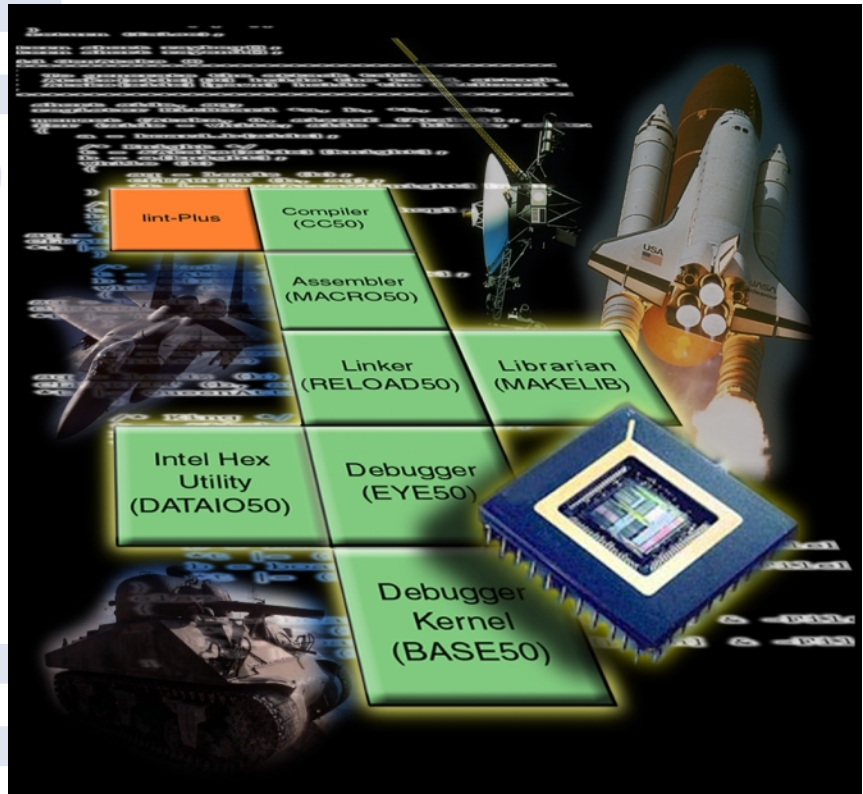


# Cleanscape XTC-1750A

“C”-language cross-development environment for MIL-STD-1750A systems

## Key Features

- Complete tool-set helps automate “C” language development for MIL-STD-1750A systems:
  - “C” compiler
  - Assembler
  - Loader/Linker
  - Libraries
  - Source code analyzer
  - Debugger
  - Intel Hex utility
- ANSI “C” prototype support
- Supports MIL-STD-1750A processors, including:
  - National Semiconductor F9450
  - Honeywell Generic VHSIC Spaceborne RH1750
  - Performance Semiconductor PACE 1750A
  - Marconi Electronic Devices MIL-STD-1750A
- Can be customized for 1750A prototype systems, like Honeywell PDU/SDU
- Supports Linux & Solaris operating platforms; others available on request
- Streamlined run-time library is easily modified for new platforms
- Bundled with Cleanscape lint-Plus source code analyzer



**Cleanscape XTC-1750A** is the first “C” and assembly language tool-set for MIL-STD-1750A system development. By allowing software engineers to transition from legacy programming languages to development in “C”, Cleanscape XTC-1750A helps developers to more easily find resources and talent, boost team productivity, improve system quality, and reduce development costs.

The Cleanscape XTC-1750A suite includes the following integrated tools for automating MIL-STD-1750A system development:

- Compiler (CC50) translates “C” code into assembly code.
- Assembler (MACRO50) translates assembly code into relocatable object files.
- Linker (RELOAD50) produces a downloadable program image from relocatable object files or libraries.
- Debugger (EYE50) debugs “C” assembly code at the source level through a downloadable kernel that works with generic target systems or prototype systems.

