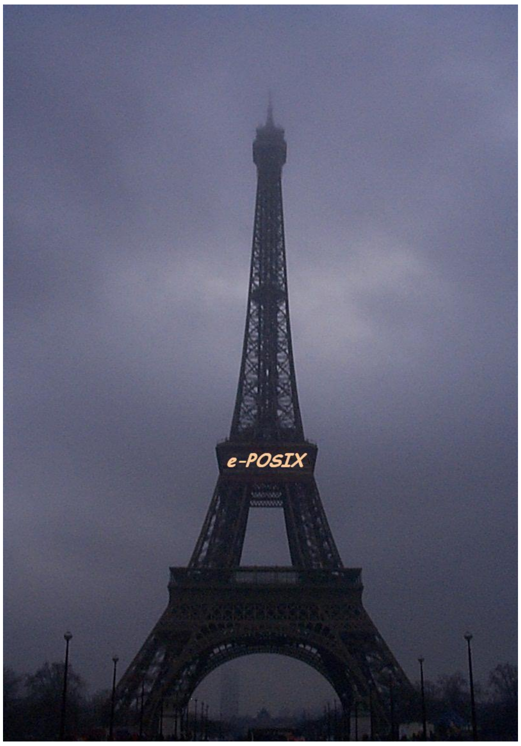

e-POSIX

eposix short-flat listing of classes

written by Berend de Boer

Contents

In this chapter:

A
Short (flat) listing of Standard C classes

A.1 Short form of STDC_BASE

```
class interface STDC_BASE
feature(s) from STDC_BASE
    -- errno
    errno: STDC_ERRNO
        -- Access to the variable that contains the error that occurred.
invariant
    accessing_real_singleton: security_is_real_singleton;
end of STDC_BASE
```

