

Mu Runtime Reference

version 0.2.14

type keywords and aliases

<i>supertype</i>	<i>T</i>
<i>bool</i>	<i>()</i> , <i>:nil</i> are false, otherwise true
<i>condition</i>	<i>keyword</i> , see exceptions
<i>list</i>	<i>:cons</i> or <i>()</i> , <i>:nil</i>
<i>ns</i>	<i>#\$(:ns #(:t fixnum symbol))</i>
<i>ns-designator</i>	<i>ns</i> , <i>:nil</i> , <i>:unqual</i>
<i>:null</i>	<i>()</i> , <i>:nil</i>
<i>:char</i>	<i>char</i>
<i>:cons</i>	<i>cons</i> , <i>list</i>
<i>:fixnum</i>	<i>fixnum</i> , <i>fix</i>
<i>:float</i>	<i>float</i> , <i>fl</i>
<i>:func</i>	<i>function</i> , <i>fn</i>
<i>:keyword</i>	<i>keyword</i> , <i>key</i>
<i>:stream</i>	<i>stream</i>
<i>:struct</i>	<i>struct</i>
<i>:symbol</i>	<i>symbol</i> , <i>sym</i>
<i>:vector</i>	<i>vector</i> , <i>string</i> , <i>str</i>
	<i>:bit</i> <i>:char</i> <i>:t</i>
	<i>:byte</i> <i>:fixnum</i> <i>:float</i>

core

<i>apply fn list</i>	<i>T</i>	apply <i>fn</i> to <i>list</i>
<i>compile form</i>	<i>T</i>	<i>mu</i> form compiler
<i>eq T T'</i>	<i>bool</i>	<i>T</i> and <i>T'</i> identical?
<i>eval form</i>	<i>T</i>	evaluate <i>form</i>
<i>type-of T</i>	<i>key</i>	type keyword
<i>view for</i>	<i>vector</i>	vector of object
<i>fix fn T</i>	<i>T</i>	fixpoint of <i>fn</i>
<i>gc</i>	<i>bool</i>	garbage collection
<i>repr T</i>	<i>vector</i>	tag representation
<i>unrepr vector</i>	<i>T</i>	tag representation

special forms

<i>:lambda list . list'</i>	<i>function</i>	anonymous <i>fn</i>
<i>:alambda list . list'</i>	<i>function</i>	anonymous <i>fn</i>
<i>:quote T</i>	<i>list</i>	quoted form
	<i>T</i>	conditional

vector is an 8 element *:byte* vector of little-endian argument tag bits.