In addition, Cleanscape XTC-1750A now comes bundled with Cleanscape lint-Plus, a leading “C” source-code analyzer that automatically identifies problems standard compilers can’t detect.

The standard version supports generic MIL-STD-1750A target systems, like Fairchild SBC. Versions for 1750A prototype systems, like Honeywell PDU/SDU, are available on request. XTC-1750A runs under Solaris and Linux. Other platforms provided on request.

# Cleanscape XTC-1750A

## “C”-language cross-development environment for MIL-STD-1750A systems

### Summary

**Cleanscape XTC-1750A** is the first “C” and assembly language tool-set for MIL-STD-1750A system development. By allowing software engineers to transition from legacy programming languages to development in “C”, Cleanscape XTC-1750A helps developers to more easily find resources and talent, boost team productivity, improve system quality, and reduce development costs.

### Applications

Cleanscape XTC-1750A is designed for developing embedded systems for Aerospace navigation systems, Aerospace control and thrust management systems, weapon delivery systems, and other applications using embedded MIL-STD-1750A chips.

### Components

#### Compiler (CC50)

Cleanscape XTC-1750A Compiler translates “C” code into MIL-STD-1750A assembly language.

- ANSI prototype support
- Generates readable assembly output
- Can invoke the assembler to generate object code
- Can interface with assembly language routines

#### Assembler (MACRO50)

Cleanscape XTC-1750A Assembler is a two-pass assembler that translates assembly-language source files into relocatable binary object files.

- Supports user-defined macros, assembler directives, and MIL-STD-1750A instructions
- Directives may be used to define new machine instructions
- Supports the GPS 1750A XIO extensions
- Other extensions can be added, on request

#### Linker (RELOAD50)

Cleanscape XTC-1750A Linker accepts one or more relocatable object files or library files, and produces a downloadable program image.

- Produces relocatable code
- Produces ROMable code

#### Debugger (EYE50)

Cleanscape XTC-1750A Debugger is a command-line debugger that communicates over a serial line with a kernel running on the target system.

- Accesses modules at the “C” level or at the assembly level
- “Print” commands support expressions at both levels
- Break-point/single-step commands can be used to control execution at either level
- Configurable output formats
- Array-range expressions
- User-definable XIO commands
- Versions for development-system monitors are available on request

#### Cleanscape lint-Plus source code analyzer

Cleanscape lint-Plus is a source-code diagnostic tool that performs advanced pre-compile analysis on “C” code.

- Automates analysis and documentation of “C” programs
- Finds problems that standard compilers can't detect
- Analyzes “C” source files individually or as a group
- Maps program structure
- Finds unused functions, subroutines, variables, non-portable code

#### Other components:

- MAKELIB: Creates and updates relocatable binary library files
- BASE50: A debugger kernel that resides on the target board; allows the debugger to communicate with the MIL-STD-1750A processor; allows engineers to easily port to different hardware implementations
- Other utilities for converting an executable to various binary formats for the different in-circuit emulators supported by XTC-1750A

### Benefits

Using “C” makes MIL-STD-1750A system development faster, better, smarter, cheaper... *cleaner*:

- Easier access to talent, resources, and tools than for Ada and Jovial
- Reduced resource allocation
- Simplified management
- Boost productivity with a modern programming language
- Streamlined development process shortens development cycle

## The System In Action

- Programs are written in "C" or assembly language on a UNIX or Linux host system, and executed or debugged on a MIL-STD-1750A target system.
- Cleanscape lint-PLUS (an optional component) can be used to analyze "C" sources before compilation.
- XTC-1750A Compiler (CC50) is used to translate "C" code into 1750A assembly code.
- XTC-1750A Assembler (MACRO50) is used to translate assembly code into relocatable object files.
- XTC-1750A Linker (RELOAD50) is used to translate object files and/or library files into program images. The resulting images can be burned to EPROM or downloaded to the target system.
- To debug a downloaded program, the developer runs XTC-1750A Debugger (EYE50) on the host. The standard version of EYE50 communicates over a serial line with a kernel (BASE50) running on the target, such as Fairchild SBC. Versions of EYE50 that communicate with development-system monitors, like Honeywell PDU/SDU, are available on request.

