**PRODUCT NAME:** VAX-11 BLISS-32 Implementation Language, Version 2.0

**DESCRIPTION:**

BLISS-32 is a high level systems implementation language for VAX-11. BLISS-32 supports development of modular software according to structured programming concepts by providing an advanced set of language features. BLISS-32 also provides access to most of the hardware features of the VAX-11/780 to facilitate programming of time-critical and/or hardware-dependent applications. BLISS-32 is especially intended for the development of operating systems, compilers, run-time system components, data base file systems, communications software, and utilities, etc., for use on a VAX-11 hardware system.

The BLISS-32 compiler runs in native mode under the VAX/VMS Operating System. It translates BLISS-32 source programs into relocatable object modules that can be linked for execution. BLISS-32 compiled code is optimized for execution efficiency.

The following features of BLISS-32, while sometimes described in VAX-11 or VAX/VMS terminology, are machine-independent concepts. Collectively, this set of common features is known as "Common BLISS" and can be used in a transportable manner to develop programs for more than one computer architecture. Common BLISS features include:

- Separately compiled modules for modularity and convenient development. Object modules are relocatable and can be linked with object modules produced by the VAX-11 MACRO assembler or other native mode languages.
- BLISS-32 provides expressions for describing the actions to be performed and declarations for allocating, describing, and initializing data, for defining macros and literals, etc.
- BLISS-32 is "type-free": all data is manipulated as longwords or 32 bits. Interpretation of data is provided by operators.
- The operators provide a set of operations for integer arithmetic, for comparison, maximization, and minimization of signed integer, unsigned integer, and address values, and for Boolean operations.
- Field references allow values to be fetched from or assigned to any contiguous field of from 1 to 32 bits located anywhere in the VAX-11 virtual address space.
- Character sequence functions provide for efficient run-time manipulation of character data. Operations include moving, concatenating, comparing, and translating character sequences, as well as searching for particular characters or substrings of characters.
- IF, CASE, SELECT, and SELECTONE allow the choice of an expression or group of expressions to be executed based on programmed tests.
- DO, WHILE, and UNTIL allow general loops that cycle as long as a programmed test is satisfied. The test can be made at either the beginning or the end of the loop.
- INCR and DECR allow counted loops that execute a computed number of times under control of a loop variable.
- LEAVE allows early termination of a named lexically containing block and continuation after the named block. LEAVE may be considered a restricted form of forward-only GOTO; there is no general GOTO in BLISS-32.
- OWN and GLOBAL declarations provide static storage allocation; GLOBAL names are made available to the Linker and resolve EXTERNAL data declarations in other modules.
- LOCAL, STACKLOCAL, and REGISTER allow dynamic stack-like allocation using either the execution stack or the general registers.
- STRUCTURE declarations allow the programmed definition of arbitrary data structures in terms of an accessing algorithm for locating elements of a structure. Built-in declarations for VECTORS, BLOCKs, BITVECTORS, and BLOCKVECTORS provide for commonly needed structures.
- ROUTINE declarations provide subroutines or procedures in BLISS-32. Routines are recursive and reentrant, and can be linked in shareable images for use by multiple processes.
- REQUIRE declarations allow sources from other files to be automatically included in the module being compiled.
- LIBRARY declarations allow special compiler-produced binary declaration files to be included in the module being compiled. The effect is substantially the same as REQUIRE, but is more efficient because a restricted set of declarations can be precompiled into internal form.
- MACRO and KEYWORDMACRO declarations allow compile-time definition of both positional and keyword oriented macros. Macro definition and replacement are in terms of source lexical units.
lexemes (atoms, tokens) rather than character text. 
Macros and macro declarations can be nested 
and recursive.

- %IF, %THEN, %ELSE, and %FI allow conditional 
compilation of BLISS source depending on pro-
grammed compile-time tests. These can be used to 
control expansion of macros or in their own right.

- Lexical functions allow a variety of compile-time 
operations such as concatenation of strings, con-
struction of names, testing properties of macro 
parameters, testing compiler qualifiers, generating 
compiler diagnostic messages, and controlling macro expansion.

The following features of BLISS-32 are specialized for 
use on VAX-11. They provide precise means to tailor 
BLISS-32 programs to the special capabilities of the 
VAX-11 and VAX/VMS environment.

- LINKAGES declarations allow definition of special-
ized calling sequences for time critical or unusual 
applications. Options allow for use of 
calls/callg/ret or jsb/bsb/rsb type call 
and return instructions, for passing parameters in 
VAX-11 general registers or in parameter blocks, for 
controlling the preservation and use of registers by 
a routine, and for the sharing of registers across a 
set of routines as high speed common storage. 
Built-in linkage declarations for BLISS and FOR-
TRAN fully support the VAX-11 calling sequence 
conventions.

- PSELECT declarations allow use of link-time program 
sections for efficient use of the virtual address 
space. By default generated code sections are position 
independent.

- BUILTIN declarations allow use of VAX-11 
machine-specific functions for access to VAX-11 
features not otherwise provided by the BLISS-32 
language. Machine specific functions generally cor-
respond to VAX-11 instructions such as: ADAWI, 
BISPUSW, CRC, HALT, INDEX, MTPR, PROBER, 
RMQUE, etc. Over 50 such functions are provided.

- ENABLE declarations, together with SIGNAL, 
SIGNAL-STOP, and SETUNWIND functions, allow 
use of the VAX/VMS condition handling and error 
message reporting mechanisms.

The BLISS-32 compiler performs global and local 
obtization to produce efficient and compact gener-
ated code. Each routine definition is treated as a com-
plete unit in compiling the code for that routine. While 
industry terminology varies, the following generally 
characterizes the optimizations employed: common 
subexpression elimination, removing loop invariants, 
constant folding, block register allocation, peephole 
replacement, test instruction elimination, jump vs. 
branch instruction resolution, branch chaining, and 
cross-jumping.

The BLISS-32 compiler optionally produces a listing 
file that shows the source text compiled and the gener-
ated code in a format that closely resembles a VAX-11
Options with no support services are only available after the purchase of one supported license.

A single-use license only option is a license to copy the software previously obtained under license.

The following key (Y, Z) represents the distribution media for the product and must be specified at the end of the order number, e.g., QE106-AY = RX01 Floppy Diskette.

Y = RX01 Floppy Diskette
Z = No hardware dependency

QE106 -A— Single-use license binaries, documentation, support services (media: Y)

QE106 -D— Single-use license only, no binaries, no documentation, no support services (media: Z)

Miscellaneous Options
QE106 -G— Documentation only kit (media: Z)

Additional Services:
Post-warranty Software Product Services for this software product are available with the prerequisite being the purchase of the VAX/VMS Self-Maintenance Service for Software. Customers should contact their local DIGITAL office for additional information.