

Message and Codes Manual

Compiler Messages

JOL

Universal Command Language

Version 5.2

Clarke Computer Software.

Jol Universal Command Language "*Messages and Codes*".

Fifth Edition (June, 1999)

This is a major revision of, and obsoletes, the
Fourth Edition of the Jol Command Language
"*Reference Guide*" September 1987

Copyright © Clem Clarke 1973-2010.

All Rights Reserved. The contents of this publication may not be reproduced in any form in any means in part or
in whole without the prior written consent of OSCAR Pty Ltd.

**Clarke Computer Software,
45 Riversdale Road,
Hawthorn, Victoria,
Australia 3022.**

**Telephone (61)-401-054-155
Email clementclarke@ozemail.com.au**

Jol Message Manual - Preface

Preface

This publication is for persons who use the Jol Universal Command Language. The *Jol Message Manual* contains all the messages that may be produced when using Jol. These messages are in alphameric order.

The messages, where appropriate, consist of:

- an expanded explanation of the message text,
- the action the system has undertaken, and
- the response required of the Operator and/or Programmer.

Format of the Manual

The book is divided into separate sections, each commencing with a new Jol message code.

It is recommended that one copy of this manual be placed with the *Jol Reference Manual*, and copies be stored with, or in, each copy of the installation's Operating System Message Manuals for the convenience of Users and Operators.

Summary of Amendments

TSO Support.

- Options to allow the job to execute under TSO or Background, using the same Command Language.

ALLOCATE Command.

- **ALLOCATE** a data set in the Preprocessor or Macro Phase for Input or Output.

Addition of OPEN, READ and WRITE Instructions.

- **OPEN** a file for input or output.
- **READ** a record into a variable.
- **WRITE** a record from a variable.

Automatic Reset of Relative Generation Numbers

- Relative Generation Numbers are automatically reset for Reruns or Restarts.

Testing if a data set exists.

- **TEXTIST** Command allows the testing of the existence of a data set at execution time.

New Installation Procedures

- The basic machine-readable material has been revised, and there are new procedures to install the product. Separate procedures are used for installation in the OS environment (using SMP) and the non-SMP Installation.

Associated Documentation includes:

- **Jol Reference Manual**
- **Jol User Guide**
- **Jol Concepts and Facilities Manual**
- **Jol Answers to Questions**
- **Jol Conversion Utilities**
- **Jol System Programmer Guide**
- **Jol Release and SMP Guide**
- **Jol Program Logic Manual**

Contents

Introduction	7
UJC00-nn Jol Compiler Phase Control Module Messages	10
UJC01-nn Job Statement Messages	12
UJC02-nn Program Declare Messages	13
UJC03-nn Data Set Declare Messages	16
UJC04-nn IF Statement Messages	19
UJC05-nn Disposition Instruction Messages	22
UJC10-nn TYPE/DISPLAY Messages	23
UJC12-nn SIGNAL ERROR Messages	24
UJC13-nn STOP Messages	25
UJC24-nn RUN Messages	27
UJC35-nn OPCNTL Messages	29
UJC50-nn READ Program Statement Messages	30
UJF00-nn Jol Compiler Termination Messages	33
UJO39-nn Jol Validation Messages	35
UJO40-nn Resolve Data Set Usage Messages	36
UJO43-nn Resolve Data Set Usage Messages	38
UJP00-nn Preprocessor Control Module Messages	40
UJP02-nn Declaration Messages	42
UJP04-nn IF Statement Messages	44
UJP12-nn SIGNAL ERROR Messages	47
UJP13-nn STOP Statement Messages	48
UJP15-nn ALLOCATE Data Set Messages	49
UJP17-nn OPEN Instruction Messages	51
UJP18-nn READ/GETFILE Instruction Messages	53
UJP19-nn WRITE/PUTFILE Instruction Messages	55
UJP20-nn CLOSE Instruction Messages	56
UJP22-nn ENQ/DEQ Messages	57
UJP25-nn EDIT Statement Messages	58
UJP26-nn GET Statement Messages	60
UJP27-nn PANEL Statement Messages	61
UJP45-nn REDO Instruction Messages	63
UJP46-nn REDO Instruction Messages	64
UJP50-nn Read Program Statement Messages	66
UJP51-nn Read Program Statement Messages	67
UJP52-nn Read Program Statements from Library Messages	68
UJP53-nn Read Macro Statement Messages	69
UJP54-nn INCLUDE and Macro Statement Messages	71
UJP60-nn and UJP61-nn READ and WRITE Instruction Messages	73
UJP85-nn Assignment Statement Messages	73
UJP93-nn, UJP95-nn Preprocessor Initialization Messages	76
UJS01-nn Catalog Search Messages	79
UJS10-nn Store Variable Data Subroutine Messages	83

UJS15-nn Jol Compiler Workfile Messages.....	83
UJS16-nn Jol Compiler READ From Workfile Messages	84
UJS19-nn Internal Compiler Messages.....	85
UJS20-nn Replace Symbolic Variables Messages.....	86
UJS61-nn Generation Data Set Messages.....	88
UJU01-nn Installation Job Validation Messages	90
UJU02-nn Installation Program Validation Messages.....	91
UJU03-nn Installation Data Set Validation Messages	92
UJV00-nn Main Validation Module Messages.....	94
UJV01-nn Job Definition Validation Messages.....	95
UJV02-nn Program Definition Validation Messages	96
UJV03-nn Data Set Validation Messages.....	97
UJV05-nn CATALOG Instruction Validation Messages	101
UJV07-nn KEEP Instruction Validation Messages	102
UJV08-nn SCRATCH Instruction Validation Messages.....	103

Introduction

JOL Message Format

Jol messages are detailed as fully as possible to permit you, the user, to understand a condition or to take necessary action for successful completion of your work. Information for each message presents appropriately:

- ☐ Explanation: what the message means, why it appears, what caused it, what its variable entries are, and the severity of the condition.
- ☐ System action: what is happening as a result of the condition causing the message, whether the system is waiting for a response.
- ☐ Operator and/or Programmer response: if a response is necessary, who performs it, what the pertinent responses are and their effect on the system or program.

Format of Execution Messages

Messages commencing with **JOLE** are produced at execution time by the Jol monitor.

For example:-

JOLE98-05 LM IEFBR14 SSI 00000000 RETURNED 0 AT 4

indicates that the program **IEFBR14** was executed at statement 23 and that it returned a 0 return-code when it finished.

Format of Compiler Messages

Messages commencing with **UJ** are produced by the compiler at compile time; all messages contain a statement number referring to the statement.

For example:

23 UJC24-01 CANNOT FIND PROGRAM 'IEFBR14'

indicates that you were attempting to **RUN** the program **IEFBR14** at statement 23 in your Jol program, but that you had not previously **DECLARED** or **DEFINED** it earlier in your program.

Part of Jol's message number indicates the name of the Jol module that produced the message. In the first example above, the message number JOLE98 specifies that the message was produced by module UJE98, while the second example of UJC24 specifies that the message was produced by module UJC24.

All Compiler messages are placed on the listing showing your input Jol statements. If you are executing Jol interactively, the messages will also be displayed on your terminal.

Execution messages are printed on your job's system log but if an error occurred, the messages also appear on the Operator's Console.

Jol Compiler – Compiler Phase Messages (UJC)

Component Name	UJC00-UJC99 Jol Compiler Modules.
Program Producing Message	Jol Command Language Compiler
Message Format	<p>For Operator: UJC<i>mm-nn text</i> For Programmer: UJC<i>mm-nn text</i></p> <p><i>where:</i></p> <p><i>mm</i> is the module number issuing the message.</p> <p><i>nn</i> is the message number.</p> <p><i>text</i> is the message text.</p> <p><i>severity</i> is indicated in the message descriptions.</p>
Associated Publications	Jol Reference Manual. Jol General Information Manual.
Problem Determination	If the problem recurs, do the problem determination action(s) to ensure that qualified personnel will have the essential programming information available to diagnose and correct the difficulty.

UJC00-nn Jol Compiler Phase Control Module Messages

UJC00-01 PREPROCESSOR DID NOT GENERATE ANY STATEMENTS

Terminal Error: No code was produced by the preprocessor phase.

This is due to either:-

- a) No input was supplied to Jol.

or

Symbolic Parameter statements.

System Action: The next job in the input stream is compiled as normal.

Programmer Response: Correct the program and rerun.

UJC00-02 DECLARATION FOUND AFTER EXECUTION 'IF'

Terminal Error: A declare statement has been found following an execution **IF** statement. **DECLARE** statements of Data Sets or Programs may not be executed after **IF** statements.

System Action: The statement is deleted.

Programmer Response: Correct the program and rerun.

UJC00-03 ELSE IGNORED

Severe Error: An **ELSE** was found, but it was not preceded by an **IF** statement or an **IF** group of statements.

System Action: The **ELSE** is ignored.

Programmer Response: Correct the program and rerun.

UJC00-04 'instruction' IS NOT YET IMPLEMENTED

Terminal Error: The instruction printed is not yet implemented but is planned for a future release.

This may also be caused by coding a Preprocessor Instruction after an Execute orientated **IF** statement.

System Action: The statement is ignored.

Programmer Response: Correct the problem and rerun the job.

UJC00-05 UNRECOGNIZABLE DECLARATION

Terminal Error: A **DECLARE** statement has been found, but it is not recognizable by Jol. For example:

DCL name **DSN=DATA.SET**

has no **DS** or **DATA SET** keyword coded after the **DCL**.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJC00-06 TOO MANY CONCURRENT DO'S

Terminal Error: The maximum number of active **DO** groups is eight for this implementation of Jol.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJC00-07 TOO MANY END'S

Error: Too many **END** statements were found.

One **END** is required for every **DO** statement, plus one for every **MACRO** statement. One or more extra **END** statements have been coded.

System Action: The **END** is ignored.

Programmer Response: Correct the program and rerun.

UJC00-08 TOO MANY LEVELS OF MACRO NESTING

Terminal Error: The maximum number of macro calls has been exceeded. A macro may call eleven other macros concurrently, i.e. a macro may issue a macro instruction and that macro may issue another macro instruction, but the eleventh macro may not issue another macro instruction, as this then increments the level number to a number greater than is permitted.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJC00-09 LABEL 'start-at-label' NOT FOUND FOR STARTAT INSTRUCTION

Terminal Error: The **STARTAT** instruction specified a label at which execution is to begin, but after the entire Jol program had been read, it was found that the specified label could not be found. For example:

STARTAT STEPONE;

STEP2 : RUN A;

STEP3 : RUN B;

Label **STEPONE** is not in the Jol program, and so a message is produced.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJC00-10 LABEL 'stop-at-label' NOT FOUND FOR STOPAT INSTRUCTION ENTIRE JOB WILL RUN

Severe Error: The label specified in the **STOPAT** instruction was not found in the Jol program.

System Action: The job is placed on the **HOLD** Queue, and if the job is released and allowed to execute, the entire job will run.

Programmer Response: Correct the program and rerun.

UJC00-11 IDENTIFIER 'dsid' EXCEEDS MAXIMUM LENGTH OF 8 BYTES, DCL IGNORED

Terminal Error: The maximum length of a DSID is 8 bytes.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJC01-nn Job Statement Messages

UJC01-01 JOB DETAILS MERGED

Information: Two or more **JOB** statements have been found.

System Action: The information from the second statement is ignored if the corresponding information was found in the first.

Programmer Response: Correct and rerun if necessary.

UJC01-02 SYMBOL 'symbol' IGNORED - UNRECOGNIZABLE

Severe Error: The symbol printed is not recognizable as:

1. A Jol keyword or parameter.
2. Recognizable by the installation user code exit.

For example:

A:JOB TYME 5,1 MINS;

The **TYME** should have been coded as **TIME**.

Programmer Response: Correct the error and rerun unless Jol overcame the problem.

UJC01-03 ERROR IN TIME PARAMETER

Severe Error: The **TIME** coded was not numeric or greater than 1440.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJC01-04 'parameter' NOT YET IMPLEMENTED

Severe Error: The parameter printed is planned for future implementation in Jol, but is currently ignored.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJC01-05 MAX OF 8 SUB-ACCTS ALLOWED:- 'symbol' IGNORED; BUT ')' MAY ALSO BE MISSING

Severe Error: The **ACCT** field in the **JOB** statement is of the type **ACCT(1,2,3)**; the ')' may be missing or there are too many sub-accounting fields. For example:

ACCT(1,2 50 K CLASS A;

will give the above message.

If more than the 8 sub accounts are required, they must be coded as **ACCT=(accounting information)**'.

System Action: The parameter is ignored.

Programmer Response: Correct the **ACCT** and rerun.

UJC02-nn Program Declare Messages

UJC02-01 'symbol' IS NOT A VALID SYMBOL - IGNORED

Terminal Error: The symbol printed is not recognizable as:

1. A Jol keyword or parameter.
2. Recognizable by the installation user code exit.

For example,

```
DCL PROG1 PROG  
LYB = TEST.LIB;
```

The **LYB** should have been coded as **LIB**.

System Action and Programmer Response: Correct the error and rerun unless Jol overcame the problem.

UJC02-02 ERROR CONVERTING 'symbol' TO NUMERIC

Severe Error: An error occurred converting the **TIME** parameter to numeric.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJC02-03 'parameter' IS NOT YET IMPLEMENTED

Severe Error: The parameter printed is planned for a future implementation of Jol but is currently ignored.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJC02-04 INTERNAL ERROR

Compiler Error: A critical error occurred when processing similar program names at different levels.

System Action: The statement is ignored.

Programmer Response: Contact your System Programmer.

UJC02-05 DUPLICATE NAME DDNAME 'ddname' IGNORED

Error: A duplicate DDNAME has been found in the declaration, and ignored. For example.

```
DCL X PROG  
SYSUT1 READS B,  
SYSUT1 READS C;
```

System Action: The statement is ignored. The second 'SYSUT1' ddname is ignored.

Programmer Response: None required.

UJC02-06 CANNOT STORE DDNAME 'ddname':- INSUFFICIENT ROOM IN TABLE

Terminal Error: The declaration has too many filenames for Jol to handle. The limit is approximately 50 filenames per declaration.

System Action: The details for the program declaration are used up to the point in error.

Programmer Response: If more filenames are required than allowed by the current implementation limits, code two or more **DECLARE** statements, each with the same program name. Jol will then merge the

statements into one, and it will be treated as if it had been coded as one declare statement.

**UJC02-07 PROGRAM DECLARE AT STMT
stmt NOT LEVEL 0 OR EXTERNAL AND
CURRENT OVERRIDE IS
EXTERNAL:- EXTERNAL IGNORED**

Severe Error: A program declared in a macro is to be merged with one that was declared as an **EXTERNAL** program. The current declare is not **EXTERNAL**, and so they are presumed to be different programs.

System Action: The statement is treated as a new program declaration.

Programmer Response: Correct the program and rerun.

**UJC02-08 ATTEMPTING TO OVER-RIDE
STATEMENT stmt WHICH IS NOT A
PROGRAM DECLARE**

Severe Error: This problem may be detected where:

1) The **LIKE** parameter is used to copy information from a previously defined item but that item is not a program definition.

or

2) A merge or override was about to be performed, but it was found that the previously defined name was not a program definition.

System Action: The statement is ignored.

Programmer Response: Correct the error and rerun.

**UJC02-09 LIKE PROCESSING
IMPOSSIBLE BECAUSE 'program'
DECLARED AT stmt IS NOT A
PROGRAM DECLARE**

Severe Error: The **LIKE** parameter specified a non-program declaration to be used to copy

information to the current declare. Because the referred to declaration was not a program, no details can be merged from it.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

**UJC02-10 OVER-RIDING (MERGING)
PROGRAM DETAILS FROM STMT stmt**

Information: Two program declarations specifying the same program identifier name have been found.

System Action: Information from the second declaration is merged with information in the first declaration. The duplicate information is ignored.

Programmer Response: None required.

**UJC02-11 'symbol' MUST BE PLACED AFTER
'PROG'**

Severe Error: **EXTERNAL** or **EXT** must be placed after the **DCL** name **PROG**.

System Action: The **EXT** parameter is ignored.

Programmer Response: Correct the **EXT** and rerun.

**UJC02-12 CANNOT FIND LIKE PROGRAM
DEFINITION, 'program'**

Severe Error: The **LIKE** parameter was specified but the program specified had not been previously defined.

System Action: The **LIKE** parameter is ignored.

Programmer Response: Correct the program and rerun.

**UJC02-13 UNITAFF FAILED BECAUSE
DDNAME 'ddname' COULD NOT BE FOUND OR
WAS CODED AFTER THE UNITAFF FILE
NAME**

Severe Error: When the **UNITAFF** or **UNITREF** option is used, the ddname being referred to must already have been used previously in the program definition.

