900 Series

Software Release Notice

SRN/AJH ALGOL/1.1 2nd April 2015

PURPOSE OF RELEASE

To consolidate several versions of Algol that have been developed from the Issue 6 release (MASD Algol, Hunter Algol) and to address errors the Issue 6 release reported by users. The consolidated system is called “AJH Algol”.

CONTENTS OF RELEASE

The release consists of three sum-checked binary tapes and two relocatable binary tapes, labelled:-

ALG1(AJH) 23/03/2015	2 Pass Algol Translator (TAPE 1)
ALG2(AJH) 23/03/2015	2 Pass Algol Interpreter (TAPE 2)
ALG3(TJF) 02/02/2015	Algol library tape (TAPE 3)
ALG16KLG(AJH) 24/03/2015	16K Algol load-and-go system
ALG16KLP(AJH) 23/03/2015	16K Algol large program system
ALG64KLP(AJH) 23/03/2015	64K Algol large program system
INTERRUPT 27/03/2015	RLB Algol code procedure
UNSIGNED 23/03/2015	RLB Algol code procedure.

DOCUMENTATION CHANGES

900 Algol is documented in Volume 2, Part 1, Section 2 of the 900 Technical Manual. There is no plan to make a new issue of this document: accordingly users should update their current copy with the information following, and retain a copy of this Notice at the front of the volume.

Revised Algol Representation (Chapter 1.1):-

The Algol alphabet has been updated to support 900 telecode and ASCII.

Codes 35 and 92 are treated as the sterling symbol (£) on input and output as code 35.

Codes 64 and 96 are treated as the close quote symbol (`) on input and output as code 96.

Codes 123 and 125 ({ and } in ASCII) are treated as alternatives to quote symbols (‘ and `) on input and output as codes 39 and 96 respectively.

This alphabet applies to all Algol input and output, including programs, error reports and data.

On program input, code 95 (_ in ASCII) deletes the current line of input. This enables programs to be entered interactively from the teletype.

Restrictions removed (Chapter 1.2.1):-

1. Recursive procedures are allowed, but with two restrictions:
    a. Procedures still have to be declared before they are used and thus mutually recursive procedures are not permitted.
    b. Local data within a recursive procedure remains static. If dynamic local data is required a nested local procedure should be introduced with the dynamic data held in value parameters.
2. Expressions are permitted in call-by-name arguments, allowing the use of, for example, Jensen’s device. An error is signalled if an attempt is made to assign a value to a formal name parameter bound to an actual parameter that is an expression or constant.
3. The parameters of a formal procedure parameter are no longer assumed to be call-by-name by default, but the translator needs help with calls of the formal procedures: the call must indicate which parameters are called by name and which by value. It assumes a call by value is meant when a closing bracket, round or square, precedes the comma or closing round bracket of the call. This means that an extra bracket must sometimes be used.
4. Labels do not have to be declared (in switch declarations) before use. An error will be reported at the end of a block if it contains a “goto” statement referencing a non-existent or out-of-scope label.

2.2.3 Format Setting Procedures

The UNSIGNED tape contains four new format setting procedures:

code procedure unsigned; algol;

Suppresses the printing of the sign for integers.

code procedure leadzero; algol;

Prints leading zeros in integers as zeros rather than as spaces.

code procedure revertu; algol;

Cancels unsigned.

code procedure revertl; algol;

Cancels leadzero.

Machine Code (Chapter 3.4)

All versions of the interpreter execute the loaded program at priority level 4. Null interrupt routines are set up at levels 1, 2 and 3 which drop back to the previous level. If a program is resumed after a wait by re-entry at 9 (or typing C to the large program system) it resumes execution on priority level 4.

In the example contained in Chapter 3.4.6 Switch Parameters, the instruction /0 6 should have been /0 9.

In Chapter 3.4.8 Use of Interpreter Subroutines, subroutine 28 (R1 := R1 ^ I2) is no longer available.

Errors During Library Scan (Chapter 4.1.3)

It is not possible to give a precise limit to the maximum size of a library program allowed during a library scan – such programs are buffered as they are read in, and the space available depends up how many names are in the translator dictionary. The suggested figure of 600 words is a practical limit for most cases. If a library program is too large to be loaded as part of the library scan, the operator should load it as a standalone code procedure.