A.2 Short form of STDC_BUFFER

```
class interface STDC_BUFFER
creation
    allocate (a_capacity: INTEGER)
        -- Allocate memory of a_capacity bytes.
        -- If is_owner then the buffer is first deallocated.
    allocate_and_clear (a_capacity: INTEGER)
        -- Allocate memory of a_capacity bytes, make sure its zeroed out.
        -- If is_owner then the buffer is first deallocated.
    make_from_pointer (a_pointer: POINTER; a_capacity: INTEGER; a_become_owner: BOOLEAN)
        -- Attach a pointer to this object. If a_become_owner is
        -- True, it will deallocate the pointer when close is
        -- called, or when this object is garbage collected.
feature(s) from STDC_BUFFER
    -- Allocation
    allocate (a_capacity: INTEGER)
        -- Allocate memory of a_capacity bytes.
        -- If is_owner then the buffer is first deallocated.
    allocate_and_clear (a_capacity: INTEGER)
        -- Allocate memory of a_capacity bytes, make sure its zeroed out.
        -- If is_owner then the buffer is first deallocated.
    make_from_pointer (a_pointer: POINTER; a_capacity: INTEGER; a_become_owner: BOOLEAN)
        -- Attach a pointer to this object. If a_become_owner is
        -- True, it will deallocate the pointer when close is
        -- called, or when this object is garbage collected.
feature(s) from STDC_BUFFER
    -- Other allocation commands
    resize (new_capacity: INTEGER)
        -- Resize memory to new_capacity bytes. Expanded memory is not
        -- guaranteed to be zeroed out.
feature(s) from STDC_BUFFER
    -- Access
    resource_usage_can_be_increased: BOOLEAN
        -- Can the number of allocated resources increased with capacity?
feature(s) from STDC_BUFFER
    -- Copy data internally or externally
    copy_from (source: STDC_BUFFER; src_offset, dest_offset, bytes: INTEGER)
        -- Move data from another buffer into ourselves.
        -- Start at offset src_offset, into
        -- offset dest_offset, moving bytes bytes
        -- Memory may overlap.
    memory_copy (source: POINTER; src_offset: INTEGER; dest_offset, bytes: INTEGER)
        -- Copy data from source, offset src_offset, to location
        -- dest_offset in this buffer, for bytes bytes.
        -- Memory may not overlap, use move to copy within buffer
        -- or memory_move to copy from potentially overlapping buffer.
    memory_move (source: POINTER; src_offset: INTEGER; dest_offset, bytes: INTEGER)
        -- Copy data from source, offset src_offset, to location
        -- dest_offset in this buffer, for bytes bytes.
        -- Memory may overlap.
    move (src_offset, dest_offset: INTEGER; bytes: INTEGER)
        -- Move data around in buffer itself from offset src_offset to
        -- offset dest_offset, moving bytes bytes.
        -- Memory may overlap.
feature(s) from STDC_BUFFER
    -- Set/get bytes (8-bit data)
    peek_uint8 (index: INTEGER): INTEGER
        -- consider memory an array of 8 bit values.
    infix "@" (index: INTEGER): INTEGER
        -- consider memory an array of 8 bit values.
    poke_uint8 (index, value: INTEGER)
    peek_int8 (index: INTEGER): INTEGER
        -- consider memory an array of 8 bit values.
    poke_int8 (index, value: INTEGER)
feature(s) from STDC_BUFFER
    -- Set/get integers (16-bit data)
    peek_int16 (index: INTEGER): INTEGER
        -- Read signed 16 bit value at offset index in native
        -- endian format.
    peek_int16_native (index: INTEGER): INTEGER
        -- Read signed 16 bit value at offset index in native
        -- endian format.
    peek_uint16 (index: INTEGER): INTEGER
        -- Read unsigned 16 bit value at offset index in native format.
    peek_uint16_native (index: INTEGER): INTEGER
        -- Read unsigned 16 bit value at offset index in native format.
    peek_int16_big_endian (index: INTEGER): INTEGER
        -- Read 16 bit value at offset index in big endian format.
    peek_int16_little_endian (index: INTEGER): INTEGER
        -- Read 16 bit value at offset index in little endian format.
    poke_int16 (index: INTEGER; value: INTEGER)
        -- Write 16 bit value at offset index, in native endian format.
    poke_int16_native (index: INTEGER; value: INTEGER)
        -- Write 16 bit value at offset index, in native endian format.
    poke_int16_big_endian (index: INTEGER; value: INTEGER)
        -- Write 16 bit value at offset index, in big endian format.
    poke_int16_little_endian (index: INTEGER; value: INTEGER)
