset recovery

variable EQUIPMENT CONFIGURATION ILLOGICAL

CLASS: Not-fatal

CAUSE: System is assembled with device *ldv* flagged as UP, or the release bit is set without also setting the not-available bit, or both an *UP and a *DOWN command reference the same device.

ACTION: Reassemble the system or correct the parameter file.

SOURCE: Startup

variable ERROR WHILE COMPRESSING SYMBOL TABLES

CLASS: Fatal

CAUSE: Error occurred when undesirable duplicate symbols (field definitions, etc.), were being deleted.

ACTION: Correct error.

SOURCE: FDUMP

variable FDUMP - AUTO FILE/ARRAY CONTAINS UNRECOGNIZABLE DIRECTIVE

CLASS: Non-fatal

CAUSE: See message.

ACTION: Not applicable

SOURCE: FDUMP

variable FDUMP - BAD CHARACTER WHILE CONVERTING OCTAL PARAMETER

CLASS: Fatal

CAUSE: Input must consist of only octal digits.

ACTION: Correct error.

SOURCE: FDUMP

variable FDUMP - CONTROL STATEMENT ERROR

CLASS: Fatal

CAUSE: See message.

ACTION: Correct control statement.

SOURCE: FDUMP

variable FDUMP - DATASET TO BE DECOMPRESSED UNRECOGNIZABLE

CLASS: Fatal

CAUSE: Bad input dataset. The input dataset must contain a dump with a recognizable header.

ACTION: Correct error.

SOURCE: FDUMP

*** CORRECTED DISK ERROR ***

CLASS: Informative

CAUSE: A hardware disk checkword error was not recovered by retry but the data was corrected by the Disk Error Correction (DEC) system task.

ACTION: Not appropriate

SOURCE: DQM

*** DATASET NOT FOUND: *dn*. *** (LINE *ln*)

CLASS: Warning

CAUSE: A dataset specified on a station SAVE or SUBMIT command could not be found.

ACTION: Determine if the dataset name was misspelled on the station command or if it does not exist. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** DEADSTART ERROR: INVALID STATUS RETURNED BY $RDC (READ CIRCULAR).*** (LINE *ln*)

CLASS: Fatal

CAUSE: The system is unable to perform the physical read.

ACTION: Check for a hardware error. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** DEADSTART ERROR: THE SYSTEM DATASET IS TOO BIG. RUN USING VIRTUAL MEMORY. *** (LINE *ln*)

CLASS: Fatal

CAUSE: The simulated system is too big to fit in real memory.

ACTION: Remove the VMEM=0 parameter from the CSIM control statement. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** DUMP FW PARAMETER IS TOO BIG. DUMP SUPPRESSED. *** (LINE *ln*)

CLASS: Warning

CAUSE: The FW parameter is greater than the LW parameter.

ACTION: Change the DUMP directive parameters. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** DUMP LW PARAMETER IS TOO BIG. DUMP SHORTENED. *** (LINE *ln*)

CLASS: Warning

CAUSE: The LW parameter is past the end of simulated memory.

ACTION: The high memory address is used for LW. If this is not the desired address, change the LW parameter on the DUMP directive. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** END OF FILE ON THE SIMULATOR DIRECTIVE FILE. ***

CLASS: Informative

CAUSE: All simulator directives have been processed.

ACTION: Not applicable

SOURCE: CSIM

*** ERROR IN STATION COMMAND INPUT AT LINE *ln*, ARGUMENT NUMBER *n*.***

CLASS: Warning

CAUSE: A syntax error was encountered in the station commands.

ACTION: Examine argument number *n* in the station command line *ln* and correct any errors. Each station command must be terminated by a semicolon.

SOURCE: CSIM

*** FLOATING POINT ERROR. *** (LINE *ln*)

CLASS: Fatal

CAUSE: A floating-point error occurred in the simulated system.

ACTION: Check the simulated system. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** I/O ERROR IN THE SIMULATED SYSTEM. *** (LINE *ln*)

CLASS: Fatal

CAUSE: The system being simulated attempted to make an invalid I/O request.

ACTION: Check the simulated system. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** ILLEGAL CHANNEL FOR DISK I/O. *** (LINE *ln*)

CLASS: Fatal

CAUSE: The channel specified for disk I/O was not 4 or 5.

ACTION: Change the simulated system to use channels 4 and 5 only. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** ILLEGAL DEADSTART CHANNEL. *** (LINE *ln*)

CLASS: Fatal

CAUSE: Simulator error - An attempt was made to deadstart CRAY-1 over a channel other than channel 2.

ACTION: See site analyst.