System Action: The **UNITAFF** is ignored.

Programmer Response: Correct the program and rerun.

UJC03-nn Data Set Declare Messages

UJC03-01 SYMBOL 'symbol' IS NOT A VALID KEYWORD :- IGNORED

Severe Error: The symbol printed is not recognizable as:

1. A Jol keyword or parameter.
2. A preferred volume.
3. An installation defined unit or a system unit.
4. An installation user defined parameter.

For example:

**DCL X DATA SET DDN =
EXT.STRG.NAME**

The **DDN** should have been coded as **DSN**.

Programmer Response: Correct the program and rerun unless Jol overcame error.

UJC03-02 UNIT SPECIFIED FOR PREFERRED VOLUME CHANGED AS IT WAS INCORRECT.

Error: A unit has been specified for one of the preferred volumes but the unit specifically declared is not of the same type as specified when Jol was installed.

For example:

If ... **VOL DISK1 UNIT TAPE** ... is coded and if volume DISK1 had been defined as a preferred disk volume, the coding of '**UNIT TAPE**' results in a conflict of unit type.

System Action: The unit specifically declared is ignored.

Programmer Response: Correct the error for subsequent runs.

UJC03-03 ERROR IN GENERATION NUMBER

Severe Error: A relative generation has been specified for a generation data set and following the '+' or the '-' is a non-numeric symbol or the number specified when referring to a specified or relative generation is greater than 9999.

System Action: The data set is ignored.

Programmer Response: Correct the program and rerun.

UJC03-04 CANNOT FIND LIKE DSID 'dsid'

Severe Error: The **LIKE** parameter was specified in a data set definition, but no previous definition of the 'dsid' printed could be found, as it was not previously defined.

System Action: The **LIKE** is ignored.

Programmer Response: Resubmit the job, placing the required DSID before the definition referring to it.

UJC03-05 ERROR IN BLOCKSIZE / RECORDSIZE:- > 32768

Severe Error: The maximum blocksize supported by the Operating System is 32768.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJC03-06 VOLUME 'volume' NAME > 6 CHARACTERS - TRUNCATED

Severe Error: The volume name printed is greater than the allowed length of six characters.

System Action: The Volume Serial Number is truncated to six characters.

Programmer Response: Correct the program and rerun.

UJC03-07 OVERRIDING/MERGING STATEMENT *statement-number*

Information: Two data set declarations specifying the same data set identifier have been found.

System Action: Information from the first declaration overwrites any information in the second declaration.

Programmer Response: None required.

UJC03-08 DSID NAME WAS PREVIOUSLY DECLARED AT '*stmt*' AND WAS NOT A DSID:- NO MERGE CAN OCCUR

Severe Error: Two declared items with the same names have been found, the second being a declaration for a data set, and the first being some other declaration.

System Action: The statement is ignored as a merge cannot be performed.

Programmer Response: Correct the program and rerun.

UJC03-09 '*name*' IS NOT YET IMPLEMENTED

Severe Error: An option has been specified in the data set declaration which is planned as a future enhancement to Jol, but is currently ignored.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJC03-10 BPI number INCORRECT: S/B 800, 1600 OR 6250

Severe Error: The BPI was incorrectly specified. For example, coding 2000 BPI is invalid.

System Action: The parameter is ignored.

Programmer Response: Correct the error and rerun.

UJC03-11 '*keyword=parameter*' IGNORED:- FOR SYSOUT ONLY

Severe Error: A parameter was validly coded, but is valid for a printer or punch data set only. For example:

LIMCT=100

can only be specified for printers or punch card files.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJC03-12 '*keyword=parameter*' IGNORED:- FOR NON-SYSOUT FILES ONLY

Severe Error: A parameter was validly coded, but it can only be used for non-printer data sets. For example:

VOL=TAPE01

cannot be specified for printer or punch files.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJC03-13 ERROR IN DSNAME

Severe Error: There is a mistake in the DSNAME.

System Action: The DSNAME parameter is ignored.

Programmer Response: Correct the program and rerun.

UJC04-nn IF Statement Messages

UJC04-01 INVALID SYMBOL '*symbol*' IGNORED

Severe Error: An invalid symbol was found in the **IF** statement.

System Action: The **IF** statement is ignored.

Programmer Response: Correct the program and rerun.

UJC04-02 TEST FOR '*name*' IMPOSSIBLE :- NOT DECLARED

Severe Error: When attempting to test a return code from a program, it was found that the program had not been declared.

System Action: The **IF** statement is ignored.

Programmer Response: Correct the program and rerun.

UJC04-03 NUMBER '*number*' IS > 4095:-NO PROGRAM CAN ISSUE A RETURN CODE > 4095

Terminal Error: A program can return a numeric value of **0** to **4095** after execution. Therefore it is not possible to test for a number greater than **4095**.

System Action: The **IF** statement is ignored.

Programmer Response: Correct the program and rerun.

UJC04-04 INVALID SPECIAL SYMBOL '*symbol*' IGNORED

Severe Error: A special character was found which is not permissible in an **IF** statement.

System Action: The **IF** statement is ignored.

Programmer Response: Correct the program and rerun.

UJC04-05 IF STATEMENT DELETED

Terminal Error: Due to other errors in the **IF** statement, it has been deleted or ignored.

Programmer Response: Correct the program and rerun.

UJC04-06 TOO MANY '(' IN 'IF'

UJC04-07 TOO MANY ')' IN 'IF'

Terminal Error: The sum of the number of left parentheses did not equal the sum of the number of right parentheses.

System Action: The **IF** statement is ignored.

Programmer Response: Correct the **IF** and rerun.

UJC04-08 STATEMENT ENDS PREMATURELY OR INCORRECTLY

Terminal Error: The **IF** statement is incomplete.

For example,

IF A= will produce the above message.

System Action: The statement is ignored.

Programmer Response: Correct the **IF** and rerun.

**UJC04-09 IF STATEMENT INTERNAL
ERROR AT 'symbol'**

Compiler Error: A literal character string has been found in the execution **IF** statement.

This is an internal problem in Jol.

System Action: The **IF** statement is deleted. This should have been deleted in an earlier phase of Jol.

Programmer Response: Contact your System Programmer.

**UJC04-10 INVALID SYNTAX AT / BEFORE
'symbol'**

Terminal Error: The **IF** statement contains invalid syntax.

For example,

IF A = 10 B = 20

does not contain '!' or an '&' and is invalid.

System Action: The **IF** is ignored.

Programmer Response: Correct the **IF** and rerun.

**UJC04-11 INVALID SYMBOL 'symbol'
AFTER OPERATOR**

Terminal Error: The **IF** statement contains invalid syntax.

For example,

IF A = 'A' will produce the above message.

System Action: The **IF** is ignored.

Programmer Response: Correct the **IF** and rerun.

UJC04-12 AFTER 'C' 'symbol' IS INVALID

Terminal Error: A problem with syntax has been detected.

For example,

IF B(& A)

would produce the message above because there is no expression preceding the 'A'.

System Action: The **IF** is ignored.

Programmer Response: Correct the **IF** and rerun.

UJC04-13 AFTER ')' 'symbol' IS INVALID

Terminal Error: A syntax error has been detected.

For example,

IF (A | B) C

would give rise to the above message, because the **C** is not connected to the expression with an '!' or an '&'.

System Action: The **IF** is ignored.

Programmer Response: Correct the **IF** and rerun.

UJC04-14 IF TOO COMPLEX

Terminal Error: The **IF** statement is too complex.

System Action: The **IF** is ignored.

Programmer Response: Split the **IF** into several smaller statements and rerun.

**UJC04-15 TEST FOR 'stepname' INVALID AS
'stepname' IS NOT A PROGRAM OR MACRO
NAME**

Severe Error: A return code test has been made, but the program being tested has not yet been run.

System Action: The **IF** is ignored.

Programmer Response: Correct the program and rerun.

**UJC04-16 STEP 'stepname' NOT
EXECUTED:- TEST MARKED FALSE**

Error: A test has been made, but the program or step being tested was not run due to being bypassed by a **STARTAT** instruction.

System Action: That portion of the **IF** test will be marked false. If one part of the **IF** is **true**, and the overall condition is **true**, then the next statement will be executed.

Programmer Response: Correct the program if necessary and rerun.

UJC05-nn Disposition Instruction Messages

UJC05-01 CANNOT PERFORM ACTION ON 'name' :- NOT DECLARED

Severe Error: When processing a **KEEP**, **CATALOG**, **UNCATALOG**, **DELETE** or **SCRATCH** instruction it was found that the variable had not been previously declared.

System Action: The requested action for the variable is ignored.

Programmer Response: Correct the program and rerun.

UJC05-02 CANNOT READ 'name'

Compiler Error: An I/O error occurred when the details for the named data set identifier was read from a work data set.

System Action: The requested action for the variable is ignored.

Programmer Response: Rerun the job. If errors continue, contact your System Programmer.

UJC05-03 'name' IS NOT A DATA SET VARIABLE - IGNORED

Severe Error: A **KEEP**, **CATALOG**, **UNCATALOG**, **DELETE** or **SCRATCH** instruction has been issued to a variable which is not a data set variable.

System Action: The requested action for the variable is ignored.

Programmer Response: Correct the program and rerun.

UJC05-04 VOLUME SPECIFIED USED INSTEAD OF DECLARED VOLUME

Warning Message: A **KEEP**, **CATALOG**, **UNCATALOG**, **DELETE** or **SCRATCH** instruction was requested for a data set identifier (DSID) that had a volume specified, but the disposition instruction also had a volume specified.

For example,

DCL IN DS SYS1.LIB VOL 111111;

DELETE IN FROM 222222 2314;

would produce the message.

System Action: The volume specified in the **DELETE** is used.

Programmer Response: Correct the program and rerun if necessary.

UJC05-05 UNIT SPECIFIED USED INSTEAD OF DECLARED UNIT

Warning Message: A **KEEP**, **CATALOG**, **UNCATALOG**, **DELETE** or **SCRATCH** instruction was requested for a data set identifier (DSID) which had a unit specified, but the disposition instruction also had a unit specified.

For example,

DCL IN DS SYS1.LIB UNIT 3330;

DELETE IN FROM 222222 2314;

would produce message above.

System Action: The volume specified in the **DELETE** is used.

Programmer Response: Correct the program and rerun if necessary.

UJC10-nn TYPE/DISPLAY Messages

UJC10-01 INVALID SYNTAX

Error: The message text for the **DISPLAY**, **RETURN** or **TYPE** is invalid because there are too many symbols.

For example,

TYPE JOB FAILED;

Coded correctly this becomes **TYPE 'JOB FAILED'**;. In the example, only the word **JOB** will be typed or displayed.

System Action: The first symbol is accepted.

Programmer Response: Correct the program and rerun.

UJC10-02 MESSAGE TEXT TRUNCATED

Error: The message text associated with the **TYPE** or **DISPLAY** is too long.

The maximum length for a **TYPE** or **DISPLAY** is **100** characters.

System Action: The message is truncated.

Programmer Response: Correct the program and rerun.

UJC10-03 MESSAGE TEXT INVALID 'symbol'

Severe Error: The symbol printed is invalid in a **TYPE** or **DISPLAY**.

For example, **TYPE =;**

Coded correctly this becomes **TYPE '=';**

System Action: The text has been accepted and will be displayed as the message text.

Programmer Response: Correct the instruction and rerun.

UJC10-04 MESSAGE TEXT SHOULD BE IN QUOTES

Error: The message text for a **TYPE** or **DISPLAY** is not a number, a literal character string or a special character.

For example, **TYPE STEP 1;**

Coded correctly this becomes **TYPE 'STEP 1'**.

System Action: The text is accepted as the message and will be typed or displayed when the instruction is executed.

Programmer Response: Correct the program and rerun.

UJC10-05 INVALID SYNTAX

Error: Too many tokens were probably found on the statement. The usual cause is a missing semi-colon.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJC12-nn SIGNAL ERROR Messages

UJC12-01 MESSAGE TEXT IS INVALID, 'symbol'

Error: The symbol printed is invalid as a message text for the **SIGNAL ERROR**.

For example, **SIGNAL 3, =;**

Coded correctly this becomes

SIGNAL 3, '=';

System Action: The text has been accepted and will be displayed as the **SIGNAL** message.

Programmer Response: Correct the program and rerun.

UJC12-02 MESSAGE TEXT SHOULD BE IN QUOTES

Warning Message: The message text for a **SIGNAL ERROR** is not a number, a literal character string or a special character.

For example, **SIGNAL ERROR 3, ERROR;**

Coded correctly this becomes

SIGNAL ERROR 3, 'ERROR';

System Action: The text has been accepted and will be displayed as the **SIGNAL** message.

Programmer Response: Correct error for subsequent runs.

UJC12-03 MESSAGE TEXT TRUNCATED

Error: The message for the **SIGNAL ERROR** is too long.

The maximum length of a **SIGNAL** message is **100** characters.

System Action: The message text is truncated.

Programmer Response: Correct the program and rerun.

UJC12-04 INVALID SEVERITY, RANGE IS 1 TO 4

Error: The severity code for the **SIGNAL ERROR** is less than **0** or greater than **4**, or is a non-numeric severity code.

System Action: The instruction is replaced with a **SIGNAL ERROR 4, 'ERROR SIGNALLED'** instruction.

Programmer Response: Correct the program and rerun.

UJC12-05 NO MESSAGE TEXT PROVIDED

Warning Message: There is no message text for the **SIGNAL ERROR** instruction.

System Action: A message text of **'ERROR SIGNALLED'** is supplied.

UJC13-nn STOP Messages

UJC13-01 MESSAGE TEXT SHOULD BE IN QUOTES

Error: The message text for a **STOP** is not a number, a literal character string or a special character.

For example, **STOP ERROR;**

Coded correctly this becomes

STOP 'ERROR';

System Action: The message text is accepted and will be displayed on the console when the **STOP** is executed.

Programmer Response: Correct the program for the next submission.

UJC13-02 STOP MESSAGE TEXT TRUNCATED

Error: The message text associated with the **STOP** is too long. The current maximum length of a **STOP** message is 100 characters.

System Action: The statement is truncated.

Programmer Response: Correct the program for the next re-submission.

UJC13-03 SYMBOL 'symbol' INVALID

Severe Error: The symbol printed is invalid as a message text.

For example, **STOP =;**

Coded correctly this becomes

STOP '=';

System Action: The statement is accepted.

Programmer Response: Correct the program for the re-submission.

UJC13-04 STOP 'symbol' NOT YET IMPLEMENTED

Terminal Error: This implementation of Jol does not yet support:-

STOP BEFORE labelor
STOP AFTER label

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJC13-05 STOP WHEN CANNOT BE EXECUTED IN A DO GROUP OR AFTER AN IF

Severe Error: The **STOP WHEN** cannot be executed after an **IF** statement.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJC13-06 INVALID STOP WHEN

UJC13-07 INVALID OPERATOR 'symbol' :- STOP IGNORED

Severe Error: A special character was found which is not permissible in a **STOP WHEN** statement.

System Action: The **STOP WHEN** instruction is ignored.

Programmer Response: Correct the program and rerun.

**UJC13-08 NUMBER EXPECTED NOT
'symbol'**

Severe Error: An invalid symbol was found where a number was expected.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

**UJC13-09 MAX NUMBER ALLOWED 4095,
FOUND 'symbol'**

Severe Error: A program can return a numeric value of **0** to **4095** after execution. Therefore it is not possible to test for a number greater than **4095**.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJC13-10 EXPECTED 'I' FOUND 'string'

Severe Error: An invalid symbol was found where an **OR** character was expected.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJC13-11 EXPECTED 'ANY' FOUND 'string'

Severe Error: An invalid symbol was found where **ANY** was expected.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJC24-nn RUN Messages

UJC24-01 PROGRAM 'program' HAS NOT YET BEEN DEFINED

Terminal Error: A **RUN** statement has been found, but the program to be run has not yet been declared.

System Action: The statement is ignored.

Programmer Response: Declare the program and rerun.

UJC24-02 CANNOT RUN 'name' IS NOT A PROGRAM

Terminal Error: A **RUN** statement has been found, but the name referred to in the **RUN** is not a program.

Most likely, the **DECLARE** or **DEFINE** command refers to a data set rather than a program.

For example:

DCL X DS SYS1.MACLIB;

RUN X;

will produce the above message.

System Action: The **RUN** is ignored.

Programmer Response: Correct the **RUN** instruction and rerun.