        -- Write 16 bit value at offset index, in little endian format.
feature(s) from STDC_BUFFER
    -- Set/get integers (32-bit data)
    peek_int32_native (index: INTEGER): INTEGER
        -- Read 32 bit value at offset index, assume its byte order
        -- is native, and return it.
    peek_integer (index: INTEGER): INTEGER
        -- Read 32 bit value at offset index, assume its byte order
        -- is native, and return it.
    peek_int32_big_endian (index: INTEGER): INTEGER
        -- Read 32 bit value at offset index, assume its byte order
        -- is big endian, and return it in native format.
    peek_int32_little_endian (index: INTEGER): INTEGER
        -- Read 32 bit value at offset index, assume its byte order
        -- is little endian, and return it in native format.
    peek_uint32_native (index: INTEGER): INTEGER
        -- Read 32 bit unsigned int at offset index, assume native
        -- byte order.
    peek_uint32_big_endian (index: INTEGER): INTEGER
        -- Read 32 bit unsigned int at offset index, assume its
        -- byte order is big endian, and return it in native format.
    peek_uint32_little_endian (index: INTEGER): INTEGER
        -- Read 32 bit unsigned int at offset index, assume its
        -- byte order is big endian, and return it in native format.
    poke_integer (index: INTEGER; value: INTEGER)
        -- Write 32 bit value at offset index, in native endian format.
    poke_int32_native (index: INTEGER; value: INTEGER)
        -- Write 32 bit value at offset index, in native endian format.
    poke_int32_big_endian (index: INTEGER; value: INTEGER)
        -- Write 32 bit value at offset index, in big endian format.
    poke_int32_little_endian (index: INTEGER; value: INTEGER)
        -- Write 32 bit value at offset index, in little endian format.
feature(s) from STDC_BUFFER
    -- Set/get characters
    append_to_string (dest: STRING; start_index, end_index: INTEGER)
        -- Append all characters from start_index to end_index
        -- inclusive to dest.
    peek_character (index: INTEGER): CHARACTER
        -- Return value at index as an 8-bit character.
    poke_character (index: INTEGER; value: CHARACTER)
        -- Set character at index index to value.
    put_to_string (dest: STRING; pos, start_index, end_index: INTEGER)
        -- Put characters from start_index to end_index inclusive
        -- in dest starting at position pos.
        -- Useful for Gobo character buffers.
    c_substring_with_string (dest: STRING; start_index, end_index: INTEGER)
        -- As c_substring but used dest as the destination.
    c_substring (start_index, end_index: INTEGER): STRING
        -- Create a substring containing all characters from
        -- start_index up to encountering a %U or when end_index is
        -- reached, whatever happens first.
    substring (start_index, end_index: INTEGER): STRING
        -- Create a substring containing all characters
```

```
		-- from start_index to end_index inclusive.
feature(s) from STDC_BUFFER
	-- Fill
	fill_at (start_index, a_count: INTEGER; byte: INTEGER)
		-- Starting at position start_index, write byte for a_count bytes
feature(s) from STDC_BUFFER
	-- Searching
	locate_character (other: CHARACTER; start_index: INTEGER): INTEGER
		-- Return index of other in buffer, or -1.
		-- Search begins at start_index.
	locate_string (other: STRING; start_index: INTEGER): INTEGER
		-- Does buffer contain other?
		-- Returns index where other is found.
		-- Returns -1 if not found
		-- searching starts at position start_index
feature(s) from STDC_BUFFER
	-- Queries
	is_valid_index (index: INTEGER): BOOLEAN
	is_valid_range (from_index, to_index: INTEGER): BOOLEAN
		-- Is from_index..to_index a valid and meaningfull range?
feature(s) from STDC_BUFFER
	-- Low level handle functions
	do_close: BOOLEAN
		-- Close resource, return error if any, or zero on
		-- success. This routine may never call another object, else
		-- it cannot be used safely in dispose.
	unassigned_value: POINTER
		-- The value that indicates that handle is unassigned.
invariant
	accessing_real_singleton: security_is_real_singleton;
	capacity_not_negative: capacity >= 0;
	valid_capacity: is_allocated = (capacity > 0);
	open_implies_handle_assigned: is_allocated = (ptr /= unassigned_value);
	owned_implies_open: is_owner implies is_allocated;
	owned_implies_handle_assigned: is_owner implies ptr /= unassigned_value;
end of STDC_BUFFER
```

