

SWIFTE FORTRAN-77

Full ANSI-77
FORTRAN
For
MC68000
Microcomputers

Swifte FORTRAN-77

Overview

The programming language FORTRAN has been in existence for no less than twenty-five years, almost as long as the electronic computer itself, and during this period it has become the lingua franca of the community of scientific and engineering computer users. The reasons for this include its simplicity, availability, and portability. Although other languages like Pascal, BASIC, and APL do many things better than FORTRAN, the software investment in the latter means it will survive for quite some time to come.

Now a full implementation of the latest version of the language, ANSI-77 FORTRAN, is available for use on MC68000 based microcomputers running under MIRAGE, the powerful multi-user network operating system. Swifte FORTRAN-77 is compatible with most mainframe FORTRAN-77 compilers and also makes special provision to handle programs written in FORTRAN-66.

Using Swifte FORTRAN-77, time wasted debugging is dramatically reduced due to comprehensive compile-time and run-time error checking. When a run-time error occurs, a back-trace by subprogram name and statement number is produced, together with an English language error message, thus allowing bugs to be trapped and eliminated very quickly.

Swifte FORTRAN-77 is a true compiler, emitting assembly language which is assembled and linked to give MC68000 machine code. The code generated is fast, reentrant, relocatable, and reusable. Object programs call a separate, shareable, run-time support package in order to minimise program size and to conserve memory space when more than one user is running FORTRAN programs.

Naturally, Swifte FORTRAN-77 source, object, and program-generated files are completely compatible with the host operating system MIRAGE; additionally, since the FORTRAN system uses IEEE-standard Floating Point Arithmetic, data can be interchanged between different language processors. Routines written in FORTRAN can be called from Swifte-Pascal and APL.68000, thus allowing standard software, e.g. the NAG/IMSL library, to be used in several environments.

Swifte FORTRAN-77 is validated by the National Computing Centre, England, in association with the Federal Software Testing Centre, Washington, U.S.A., as being an error-free implementation of the Full FORTRAN language according to ANSI X3.9-1978; it is thus eligible for procurement under the terms of reference of the CCTA and GSA.

Note that there are no oppressive restrictions on either code or data size in Swifte FORTRAN-77 — either may be up to 16 Mbytes in size on 68000 based machines.



First and Foremost

ANSI-77 Standard Facilities

- * INTEGER data type — 4 bytes, range -2147483648 to +2147483647.
- * REAL data type — 4 bytes, IEEE format, 7 digits precision, range $10E-35$ to $10E+35$.
- * DOUBLE PRECISION data type — 8 bytes, IEEE format, 16 digits precision, range $10E-308$ to $10E+308$.
- * CHARACTER data type — up to 32000 elements; // concatenation operator supplied; user functions may return character strings.
- * PARAMETER statement — allows named constants to be defined.
- * Input-Output — a powerful set of statements to open, close, and enquire about units and files is provided.
- * List-Directed Input-Output — explicit FORMATS may be omitted.
- * Internal Files — text lines may be written to, or read from, character variables instead of disc files.
- * Array Bounds — may be expressions, and lower bounds can be other than unity.
- * Block IF, ELSE IF, ELSE, and END IF — these statements permit more structured programs to be created.
- * ENTRY statement — allows multiple entry points to a subroutine or function.
- * Generic Intrinsic Functions — where the argument type specifies which function is meant, e.g. LOG(x) implies ALOG(x), DLOG(x), if x is real, double precision.
- * Alternate RETURN — exit from SUBROUTINES may be to places other than after the call.
- * IMPLICIT statement — allows the default type rules to be overridden.
- * ANSI-66 compatibility mode.

Language Enhancements

- * Length Specification in Type Statements, e.g. INTEGER*1, INTEGER*2 denote signed byte and signed word integers.
- * Double Precision Complex — COMPLEX*16 type with full set of operators and intrinsic functions.
- * Identifiers may be up to 31 characters long and contain "\$" (dollar) and "_" (underscore), all characters being significant.
- * Based Constants — hexadecimal, octal and binary, integer constants supported, e.g. \$FF, 8\$377, 2\$11111111.