UJC24-03 CANNOT FIND DATA SET VARIABLE 'name'

Terminal Error: The data set identifier printed has been defined in the program declaration, but it has not yet been declared.

System Action: The data set variable is ignored.

Programmer Response: Correct the program and rerun.

UJC24-04 'name' IS NOT A DATA SET VARIABLE AND CANNOT BE USED BY A PROGRAM

Terminal Error: The name printed is not a data set variable and cannot be read or written as the program declaration specifies.

System Action: The data set variable is ignored. Any commands produced to the Operating System will ignore that file name.

Programmer Response: Correct the program and rerun.

UJC24-05 CANNOT FIND VOLREF 'name'

Terminal Error: A volume reference cannot be resolved as the referenced item has not been declared.

System Action: The data set variable is ignored. Any commands produced to the Operating System will ignore that file name.

Programmer Response: Correct the error and rerun.

UJC24-06 VOLREF PARAMETER 'name' IS NOT A DATA SET

Terminal Error: A **VOLREF** has been specified in a data set declare, but the **DECLARE** to which the **VOLREF** points was not a data set identifier (DSID).

For example:

SYSUT1 READS X VOLREF Y

If Y is not a DSID, the message will be produced.

System Action: The **VOLREF** is ignored.

Programmer Response: Correct the program and rerun.

**UJC24-07 EXCESS INFORMATION
STARTING 'symbol' IGNORED**

Severe Error: Excess information was found on a **RUN** statement.

This may be due to a missing semi-colon.

System Action: The excess information is ignored.

Programmer Response: Correct the program and rerun.

**UJC24-08 LABEL 'label' HAS ALREADY
BEEN DEFINED:- IGNORED**

Severe Error: Two identical labels have been defined; the second label is ignored.

For example:

A:SORT X TO Y USING B;

A:RUN IEFBR14;

Label A is duplicated.

Similarly, the following is not permissible

DCL A DS SYS1.MACLIB;

A: RUN IEBGENER;

System Action: The label is ignored.

Programmer Response: Correct the program and rerun.

**UJC24-09 CANNOT FIND DSID 'name' FOR
PARAMETER**

A **RUN** statement was found requesting that parameter information be provided from a Data Set, but the DSID cannot be found.

For example:

RUN IEMAA DSIDI;

where the data set DSIDI cannot be found.

System Action: The variable is ignored.

Programmer Response: Correct the program and rerun.

UJC35-nn OPCNTL Messages

UJC35-01 'symbol' INVALID, 'AFTER'
ASSUMED

Severe Error: An **OPCNTL** instruction was found,
but the optional position specified was not
BEFORE or **AFTER**.

For example:

OPCNTL /*SETUP 111111'
AFTE;

The 'AFTE' is incorrectly spelt.

System Action: The default for **OPCNTL** applies.

Programmer Response: Correct the program and
rerun.

UJC35-02 'symbol' INVALID

Severe Error: An **OPCNTL** instruction was found, but the
optional keyword after the **BEFORE** or **AFTER** was
not **JOB**CARD.

This may be due to a missing semi-colon after the **BEFORE**
or **AFTER**.

For example:

OPCNTL /*PRIORITY 10' BEFORE JOB;

The '**JOB**' is invalid, and should have been coded as
JOBCARD.

System Action: The symbol is ignored.

Programmer Response: Correct the program and rerun.

UJC50-nn READ Program Statement Messages

UJC50-01 INPUT STATEMENT TOO LONG:- GREATER THAN 2048 BYTES

Warning Message: The maximum length of a Jol program statement is 2048 bytes. This has been exceeded.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJC50-02 TOO MANY TOKENS (SYMBOLS):- GREATER THAN 511 TOKENS

The statement contains more symbols than is permitted.

Each statement is broken into small components called **symbols** or **tokens** so that Jol can process the parts individually.

For example:

IF A=1 THEN STOP;

consists of 6 tokens:-

**IF
A
=
1
THEN
STOP**

The current statement has too many tokens for Jol to process.

System Action: The statement is ignored.

Programmer Response: Check for missing semi-colons. If the statement is correct, it must be broken into smaller statements and the program rerun.

UJC50-03 SYMBOL STARTING 'symbol' TOO LONG, GREATER THAN 253 BYTES

Severe Error: The statement contains a symbol or character string which is longer than 253 characters.

The maximum length of a symbol is 253 characters.

System Action: The statement is ignored.

Programmer Response: This problem is usually caused by a missing quote on character strings. Correct the statement and rerun.

UJC50-04 LITERAL STRING STARTING 'stmt' HAS NO ENDING QUOTE

Severe Error: The statement contains a character string which does not contain a matching quote.

System Action: The compilation ceases - recovery is not possible.

Programmer Response: Place quotes where appropriate in the program and rerun.

UJC50-05 ERROR IN STATEMENT

Severe Error: An error was detected by the Compiler Read Statement Routine. One of the error messages above (UJC50-01 to UJC50-04) will also have been issued, indicating the real nature of the problem.

System Action: The statement is ignored.

Programmer Response: Correct the problem as indicated by the earlier messages and rerun.

**UJC50-06 LABEL 'label' IGNORED FOR
STARTAT/STOPAT**

Error: The label specified in the **STARTAT** or **STOPAT** instruction is ignored because it was found on a **DECLARE** statement. Target labels of the **STARTAT** and **STOPAT** instructions must be labels on executable statements.

System Action: The label is ignored. Other messages may be produced during the compilation.

Programmer Response: Place the label on an executable statement, even if that statement is a **null** statement (a semi-colon) and rerun.

Jol Compiler Termination Messages (UJF)

Component Name	UJF00-UJF99 Jol Compiler Modules.
Program Producing Message	Jol Command Language Compiler
Message Format	<p>For Operator: UJF<i>mm-nn text</i> For Programmer: UJF<i>mm-nn text</i></p> <p><i>where:</i></p> <p><i>mm</i> is the module number issuing the message.</p> <p><i>nn</i> is the message number.</p> <p><i>text</i> is the message text.</p> <p><i>severity</i> is indicated in the message descriptions.</p>
Associated Publications	Jol Reference Manual. Jol General Information Manual.
Problem Determination	If the problem recurs, do the problem determination action(s) to ensure that qualified personnel will have the essential programming information available to diagnose and correct the difficulty.

UJF00-nn Jol Compiler Termination Messages

UJF00-01`jobname` WILL NOT EXECUTE:- JOL ERROR

Terminal Error: Severe errors were detected while compiling the job.

System Action: No JCL or control instructions were produced so that the Operating System could execute the job.

Programmer Response: Correct the errors as indicated by earlier messages and rerun the job.

UJF00-02`'jobname'` HAS JOL ERRORS:- PLACED ON HOLD QUEUE

The Jol Compiler detected errors while compiling the job.

System Action: Jol generated the JCL or control statements required to execute the job, but placed it in HOLD Status.

Programmer Response: Depending on the severity of the errors, either have the Operator release the job, or cancel it from the System Job Queue.

Jol Compiler Data Set Resolution Messages (UJO)

Component Name	UJO00-UJO99 Jol Compiler Modules.
Program Producing Message	Jol Command Language Compiler
Message Format	<p>For Operator: UJOmm-nn $text$ For Programmer: UJOmm-nn $text$</p> <p><i>where:</i></p> <p style="padding-left: 40px;">mm is the module number issuing the message.</p> <p style="padding-left: 40px;">nn is the message number.</p> <p style="padding-left: 40px;">$text$ is the message text.</p> <p style="padding-left: 40px;">$severity$ is indicated in the message descriptions.</p>
Associated Publications	Jol Reference Manual. Jol General Information Manual.
Problem Determination	If the problem recurs, do the problem determination action(s) to ensure that qualified personnel will have the essential programming information available to diagnose and correct the difficulty.

UJO39-nn Jol Validation Messages

**UJO39-01 LOWEST RELATIVE
GENERATION NUMBER FOUND
FOR DSNAME '*dsname*' WAS
generation-number: ALL
GENERATIONS RESET SO THAT
LOWEST OUTPUT DATA SET IS (+1)**

Warning Message: While examining the total Job, Jol found that the first reference to the named OUTPUT data set was other than (+1).

This is most likely caused by using the STARTAT instruction to skip a RUN statement of the program that produced the (+1) generation.

System Action: Jol has reset all the relative generations so that the lowest relative generation is (+1).

Programmer Response: None required.

**UJO39-02 LOWEST RELATIVE GENERATION
NUMBER FOUND FOR DSNAME '*dsname*'
WAS *generation-number*: ALL
GENERATIONS RESET SO THAT THE
LOWEST INPUT DATA SET IS (0).**

Warning Message: This message is produced when the lowest INPUT generation number referring to a Data Set is not (0) or (**nn**).

This is most likely caused by using the **STARTAT** instruction to skip a **RUN** statement of the program that produced the (+1), etc., generation of the data set(s) being read.

System Action: Jol has reset all the relative generations so that the lowest generation is (0).

Programmer Response: None required.

UJO40-nn Resolve Data Set Usage Messages

**UJO40-01 DSID '*dsid*' DSNAME '*dsname*'
DECLARED AT *stmt* IS A NEW
GENERATION DATA SET BUT
THERE IS NO CATALOG
INSTRUCTION FOR IT**

Warning Message: A DSID specified a new GDG entry but did not issue a **KEEP** or **CATLG** instruction.

System Action: The data set name will be deleted at the end of the job.

Programmer Response: Check the program logic and if necessary alter the program and rerun.

**UJO40-02 DSID '*dsid*' DSNAME '*dsname*'
DECLARED AT *stmt* IS BEING READ
BUT HAS NOT YET BEEN WRITTEN.**

Terminal Error: An attempt has been made to read a new generation data set which has not yet been created.

Before any attempt is made to read a + GDG, it must have been output earlier in the job.

System Action: The job is marked as having errors.

Programmer Response: Check your program logic, correct the error and rerun.

UJO43-nn Resolve Data Set Usage Messages

UJO43-01 NO DSNAME:- DCL STMT = *stmt*

Compiler Error: A compiler error has occurred.

System Action: The job is marked as having errors.

Programmer Response: Notify your System Programmer.

System Action: The job is marked as having errors.

Programmer Response: To circumvent the problem, code the unit on the data set declare statement. Contact your System Programmer.

UJO43-02 DSID '*dsid*' DSNAME '*dsname*' DECLARED AT *stmt* HAS UNKNOWN INTERNAL UNIT TYPE.

Error: This is an installation problem. The problem occurs because the Jol Volume Table (generated when Jol was installed) shows an incorrect entry in the table.

UJO43-03 DSID '*dsid*' DSNAME '*dsname*' DECLARED AT *stmt* HAS UNKNOWN UNIT TYPE OF '*unit-name*'

Error: Jol did not recognize the UNIT name coded on a data set Declare.

System Action: The UNIT specified is copied to the generated control instructions.

Programmer Response: Ask your System Programmer to respecify to Jol all the unit names your system uses.

Jol Compiler Preprocessor Messages (UJP)

Component Name	UJP00-UJP99 Jol Compiler Modules.
Program Producing Message	Jol Command Language Compiler
Message Format	<p>For Operator: UJPmm-nn <i>text</i> For Programmer: UJPmm-nn <i>text</i></p> <p><i>where:</i></p> <p>mm is the module number issuing the message.</p> <p>nn is the message number.</p> <p>$text$ is the message text.</p> <p><i>severity</i> is indicated in the message descriptions.</p>
Associated Publications	Jol Reference Manual. Jol General Information Manual.
Problem Determination	If the problem recurs, do the problem determination action(s) to ensure that qualified personnel will have the essential programming information available to diagnose and correct the difficulty.

UJP00-nn Preprocessor Control Module Messages

UJP00-01 NO STATEMENTS FOUND TO EXECUTE

Terminal Error: While attempting to read the first statement of the job, end of file was detected or a new job was found in the input stream (* JOL; card).

System Action: The current job is ignored, and the next job in the input stream is compiled as normal.

Programmer Response: Correct the program and rerun.

UJP00-02 INVALID SYSIN FOUND

Severe Error: A card image file was found in the input stream which was not preceded by a **DCL** name * statement.

System Action: The card image data is ignored.

Programmer Response: Correct the program and rerun.

UJP00-03 ELSE IGNORED, AND NEXT INSTRUCTION(S)

Severe Error: An **ELSE** was found, but it was not preceded by an **IF** statement or an **IF** group of statements.

System Action: The **ELSE** is ignored.

Programmer Response: Correct the program and rerun.

UJP00-04 TOO MANY DO'S

Terminal Error: The maximum number of active **DO** groups is eight for this implementation of Jol.

System Action: The statement is ignored. Processing

continues as though the **DO** had not been coded. The message '**TOO MANY ENDS**' may also appear later.

Programmer Response: Correct the program and rerun.

UJP00-05 SYMBOLIC ASSIGNMENT STATEMENT FOUND IN EXECUTION 'IF' LOGIC

Severe Error: An assignment statement was found after an Execution oriented **IF**. An execution oriented **IF** is one that tests a program or command return code. After an execution form of the **IF** statement, an assignment to a symbolic variable is not allowed.

For example:

IF PROGRAM =0 THEN %A=1;

is such a mistake.

Programmer Response: Correct the program and rerun.

UJP00-06 TOO MANY ENDS

Severe Error: Too many **END** statements were found.

One **END** is required for every **DO** statement, plus one for every **MACRO** statement.

System Action: The **END** is ignored.

Programmer Response: Correct the program and rerun.

UJP00-07 'instruction' IS AN INVALID INSTRUCTION

Terminal Error: The instruction printed is not known to Jol.

Valid instructions are:-

- 1) Jol **internal** instructions
- 2) **MACRO** Commands stored in the Macro Library.
- 3) **INVOKE** routines that are stored in the Load Module data set.

System Action: The instruction is ignored.

Programmer Response: Correct the program and rerun.

UJP00-08 EXCESS INFORMATION STARTING 'symbol' IGNORED

Severe Error: Too much information was found on a statement, and was ignored. The most likely reason for this message is a missing semi-colon.

For example:

DO

CATLG 'MY.DATA.SET';

is such an error as there is no semi-colon after the '**DO**'.

System Action: The statement is ignored.

Programmer Response: Resubmit the Job with the missing semi-colon inserted.

UJP00-09 INSUFFICIENT END STATEMENTS FOUND

Terminal Error: After completion of the Preprocessor Phase, it was found that there were not as many **END** statements as there were **DO** and/or **MACRO** statements.

System Action: As with all **SEVERE** or **TERMINAL** errors, the job is marked in error, and will not be allowed to execute unless the **LET** option is specified.

Programmer Response: Check the logic of your Jol program carefully, and insert the missing **END** statement(s).

UJP02-nn Declaration Messages

UJP02-01 INVALID SYMBOL '*symbol*' IGNORED

Terminal Error: The symbol printed is invalid in the **DECLARE** statement.

System Action: The symbol is ignored.

Programmer Response: Correct the program and rerun.

Coded correctly this becomes:-

DCL %SYM INIT '=';

System Action: The symbol is accepted as the initial value.

Programmer Response: Correct the program for the next submission.

UJP02-02 SYMBOL '*symbol*' SHOULD BE IN QUOTES

Warning Message: The symbol printed is not a number or a literal character string. The symbol is accepted as the initial value.

For example:

DCL %SYM INIT A;

will cause the above error message.

Coded correctly this becomes:-

DCL %SYM INIT 'A';

System Action: The value coded is accepted as though quotes had been coded correctly.

Programmer Response: Correct the program for the next submission.

UJP02-04 SYMBOLIC NAME '*name*' HAS BEEN DECLARED AS EXTERNAL, AND ALSO IN A MACRO, THEN EXTERNAL AGAIN

Terminal Error: An **EXTERNAL** name has been used incorrectly.

Consider the following:

1. **DCL %A INIT 'abc';**
2. **DEFJOB : MACRO;**
3. **DCL %A INIT 'xyz';**
4. **MACROCAL:MACRO;**
5. **DCL %A INIT '123' EXT;**

where **MACROCAL** is a macro and has defined **%A** as **EXTERNAL**. Thus the situation is:

%A level **0** (**EXT** is the default in main-line)

%A level **1** in the **DEFJOB** macro

%A level **0** in the second macro invocation of **MACROCAL** at statement 4, because of the **EXTERNAL** parameter.

Jol does not allow multiple **EXTERNAL** statements if there are intervening **DECLARE** statements at different levels.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

UJP02-03 SYMBOL '*symbol*' ACCEPTED

Warning Message: The symbol printed is a special symbol coded where an initial value is expected.

For example:

DCL %SYM INIT = ;

will cause the above problem.