A.3 Short form of STDC_CONSTANTS

```
class interface STDC_CONSTANTS
feature(s) from STDC_CONSTANTS
    -- Error codes
    edom: INTEGER
        -- Math argument out of domain of function
    erange: INTEGER
        -- Math result not representable
    emfile: INTEGER
        -- Too many open files
feature(s) from STDC_CONSTANTS
    -- Standard streams
    stream_stdin: POINTER
    stream_stdout: POINTER
    stream_stderr: POINTER
feature(s) from STDC_CONSTANTS
    -- Special characters
    const_eof: INTEGER
        -- signals EOF
feature(s) from STDC_CONSTANTS
    -- I/O buffering
    iofbf: INTEGER
        -- full buffering
    iolbf: INTEGER
        -- line buffering
    ionbf: INTEGER
        -- no buffering
feature(s) from STDC_CONSTANTS
    -- file positioning
    seek_set: INTEGER
    seek_cur: INTEGER
    seek_end: INTEGER
feature(s) from STDC_CONSTANTS
    -- Signal related constants
    sig_dfl: POINTER
    sig_err: POINTER
    sig_ign: POINTER
feature(s) from STDC_CONSTANTS
    -- Signals
    sigabrt: INTEGER
    sigfpe: INTEGER
        -- erroneous arithmetic operation, such as zero divide or an
        -- operation resulting in overflow
    sigill: INTEGER
        -- illegal instruction
    sigint: INTEGER
```

```
        -- receipt of an interactive attention signal
    sigsegv: INTEGER
        -- invalid access to storage
    sigterm: INTEGER
feature(s) from STDC_CONSTANTS
    -- random numbers
    rand_max: INTEGER
        -- maximum value returned by the random function
feature(s) from STDC_CONSTANTS
    -- category constants
    lc_ctype: INTEGER
    lc_numeric: INTEGER
    lc_time: INTEGER
    lc_collate: INTEGER
    lc_monetary: INTEGER
    lc_all: INTEGER
feature(s) from STDC_CONSTANTS
    -- various
    clocks_per_sec: INTEGER
feature(s) from STDC_CONSTANTS
    -- exit codes
    exit_failure: INTEGER
        -- exit status when something has gone wrong
    exit_success: INTEGER
        -- exit status upon success
end of STDC_CONSTANTS
```