Program Development Facilities

- * Free-format source option — eases creation of program source text at a VDU.
- * Source file inclusion facility — copies in common declarations, code.
- * Separate compilation — allows subprograms to be compiled independently and combined together later using the standard MIRAGE linker.
- * Numerous compile-time toggles, set from either the command line or embedded in the source files, allow control of compiler operation, e.g. list file output or array bounds checking.

MIRAGE Environment Support

- * Full access to the advanced facilities of MIRAGE — e.g. Networking Filing System, Spooler, Time, and Date.
- * Interface to Assembly Language — allows special purpose or ultra-fast hand coded routines written in MC68000 Macro Assembly Language to be accessed.
- * Callable from other languages — FORTRAN programs, subroutines, and functions, can be called from Swifte-Pascal and APL.68000, e.g. NAG library functions can be called from inside APL application programs.
- * Interface to GKS — an ANSI-compatible FORTRAN binding to the Graphics Kernel System Bolton is provided.
- * NAG double precision mathematical software library is available.

Built-in Subroutines and Functions

Type Conversion:

INT, IFIX, IDINT, REAL, FLOAT, SNGL,
DREAL, DBLE, DFLOAT, CMPLX, DCMLX

Maximum/Minimum:

MAX, MAX0, AMAXI, DMAXI, AMAX0,
MAXI, MIN, MIN0, AMINI, DMINI, AMIN0,
MINI

Character Manipulation:

ICHAR, CHAR, LEN, INDEX, LGE, LGT,
LLE, LLT

Miscellaneous Mathematical:

ABS, IABS, DABS, CABS, CDABS, IMAG,
AIMAG, DIMAG, CONJG, DCONJG, AINT,
DINT, ANINT, DNINT, NINT, IDNINT, MOD,
AMOD, DMOD, DPROD, SQRT, DSQRT,
CSQRT, CDSQRT

Logarithm and Exponential:

LOG, ALOG, DLOG, CLOG, CDLOG, LOG10,
ALOG10, DLOG10, EXP, DEXP, CEXP,
CDEXP

Transcendentals:

SIN, DSIN, CSIN, CDSIN, COS, DCOS, CCOS,
CDCOS, TAN, DTAN, ASIN, DASIN, ACOS,
DACOS, ATAN, DATAN, ATAN2, DATAN2

Hyperbolics:

SINH, DSINH, COSH, DCOSH, TANH,
DTANH

Bit Manipulation:

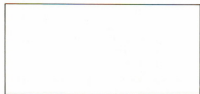
IAND, IOR, IEOR, NOT ISHFT, BTEST,
IBSET, IBCLR, ISHFTC, ICNT

MIRAGE Interface:

CHAIN, SPOOL, CRT, TIME, DATE, SLEEP,
GETCML, IGETCC, IGETCH

Heap Storage Management:

MEMAVS, MEMRQS, MEMRTS



Documentation

Swift Fortran-77 comes complete with a comprehensive User's Guide and Language Reference Manual. The documentation may be purchased separately from your supplier.

Hardware Requirements

The Swift Fortran-77 compiler requires a MC68000-based microcomputer running the MIRAGE Operating System (Version 1.2 or later) with at least 250 Kbytes of user memory and one disc drive. Generated programs require 20 Kbytes for the run-time support package (if not shared) plus the program's code and data space.

Notes

ANSI — American National Standards Institution

NCC — National Computing Centre

FTSC — Federal Software Testing Centre

GSA — General Services Administration

IEEE — Institute of Electrical and Electronic Engineers

ISO — International Standards Organisation

NAG — Numerical Algorithms Group

IMSL — International Mathematical Software Library

Swift Fortran-77, Swift-Pascal and the MIRAGE Operating System were written by Swift Computer Systems Limited and are exclusively marketed worldwide by Sahara Software Limited.

Copyright © 1984 Swift Computer Systems Limited.

SAHARA SOFTWARE LIMITED

UNIT 1F, TIDEWAY INDUSTRIAL
ESTATE, 87 KIRTLING ST.,
LONDON SW8 5BP.
Tel: 01-627 1733
Telex: 896885 IOTA