**UJP02-05 SYMBOLIC PARAMETER TABLE
IS FULL**

Terminal Error: When attempting to enter a new symbolic variable name in the table, it was found that there was insufficient space in the table to enter the name.

System Action: The statement is ignored. Any reference to that Symbolic Name will produce the 'CANNOT FIND SYMBOLIC' message.

Programmer Response: Reduce the number of names or ask your System Programmer to regenerate Jol with a larger Symbolic table.

**UJP02-06 SYMBOLIC PARAMETER TABLE
IS FULL**

Terminal Error: There is insufficient space in the symbolic variable table to enter the **value** of the symbolic parameter.

System Action: The statement is ignored.

Programmer Response: Reduce the size of your symbolic variables or ask your System Programmer to regenerate Jol with a larger table.

**UJP02-07 DEFINITION OF 'name' IGNORED:
PREVIOUSLY DEFINED**

Warning Message: The symbolic variable named has been previously declared either explicitly or implicitly (with an assignment statement).

System Action: The statement is ignored.

Programmer Response: This message is often produced when **OVER-RIDES** are used, and is usually ignored.

**UJP02-08 INVALID SYMBOL 'symbol'
IGNORED FOR CARD FILE**

Warning Message: When processing a card file declaration, an invalid symbol was detected. For example:

DCL X * FBA 80;

FBA is not allowable in a card file declaration, and an error occurs.

System Action: The parameter is ignored. The default for card image data set is applied.

Programmer Response: Correct the program and rerun.

**UJP02-09 THE DECLARED DSID * AT stmt
HAS A CARD THAT, WITH SYMBOLICS
REPLACED, IS LONGER THAN xx BYTES AND
THEREFORE WILL BE TRUNCATED**

Severe Error: After replacing Symbolic Parameters on a card image, the length of the data was larger than the record length.

System Action: The card image is truncated to the specified record length.

Programmer Response: Either:-

1. Change the length of the record.
- or
2. Reduce the size of the card image.

UJP04-nn IF Statement Messages

UJP04-01 TOO MANY '('

UJP04-02 TOO MANY ')'

Terminal Error: The sum of the number of left parentheses did not equal the sum of the number of right parentheses.

This is a programming error.

System Action: The **IF** statement and the next statement(s) is ignored.

Programmer Response: Correct the **IF** and rerun.

UJP04-03 IF TOO LONG WHILE PUTTING ' ' ROUND SYMBOLICS

Terminal Error: The **IF** statement is too complex for Jol to process. After replacement of symbolic parameters, the statement exceeded 2048 characters.

System Action: The **IF** is ignored.

Programmer Response: Split the **IF** statement into several statements and rerun.

UJP04-04 ERROR AFTER REPLACING SYMBOLICS

Terminal Error: After replacing the symbolic parameters in an **IF** statement, the statement exceeded the maximum permissible statement length.

System Action: The **IF** statement is ignored.

Programmer Response: Break the **IF** into several shorter statements.

UJP04-05 ERROR REPLACING SYMBOLIC PARAMETERS

Terminal Error: An error occurred replacing the symbolic parameters in the **IF** statement.

The actual message would have been identified by one of the UJS20 messages.

System Action: The **IF** statement is deleted.

Programmer Response: Correct the program and rerun.

UJP04-06 SYMBOL 'symbol' INVALID AFTER AN OPERATOR

Severe Error: The symbol printed is invalid after an arithmetic operator.

For example:

IF A = (

had been coded, the symbol printed would be the '('

System Action: The **IF** is ignored.

Programmer Response: Correct the **IF** and rerun.

UJP04-07 SYMBOL 'symbol' IGNORED

Severe Error: The symbol printed is invalid in the **IF** Statement.

For example:

IF A = .

had been coded, the symbol printed would be '.'

System Action: The statement is deleted.

Programmer Response: Correct the program and rerun.

**UJP04-08 NUMBER *number* NOT
FOLLOWED BY AN OPERATOR BUT '*symbol*'**

Terminal Error: A number has been found in the **IF** statement which is not followed by an operator.

For example:

IF 10

would give rise to the above message.

System Action: The statement is ignored.

Programmer Response: Correct the **IF** and rerun.

**UJP04-09 SYMBOL '*symbol*' INVALID
AFTER AN OPERATOR**

Terminal Error: An execution time form of the **IF** is being compared with a symbol which is not a number or another program return code.

For example:

IF A ='A'

would give rise to the above message.

System Action: The statement is ignored.

Programmer Response: Correct the program and rerun.

**UJP04-10 SYMBOL '*symbol*' INVALID
AFTER AN OPERATOR**

Terminal Error: A symbolic variable is being compared with an invalid symbol.

For example:

IF '%SYM' = (

is invalid.

System Action: The statement is deleted.

Programmer Response: Correct the program and rerun.

**UJP04-11 '*symbol*' IS AN INVALID
OPERATOR**

Terminal Error: This is a compiler problem.

System Action: The statement is ignored.

Programmer Response: Contact your System Programmer.

**UJP04-12 IF TOO COMPLEX TO
EVALUATE**

Terminal Error: When attempting to evaluate whether the **IF** was true, false or had to be reconstructed for the compiler phase (i.e. an execution **IF**), the statement had too many comparisons to be evaluated.

System Action: The statement is ignored.

Programmer Response: Split the **IF** into several statements and rerun.

UJP04-13 #OP NEGATIVE

UJP04-14 #OP TOO BIG

Compiler Error: When attempting to evaluate the **IF** statement, an internal error occurred.

System Action: The statement is ignored.

Programmer Response: Contact your System Programmer.

UJP04-15 > 150 TOKENS

UJP04-16 > 3 TOKENS TO RECREATE

**UJP04-17 RECREATED STRING TOO
LONG**

Compiler Error: When attempting to recreate the **IF** for the main compiler phase, an internal error occurred.

System Action: The statement is ignored.

Programmer Response: Contact your System Programmer.

UJP04-18 SYMBOL 'symbol' SHOULD BE IN QUOTES

Warning Message: A statement of the type **IF '%SYM' = A** was coded.

System Action: In the above example the **A** is assumed to be 'A' and the comparison carried out as usual.

Programmer Response: Correct the program and rerun.

UJP04-19 INVALID PARSE 'symbol'

Compiler Error: An internal error occurred when recreating the **IF** statement for the main compiler phase.

This error is caused when a statement of the form **IF A=10 & 2** is coded.

System Action: The statement is ignored.

Programmer Response: Before contacting your System Programmer, check the **IF** for such an error.

UJP04-20 AFTER 'symbol1' 'symbol2' IS INVALID

Severe Error: The **IF** statement has been incorrectly coded.

For example:

IF %A %B (no operator or connector)

IF %A & & %B (two connectors)

System Action: The **IF** is ignored.

Programmer Response: Correct the **IF** and rerun.

UJP12-*nn* SIGNAL ERROR Messages

UJP12-01 MINOR ERROR '*error message*' SIGNALLED

Warning Message: A **SIGNAL ERROR 1**, '**error message**' instruction was executed by the preprocessor.

System Action: The message was printed, as requested.

Programmer Response: None required.

UJP12-02 SEVERE ERROR '*error message*' SIGNALLED

Error: A **SIGNAL ERROR 2**, '**error message**' instruction was executed by the preprocessor.

System Action: The message was printed, as requested. The job is placed on the **HOLD** queue.

Programmer Response: Determine the reason for the message, correct and rerun.

UJP12-03 MAJOR ERROR '*error message*' SIGNALLED

Severe Error: A **SIGNAL ERROR 3**, '**error message**' instruction was executed by the preprocessor.

System Action: The message was printed, as requested. The job is placed on the **HOLD** queue.

Programmer Response: Determine the reason for the message, correct and rerun.

UJP12-04 TERMINAL ERROR '*error message*' SIGNALLED

Terminal Error: A **SIGNAL ERROR 4**, '**error message**' instruction was executed by the preprocessor.

System Action: The message was printed, as requested. The job will not execute unless the **LET** compiler option was specified

Programmer Response: Determine the reason for the message, correct and rerun.

UJP12-05 INCORRECT SEVERITY NUMBER '*number*'

Error: The severity code for the **SIGNAL ERROR** instruction must be **1, 2, 3, or 4**.

System Action: The severity is assumed to be 4 and the job will not execute unless **LET** was specified.

Programmer Response: Determine the reason for the message, correct and rerun.

UJP12-06 INVALID SYMBOL '*symbol*' IGNORED

Severe Error: A symbol was found after the message in the **SIGNAL** statement. The most likely cause is a missing semi-colon at the end of the statement.

System Action: The symbol printed is ignored.

Programmer Response: Correct the program and rerun.

UJP13-nn STOP Statement Messages

UJP13-01 STOP:- 'error message' EXECUTED

Terminal Error: The preprocessor executed a **STOP** instruction.

System Action: The message was printed, as requested. Your job will not execute unless **LET** was specified.

Programmer Response: Determine the reason for the message, correct and rerun.

UJP13-02 EXCESS INFORMATION IGNORED

Severe Error: A symbol was found after the message in the **STOP** statement. The most likely cause is a missing semi-colon at the end of the statement.

System Action: The symbol and possibly the next statement is ignored.

Programmer Response: Correct the program and rerun.

UJP15-*nn* ALLOCATE Data Set Messages

UJP15-01 SYMBOL '*symbol*' IS NOT A VALID KEYWORD :- IGNORED

Severe Error: The symbol printed is not recognizable as:

1. A Jol keyword or parameter.
2. A preferred volume.
3. An installation defined unit or a system unit.
4. An installation user defined parameter.

For example:

ALLOCATE F(X) DDN = EXT.STRG.NAME

The **DDN** should have been coded as **DSN**.

System Action: The symbol is ignored.

Programmer Response: Correct the program and rerun unless Jol overcame error.

UJP15-02 UNIT SPECIFIED FOR PREFERRED VOLUME CHANGED AS IT WAS INCORRECT.

Error: A unit has been specified for one of the preferred volumes but the unit specifically declared is not of the same type as specified when Jol was installed.

For example:

If ... **VOL DISK1 UNIT TAPE** ... is coded and if volume DISK1 had been defined as a preferred disk volume, the coding of '**UNIT TAPE**' results in a conflict of unit type.

System Action: The unit specifically declared is ignored.

Programmer Response: Correct the error for subsequent runs.

UJP15-03 ERROR IN GENERATION NUMBER

Severe Error: A relative generation has been specified for a generation data set and following the '+' or the '-' is a non-numeric symbol or the number specified when referring to a specified or relative generation is greater than 9999.

System Action: The data set is ignored.

Programmer Response: Correct the program and rerun.

UJP15-05 ERROR IN BLOCKSIZE / RECORDSIZE:- > 32768

Severe Error: The maximum blocksize supported by the Operating System is 32768.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJP15-06 VOLUME '*volume*' NAME > 6 CHARACTERS - TRUNCATED

Severe Error: The volume name printed is greater than the allowed length of six characters.

System Action: The Volume Serial Number is truncated to six characters.

Programmer Response: Correct the program and rerun.

UJP15-09 '*name*' IS NOT YET IMPLEMENTED

Severe Error: An option has been specified in the data set declaration which is planned as a future enhancement to Jol, but is currently ignored.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJP15-10 BPI number INCORRECT: S/B 800, 1600 OR 6250

Severe Error: The BPI was incorrectly specified. For example, coding 2000 BPI is invalid.

System Action: The parameter is ignored.

Programmer Response: Correct the error and rerun.

UJP15-11 'keyword=parameter' IGNORED:- FOR SYSOUT ONLY

Severe Error: A parameter was validly coded, but is valid for a printer or punch data set only. For example:

LIMCT=100

can only be specified for printers or punch card files.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJP15-12 'keyword=parameter' IGNORED:- FOR NON-SYSOUT FILES ONLY

Severe Error: A parameter was validly coded, but it can only be used for non-_printer data sets. For example:

VOL=TAPE01

cannot be specified for printer or punch files.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJP15-13 ERROR IN DSNAME

Severe Error: There is a mistake in the DSNAME.

System Action: The DSNAME parameter is ignored.

Programmer Response: Correct the program and rerun.

UJP17-nn OPEN Instruction Messages

UJP17-01 NO FILE SPECIFIED ON OPEN STATEMENT

Severe Error: The statement was coded incorrectly. For example:

OPEN; will produce the message above, as no filename was specified.

System Action: The OPEN is ignored.

Programmer Response: Correct the program and rerun.

UJP17-02 INVALID OPEN OPTION 'option'

Severe Error: The statement was coded incorrectly. For example:

OPEN SYSUT1 FOR INPUT ; will produce the message above, because the word **FOR** is not a valid option on the **OPEN** statement.

System Action: The invalid option is ignored.

Programmer Response: Correct the program and rerun.

UJP17-03 TOO MANY SYMBOLS ON OPEN

Severe Error: The statement was coded incorrectly. The most likely cause is a missing semi-colon at the end of the statement. For example:

**OPEN SYSUT1 INPUT
DCL %NAME INIT '';**

will produce the message above, because the semi-colon is missing.

System Action: The instruction is ignored.

Programmer Response: Correct the program and rerun.

UJP17-04 FILE 'filename' ALREADY OPEN

Severe Error: The file name listed cannot be opened as it is already open.

System Action: The instruction is ignored.

Programmer Response: This is probably a program logic error. Correct the program and rerun.

UJP17-05 INTERNAL ERROR: DSNAME ALL BLANK OR LENGTH < 0

Severe Error: An internal error occurred when Jol was reading the JFCB to determine the name of the data set to ENQUEUE for a data set opened for output.

System Action: The instruction is ignored.

Programmer Response: Contact your System Programmer.

UJP17-06 RECORD LENGTH OF FILE 'filename' TOO LARGE:- > 3000 CHARACTERS'

Severe Error: The record length of a file opened for input is too large.

System Action: The file is left open; any subsequent READ or WRITES will produce another message.

Programmer Response: Use a file with a smaller record length.

UJP17-08 OPEN FAILED FOR 'filename'

Severe Error: The **OPEN** statement failed. The most likely cause is the file was not previously allocated with an **ALLOCATE** statement.

System Action: The instruction is ignored.

Programmer Response: Check that the file is correctly allocated, and if not, allocate it. An operating system

message should also have been produced to help you find the error. If no operating system message was produced, check your TSO profile, and turn the WTP option, and rerun.

UJP18-nn READ/GETFILE Instruction Messages

UJP18-01 TOO MANY SYMBOLS ON GETFILE STATEMENT

Severe Error: The statement was coded incorrectly. The most likely cause is a missing semi-colon at the end of the statement. For example:

```
GETFILE SYSUT1  
DCL %NAME INIT '';
```

will produce the message above, because the semi-colon is missing.

System Action: The instruction is ignored.

Programmer Response: Correct the program and rerun.

UJP18-02 FILE 'filename' NOT OPEN

Severe Error: When attempting to read a file with the **READ** or **GETFILE** instruction, Jol found that the file had not been opened.

System Action: The instruction is ignored.

Programmer Response: Use the **OPEN** instruction to open the file for input, and rerun the job.

UJP18-03 FILE 'filename' NOT OPENED FOR INPUT

Severe Error: When attempting to read a file with the **READ** or **GETFILE** instruction, Jol found that the file had been opened, but not for input.

System Action: The instruction is ignored.

Programmer Response: Use the **OPEN** instruction to open the file for input, and rerun the job. The error is probably caused by opening the file for output, instead of input.

UJP18-04 FILE 'filename' HAD OPEN ERROR:- IT CANNOT BE USED

Severe Error: When attempting to read a file with the **READ** or **GETFILE** instruction, Jol found that the file had not been opened correctly.

System Action: The instruction is ignored.

Programmer Response: Check for other messages that indicate the reason for the OPEN failure, correct and rerun.

UJP18-05 FILE 'filename' IS AT END OF FILE

Severe Error: When attempting to read a file with the **READ** or **GETFILE** instruction, Jol found that there were no more records to read in the file.

System Action: The instruction is ignored.

Programmer Response: You can use the **IF EOF(filename)** function to check if the file is finished before using any read instructions. Correct your program logic, and rerun the job.

UJP18-06 FILE 'filename' RECORD IS TOO LONG

Severe Error: When attempting to read a file with the **READ** or **GETFILE** instruction, Jol found that the record to be read was greater than 253 characters in length.

System Action: The record is read but truncated to 253 bytes.

Programmer Response: Jol cannot read records longer than the maximum length of a symbolic variable. Check that the file you are using is correct, otherwise you must rearrange the data in the file so that no record is greater

than 253 characters.