A.4 Short form of STDC_CURRENT_PROCESS

```
class interface STDC_CURRENT_PROCESS
feature(s) from STDC_SECURITY_ACCESSOR
   -- The singleton, available to any because its used in preconditions
   security: STDC_SECURITY
      -- Singleton entry point for security.
feature(s) from STDC_BASE
   -- errno
   errno: STDC_ERRNO
      -- Access to the variable that contains the error that occurred.
feature(s) from STDC_CURRENT_PROCESS
   -- My standard input/output/error
   stdin: STDC_TEXT_FILE
   stdout: STDC_TEXT_FILE
   stderr: STDC_TEXT_FILE
feature(s) from STDC_CURRENT_PROCESS
   -- various
   clock: INTEGER
      -- return approximation of processor time used by the
      -- program, or -1 if unknown
feature(s) from STDC_CURRENT_PROCESS
   -- Random numbers
   random: INTEGER
      -- Returns a pseudo-random integer between 0 and RAND_MAX.
   set_random_seed (a_seed: INTEGER)
      -- Sets a_seed as the seed for a new sequence of
      -- pseudo-random integers to be returned by random. These
      -- sequences are repeatable by calling set_random_seed with
      -- the same seed value. If no seed value is provided, the
      -- random function is automatically seeded with a value of
      -- 1.
invariant
   accessing_real_singleton: security_is_real_singleton;
end of STDC_CURRENT_PROCESS
```

A.5 Short form of STDC_ENV_VAR

```
class interface STDC_ENV_VAR
creation
    make (a_name: STRING)
feature(s) from STDC_ENV_VAR
    -- Initialization
    make (a_name: STRING)
feature(s) from STDC_ENV_VAR
    -- Access
    exist: BOOLEAN
        -- Is this environment variable defined?
    name: STRING
        -- Name of environment variable.
    value: STRING
        -- Current value of environment variable.
invariant
    accessing_real_singleton: security_is_real_singleton;
end of STDC_ENV_VAR
```