frames			vectors		
		frame binding: <i>(fn . #(:t ...))</i>			
<i>%frame-stack</i>	<i>list</i>	active frames	<i>make-vector</i>	<i>key list</i>	specialized vector from list
<i>%frame-pop fn</i>	<i>frame</i>	pop function's top frame binding	<i>vector-length</i>	<i>vector</i>	length of vector
<i>%frame-push frame</i>	<i>cons</i>	push frame	<i>vector-type</i>	<i>vector</i>	type of vector
<i>%frame-ref fn fix</i>	<i>T</i>	function, offset	<i>sref</i>	<i>vector fix</i>	<i>n</i> th element
symbols			namespaces		
<i>boundp sym</i>	<i>bool</i>	is symbol bound?	runtime namespaces: <i>mu</i> (static), <i>keyword</i>		
<i>make-symbol string</i>	<i>sym</i>	uninterned symbol	<i>make-namespace</i>	<i>str</i>	make namespace
<i>symbol-namespace sym</i>	<i>ns-designator</i>	namespace designator	<i>namespace-name</i>	<i>ns</i>	namespace name
<i>symbol-name symbol</i>	<i>string</i>	name binding	<i>intern ns str value</i>	<i>symbol</i>	intern symbol in non-static namespace
<i>symbol-value symbol</i>	<i>T</i>	value binding	<i>find-namespace</i>	<i>str</i>	map string to namespace
fixnums			<i>find ns string</i>	<i>symbol</i>	map string to symbol
<i>add fix fix'</i>	<i>fixnum</i>	sum			
<i>ash fix fix'</i>	<i>fixnum</i>	arithmetic shift			
<i>div fix fix'</i>	<i>fixnum</i>	quotient			
<i>less-than fix fix'</i>	<i>bool</i>	<i>fix < fix'?</i>			
<i>logand fix fix'</i>	<i>fixnum</i>	bitwise and			
<i>lognot fix</i>	<i>fixnum</i>	bitwise complement			
<i>logor fix fix'</i>	<i>fixnum</i>	bitwise or			
<i>mul fix fix'</i>	<i>fixnum</i>	product			
<i>sub fix fix'</i>	<i>fixnum</i>	difference			
floats					
<i>fadd fl fl'</i>	<i>float</i>	sum	<i>*standard-input*</i>	<i>stream</i>	std input stream
<i>fdiv fl fl'</i>	<i>float</i>	quotient	<i>*standard-output*</i>	<i>stream</i>	std out stream
<i>fless-than fl fl'</i>	<i>bool</i>	<i>fl < fl'?</i>	<i>*error-output*</i>	<i>stream</i>	std error stream
<i>fmul fl fl'</i>	<i>float</i>	product	<i>open type dir str bool</i>	<i>stream</i>	open stream, raise error if bool
<i>fsub fl fl'</i>	<i>float</i>	difference			
conses/lists					
<i>append list</i>	<i>list</i>	append lists			
<i>car list</i>	<i>T</i>	head of <i>list</i>			
<i>cdr list</i>	<i>T</i>	tail of <i>list</i>			
<i>cons T T'</i>	<i>cons</i>	(<i>T</i> , <i>T'</i>)			
<i>length list</i>	<i>fixnum</i>	length of <i>list</i>			
<i>nth fix list</i>	<i>T</i>	<i>n</i> th car of <i>list</i>			
<i>nthcdr fix list</i>	<i>T</i>	<i>n</i> th cdr of <i>list</i>			
streams					
<i>*standard-input*</i>	<i>stream</i>				
<i>*standard-output*</i>	<i>stream</i>				
<i>*error-output*</i>	<i>stream</i>				
<i>open type dir str bool</i>	<i>stream</i>				
	<i>type</i>	<i>:file</i>	<i>:string</i>		
	<i>dir</i>	<i>:input</i>	<i>:output</i>	<i>:bidir</i>	
<i>close stream</i>	<i>bool</i>				
<i>openp stream</i>	<i>bool</i>				
<i>flush stream</i>	<i>bool</i>				
<i>get-string stream</i>	<i>string</i>				
<i>read-byte stream bool T</i>	<i>byte</i>				
streams					
<i>read-char stream bool T</i>	<i>char</i>				
<i>unread-char char stream char</i>					
<i>write-byte byte stream</i>	<i>byte</i>				
<i>write-char char stream</i>	<i>char</i>				
<i>read stream bool T</i>	<i>T</i>				
<i>write T bool stream</i>	<i>T</i>				