UJP19-nn WRITE/PUTFILE Instruction Messages

UJP19-01 TOO MANY SYMBOLS ON PUTFILE STATEMENT

Severe Error: The statement was coded incorrectly. The most likely cause is a missing semi-colon at the end of the statement. For example:

```
PUTFILE SYSUT1  
DCL %NAME INIT '';
```

will produce the message above, because the semi-colon is missing.

System Action: The instruction is ignored.

Programmer Response: Correct the program and rerun.

UJP19-02 FILE 'filename' NOT OPEN

Severe Error: When attempting to write data to a file with the **WRITE** or **PUTFILE** instruction, Jol found that the file had not been opened.

System Action: The instruction is ignored.

Programmer Response: Use the **OPEN** instruction to open the file for output, and rerun the job.

UJP19-03 FILE 'filename' NOT OPENED FOR OUTPUT

Severe Error: When attempting to write data to a file with the **WRITE** or **PUTFILE** instruction, Jol found that

the file had not been opened for output.

System Action: The instruction is ignored.

Programmer Response: Use the **OPEN** instruction to open the file for output, and rerun the job. This is probably a program logic error.

UJP19-04 RECORD 'record-details' TRUNCATED ON OUTPUT TO FILE 'filename'

Severe Error: When attempting to write data to a file with the **WRITE** or **PUTFILE** instruction, Jol found that the maximum record allowed to be written to the file was length than the current length of the symbolic variable specified.

System Action: Truncated data is written to the data set.

Programmer Response: Either write smaller records to the file, or increase the maximum record length allowed to be written to the file.

UJP19-05 SYMBOLIC VARIABLE 'symbolic' NOT FOUND FOR OUTPUT

Severe Error: When attempting to write data to a file with the **WRITE** or **PUTFILE** instruction, Jol could not find the variable to be written.

System Action: The instruction is ignored.

Programmer Response: Check your program logic, correct the error and rerun.

UJP20-nn CLOSE Instruction Messages

UJP20-01 NO FILE SPECIFIED ON CLOSE STATEMENT

Severe Error: No filename was coded on a **CLOSE** statement.

System Action: The instruction is ignored.

Programmer Response: Correct the error, and rerun.

will produce the message above, because the semi-colon is missing.

System Action: The instruction is ignored.

Programmer Response: Correct the program and rerun.

UJP20-02 TOO MANY SYMBOLS ON CLOSE

Severe Error: The statement was coded incorrectly. The most likely cause is a missing semi-colon at the end of the statement. For example:

```
CLOSE SYSUT1  
DCL %NAME INIT ";
```

UJP20-03 CANNOT CLOSE FILE 'filename'

Severe Error: When attempting to close a file, an error was detected. An operating system message will usually accompany this message and give details of the problem.

System Action: The instruction is ignored.

Programmer Response: Correct the problem, and rerun. Note that Jol will implicitly close files if they are not explicitly closed with the **CLOSE** statement, and this message could appear at that time.

UJP22-nn ENQ/DEQ Messages

UJP22-01 ENQUEUE/DEQUEUE:- INVALID PARAMETERS, MUST BE *major-name/'minor-name'*

Severe Error: The statement was coded incorrectly. For example:

ENQ SPFDSN/SYS1.PROCLIB will produce the message above, as the dsname must be coded in quotes.

System Action: The instruction is ignored.

Programmer Response: Correct the program and rerun unless Jol overcame error.

UJP22-02 WAIT OR NOWAIT EXPECTED, NOT '*symbol*'

Severe Error: The statement was coded incorrectly. For example:

ENQ SPFDSN/'SYS1.PROCLIB' NOWATE will produce the message above, as the **NOWATE** should have been coded as **WAIT**.

Programmer Response: Correct the program and rerun unless Jol overcame error.

UJP22-03 WAITING FOR '*qname/minor-name*' REPLY 'W' FOR WAIT, 'N' FOR NOWAIT OR PRESS ENTER TO RETRY

Information: An **ENQ** instruction without the **WAIT** option was executed, but the resource specified is in use by another job or user.

System Action: No further processing occurs until a response has been made.

Programmer Response: Take one of the actions specified in the message. If you specify **WAIT** you will be unable to access your terminal until the resources have become available.

Note: This message may also appear when a data set is opened for output because Jol automatically enques output data sets with the name **SPFDSN/dsname** to prevent accidental destruction of data sets or data.

UJP25-nn EDIT Statement Messages

UJP25-01 SYMBOLIC NAME '*name*' TOO LONG

Terminal Error: The name containing the information to be edited is not a valid symbolic name, as it is greater than 8 characters in length.

For example:

EDIT VERYLONGNAME ...

would give the above error.

System Action: The **EDIT** is terminated.

Programmer Response: Correct the program and rerun.

UJP25-02 INPUT STRING NAME '*name*' CANNOT BE FOUND OR NO NAME GIVEN AS INPUT

Terminal Error: The **EDIT** is invalid because it either had no name specified, or the symbolic name had not been previously defined.

For example:

EDIT A ...

where **%A** has not yet been defined would produce this message.

System Action: The **EDIT** is terminated.

Programmer Response: Correct the program and rerun.

UJP25-03 NO NAME LIST IN EDIT

Terminal Error: No names have been specified into which to edit the information.

For example:

EDIT A ;

will give the above error.

System Action: The **EDIT** is terminated.

Programmer Response: Correct the program and rerun.

UJP25-04 INVALID SYMBOL '*symbol*' IN NAME LIST

Terminal Error: A list may be:-

name1, name2

or

(name1, name2)

but if (*name1* + *name2*) were coded, the '+' would be invalid.

System Action: The **EDIT** is terminated; all previous names in the name list are set to " (null)

Programmer Response: Correct the **EDIT** and rerun.

UJP25-05 FORMAT ITEM '*symbol*' INVALID

Terminal Error: An invalid format item was coded. The only valid format items are **A** or **X**.

System Action: The **EDIT** is terminated; all the names in the name list are set to null strings (")

Programmer Response: Correct the program and rerun.

UJP25-06 FORMAT LIST INVALID AT '*symbol*'

Terminal Error: The Format list is invalid.

In this example:-

EDIT X Y (A1 X(1));

there is no number in parentheses after the **A** format item.

System Action: The **EDIT** is terminated; some of the edit may have been performed, depending on where the error was detected.

Programmer Response: Correct the program and rerun.

**UJP25-07 EXPECTED A NUMBER IN
FORMAT:- 'symbol' INVALID**

Terminal Error: For example

EDIT INPUT OUTPUT X(A);

The **X(A)** is invalid; the **A** should be a number.

System Action: The **EDIT** is terminated; some of the edit

may have been performed, depending upon where the error was detected.

**UJP25-08 FORMAT LENGTH OF *number*
TOO BIG**

Terminal Error: The number of characters to be skipped or transferred is greater than 253.

For example:

EDIT INPUT OUTPUT A(500);

The 500 is too large because the maximum length of a symbolic variable is 253 characters.

System Action: The **EDIT** is terminated; some of the edit may have been performed, depending upon where the error was detected.

Programmer Response: Correct the program and rerun.

UJP26-nn GET Statement Messages

UJP26-01 GET HAS NO FILE NAME

Severe Error: The **GET** was incorrectly coded as **GET**;

System Action: Jol cannot know which file name to open, hence no data is obtained.

Programmer Response: Correct the program and rerun.

UJP26-02 KEY LENGTH 'number' IS NOT NUMERIC

Terminal Error: The keylength specified is not numeric.

System Action: The **GET** is terminated.

Programmer Response: Correct the program and rerun.

UJP26-03 KEY POSITION 'number' IS NOT NUMERIC

Terminal Error: The key position specified is not numeric.

System Action: The **GET** is terminated.

Programmer Response: Correct the program and rerun.

UJP26-04 KEY POSITION SPECIFIED WAS ZERO, ONE ASSUMED

Error: The **GET** assumes that the first byte of any file is 1, you specified 0 and Jol changes this to 1.

System Action: The **GET** then continues.

Programmer Response: Correct the program and rerun.

UJP26-05 KEY NAME OF 'name' IS LONGER THAN THE KEYLENGTH YOU SPECIFIED

Severe Error: The name printed is longer than the keylength and therefore will never be found in the file.

System Action: The **GET** is terminated.

Programmer Response: Correct the program and rerun.

UJP26-06 'symbol' IS INVALID IN A KEYNAME LIST

Terminal Error: Only names and numbers are allowed in a keyname list.

System Action: The **GET** is terminated.

Programmer Response: Correct the program and rerun.

UJP26-07 KEY 'key' FOUND IN RECORD 'record' WAS NOT SPECIFIED IN YOUR KEY LIST, OR THE RECORD IS DUPLICATED

Warning Message: A record was read, but the key found in the record did not match any keys specified in your key list, or the record is duplicated.

System Action: The record is ignored.

Programmer Response: Correct the program and rerun.

UJP26-08 'symbol' IS NOT SUPPORTED

Terminal Error: A **GET** was found, but the option did not specify **KEYS**, **DATA** or **LIST**.

System Action: The **GET** is terminated.

Programmer Response: Correct the program and rerun.

UJP26-09 MEMBER 'name' CANNOT BE FOUND

Terminal Error: The **GET** specified a member to be read; the member could not be found.

System Action: The **GET** is terminated.

Programmer Response: Determine the correct member, correct the program and rerun.

UJP26-10 RECORDS TOO LARGE, MAX IS 200

Severe Error: A record was read from the specified file and it was found that its length was greater than the maximum allowed.

System Action: The **GET** is terminated.

Programmer Response: Correct the data in the file and rerun.

UJP26-11 FILE SPECIFIED CANNOT BE OPENED

Terminal Error: The **GET** specified a file that the Operating System could not open.

This is probably due to the necessary **DDCARD** missing, hence Jol cannot open the file.

System Action: The **GET** is terminated.

Programmer Response:

1. Ensure that the file is allocated and rerun.

2. If the file is allocated, check that the file name in the **GET** is specified correctly, and change if necessary,

3. If problems still occur, contact your System Programmer.

UJP26-12 NO DATA IN FILE

Information: This is for information only - the file specified had no data.

System Action: The **GET** is effectively ignored.

Programmer Response: None required.

UJP26-13 GET 'DSNAME' NOT SUPPORTED - MVS ONLY

Terminal Error: The **GET** specified a DSNAME instead of a file name - this facility is not yet implemented.

System Action: The statement is ignored.

Programmer Response: Allocate the file, and use the **GET filename** instruction.

UJP26-14 I/O ERROR - CHECK FAILED

Terminal Error: An I/O Error was detected when reading the file.

System Action: The **GET** is terminated.

Programmer Response: Try running the program again. If errors still occur, recreate the data set before rerunning.

UJP27-nn PANEL Statement Messages

UJP27-02 NEW SUB-COMMAND FOUND

Terminal Error: An ending bracket (closing parenthesis) was found in an unexpected position.

System Action: The **PANEL** is ignored.

Programmer Response: Correct the instruction and rerun.

**UJP27-04 SYMBOLIC NAME 'symbolic-name'
TOO LONG**

Terminal Error: A Symbolic name greater than 8 characters in length was found. Symbolic parameter names must be 8 characters or less.

System Action: The **PANEL** is ignored.

Programmer Response: Correct the name and rerun.

**UJP27-05 REPLY LENGTH IN ERROR:-
'length'**

Error: A non-numeric reply length was found. Reply lengths must be numeric.

System Action: The **PANEL** is ignored.

Programmer Response: Correct the instruction and rerun.

**UJP27-06 ROW NUMBER 'row-number'
INVALID - DEFAULTS USED**

Error: A non-numeric row number was found. Row numbers must be numeric.

System Action: The statement is ignored.

Programmer Response: Correct the row number and rerun.

**UJP27-07 COLUMN NUMBER 'column-
number' INVALID - DEFAULTS USED**

Error: A non-numeric column number was found. Column numbers must be numeric.

System Action: The **PANEL** is ignored.

Programmer Response: Correct the column number and rerun.

**UJP27-08 REPLY ROW NUMBER 'reply-row-
number' INVALID - DEFAULT USED**

Error: A non-numeric row number was found. Row numbers must be numeric.

System Action: The **PANEL** is ignored.

Programmer Response: Correct the row number and rerun.

**UJP27-09 REPLY COLUMN NUMBER 'reply-
row-number' INVALID - DEFAULT USED**

Error: The reply column number must be numeric.

System Action: The **PANEL** is ignored.

Programmer Response: Correct the reply column number and rerun.

**UJP27-10 ERROR OCCURRED IN SCREEN
HANDLER**

Terminal Error: A problem occurred when the **PANEL** was being displayed on the terminal. An Operating System message should have been displayed prior to this with full explanation of the error.

System Action: Another attempt is made to display the **PANEL**.

Programmer Response: Advise the Terminal Controller.

UJP45-nn REDO Instruction Messages

UJP45-01 LABEL FOR REDO MUST BE 7 CHARACTERS OR LESS

Terminal Error: A **label** was found that was longer than 7 characters. The **LABEL** instruction does not allow you to specify a label of greater than 7 characters.

System Action: The **LABEL** is ignored.

Programmer Response: Use a label of 7 characters or less.

UJP45-02 CANNOT TAKE ADDRESS OF SYSIN FILE FOR LOOPING

Terminal Error: A **REDO** instruction was coded in the Jol primary input data set. This cannot be done, and is an implementation restriction.

REDO instructions can only be coded in **included** or **macro** members.

System Action: The **REDO** is ignored.

Programmer Response: Restructure the program and rerun. Alternatively, place the code into a member of the **INCLUDE** data set, and **INCLUDE** it.

UJP45-03 YOU ALREADY HAVE A LABEL, 'label-name', THIS ONE IGNORED

Terminal Error: The label instruction has been executed already with the name specified. Duplicate labels are not allowed.

System Action: The label is ignored.

Programmer Response: Use a different label name.

UJP46-nn REDO Instruction Messages

UJP46-01 TOO MANY PARAMETERS IN REDO INSTRUCTION

Terminal Error: Only one parameter or label name is allowed on the **REDO** instruction.

System Action: The **REDO** is executed using the first name as the label to branch to.

Programmer Response: Check for possible missing semi-colons after the **REDO** instruction.

UJP46-02 BLANK PARAMETER ON REDO - NEXT INSTRUCTION WILL BE EXECUTED

Warning Message: The **REDO** instruction was executed but did not have a parameter.

Sometimes, a symbolic parameter name is coded on the **REDO** instruction, and this may be null or blank.

System Action: The **REDO** is ignored.

Programmer Response: Check for null or blank symbolic names or code a parameter on the **REDO** and re-execute.

UJP46-03 LABEL 'label' NOT FOUND FOR LOOPING

Terminal Error: A **REDO** instruction was found specifying a label, however the label had not been previously specified with the **LABEL** instruction.

System Action: The **REDO** is ignored.

Programmer Response: Correct the **REDO** and rerun.

UJP46-04 SYMBOLIC NAME 'symbolic-name' FOUND AT WRONG LEVEL. ARE YOU REDOING CODE IN A MACRO WITHOUT A PREVIOUS LABEL?

Terminal Error: An outer **MACRO** had a label instruction specified and an inner **MACRO** has attempted to **REDO** that label.

A **REDO** can only **REDO** code in the include or **MACRO** member in which the label is coded.

System Action: The **REDO** is ignored.

Programmer Response: Correct the label name on the **REDO** and re-run.

UJP46-05 LENGTH OF LABEL SYMBOLIC INVALID

Terminal Error: This is an internal error.

System Action: The **REDO** is ignored.

Programmer Response: Contact your System Programmer.

UJP46-06 INCORRECT FILE LEVELS: LABEL FILE ='label-number:' CURRENT FILE ='current-file-number'. ARE YOU REDOING CODE FROM INCLUDED TEXT?

Terminal Error: A **MACRO** or **INCLUDED** text had a **LABEL** statement and it **INCLUDED** some more Jol text. The **INCLUDED** text then attempted to

re-execute the **LABEL** instruction that had been specified in the (outer) macro.

This is not allowed as a **REDO** instruction can only **REDO** instructions in the same macro command or include member.

System Action: The **REDO** is ignored.

Programmer Response: Correct the **REDO** and rerun.

UJP46-07 YOUR REDO INSTRUCTION IS BEING EXECUTED FROM A DIFFERENT MEMBER OR DATA SET

Terminal Error: A **macro** or **INCLUDEd** text had a **LABEL** statement and it **INCLUDEd** more Jol text. The included text then attempted to re-execute the label instruction that had been specified in the macro.

This is not allowed as a **REDO** instruction can only **REDO** instructions in the same macro command or **INCLUDE** member.

System Action: The **REDO** is ignored.

Programmer Response: Correct the **REDO** and rerun.