A.6 Short form of STDC_FILE

STDC_FILE is a deferred class. Use STDC_TEXT_FILE for accessing and creating text files, or STDC_BINARY_FILE for binary files.

```
deferred class interface STDC_FILE
feature(s) from STDC_FILE
    -- Initialization
    create_read_write (path: STRING)
        -- Open file for update (reading and writing). If the file
        -- already exists, it is truncated to zero length.
        -- So permissions seem to remain.
    create_write (path: STRING)
        -- create new file for writing. If the file already exists,
        -- it is truncated to zero length.
        -- So permissions seem to remain.
    open (path, a_mode: STRING)
        -- open file in given mode
    open_append (path: STRING)
        -- Append to exiting file or create file if it does not exist.
    open_read (path: STRING)
        -- open file for reading
    open_read_write (path: STRING)
        -- Open file for reading and writing.
feature(s) from STDC_FILE
    -- Work with existing streams
    attach_to_stream (a_stream: POINTER; a_mode: STRING)
        -- Attach to a_stream. Does not become owner of stream so
        -- it will not close on close or when garbage collected.
feature(s) from STDC_FILE
    -- Close
    detach
        -- Forget the resource. Resource is not closed.
        -- You cannot read and write anymore.
feature(s) from STDC_FILE
    -- Reopen
    reopen (path, a_mode: STRING)
        -- Closes and then opens a stream.
feature(s) from STDC_FILE
    -- Control over buffering
    flush
        -- Updates this stream
    setbuf (buffer: POINTER)
        -- Determines how the stream will be buffered
        -- gives you a fully buffered input and output.
        -- Not sure: buffer should have at least BUFSIZ bytes?
        -- No operation should yet been performed on this file
        -- buffer = default_pointer: default buffer will be allocated
```

```
            -- buffer /= default_pointer implies buffer size = BUFSIZ
      set_buffer (buffer: POINTER)
            -- Determines how the stream will be buffered
            -- gives you a fully buffered input and output.
            -- Not sure: buffer should have at least BUFSIZ bytes?
            -- No operation should yet been performed on this file
            -- buffer = default_pointer: default buffer will be allocated
            -- buffer /= default_pointer implies buffer size = BUFSIZ
      set_full_buffering (buffer: POINTER; size: INTEGER)
            -- Determines buffering for a stream.
            -- If buffer is default_pointer, a buffer of size bytes
            -- will be allocated by this routine.
      set_line_buffering (buffer: POINTER; size: INTEGER)
            -- Determines buffering for a stream.
            -- Give NULL buffer so setvbuf will allocate a buffer.
      set_no_buffering
            -- Turn buffering off.
feature(s) from STDC_FILE
      -- read, C like
      last_byte: INTEGER
            -- Last read character of get_character.
            -- Can be negative, so is more a last_shortint or so!
      getc
            -- Reads a C unsigned char and converts it to an integer,
            -- the result is left in last_byte.
            -- This function probably can be used to read a single
            -- byte.
      get_character
            -- Reads a C unsigned char and converts it to an integer,
            -- the result is left in last_byte.
            -- This function probably can be used to read a single
            -- byte.
      read (buf: POINTER; offset, bytes: INTEGER)
            -- Read chunk, set last_read. offset determines how far
            -- in buf you want to start writing.
feature(s) from STDC_FILE
      -- Write, C like
      putc (c: INTEGER)
            -- Write a single character.
      write (buf: POINTER; offset, bytes: INTEGER)
            -- write bytes bytes from buf at offset offset
            -- we do not really care if offset is positive or negative...
feature(s) from STDC_FILE
      -- read, Eiffel like
      last_boolean: BOOLEAN
            -- last boolean read by read_boolean
      last_character: CHARACTER
```

```
          -- last character read by read_character
     last_double: DOUBLE
          -- last double lread by read_double
     last_integer: INTEGER
     last_real: REAL
          -- last real read by read_real
     read_boolean
          -- Attempt to read back a boolean written by write_boolean.
     read_buffer (buf: STDC_BUFFER; offset, bytes: INTEGER)
          -- More safe version of read in case you have a
          -- STDC_BUFFER object. Read starts at offset bytes in buf.
          -- Check last_read for number of bytes actually read.
     read_double
     read_character
          -- Read a single character and set last_character.
          -- If end-of-file encountered, eof is True.
     read_integer
     read_real
     read_string (nb: INTEGER)
          -- Read at most nb characters from input stream.
          -- Make the characters that have actually been read
          -- available in last_string.
          -- The input stream should not contain %U characters.
feature(s) from STDC_FILE
     -- write, Eiffel like
     put (any: ANY)
          -- Write object as string.
     put_buffer (buf: STDC_BUFFER; offset, bytes: INTEGER)
          -- more safe version of write in case you have a
          -- STDC_BUFFER object
          -- Check last_written for number of bytes actually written,
          -- if you use asynchronous writing.
     write_buffer (buf: STDC_BUFFER; offset, bytes: INTEGER)
          -- more safe version of write in case you have a
          -- STDC_BUFFER object
          -- Check last_written for number of bytes actually written,
          -- if you use asynchronous writing.
     put_boolean (b: BOOLEAN)
          -- Write "True" to output stream if
          -- b is true, "False" otherwise.
     write_boolean (b: BOOLEAN)
     write_character (c: CHARACTER)
          -- Write a single character.
     put_double (d: DOUBLE)
          -- Write a double in Standard C %f format.
     write_double (d: DOUBLE)
          -- Write a double in Standard C %f format.
```

```
        put_integer (i: INTEGER)
            -- Write an integer in Standard C %d format.
        write_integer (i: INTEGER)
            -- Write an integer in Standard C %d format.
        put_real (r: REAL)
            -- Write a real in Standard C %f format.
        write_real (r: REAL)
            -- Write a real in Standard C %f format.
        put_string (a_string: STRING)
            -- Write a string. a_string should not
            -- contain the null character.
        write_string (s: STRING)
        puts (s: STRING)
    feature(s) from STDC_FILE
        -- Unreading
        ungetc (c: INTEGER)
            -- Pushes c back to the stream. Only one push back is guaranteed.
            -- Note that file positioning functions discard any
            -- pushed-back characters.
        unread_character (an_item: CHARACTER)
            -- Put an_item back in input stream. Only one push back is
            -- guaranteed.
            -- This item will be read first by the next
            -- call to a read routine.
            -- Note that file positioning functions discard any
            -- pushed-back characters.
    feature(s) from STDC_FILE
        -- File position
        get_position: STDC_FILE_POSITION
            -- Get the current position. Use set_position to return to
            -- this saved position
        rewind
            -- Sets the file position to the beginning of the file.
        seek (offset: INTEGER)
            -- Set file position to given absolute offset.
        seek_from_current (offset: INTEGER)
            -- Set file position relative to current position.
        seek_from_end (offset: INTEGER)
            -- Set file position relative to end of file.
        set_position (a_position: STDC_FILE_POSITION)
            -- Set the current file position.
        tell: INTEGER
            -- The current position.
    feature(s) from STDC_FILE
        -- Other
        clearerr
            -- Clears end-of-file and error indicators for a stream.
```

```
   clear_error
      -- Clears end-of-file and error indicators for a stream.
feature(s) from STDC_FILE
   -- Status report
   eof: BOOLEAN
      -- Is eof encountered by getc or is the end-of-file indicator
      -- is set?
   error: BOOLEAN
      -- Is the error indicator is set?
   resource_usage_can_be_increased: BOOLEAN
      -- Is it allowed to open another file?
feature(s) from STDC_FILE
   -- Access
   filename: STRING
      -- The filename of this file.
   mode: STRING
      -- Mode in which the file is opened/created.
feature(s) from STDC_FILE
   -- is mode binary or text
   is_binary_mode_specification (a_mode: STRING): BOOLEAN
      -- Is the last character of a_mode equal to b?
   is_text_mode_specification (a_mode: STRING): BOOLEAN
      -- Is the last character of a_mode equal to t?
invariant
   open_in_sync: is_open_read or is_open_write implies is_open; -- The reverse is not true, for examples sockets
 -- closed for reading/writing, but still open.
   accessing_real_singleton: security_is_real_singleton;
   capacity_not_negative: capacity >= 0;
   valid_capacity: is_open = (capacity > 0);
   open_implies_handle_assigned: is_open = (stream /= unassigned_value);
   owned_implies_open: is_owner implies is_open;
   owned_implies_handle_assigned: is_owner implies stream /= unassigned_value;
   last_string_valid: last_string /= Void;
   gets_buf_valid: gets_buf /= Void;
end of deferred STDC_FILE
```