exceptions			environment		Reader Syntax
with-exception <i>fn fn'</i>	<i>T</i>	catch exception	JSON config format:	;	comment to end of line
		<i>fn - (:lambda (obj cond src) . body)</i>	{	# ... #	block comment
		<i>fn' - (:lambda () . body)</i>	"pages": <i>N</i> ,	'form	quoted form
raise <i>T keyword</i>		raise exception on <i>T</i> with <i>keyword</i> condition	"gc-mode": "none" "auto",	`form	backquoted form
raise-from <i>T symbol keyword</i>		raise exception on <i>T</i> with <i>keyword</i> condition	}	(...)	backquoted list (proper lists)
				,form	eval backquoted form
				,@form	eval-splice backquoted form
			Mu library API		
			<i>[dependencies]</i>		
			<i>mu = {</i>		
			<i>git</i> = " https://github.com/Software-Knife-and-Tool/mu.git ",	(...)	constant list
			<i>branch</i> = "main"	()	empty list, prints as :nil
			<i>}</i>	(... . .)	dotted list
				"..."	string, char vector
					single escape in strings
			<i>use mu::{ Condition, Core, Env, Exception,</i>	<i>ns:name</i>	qualified symbol, where <i>ns</i> and <i>name</i> are symbol constituents
			<i>Mu, Result, Tag }</i> ;	<i>name</i>	lexical symbol
			<i>impl Mu {</i>		
			<i>fn apply(_ : &Env, _ : Tag, _ : Tag) -> Result<Tag></i>		
			<i>fn compile(_ : &Env, _ : Tag) -> Result<Tag></i>		
			<i>fn config(_ : Option<String>) -> Option<Config></i>		
			<i>fn core() -> &Core</i>		
			<i>fn eq(_ : Tag, _ : Tag) -> bool</i>	#*	bit vector
			<i>fn err_out() -> Tag</i>	#X	hexadecimal fixnum
			<i>fn eval_str(_ : &Env, _ : &str) -> Result<Tag></i>	#.	read-time eval
			<i>fn eval(_ : &Env, _ : Tag) -> Result<Tag></i>	#\	char
			<i>fn exception_string(_ : &Env, _ : Exception) -> String</i>	(:type ...)	vector
			<i>fn lo8_ptad(_ : &Env, _ : &str) -> Result<bool></i>	#\$(:type ...)	struct
			<i>fn make_env(_ : &Config) -> Env</i>	#::...	uninterned symbol
			<i>fn read_str(_ : &Env, _ : &str) -> Result<Tag></i>	"` , ;	terminating macro char
			<i>fn read(_ : &Env, _ : Tag, _ : bool, _ : Tag) -> Result<Tag></i>	#	non-terminating macro char
			<i>fn std_in() -> Tag</i>		
			<i>fn std_out() -> Tag</i>		
			<i>fn version() -> &str</i>		
			<i>fn write_str(_ : &Env, _ : &str, _ : Tag) -> Result<()></i>	!\$%&*- .	
			<i>fn write_to_string(_ : &Env, _ : Tag, _ : bool) -> String</i>	<>=?@[]	
			<i>fn write(_ : &Env, _ : Tag, _ : bool, _ : Tag) -> Result<()></i>	:^_{ }~/	
			}	A..Za..z	
				0..9	
				0x09 #\tab	character designators
				0x0a #\linefeed	
				0x0c #\page	
				0x0d #\return	
				0x20 #\space	
			Features		
			<i>[features]</i>		
			<i>default = ["core", "env", "system"]</i>		
feature/core	<i>core</i>	<i>list</i>	core state		
	<i>delay</i>	<i>fixnum</i>	microseconds		
	<i>process-mem-virt</i>	<i>fixnum</i>	vmem		
	<i>process-mem-res</i>	<i>fixnum</i>	reserve		
	<i>process-time</i>	<i>fixnum</i>	microseconds		
	<i>time-units-per-sec</i>	<i>fixnum</i>			
	<i>ns-symbols ns :nil</i>				
		<i>list</i>	<i>symbol list</i>		
feature/env	<i>env</i>	<i>list</i>	env state		
	<i>heap-info</i>	<i>()</i>	heap info to		
	<i>heap-room</i> <i>key</i>	<i>vector</i>	stdout		
		<i>#(:t size total free ...)</i>	allocations		
	<i>heap-size</i> <i>key</i>	<i>fixnum</i>	type size		
	<i>cache-room</i>	<i>vector</i>	allocations		
		<i>#(:t size total ...)</i>			
feature/system	<i>uname</i>	<i>:t</i>	system info		
	<i>shell string list</i>	<i>fixnum</i>	shell command		
	<i>exit</i> <i>fixnum</i>				
	<i>sysinfo</i>	<i>:t</i>	not on macOS		
feature/instrument					
	<i>instrument-control</i> <i>key</i>				
		<i>:on :off :get</i>			
		<i>key vec</i>			