UJP46-08 YOU HAVE EXCEEDED 50 LOOPS. YOU CAN INTERRUPT IF YOU WISH TYPE C TO CONTINUE

Warning Message: The **REDO** instruction was executed more than 50 times. To prevent uncontrolled looping, Jol places a limit of 50 on loops, unless you use the **FOREVER** option with the **REDO** instruction.

System Action: The **REDO** temporarily halts.

Programmer Response: To continue, enter a C.

UJP46-09 END OF FILE WHEN RESETTNG FILE FOR REDO

Compiler Error: This is an internal error.

System Action: The **REDO** is ignored.

Programmer Response: Contact your System Programmer.

UJP46-10 YOU HAVE EXCEEDED 50 LOOPS. YOU CAN INTERRUPT IF YOU WISH TYPE C TO CONTINUE

Warning Message: The **REDO** instruction was executed more than 50 times. To prevent uncontrolled looping, Jol places a limit of 50 on loops, unless you use the **FOREVER** option with the **REDO** instruction.

System Action: The **REDO** temporarily halts.

Programmer Response: To continue merely enter a C.

UJP50-nn Read Program Statement Messages

UJP50-01 STATEMENT TOO LONG

Severe Error: The maximum length of a Jol program statement is 2048 bytes. This has been exceeded.

System Action: The job is aborted. No more statements are read from the input stream.

Programmer Response: Check for missing semi-colons, or missing quotes. Correct the statement and rerun.

UJP50-02 ERROR DETECTED IN MAIN READ PHASE

Terminal Error: An error was detected while reading a statement. Other messages (UJC50-01 to UJC50-02) would also have been issued indicating the specific error.

System Action: The statement is ignored.

Programmer Response: Correct the statement as indicated by earlier messages and rerun.

UJP50-03 DUPLICATE LABEL 'label' IGNORED

Severe Error: Duplicate labels were found. A Jol statement may have one label per statement.

System Action: The label is ignored.

Programmer Response: Remove the second label and rerun.

UJP50-04 LABEL 'label' IS > 8 CHARACTERS

Severe Error: A label must be less than 8 characters in length.

System Action: The label is ignored.

Programmer Response: Correct the label and rerun.

UJP50-05 MISSING ';' :- STATEMENT LOST

UJP50-06 MISSING ';' :- INSERTED

Error: A semi-colon was found to be missing.

System Action: A semi-colon was inserted for you, or the statement was lost.

Programmer Response: Insert the semi-colon and rerun.

UJP51-nn Read Program Statement Messages

UJP51-01INVALID LITERAL STRING '*string*' FOUND

Terminal Error: No ending quote was found for the string printed.

System Action: The compilation terminates.

Programmer Response: Correct the literal string and rerun.

UJP51-02COMMENT HAS NO END

Terminal Error: While reading a comment, the end of the program (end of the program file or a new * JOL; card) or a card file was detected.

System Action: The job is aborted.

Programmer Response: Correct the program and rerun.

UJP51-03SYMBOL '*symbol*' TOO LONG

Terminal Error: The length of the symbol printed is greater than the maximum allowed of 253 characters.

System Action: The compilation is terminated.

Programmer Response: Correct the program and rerun.

UJP52-nn Read Program Statements from Library Messages

UJP52-01 ERROR REPOSITIONING INPUT FILES

Compiler Error: When the end of a member in either the **MACRO** or **INCLUDE** library was found, an error occurred resetting the read in module to the previous member or MACRO. This is a compiler error.

System Action: The statement is ignored. Processing continues at the next statement.

Programmer Response: Contact your System Programmer.

UJP52-02 INVALID NUMBER OF LINES TO SKIP *number* DEFAULTING TO SKIP 1

Warning Message: An invalid **SKIP** number was coded.

System Action: The skip number is altered to 1.

Programmer Response: Correct the error and rerun.

UJP53-nn Read Macro Statement Messages

UJP53-01 FIRST STATEMENT NOT A MACRO BUT 'statement'

Severe Error: The first statement read from the macro library member was not a **MACRO** statement.

System Action: The statement is processed as though it had come from an **INCLUDE** library. Other errors will most likely be produced, particularly if the macro attempts to use parameters.

Programmer Response: Code a **MACRO** statement as the first instruction in the macro.

For example:

SORT : MACRO (=);

The = is invalid in the macro proto-type instruction.

System Action: The symbol is ignored. Processing continues with the next parameter.

Programmer Response: Correct the macro and rerun.

UJP53-02 TOO MANY LEVELS (IE CONCURRENTLY EXECUTING MACROS)

Terminal Error: The maximum number concurrent macro calls has been exceeded. A macro may issue a macro call, and that macro in turn may issue another macro call, and so on, to a depth of 11 macros.

System Action: The statement is processed as though it had come from an **INCLUDE** library. Other errors will most likely be produced, particularly if the macro attempts to use parameters.

Programmer Response: Restructure the program so that it does not issue as many concurrent macros.

UJP53-04 NAME 'name' INVALID IN MACRO PROTO-TYPE

Terminal Error: A symbolic parameter variable name was expected but an ordinary name was found.

For example:

SORT : MACRO (A=1);

The **A** is invalid in the macro proto-type instruction.

System Action: The symbol is ignored. Processing continues with the next parameter.

Programmer Response: Correct the macro and rerun.

UJP53-03 SYMBOL 'symbol' INVALID IN MACRO PROTOTYPE

Terminal Error: The macro proto-type statement is invalid.

UJP53-05 STRING TOO LONG

Terminal Error: While attempting to form up the items in a list. i.e. **FIELDS = (fields)**, during macro processing, the length of the string exceeded 2048 characters.

System Action: The list is ignored.

Programmer Response: Check for missing semi-colons, correct the program and rerun.

UJP54-*nn* INCLUDE and Macro Statement Messages

UJP54-01 TOO MANY ACTIVE MACROS/INCLUDES

Terminal Error: The maximum number of **INCLUDE** and **MACRO** members concurrently being read is greater than 11.

The maximum depth of concurrent action with **INCLUDE** and **MACRO** files is 11.

For example if an **INCLUDE A** statement was executed, and member A included B and member B included C, and so on there would be 8 currently active **INCLUDES**.

If member C also had a macro instruction then during processing of the macro, the current number of active members would be 11.

This implementation cannot process more than 11 such processes.

System Action: The statement is ignored.

Programmer Response: Restructure the program and rerun.

UJP54-02 CANNOT FIND MEMBER '*name*'

Terminal Error: Either the member to be **INCLUDE**d or the JOL Macro member could not be found in the library.

System Action: The statement is ignored.

Programmer Response: Correct the member name you are **INCLUDING** or change the JOL instruction name and rerun

UJP54-03 PERM I/O ERROR IN DIRECTORY LOOKING FOR '*name*'

Terminal Error: While searching the **MACRO** or **INCLUDE** library, a permanent I/O error occurred.

System Action: The statement is ignored.

Programmer Response: Run the program again. If errors continue, recreate the library.

UJP54-04 INTERNAL ERROR

Compiler Error: While saving certain information about the previous member being processed, a compiler problem occurred.

System Action: The compilation is terminated.

Programmer Response: Contact your System Programmer.

UJP54-05 MODULE '*name*' CANNOT BE INVOKED - ABSENT

Terminal Error: An **INVOKE** instruction was about to be executed but the specified load module could not be found in the library pointed to by the JOLLOAD DD Card.

System Action: The statement is ignored.

Programmer Response: Correct the name in the **INVOKE** statement, or move the module required to the library referred to by the JOLLOAD, or concatenate the library in which the required module resides to the JOLLOAD DD Card.

**UJP54-06 DYNAMIC LOADED MODULE
RETURNED NON-ZERO**

Terminal Error: An invoked module terminated processing, but on returning to JOL register 15

contained a non-zero return code, indicating that a problem had occurred in the **INVOKE** routine.

System Action: Any output from the invoked instruction is ignored.

Programmer Response: The **INVOKE**d program (which is usually written by the installation) should have issued messages to help you pinpoint the problem.

UJP60-nn and UJP61-nn READ and WRITE Instruction Messages

UJP60-01 TOO MANY SYMBOLS ON READ STATEMENT

Severe Error: The statement was coded incorrectly.
Either a semi-colon was missing after the **READ**
or **WRITE** statement, or the text was not enclosed
in quotes. The most likely cause is a missing
semi-colon at the end of the statement. For
example:

**READ INPUT
DCL %NAME INIT ";**

will produce the message above, because the semi-colon
is missing.

System Action: The extra symbols are ignored.

Programmer Response: Correct the program and rerun.

UJP61-01 TOO MANY SYMBOLS ON WRITE STATEMENT

Severe Error: The statement was coded incorrectly.
Either a semi-colon was missing after the **READ**
or **WRITE** statement, or the text was not enclosed
in quotes. The most likely cause is a missing
semi-colon at the end of the statement. For
example:

**WRITE 'ERROR, DOLLAR VALUE NOT
NUMERIC'
INPUT DOLLVAL;**

will produce the message above, because the semi-colon
is missing after the **WRITE** instruction.

System Action: The extra symbols are ignored.

Programmer Response: Correct the program and rerun.

UJP60-02 INVALID SYNTAX ON READ FILE STATEMENT

Severe Error: The statement was coded incorrectly. For
example:

READ FI(X) INTO(INPUT);

will produce the message above, because the **FI** keyword
should have been coded as **F** or **FILE**.

System Action: The instruction is ignored.

Programmer Response: Correct the program and rerun.

UJP60-02 INVALID SYNTAX ON WRITE FILE STATEMENT

Severe Error: The statement was coded incorrectly. For
example:

WRITE F(X) FRM(XYZ);

will produce the message above, because the **FRM**
keyword should have been coded as **FROM**.

System Action: The extra symbols are ignored.

Programmer Response: Correct the program and rerun.

UJP85-nn Assignment Statement Messages

UJP85-01 'operator' IS AN INVALID ARITHMETIC OPERATOR

Severe Error: The only valid operators in an assignment are a plus (+) minus (-), divide (/) multiply (*) and concatenation (||)

System Action: As a result of this assignment, the symbolic variable contains a zero (0).

Programmer Response: Correct the statement and rerun.

UJP85-02 ERROR OCCURRED CONVERTING 'symbol' TO ARITHMETIC

Severe Error: In order to perform an arithmetic operation on the symbol, an attempt was made to convert it to arithmetic form, but one or more non numeric characters were found.

System Action: As a result of this assignment, the symbolic variable contains a zero (0).

Programmer Response: Correct the program and re-submit.

UJP85-03 LENGTH ERROR CONCATENATING 'symbol'

Terminal Error: When concatenating the symbol, it was found that the length of the resulting concatenated string exceeded 253 characters. The maximum length of a symbolic variable is 253 characters.

System Action: The concatenation is not performed.

Programmer Response: Correct the program and rerun.

UJP85-04 ZERO DIVIDE CONDITION

Severe Error: When performing a division it was found that the divisor was zero.

System Action: As a result of this assignment, the symbolic variable contains a zero (0).

Programmer Response: Correct the program and rerun.

UJP85-05 SYMBOLIC NAME 'name' CANNOT BE STORED.

Terminal Error: When attempting to store the symbolic variable named, it was found that the table containing the names was full.

System Action: The statement is ignored.

Programmer Response: Reduce the number of symbolic names but your System Programmer may elect to regenerate JOL with a larger table should this problem occur frequently.

UJP85-06 CANNOT STORE VALUE OF 'name'

Terminal Error: There is insufficient space in the symbolic variable table to enter the value of the named symbolic parameter.

System Action: The statement is ignored.

Programmer Response: Reduce the size of your symbolic variables but your System Programmer may elect to regenerate JOL with a larger table should this problem occur frequently.

**UJP85-07 ERROR AFTER SYMBOLICS
REPLACED 'string'**

Terminal Error: After the symbolic variables were replaced in the assignment or SET statement, an error was detected when the statement was again split into tokens.

A message (UJC50-01 to UJC50-04) would also have been issued.

System Action: The symbolic variable is set to nulls.

Programmer Response: Correct the error and rerun.

UJP85-08 INVALID ASSIGN : - STRING IS 'string'

Severe Error: An invalid assignment was found.

For example:

%C = 'ABC' + 2;

will give the above error, as 'ABC' cannot be converted to numeric form.

System Action: The symbolic variable is set to nulls.

Programmer Response: Correct the statement and rerun.

UJP85-09 SUBSTR IN ERROR:- string

Terminal Error: The **SUBSTR** function is in error. The position or length parameter is incorrect.

System Action: The symbolic variable is set to nulls.

Programmer Response: Correct the statement and rerun.

UJP85-10 %symbolic NOT FOUND FOR function

Terminal Error: The symbolic name specified for the **SUBSTR, TYPE, LENGTH** or **INDEX** function could not be found.

System Action: The symbolic variable is set to nulls.

Programmer Response: Correct the statement and rerun.

UJP93-nn, UJP95-nn Preprocessor Initialization Messages

UJP93-01 YEAR INVALID

UJP93-07 DAY INVALID

Error: The machine date set by the computer operator is invalid.

System Action: The compile continues, but any reference to %SYSDATE, %DAY etc will return invalid or null data.

UJP95-01 COMPILER OPTION '*option*' IGNORED

Warning Message: The option printed is invalid.

System Action: The option is ignored.

Programmer Response: Correct the option and rerun.

UJP95-02 INVALID NAME '*name*' IGNORED IN * JOL OR PARAMETER FIELD OF JOL COMPILER

Severe Error: An assignment has been specified, via the PARM field in the EXEC JCL statement or the * JOL card, that has an invalid symbolic variable name.

For example:

%2 = 'STRING';

Symbolic names must commence with an alphabetic character.

System Action: The assignment is ignored.

Programmer Response: Correct the error and rerun.

UJP95-03 VALUE '*name*' IGNORED IN * JOL OR PARAMETER FIELD OF JOL COMPILER

Severe Error: An assignment has been specified via the PARM field in the EXEC JCL statement or the * JOL card that is invalid.

System Action: The symbolic variable is set to nulls.

Programmer Response: Correct the statement and rerun.

UJP95-04 LINE COUNT OPTION NOT NUMERIC:- *symbol*

Warning Message: A LINECNT = option was specified, but the number-of-lines option was not numeric.

System Action: The default value is used.

Programmer Response: Correct the value and rerun.

UJP95-05 SVC NUMBER OPTION '*symbol*' NOT NUMERIC

Severe Error: The SVC = option was specified but the SVC number was not numeric.

System Action: The default SVC number is used.

Programmer Response: Correct the option and rerun.

**UJP95-06SM2 NOT GREATER SM1 : DEFAULT
SOURCE MARGINS USED**

Error: The SM option is incorrect, for example SM=31, 12, 15 is incorrect as the end column (12) is less than the start column (31).

System Action: The default value is used.

Programmer Response: Correct the value and rerun.

**UJP95-07SM(3) IS WITHIN SM(1) - SM(2)
RANGE:- DEFAULT SOURCE MARGINS
USED**

Error: The SM option is incorrect, for example SM=(1,72,5) is wrong because the SM(3) column is within the source columns containing the JOL program source code.

System Action: The parameter is ignored.

Programmer Response: Correct the SM parameter and rerun.

UJP95-08FLAG OPTION 'option' NOT NUMERIC

Error: The **FLAGM** option does not have a valid number following it.

For example:

FLAGM=XYZ will give such an error.

System Action: The **FLAGM** is ignored.

Programmer Response: Correct the **FLAGM** and rerun.

**UJP95-09DUMPCARD OPTION > 80 CHARS:-
IGNORED**

Severe Error: The **DUMPCARD** option was used to alter the default **DUMPCARD** generated by JOL, but the maximum length of the card specified was too long. JCL cards must be less than 81 characters.

System Action: The default Dumpcard specified when JOL was installed is used.

Programmer Response: Correct the parameter and rerun.

**UJP95-10SM OPTION ERROR AT 'symbol' :-
DEFAULT SOURCE MARGINS USED**

Severe Error: An error was detected in the SM option. One of the columns specified was not numeric.

For example:

SM=(1,72,EIGHTY).

System Action: SM is unchanged.

Programmer Response: Correct the option and rerun.

**UJP95-11FLAG COLUMN IS WITHIN SOURCE
MARGINS RANGE**

Warning Message: If the Source Margins were set up as **1,73,0** and **FLAGM=55** was specified, the above error message would result.

System Action: The **FLAGM** option is cancelled.

Programmer Response: Correct the option and rerun.