Error Messages (Chapter 4.2)

Two new loader error messages have been added to the Large Program system:-

FG Attempt to locate a global label in a code procedure at location 8151 or above.

FZ Attempt to load program or code procedures at location 8180 or above.

Translation (Chapter 5.2)

Two new translator entry points are available:

17	Enable call-by-name extensions (default)
18	Disable call-by-name extensions

Disabling the call-by-name extensions can result in shorter and faster programs.

When translating a program in library mode, the translator (in both the 2-pass and Load-and-Go systems) halts at the end of scanning the submitted library tape (e.g., TAPE 3) rather than going on to copy data up to a terminating halt code.

Dumping (Chapter 5.2)

The facility to dump a copy of an executing program with the 2-pass interpreter has been removed, although the facility to take a dump by entry at 14 when the system is quiescent remains. In place of this a new code procedure called “interrupt” has been made available and is distributed in RLB form. If interrupt is added to a program e.g., by including code procedure interrupt; algol; in the users program and called at the start of the program an “interrupt and wait” facility is provided. After interrupt has been called, generating a level 1 interrupt injects a wait function call as the next Algol primitive to be executed. The operator then has the opportunity to request a dump be punched and can if desired then continue execution from the point of interruption. (In the 2-pass system this is by use of entry at 14 and 9 respectively; in the Large Program system by use of the D and C commands; in the Load-and-Go system there is no dump facility, but interrupt can still be used to stop a program temporarily.)

Use of Algol on 920 Computers (Appendix 2)

AJH Algol does not support 503/920 telecode as this is now considered obsolete.

Use of the 903 Digital Plotter

The new Tape 3 includes the standard library issue 7 and an improved implementation of the Algol graph plotting routines made by Mr Froggatt.
The character set for plotting has been extended to include the characters: 39 ´ (acute), 64 ` (grave), 60 <, 62 >, 40 (, 41), 47 /, 91 [, 93], and 92 \. (This choice of symbols is a compromise between 903 and 900 telecode).

Operation of the 903 Algol 16K (LG) System (Appendix 4).

Three new entry points are available:-

23	Enable call-by-name extensions
24	Disable call-by-name extensions
25	Checksum store and lock.

Entry points 23 and 24 correspond to 17 and 18 for the 2-pass system translator. Entry point 25 provides session-to-session store integrity checking. On entry at 25 the entire store is sum checked and entry point 8 diverted to code that validates the checksum before starting a new translation. Thus if the computer is turned off after entry at 25 and then on again a alter time, translation will only be possible if the store has not been corrupted in the interim.

Use of Non-standard Peripheral Devices (Appendix 5)

Programmers should note that input device routines must copy the input character to the locations BUFFER+5 and 173 in addition to returning them in the accumulator as the result of the routine to ensure the correct behaviour of read statements and the instring procedure. Programmers should also note that all version of AJH Algol follow the 903 Algol Issue 6 interface, including the Load-and-Go version.

PERFORMANCE

Benchmarking has shown that AJH Algol runs at generally the same speed as 903 Algol, or at worst with a penalty of 1-3%.

For each of the versions AJH Algol has more space available for user programs and data: for the 2-pass system, an additional 50 words (approx.), for the Load-and-Go system, an additional 300 words (approx.) and for the Large Program system, an additional 190 words (approx.).

ERRORS CLEARED

The following corrections have been made to the interpreter code in all versions:

1. Corrected integer division, including for a divisor of -131072
2. Faster and shorter code for relational operations, with integer overflow detection.
3. Corrections to the Hunter call-by-name and recursion extensions.
4. Removal of redundant code to make more space for user programs and data.

ACKNOWLEDGEMENTS

Thanks are due to Mr Froggatt of Fleet, Hants and Mr Hunter of Elmdon, Essex for making their versions of Algol available for incorporation in AJH Algol.

QUERIES

Any queries relating to this release should be made on a software enquiry form to:-

Andrew Herbert
17 Latham Road
Cambridge CB2 7EG
Email: Andrew@Herbertfamily.org.uk

Release Authorized by: Andrew Herbert
Chief Author (AJH Algol)