## Features & Specifications

### Languages

- Pre-ANSI C with ANSI C prototype support

### Target processors

- National Semiconductor F9450
- Honeywell Generic VHSIC Spaceborne RH1750
- Honeywell PDU/SDU
- Performance Semiconductor PACE 1750A
- Marconi Electronic Devices MIL-STD-1750A
- Fairchild SBC

### Basic data types

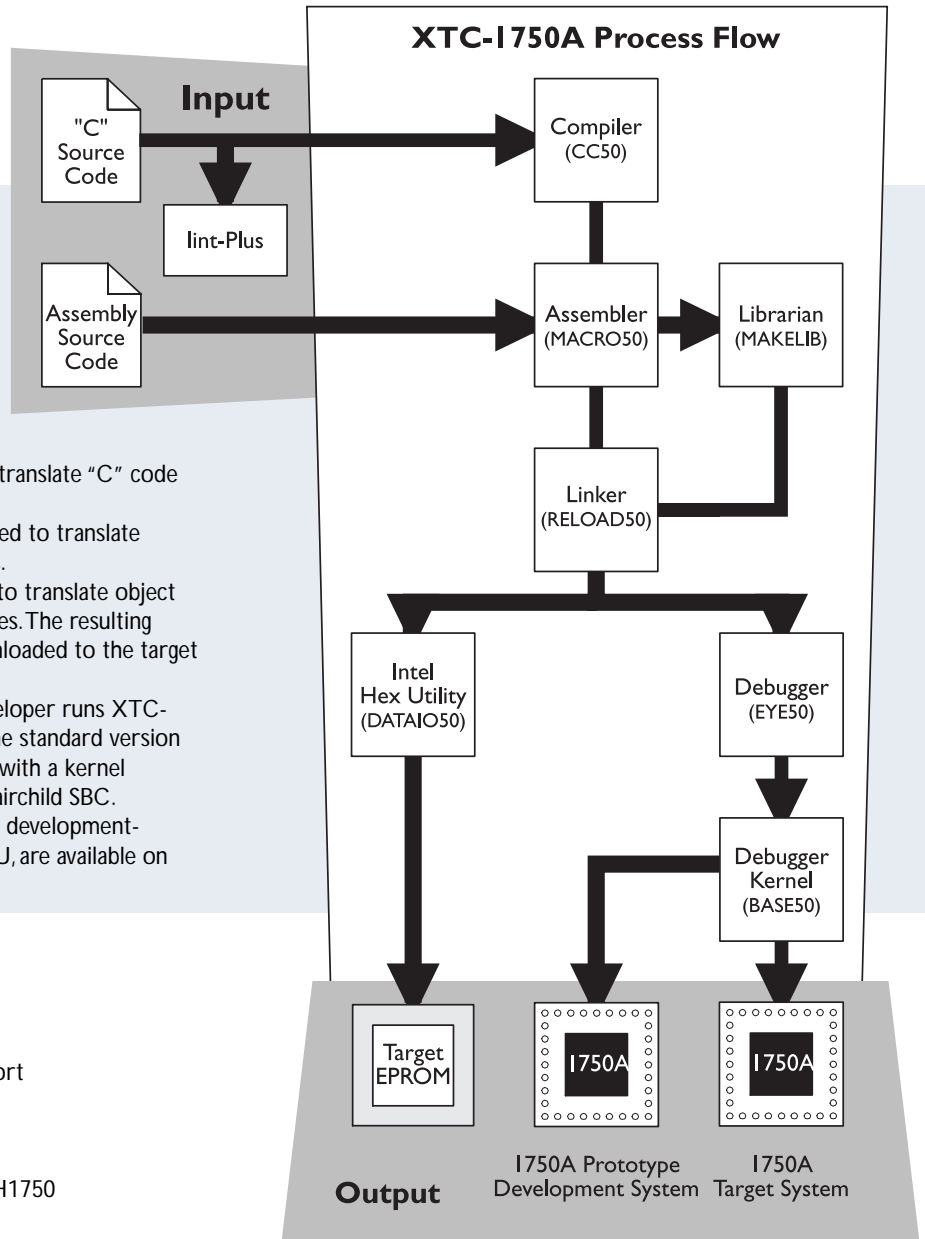
Data type	Description
char	unsigned 8-bit integer
short	signed 16-bit integer
long	signed 32-bit integer
int	same as "short"
float	32-bit MIL-STD-1750A floating-point value
double	48-bit MIL-STD-1750A floating-point value
enum	enumerated data type(s)
unsigned	same as "char"
unsigned short	unsigned 16-bit integer
unsigned long	unsigned 32-bit integer
unsigned int	same as "unsigned short"