SOURCE: CSIM

*** ILLEGAL PARAMETER VALUE ON DIRECTIVE. *** (LINE *ln*)

CLASS: Warning

CAUSE: An invalid value was coded on a keyword in the simulator directive immediately preceding the message.

ACTION: Check and correct the directive. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** INVALID CHANNEL NUMBER: *n* . *** (LINE *ln*)

CLASS: Fatal

CAUSE: An unavailable channel number was specified on a station CHANNEL command.

ACTION: Correct the CHANNEL command to use channel 1 or 12. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** INVALID DISK REQUEST. *** (LINE *ln*)

CLASS: Fatal

CAUSE: The simulated system made an invalid disk request.

ACTION: Check the disk I/O logic in the simulated system. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** INVALID SIMULATOR DUMP PARAMETER. DUMP SUPPRESSED. *** (LINE *ln*)

CLASS: Warning

CAUSE: An invalid parameter was included on the DUMP directive.

ACTION: Check the parameters on the DUMP directive. The valid parameters are:

```
FW=address  first word to dump
LW=address  last word to dump
M=E/J/T  mode is EXEC, JOB, or TASK
```

ln is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** KEYWORD NOT RECOGNIZED ON DIRECTIVE. *** (LINE *ln*)

CLASS: Warning

CAUSE: An invalid keyword was used on the simulator directive immediately preceding the message.

ACTION: Correct the directive. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** NO DNT EXISTS FOR THE *dn* DATASET. *** (LINE *ln*)

CLASS: Fatal

CAUSE: The simulator cannot access the dataset.

ACTION: Check the spelling of *dn*. Determine if *dn* is available to the simulator. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** OPERAND RANGE ERROR. *** (LINE *ln*)

CLASS: Fatal

CAUSE: The simulated system attempted to access memory outside the simulated address range.

ACTION: Check the simulated system. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** PROGRAM RANGE ERROR. *** (LINE *ln*)

CLASS: Fatal

CAUSE: The simulated system attempted to execute instructions outside of the simulated program range.

ACTION: Check the simulated system. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** RECOVERED DISK ERROR ***

CLASS: Informative

CAUSE: A hardware disk error was recovered by retry.

ACTION: Not applicable

SOURCE: DQM

*** RELOAD FROM DATASET *dn* ***

CLASS: Informative

CAUSE: A restart was initiated by an RST directive.

ACTION: Not applicable

SOURCE: CSIM

*** RUN ENCOUNTERED BEFORE START OR RELOAD. *** (LINE *ln*)

CLASS: Fatal

CAUSE: The simulator directives are not properly ordered.

ACTION: Correct the directive file. *ln* is the line in the simulator listing that prints the messages.

SOURCE: CSIM

*** SIMABORT MACRO ENCOUNTERED IN THE SIMULATED SYSTEM. *** (LINE *ln*)

CLASS: Fatal

CAUSE: The SIMABORT macro was executed in the simulated system.

ACTION: Check the portion of the simulated system where the SIMABORT macro was executed. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** SIMULATION COMPLETE ***

CLASS: Informative

CAUSE: All station commands have been processed.

ACTION: Not applicable

SOURCE: CSIM

*** SIMULATION TIME LIMIT EXCEEDED. *** (LINE *ln*)

CLASS: Fatal

CAUSE: The simulation exceeded the amount of time specified on the CSIM control statement.

ACTION: Check the simulated system for a loop or increase the amount of time specified on the CSIM control statement. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** THE DS DIRECTIVE WAS CHANGED TO START. *** (LINE *ln*)

CLASS: Warning

CAUSE: The DS directive is no longer used to deadstart the simulated system.

ACTION: Use the START directive. It has the following parameters:

OPSYS = system (system to simulate, default is COS)
OSPAR = parameters (startup parameter file, default is COSPAR)

ln is the line in the simulator that prints the message.

SOURCE: CSIM

*** THE SIMULATOR TERMINATED AT AN EXIT DIRECTIVE. ***

CLASS: Informative

CAUSE: An EXIT directive was processed.

ACTION: Not applicable

SOURCE: CSIM

*** THE TIMER TABLE OVERFLOWED. *** (LINE *ln*)

CLASS: Fatal

CAUSE: Simulator error - too many requests are queued up.

ACTION: Use fewer STATION commands or recompile the simulator with a larger table.

SOURCE: CSIM

*** UNCORRECTABLE DISK ERROR ***

CLASS: Informative

CAUSE: A hardware disk error was not recoverable by retry and the data not correctable by the Disk Error Correction (DEC) system task.

ACTION: Not appropriate

SOURCE: DQM

*** UNRECOGNIZED SIMULATOR DIRECTIVE: *dir* *** (LINE *ln*)

CLASS: Warning

CAUSE: A non-implemented simulator directive was given.