Jol Compiler General Purpose Subroutine Messages (UJS)

Component Name	UJS00-UJS99 Jol Compiler Modules.
Program Producing Message	Jol Command Language Compiler
Message Format	<p>For Operator: UJSmm-nn $text$ For Programmer: UJSmm-nn $text$</p> <p><i>where:</i></p> <p>mm is the module number issuing the message.</p> <p>nn is the message number.</p> <p>$text$ is the message text.</p> <p>$severity$ is indicated in the message descriptions.</p>
Associated Publications	Jol Reference Manual. Jol General Information Manual.
Problem Determination	If the problem recurs, do the problem determination action(s) to ensure that qualified personnel will have the essential programming information available to diagnose and correct the difficulty.

UJS01-nn Catalog Search Messages

**UJS01-01 DSID 'dsid' DSNNAME 'dsname'
DECLARED AT stmt HAS TOO MANY
VOLUMES:- OS WILL SEARCH
CATLG AT EXECUTION**

Warning Message: After searching the system catalog, the number of volumes that the data set resides on was found to be greater than the limit imposed by the Jol processor.

System Action: The initiator will search the catalog at execution time.

Programmer Response: None required.

**UJS01-02 DSID 'dsid' DSNNAME 'dsname'
DECLARED AT stmt CONTAINS
DIFFERING UNIT TYPES:- OS WILL
SEARCH CATLG AT EXECUTION**

Warning Message: After searching the system catalog it was found that part of the data set resides on a device type that is different to that which subsequent sections of the data set resides on.

System Action: The initiator will search the catalog at execution time.

Programmer Response: None required.

**UJS01-03 DSID 'dsid' DSNNAME 'dsname'
DECLARED AT stmt CATLG VOL
NOT MOUNTED:- OS WILL SEARCH
CATLG AT EXECUTION**

Warning Message: The catalog volume(s) containing pointers to the data set was not mounted.

System Action: The required catalog volumes will be mounted and interrogated by the Operating System at execution time.

Programmer Response: None required.

**UJS01-04 DSID 'dsid' DSNNAME 'dsname' DECLARED
AT stmt:- CATALOG SEARCH FAILED**

Terminal Error: A data set that will be required at execution time could not be found.

A catalog search was initiated for a data set because:-

1. The data set name was part of a generation data set group and a relative generation was specified which had to be resolved to obtain the absolute generation number.
2. The data set named was to be read but had not been output in an earlier part of the job.

System Action: The job is marked as not to be executed unless the **LET** option was specified.

Programmer Response:

Case 1: Either catalog the data set or build an index. If the data set name printed contains a positive generation number, an index must be built using the **BUILDGDG** command or appropriate utility. If the data set name contains a reference to the latest generation (i.e. 0) or to an earlier generation, the data set name must be cataloged.

Case 2: The data set name must be cataloged or the appropriate unit and volumes supplied as an over-ride or as part of the data set declaration.

The Jol program must be rerun.

**UJS01-05 DSID 'dsid' DSNAME 'dsname'
DECLARED AT stmt SPECIFIED AN
INDEX BLOCK**

Terminal Error: When searching the catalog for a data set whose volume or unit had not been specified, it was found that the name was not a data set name, but an index which contains data set names at a lower level.

System Action: The job is marked as not to be executed unless the **LET** option was specified.

Programmer Response: Correct the data set name and rerun.

**UJS01-06 DSID 'dsid' DSNAME 'dsname'
DECLARED AT stmt ANOTHER
DATA SET WAS FOUND AT A
LOWER INDEX LEVEL**

Terminal Error: When searching the catalog for a data set whose volume or unit had not been specified a data set name was found at a higher level than the data set name in the associated declare statement.

System Action: The job is marked as not to be executed unless the **LET** option was specified.

Programmer Response: Correct the data set name or supply volume and unit information and rerun the job.

**UJS01-07 DSID 'dsid' DSNAME 'dsname' HAS
DECLARED AT stmt HAS
INCORRECT SYNTAX IN THE
DSNAME**

Terminal Error: When searching the catalog for a data set whose volume or unit had not been specified, it was found that the data set name did not conform to the syntax specified for data set names.

A data set name is made up of a series of variable names separated by periods except for the last which may be followed by a relative generation number.

System Action: The job is marked as not to be executed unless the **LET** option was specified.

Programmer Response: Correct the data set name in the declare statement and rerun the program.

**UJS01-08 DSID 'dsid' DSNAME 'dsname' DECLARED
AT stmt:- PERM I/O ERR ON CATLG**

Terminal Error: When searching the catalog a permanent input/output error occurred.

System Action: The job is marked as not to be executed unless the **LET** option was specified. Jol will also notify the operator that the error occurred.

Programmer Response: Contact the System Programmer.

**UJS01-09 DSID 'dsid' DSNAME 'dsname' DECLARED
AT stmt:- CAT RTN RETND 28**

Terminal Error: This message should not occur. This is a serious error.

System Action: The job is marked as not to be executed unless the **LET** option was specified.

Programmer Response: Contact the System Programmer.

**UJS01-10 DSID 'dsid' DSNAME 'dsname' DECLARED
AT stmt:- INV WORK AREA**

Compiler Error: Jol supplied the Operating System with an invalid work area address.

This is a compiler error.

System Action: The job is marked not executable.

Programmer Response: Contact your System Programmer.

**UJS01-11 DSID 'dsid' DSNAME 'dsname'
DECLARED AT stmt CATALOG
SEARCH FAILED:- INSTRUCTION
IGNORED FOR DISPOSITIONS**

Information: A **KEEP, CATALOG, UNCATALOG, SCRATCH** or **DELETE** instruction was made referring to a data set that has not been used or created earlier in that job. The data set was not cataloged and cannot be **SCRATCHED**.

System Action: The statement is ignored.

Programmer Response: This may be a normal warning message, as you may find it desirable to **DELETE** or **SCRATCH** any new data set that is about to be created within the job.

Alternatively, if it is desired to really **DELETE** or **SCRATCH** the data set concerned, Jol found that it was not cataloged: you must either catalog the data set, or alternatively provide Jol with the Unit and Volume on which the data set resides.

**UJS01-12 DSID 'dsid' DSNAME 'dsname'
DECLARED AT stmt HAS DIFFERING
FILENOS/LABELNOS:- OS WILL
SEARCH THE CATLG AT
EXECUTION**

Warning Message: After a catalog search was performed, it was found that a data set resided on different labels of a Tape Volume.

System Action: The OS initiator will find the data sets and allocate them for you itself; Jol will not use such data sets.

Programmer Response: None required, but this may also because a data set was incorrectly cataloged.

**UJS01-13 DSID 'dsid' DSNAME 'dsname' DECLARED
AT stmt HAD INCORRECT UNIT
SPECIFIED: CATALOG ONE USED**

Warning Message: Jol searched the catalog for a data set to find which Volume it was on; the Unit had been specified on the **DECLARE** but Jol found that the unit specified differed from the one in the catalog.

System Action: The unit from the catalog is used instead of the unit specified.

Programmer Response: None required.

NOTE: If you have both Volume and Unit information specified, Jol will not search the catalog, hence if you have two data sets of the same name residing on different Volumes, you must supply both the Unit and Volume information for the data set that is not cataloged, or the above error is liable to result.

**UJS01-14 DSID 'dsid' DSNAME 'dsname' DECLARED
AT stmt HAD INCORRECT FILENO/LABEL
SPECIFIED: CATALOG ONE USED**

Warning Message: Jol searched the catalog for a data set to find its Unit and/or Volume and found that the Label or File number referred from the catalog differed from the one specified.

System Action: The catalog is assumed to be correct, and the File number from the catalog is used.

Programmer Response: None required.

**UJS01-15 DSID 'dsid' DSNAME 'dsname' DECLARED
AT stmt HAD INCORRECT VOLUME
SPECIFIED: CATALOG ONE USED**

Warning Message: Jol searched the catalog for a data set to find which Unit it was on; the Volume had been specified but Jol found that the Volume specified differed from the one in the catalog.

System Action: The volume from the catalog is used instead of the volume specified.

Programmer Response: None required.

NOTE: If you have both Volume and Unit specified, Jol will not search the catalog, hence if you have two data sets of the same name residing on different Volumes, you must supply both the Unit and Volume information for the data set that is not cataloged, or the above error is liable to result.

**UJS01-16 DSID '*dsid*' DSNNAME '*dsname*'
DECLARED AT *stmt* IS A
TEMPORARY DATA SET THAT HAS
NOT YET BEEN CREATED**

Terminal Error: A temporary data set is to be read, but it has not yet been created by an earlier program step within the job.

Temporary data sets cannot be **KEPT** or **CATALOGED**, hence they cannot be passed from job to job, and must, therefore, be created in the same job in which they are to be read. Furthermore, before they can be read they must be written or created by a program, COPY command or some similar instruction.

System Action: The job is marked not executable.

Programmer Response: Correct the logic error in the program, and rerun.

**UJS01-17 DSID '*dsid*' DSNNAME '*dsname*'
DECLARED AT *stmt* SPECIFIES UNIT
unit WHICH IS UNKNOWN, OR**

**DEVICE TYPE RETURNED FROM
CATALOG NOT IN DEVNAMET**

Error: Jol searched the system catalog for an input data set, but found that the Unit returned from the catalog was not known to it.

This is a Compiler Error as Jol should know all the IBM supplied unit types.

System Action: The job is marked as not to be executed unless the **LET** option was specified.

Programmer Response: Notify your System Programmer.

To circumvent code the correct **UNIT** on the declare statement, and rerun the job.

**UJS01-18 DSID '*dsid*' DSNNAME '*dsname*' DECLARED
AT *stmt* USER DCB CANCELLED FOR
EXISTING MOD DATASET**

Warning Message: Information was specified on a data set declare statement which is **ADDED** to or **MODDED** onto; however the data set was found in the system catalog.

System Action: Jol cancels the DCB information coded as this would destroy the DCB information in the data set itself if it was different to the information coded on the **DECLARE** statement.

Programmer Response: None required.

**UJS01-19 DSID '*dsid*' DSNNAME '*dsname*' DECLARED
AT *stmt* IS BEING OVER-WRITTEN**

Warning Message: A (0) or (-ve) generation of a data set is to be written upon and destroyed - a warning is issued.

UJS10-nn Store Variable Data Subroutine Messages

UJS10-01 *symbol* NOT STORED AS *field* ALREADY CONTAINS DATA

Error: When Jol was storing the value of a parameter, it found that the parameter had already be specified.

For example:

**JOB1: JOB ACCT X
ACCT Y ...**

specifies two account codes to be stored.

In this case, X is taken to be the account code and is stored. However, the account code is then erroneously specified again and so Jol produces the above message.

System Action: The symbol specified is ignored.

Programmer Response: Correct the program and rerun.

UJS10-02 *symbol* NOT STORED AS *field* IS TOO SHORT TO RECEIVE THE DATA. MAXIMUM LENGTH IS *length*

Error: A symbol has been found that is larger than the permitted length.

System Action: The symbol is ignored.

Programmer Response: Adjust the symbol to be less than or equal to the maximum permitted length.

UJS10-03 *number1* NOT STORED IN THE FIELD AS IT IS > *number2*

Error: *number1* is larger than the maximum allowed for the keyword printed.

For example:

JOB1 : JOB PRTY 32;

specifies a priority number that is too large. The maximum permitted value for priority is 15 and so the above error message would be produced.

System Action: The parameter is ignored.

Programmer Response: Correct the program and rerun.

UJS10-04 *symbol* NOT STORED IN THE *field* AS IT IS NOT NUMERIC

Error: The symbol printed should be a number but it is not.

For example:

DCL X DS BLOCKSIZE XYZ;

specifies 'XYZ' as a Blocksize; the blocksize must be numeric.

System Action: The parameter is ignored.

Programmer Response: Correct the error and rerun.

UJS15-nn Jol Compiler Workfile Messages

UJS15-01 -VE OR 0 LEN RECORD

Severe Error: This is a compiler error.

System Action: The compilation is terminated.

Programmer Response: Contact your System Programmer.

UJS16-nn Jol Compiler READ From Workfile Messages

UJS16-01 IDENTIFIER '*identifier*' IS A NAME ONLY AND CAN ONLY BE TESTED WITH AN 'IF'

Error: An action has been requested on a variable but cannot be performed. For example:

A : SORT X TO Y etc;

CATLG A;

The label 'A' is a **label** name, and therefore cannot be used in a **CATALOG** statement.

Similarly, if you said **RUN A;** the same message would be produced.

System Action: The statement is ignored.

Programmer Response: Correct the error and rerun.

UJS16-02 RECORD TOO LARGE

Severe Error: This is a compiler error.

System Action: The compilation is terminated.

Programmer Response: Notify your System Programmer.

UJS19-nn Internal Compiler Messages

UJS19-01 NUMB SYM PARMS < 0

Compiler Error: When reorganising the symbolic parameter table, the number of symbolic parameters was less than zero. This is a compiler error.

System Action: The compilation is terminated.

Programmer Response: Contact your System Programmer.

UJS19-02 LENGTH SYM PARM < 0

Compiler Error: When reorganising the symbolic parameter table, a symbolic parameter had a length less than zero. This is a compiler error.

System Action: The compilation is terminated.

Programmer Response: Contact your System Programmer.

UJS20-nn Replace Symbolic Variables Messages

UJS20-01 SYMBOLIC NAME COMMENCING WITH '*name*' IS TOO LONG

Terminal Error: A string commencing with the '*name*' is greater than 8 characters. Symbolic parameter names must be less than 9 characters.

System Action: The invalid text is replaced with a null string.

Programmer Response: Correct the program using shorter names.

UJS20-02 CANNOT FIND '*%name*' SET TO NULL

Terminal Error: When attempting to replace the symbolic parameter named, it was found that the name has not been declared or an assignment had not been made to the symbolic name, thus the symbolic parameter cannot be found.

System Action: The text is replaced with a null string.

Programmer Response: Either **DECLARE** the Symbolic Variable, or assign a value to it before attempting to use it in a replacement context.

UJS20-03 %LIST | %CARD SUBSCRIPT NEGATIVE

UJS20-04 %LIST | %CARD SUBSCRIPT NOT NUMERIC OR TOO LARGE

UJS20-05 %LIST SUBSCRIPT NOT NUMERIC OR TOO LARGE

Terminal Error: A conversion error occurred, or the subscript was too large (or negative) for a %LIST or %CARD item.

For example:

%LIST (A)

will give rise to the above message.

System Action: The replacement action for that variable is ignored.

Programmer Response: Correct the subscript and rerun.

UJS20-06 STRING -VE

Compiler Error: This is an internal error.

System Action: The statement is ignored.

Programmer Response: Contact your System Programmer.

UJS20-07 STRING TOO LARGE AFTER REPLACING SYMBOLICS

Terminal Error: After replacing the symbolic variables in a string, the length of the string was greater than 2048 characters.

The maximum length of any statement is 2048 characters.

Programmer Response: Split the statement into several smaller statements, and rerun.

System Action: The statement is ignored.

UJS61-nn Generation Data Set Messages

**UJS61-01 DSNAME '*dsname*' DECLARED AT
stmt SPECIFIES ABSOLUTE
GENERATION, AND ABSOLUTE
GENERATION ALSO SPECIFIED ON
PROGRAM OR DISPOSITION; DSID
GEN IGNORED**

Error: When attempting to perform the arithmetic processing specified on the DSID and the program definition, it was found that both definitions of the generation specified absolute generation numbers.

System Action: No arithmetic is performed, instead the absolute generation is copied from the program definition.

Programmer Response: Correct the program and rerun.

**UJS61-02 DSNAME '*dsname*' DECLARED AT *stmt* IS
NOT A REAL DATA SET, NO
GENERATION OVERRIDE CAN OCCUR.**

Error: A **DSID** was found that was not a real data set (eg Temporary, Print or Punch) and the program definition specified either a specific or relative generation number.

System Action: No generation number is copied to the output data set.

Programmer Response: Correct the program and rerun.

Jol Compiler – User Installed Validation Messages (UJU)

Component Name	UJU00-UJU99 Jol User Installed Validation Compiler Modules.
Program Producing Message	Jol Command Language Compiler
Message Format	<p>For Operator: UJUmm-nn <i>text</i> For Programmer: UJUmm-nn <i>text</i></p> <p><i>where:</i></p> <p>mm is the module number issuing the message.</p> <p>nn is the message number.</p> <p><i>text</i> is the message text.</p> <p><i>severity</i> is indicated in the message descriptions.</p>
Associated Publications	Jol Reference Manual. Jol General Information Manual.
Problem Determination	If the problem recurs, do the problem determination action(s) to ensure that qualified personnel will have the essential programming information available to diagnose and correct the difficulty.