### Additional data types

XTC-1750A also supports pointers, arrays, structures, and unions. These data types can be nested to three levels.

### Address space

- Storage unit: 16-bit word
- Addressing range: 64K words
- User-defined address states specify virtual address space, which is mapped into arbitrary blocks of physical memory
- Code and data mapped into the same 64K-word address space
- "C" used to access multiple address spaces through overlays



**Relocatable memory segments**

Segment	Description	Mapping
IREL	Run-time initialization code	ROM
KREL	Constants (excluding in-line constants)	ROM
NREL	Normal user code	ROM
SREL	Global and static variables	RAM
ZREL	General-purpose segment	RAM or ROM

**Notes on memory segments:**

- The XTC-1750A compiler generates most 16-bit numeric constants in-line using 1750A instructions.
- The IREL segment contains code that copies constants from the KREL segment to variables located in the SREL segment; the XTC-1750A program start-up module calls the IREL code at start-up time.

**Core library routines****General-purpose library routines**

atof	atoi	atol	index	isalnum	isalpha
isascii	iscntrl	isdigit	iseol	islower	isodigit
isprint	ispunct	isspace	isupper	isxdigit	itoa
longjmp	rindex	setjmp	strcat	strchr	strcmp
strcpy	strcspn	strlen	strncat	strncmp	strncpy
strpbrk	strrchr	strspn	toascii	tolower	toupper

**Math routines**

abs	acos	acosh	acot	acoth	acsc
acsch	asec	asin	asinh	atan	atan2
atanh	cbrt	ceil	cos	cosh	cot
coth	csc	csch	dtor	exp	fabs
floor	frac	hypot	ipow	log	log10
pow	powe	powi	rand	rtod	sec
sech	sin	sinh	sqrt	srand	tan
tanh					

**Memory allocation routines**

calloc	free	malloc	mflush	realloc
--------	------	--------	--------	---------

**Standard I/O routines**

fprintf	getc	getchar	getcnb	printf	putc
putchar	putcnb	scanf	sprintf		

**Port-Level I/O routines**

inb	outb
-----	------

- Additional routines provided on request

**Interrupt processing**

Interrupt handlers are written in assembly language; "C" subroutines may be used

**See how clean software development can be**

**Cleanscape** is a leading innovator of automated software development and testing solutions that simplify build environments for enabling customers to shorten development cycles, optimize resource utilization, increase product quality, and boost Return on Investment.

Cleanscape's product line includes the following automation tools for simplifying the software development process:

- **qef.** Team leaders can replace the *make* utility with this advanced software construction management system that generates complete and accurate builds.
- **ATTOL test automation tool kit.** Automated test generation, report generation, and coverage analysis for software components and systems.
- **FORTTRAN-lint & lint-Plus.** Static source code analyzers that can automatically identify problems that your compilers can't detect.
- **XTC-1750A.** The first "C" language cross-development environment for MIL-STD-1750A processors.
- **SNIP.** Bridges design and coding by automatically generating C++ classes from object models.



1530 Meridian Avenue Suite 200  
San Jose, CA 95125

408 978-7000 **Main**  
800 944-5468 **Sales**

[www.cleanscape.net](http://www.cleanscape.net)