ACTION: Check the spelling and availability of the directive, *dir*. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** UNRECOGNIZED STATION COMMAND. *** (LINE *ln*)

CLASS: Warning

CAUSE: An invalid station command was detected by the SKOL station command macros.

ACTION: Check the spelling; determine whether the command is implemented in the simulator. *ln* is the line in the simulator listing that prints the message.

SOURCE: CSIM

*** UNRECOVERED DISK ERROR ***

CLASS: Informative

CAUSE: A hardware disk error was not recovered by retry and was not a checkword error.

ACTION: Not appropriate

SOURCE: DQM

*---------- - ERROR - BAD PARAMETER NUMBER *c*

CLASS: Fatal

CAUSE: The character *c* following a # operator in the replacement portion of a macro definition is not a digit in the range from 1 to *n* where *n* is the number of formal parameters in the pattern portion of the macro.

ACTION: Correct the replacement portion of the macro definition. If necessary, turn on SKOL's macro trace feature.

SOURCE: SKOL

*---------- - ERROR - MISSING RIGHT BRACKET

CLASS: Fatal

CAUSE: An error occurred in a user-defined structural macro.

ACTION: Correct the user-defined macro or change the patterns of any user-defined macros that may conflict with the standard macros.

SOURCE: SKOL

*---------- - ERROR - UNCLOSED LOOP OR BLOCK

CLASS: Fatal

CAUSE: An error occurred in a user-defined structural macro.

ACTION: Correct the user-defined macro or change the patterns of any user-defined macros that conflict with the standard macros.

SOURCE: SKOL

*---------- - ERROR - UNCLOSED STRING

CLASS: Fatal

CAUSE: The program ended without the closing apostrophe for a literal string.

ACTION: Check the column between the sequence number and the nesting level for an apostrophe that indicates a literal string continuation.

SOURCE: SKOL

*---------- - ERROR - UNCLOSED COMMENT

CLASS: Fatal

CAUSE: The program ended without the closing quotation mark for a comment.

ACTION: Check the column between the sequence number and the nesting level for a quotation mark that indicates a comment continuation.

SOURCE: SKOL

*---------- ERROR *n*: LEFTOVER '>'

CLASS: Fatal

CAUSE: The character ">" does not match any macro.

ACTION: Check for a missing IF or ELSEIF. Since this character is not generated by any of the standard macros, it is not necessary to turn on the macro trace.

SOURCE: SKOL

*---------- ERROR *n*: LEFTOVER '!'

CLASS: Fatal

CAUSE: The character "!" does not match any macro. Because this character occurs commonly in the standard macros, a failure in the translation process is indicated, perhaps triggered by a user-defined macro.

ACTION: Correct the user-defined macro or change the patterns of any user-defined macros that conflict with the standard macros.

SOURCE: SKOL

*---------- ERROR *n*: LEFTOVER '$%*c*'

CLASS: Fatal

CAUSE: The character string "$%" does not match any macro. Because this character string occurs commonly in the standard macros, a failure in the translation process is indicated, perhaps triggered by a user-defined macro.

ACTION: Correct the user-defined macro, or change the patterns of any user-defined macros that conflict with the standard macros.

SOURCE: SKOL

*---------- ERROR *n*: LEFTOVER '$%%*c*'

CLASS: Fatal

CAUSE: The character string "$%%" does not match any macro. Because this character string occurs commonly in the standard macros, a failure in the translation process is indicated, perhaps triggered by a user-defined macro.

ACTION: Correct the user-defined macro, or change the patterns of any user-defined macros that conflict with the standard macros.

SOURCE: SKOL

*---------- ERROR *n*: LEFTOVER '?'

CLASS: Fatal

CAUSE: The character "?" does not match any macro. Because this character occurs commonly in the standard macros, a failure in the translation process is indicated, perhaps triggered by a user-defined macro.

ACTION: Correct the user-defined macro or change the patterns of any user-defined macros that conflict with the standard macros.

SOURCE: SKOL

*---------- ERROR *n*: MATCHING 'WHEN' STATEMENT EITHER BAD OR SUPPRESSED

CLASS: Fatal

CAUSE: An ENDWHEN statement matches an earlier WHEN statement because they are at the same nesting level. The WHEN statement is enclosed in conditional compilation brackets and is suppressed or refers to an undeclared scalar type.

ACTION: Fix the WHEN statement or ensure that the WHEN and ENDWHEN statements are both inside or both outside any conditional code brackets.

SOURCE: SKOL

*---------- ERROR *n*: MISSING 'COROUTINE'

CLASS: Fatal

CAUSE: An ENDCOROUTINE statement is not preceded by a matching COROUTINE statement.

ACTION: See the SKOL Reference Manual for correct usage.

SOURCE: SKOL