UJU01-nn Installation Job Validation Messages

**UJU01-01 JOBNAME 'jobname' DOES
NOT COMMENCE WITH 'T' OR 'P'**

Explanation: These and similiar messages may be produced by installation exits. They are not ordinarily produced by Jol.

**UJU01-02 JOBCLASS CHANGED TO 'B'
FROM 'jobclass'**

Programmer Response: If you have any difficulty with installation produced messages, contact your System Programmer.

**UJU01-03 ACCOUNT CODE 'acct' IS
INVALID**

UJU02-nn Installation Program Validation Messages

**UJU02-01 IMASPZAP EXECUTED BY
 INVALID PERSONNEL**

Terminal Error: Your installation exit has
determined that you are not authorized to run
the IMASPZAP system utility.

Explanation: These and similiar messages may be produced
by installation exits. They are not ordinarily produced
by Jol.

Programmer Response: If you have any difficulty with
installation produced messages, contact your System
Programmer.

UJU03-nn Installation Data Set Validation Messages

**UJU03-02 DSID 'dsid' HAD A MEMBER
SPECIFIED AND DSORG 'dsorg'
DSORG MEMBER CANCELLED**

Warning Message: Your installation exit has determined that the blocksize for an output data set is too small for efficient use.

**UJU03-03 BLOCKSIZE FOR DSID 'dsid'
DSNAME 'dsname' DECLARED AT
stmt IS TOO SMALL FOR EFFICIENT
USE**

Explanation: These and similar messages may be produced by installation exits. They are not ordinarily produced by Jol.

Programmer Response: If you have any difficulty with installation produced messages, contact your System Programmer.

Jol Compiler Validation Messages (UJV)

Component Name	UJV00-UJV99 Jol Compiler Modules.
Program Producing Message	Jol Command Language Compiler
Message Format	<p>For Operator: UJVmm-nn <i>text</i> For Programmer: UJVmm-nn <i>text</i></p> <p><i>where:</i></p> <p>mm is the module number issuing the message.</p> <p>nn is the message number.</p> <p><i>text</i> is the message text.</p> <p><i>severity</i> is indicated in the message descriptions.</p>
Associated Publications	Jol Reference Manual. Jol General Information Manual.
Problem Determination	If the problem recurs, do the problem determination action(s) to ensure that qualified personnel will have the essential programming information available to diagnose and correct the difficulty.

UJV00-nn Main Validation Module Messages

**UJV00-01 TOO MUCH DATA IN
GENERATE PHASE :- RERUN IN
LARGER REGION EXTRA REQD
APPROX *nn* BYTES**

Terminal Error: The program being compiled is too large.

System Action: The compilation is terminated.

Programmer Response: Resubmit the job, but allocate more virtual storage to the Jol Compiler.

**UJV00-02 NO PROGRAM TEXT FOUND
IN GENERATOR PHASE :-
COMPILER PRODUCED NO TEXT**

Terminal Error: No code was found in the generate phase. This is probably as a result of a logic problem in the preprocessor phase.

For example, if no **RUN** statements were found, the message may be produced.

System Action: The job is marked non-executable.

Programmer Response: Check the program logic, correct the problem and rerun.

UJV00-03 CANNOT GET MINIMUM 20K

Error: Jol cannot obtain a minimum storage required of 20K bytes in which to perform disposition and catalog searching in the Optimization Phase.

System Action: The job is marked not executable.

Programmer Response: Rerun the job, but increase the **REGION** size for the Jol Compiler.

**UJV00-04 DSID '*dsid*' DSNAME '*dsname*'
DECLARED AT *stmt* IS ALSO OUTPUT
HERE:- CHANGED TO 'RE-WRITES'**

Error: Jol has detected a probable logic problem in your Jol program:- you are attempting to write a data set twice in the same job.

System Action: Jol creates special Operating Instructions so that it will reuse the data set if it already exists, otherwise the Operating System will create the data set for you as usual.

Programmer Response: Check the logic of your program, and if there is a problem, alter your program, otherwise ignore the warning message.

UJV01-nn Job Definition Validation Messages

UJV01-01 NO JOBNAME :- NOW 'JOLJOB'

Severe Error: No **JOBNAME** was specified. The operating system must have a jobname before it can run your job.

System Action: A jobname of **JOLJOB** is provided.

Programmer Response: Supply a JOB statement and rerun.

UJV01-03 DEFAULTS MADE TO JOB DETAILS

Information: Defaults were copied from the details provided by the JOLGEN.

System Action: Defaults were applied to the Job statement.

Programmer Response: None required.

UJV01-02 JOB CLASS 'class' IS INVALID

Severe Error: An invalid **CLASS** was found. Valid job classes are A-Z and 0-9.

System Action: The job is marked not executable.

Programmer Response: Correct the **CLASS** and rerun.

UJV01-04 JOB CPU TIME NOT NUMERIC OR > 1440

Severe Error: The CPU time specified on the Job Definition is not numeric or is greater than 1440 Minutes.

System Action: The CPU time is cancelled.

Programmer Response: Correct the error and rerun.

UJV02-nn Program Definition Validation Messages

UJV02-01 PROGRAM NAME '*program-name*' IS INVALID IN FIRST CHARACTER.

Information: Defaults were copied from the details provided by the JOLGEN.

Programmer Response: None required.

Terminal Error: The program name is invalid.

System Action: The job is marked not executable.

Programmer Response: Correct the program and rerun.

**UJV02-03 PROGRAM CPU TIME NOT
NUMERIC OR > 1440**

Terminal Error: The CPU time is invalid.

System Action: The CPU time is cancelled.

Programmer Response: Correct the error and rerun.

**UJV02-02 DEFAULTS MADE TO
PROGRAM DEFINITION**

UJV03-nn Data Set Validation Messages

UJV03-01	THE	does not commence with an alpha character or contains two consecutive periods.
BLKSIZE NOT NUMERIC OR <i>value</i>		<i>System Action:</i> The job is set non executable.
RECSIZE NOT NUMERIC OR <i>value</i>		<i>Programmer Response:</i> Correct the problem and rerun.
KEYPOSN NOT NUMERIC OR <i>value</i>		UJV03-03 RECORD FORMAT CHARACTER 'character' IN FORMAT 'record format' SET TO AS BLANK IT WAS INVALID.
KEYLEN NOT NUMERIC OR <i>value</i>		
LABEL NOT NUMERIC OR <i>value</i>		
PRIMARY SPACE NOT NUMERIC OR <i>value</i>		
SECONDARY SPACE NOT NUMERIC OR <i>value</i>		
DIRECTORY SPACE NOT NUMERIC OR <i>value</i>		<i>Severe Error:</i> The record format is not acceptable to the Operating System.
RETENTION PERIOD NOT NUMERIC OR <i>value</i>		<i>System Action:</i> The record format is cancelled.
:- NULLIFIED		<i>Programmer Response:</i> Correct the record format and rerun.
<i>Severe Error:</i> These messages are essentially the same. The parameter specified must be numeric and less than or equal to the value shown.		
<i>System Action:</i> The parameter is nullified.		
<i>Programmer Response:</i> Correct the parameter and rerun.		
UJV03-02	DSNAME ' <i>dsname</i> ' INVALID	UJV03-04 DUPLICATE CHARACTER '<i>symbol</i>' IGNORED IN RECORD FORMAT
<i>Terminal Error:</i> The data set name is invalid. A qualifier is either greater than 8 characters,		
		<i>Error:</i> The record format is not acceptable to the Operating System.
		<i>System Action:</i> The record format is cancelled.
		<i>Programmer Response:</i> Correct the record format and rerun.
		UJV03-05 OPTCD FORMAT CHARACTER 'character' IN OPTION '<i>optcd</i>' SET TO BLANK AS IT WAS INVALID
		<i>Severe Error:</i> The OPTCD fields is not acceptable to the Operating System.
		<i>System Action:</i> The OPTCD format is cancelled.

Programmer Response: Correct the **OPTCD** and rerun.

**UJV03-06 DUPLICATE CHARACTERS
'symbol' IN OPTCD IGNORED IN DCB
OPTION FIELD**

Error: The OPTCD parameter had duplicate options.

System Action: The duplicate options are cancelled.

Programmer Response: Correct the OPTCD and rerun.

**UJV03-07 BLOCKSIZE ALTERED TO
value**

Error: A fixed blocked DCB had a blocksize that was not evenly divisible by the record length.

System Action: The blocksize is altered to the blocksize in the message.

Programmer Response: Correct the error and rerun if necessary.

**UJV03-08 RECORD FORMAT NOW VB,
AND BLOCKSIZE NOW value**

Severe Error: A DCB of F or FB had been specified and there was no record length.

System Action: Jol substitutes VB, blocksize and record length.

Programmer Response: Correct the error and rerun if necessary.

UJV03-09 RECORD FORMAT NOW VB

Severe Error: An error was detected with the Record Format, Blocksize and Record Size. Other messages would have been issued with further details.

System Action: The Record Format is changed to VB.

Programmer Response: Correct the errors and rerun.

**UJV03-10 PRINTER/PUNCH HAD
EXPIRATION DATE:- NULLIFIED**

**UJV03-11 TEMPORARY DATA SET HAD
EXPIRATION DATE:- NULLIFIED**

Warning Message: A **PRINT, PUNCH** or Temporary data set (one with no data set name, or one commencing with &) had a **RETAIN** parameter specified.

Because these data sets must be deleted at the end of the job, a retention period is not allowed.

System Action: The retention period was cancelled.

Programmer Response: Correct the program and rerun if necessary.

**UJV03-12 INPUT DATA SET HAS
CONFLICTING DCB PARAMETERS
RECFM = *recfm* BLOCKSIZE = *blocksize*
RECORDSIZE = *recordsize***

Terminal Error: A data set is being read, but there is a mistake in the DCB, or Data Control Block.

For example:

FB 80,900

would be incorrect, as 80 will not divide evenly in 900.

System Action: The job is marked not executable.

Programmer Response: Correct the program and rerun.

**UJV03-13 CANNOT CALC SPACE - NO
BLOCKSIZE**

**UJV03-14 CANNOT CALC SPACE - NO
NUMBER OF RECORDS**

**UJV03-15 CANNOT CALC SPACE - NO
RECORD SIZE**

Error: When specifying the space required for an OUTPUT data set of the form *x-y RECORDS*, some of the information necessary for Jol to calculate the space requirements was missing.

System Action: The job is marked not executable.

Programmer Response: Supply the information, and rerun.

**UJV03-16 VARIABLE LENGTH
RECORDS AND LRECL/2 IS USED
TO CALC SPACE**

Warning Message: When specifying the space required for an OUTPUT data set of the form *x-y RECORDS*, the keywords OF *xxx* was missing for a Variable Blocked file. Jol does not know the average length of the output records.

System Action: The average is used for the calculations.

Programmer Response: Supply the information, and rerun, unless the average is acceptable.

UJV03-17 RECORD RANGE INVALID

Error: When specifying the space required for an OUTPUT data set of the form *x-y RECORDS*, an error was detected. Probably, the maximum number of records specified was less than the minimum.

System Action: The job is marked not executable.

Programmer Response: Correct the information, and rerun.

**UJV03-18 SPACE IGNORED FOR VSAM DATA
SET**

Severe Error: For this implementation of Jol, VSAM files must be pre-allocated.

System Action: The job is marked not executable.

Programmer Response: Use the IDCAMS utility to pre-allocate the data set, and rerun.

UJV03-19 DEFAULTS MADE TO DSID *dsid*

Information: Required information for an output data set was missing.

System Action: Information supplied by your System Programmer when Jol was installed is used as a default. Typically, this would be UNIT, SPACE or DCB information.

Programmer Response: Correct the program and rerun if necessary.

UJV03-20 NO UNIT SPECIFIED FOR *dsid*

Terminal Error: The required UNIT information for an output data set was missing. All output data sets must have a UNIT such as TAPE or DISK coded.

In this case, information was not supplied by your System Programmer when Jol was installed that could be

used as a default, and therefore Jol does not know whether the data set is to go to DISK or TAPE.

System Action: The job is marked not executable.

Programmer Response: Supply the missing information and rerun.

**UJV03-21 NO SPACE SPECIFIED FOR
*dsid***

Terminal Error: SPACE information for an output data set was missing. All output DISK data sets must have the SPACE required so that the Operating System can allocate the space for the data set on Disk.

In this case, information was not supplied by your System Programmer when Jol was installed that could be used as a default, and therefore Jol could not supply a default.

System Action: The job is marked not executable.

Programmer Response: Supply the missing information and rerun.

**UJV03-22 NO RECORD FORMAT, BLOCK-
SIZE OR RECORD-SIZE FOR DSID '*dsid*'**

Warning Message: DCB information for an output data set was missing. All output data sets must have DCB or Data Control Block information. This is usually coded in your Jol or JCL statements as this gives a great deal of flexibility to your Control Language statements.

It can, however, be "hard coded" in a program, and in that case it is not necessary to code it on your control language statements.

Jol can supply a default DCB, if required. In this case, information was not supplied by your System Programmer when Jol was installed that could be used as a default, and therefore Jol could not supply a default.

System Action: None.

Programmer Response: Supply the missing information and rerun, or, if the information is "hard-coded" in the program, use the **NODEF** parameter on the DCL statement to suppress the message.

UJV05-nn CATALOG Instruction Validation Messages

**UJV05-01 DSID '*dsid*' DSNAME '*dsname*'
CANNOT BE CATALOGED : - NO
UNIT AND/OR VOLUME SPECIFIED
AND DATA SET NOT CREATED
EARLIER IN JOB.**

Programmer Response: In all probability, a logic mistake has occurred. Check the program to ensure that the **CATLG** instruction is in the correct place and that there are no missing statements associated with this request.

Error: A **CATLG** instruction has been issued for a data set that has no volume or unit.

For example: No **RUN** statement was found prior to the **CATLG** instruction.

System Action: The instruction is ignored.

UJV07-nn KEEP Instruction Validation Messages

**UJV07-01 DSID '*dsid*' IS A TEMPORARY
DATA SET AND CANNOT BE KEPT
OR CATALOGED.**

Error: A **KEEP** or **CATALG** instruction has been
issued for a temporary data set. This is illegal.

System Action: The instruction is ignored.

Programmer Response: Check the program to ensure
that the data set name should be temporary or
that the **KEEP/CATLG** references the correct
DSID.

**UJV07-02 DSID '*dsid*' IS A PRINT /
PUNCH / CARD / UNIT RECORD-
FILE AND CANNOT BE KEPT OR
CATALOGED.**

Error: An erroneous **KEEP/CATLG** instruction has
been issued for a unit record file.

System Action: The instruction is ignored.

Programmer Response: Correct the error and rerun.

**UJV07-03 DSID '*dsid*' DSNAME '*dsname*' HAS
NOT YET BEEN OUTPUT AND CANNOT
BE KEPT**

Error: A **KEEP** or **CATLG** instruction has erroneously
been issued for a data set that has not yet been
created. A data set that is not produced in job may be
KEPT or **CATALOGED** if the Volume and Unit are
coded on the DSID or the **KEEP** or **CATALOG**
instruction.

System Action: The instruction is ignored.

Programmer Response: Check the program logic. Correct
the error and rerun.

**UJV07-04 DSID '*dsid*' DECLARED AT *stmt* IS A
DUMMY DSID AND CANNOT BE KEPT OR
CATALOGED**

Warning Message: An erroneous **KEEP/CATLG**
instruction has been issued for a DUMMY data set.

System Action: The instruction is ignored.

Programmer Response: Correct the error and rerun.

UJV08-nn SCRATCH Instruction Validation Messages

**UJV08-01 DSID '*dsid*' DSNAME '*dsname*'
CANNOT BE SCRATCHED : - NO UNIT
AND/OR VOLUME SPECIFIED AND
NOT OUTPUT EARLIER IN JOB.**

yet been output or has no volume and/or unit
coded.

System Action: The instruction is ignored

Programmer Response: Check the problem and rerun.

Warning Message: A **SCRATCH** or **DELETE**
instruction was issued for a data set that has not

Jol Messages and Codes

Readers Comment Form

This manual is part of the Jol library that serves as a reference source for Systems Analysts, Programmers and Operators. This form may be used to communicate your views about this publication. They will be sent to the authors department for whatever review and action, if any, is deemed appropriate.

CCS-Jol may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use any information you supply.

Possible topics for comments are:

Clarity Accuracy Completeness Organization Legibility

If comments apply to a Selectable Unit, please supply the name of the Selectable Unit _____.

If you wish a reply, give your name and address

Number of latest Newsletter associated with this publication: _____

Please send your comments to:

Clarke Computer Software,
45 Riversdale Road,
Hawthorn,
AUSTRALIA, 3122.