A.7 Short form of STDC_FILE_SYSTEM

```
class interface STDC_FILE_SYSTEM
feature(s) from STDC_FILE_SYSTEM
    -- Path names
    expand_path (a_path: STRING): STDC_PATH
        -- returns a new path
feature(s) from STDC_FILE_SYSTEM
    -- Rename files/directories, remove files/directories
    remove_file (a_path: STRING)
        -- Removes a file from a directory.
        -- For Standard C, its implementation defined what
        -- remove_file does if file is opened by some process
        -- (remove_file fails on Windows for example).
        -- doesnt remove a directory.
    rename_to (current_path, new_path: STRING)
        -- Rename a file or a directory.
        -- new_path should not be an existing path.
feature(s) from STDC_FILE_SYSTEM
    -- Accessibility of files
    is_modifiable (a_path: STRING): BOOLEAN
        -- Is a_path readable and writable by this program?
        -- Does this by attemting to open a_path file read/write.
    is_readable (a_path: STRING): BOOLEAN
        -- Is a_path readable by this program?
        -- Does this by attemting to open a_path file read-only.
invariant
    accessing_real_singleton: security_is_real_singleton;
end of STDC_FILE_SYSTEM
```

A.8 Short form of STDC_SECURITY

```
class interface STDC_SECURITY
feature(s) from STDC_SECURITY
    -- Modes
    make_allow_all
        -- Just allow everything.
    make_allow_sandbox
        -- Allow very little, use for setuid root programs.
feature(s) from STDC_SECURITY
    -- The security aspects
    cpu: STDC_SECURITY_CPU
    error_handling: STDC_SECURITY_ERROR_HANDLING
    files: STDC_SECURITY_FILES
    memory: STDC_SECURITY_MEMORY
feature(s) from STDC_SECURITY
    -- Various
    assert_once_memory_allocated
        -- Make sure that certain once functions in STDC_BASE are
        -- called. These once functions are called when an error
        -- occurs, at that time there might not be memory left to
        -- create them.
invariant
    accessing_real_singleton: security_is_real_singleton;
    remain_single: Current = singleton;
end of STDC_SECURITY
```

A.9 Short form of STDC_SIGNAL

```
class interface STDC_SIGNAL
creation
    make (a_value: INTEGER)
feature(s) from STDC_SIGNAL
    -- creation
    make (a_value: INTEGER)
feature(s) from STDC_SIGNAL
    -- set signal properties, make effective with apply
    apply
        -- Make changes effective.
    set_default_action
        -- Install signal-specific default action.
        -- Call apply to make changes effective.
    set_ignore_action
        -- Set action to ignore signal.
        -- Call apply to make changes effective.
    set_handler (a_handler: STDC_SIGNAL_HANDLER)
        -- Install ones own signal handler.
feature(s) from STDC_SIGNAL
    -- signal functions
    raise
        -- raise the signal
feature(s) from STDC_SIGNAL
    -- signal state
    is_ignorable: BOOLEAN
        -- All signals Standard C knows about are ignorable...
    value: INTEGER
        -- the signal
invariant
    accessing_real_singleton: signal_switch_is_real_singleton;
    accessing_real_singleton: security_is_real_singleton;
    valid_signal_value: value >= 1;
end of STDC_SIGNAL
```

A.10 Short form of STDC_SIGNAL_HANDLER

```
deferred class interface STDC_SIGNAL_HANDLER
invariant
    accessing_real_singleton: signal_switch_is_real_singleton;
end of deferred STDC_SIGNAL_HANDLER
```

A.11 Short form of STDC_SYSTEM

```
class interface STDC_SYSTEM
feature(s) from STDC_SYSTEM
   -- run-time determined queries
   is_shell_available: BOOLEAN
      -- Return True if command interpreter is available
feature(s) from STDC_SYSTEM
   -- compile time determined queries
   clocks_per_second: INTEGER
      -- number per second of the value returned by the clock function
feature(s) from STDC_SYSTEM
   -- endianess
   is_big_endian: BOOLEAN
      -- True if this is a big endian architecture
   is_little_endian: BOOLEAN
      -- True if this is a little endian architecture
invariant
   accessing_real_singleton: security_is_real_singleton;
end of STDC_SYSTEM
```

A.12 Short form of STDC_TIME

```
class interface STDC_TIME
creation
   make_date (a_year, a_month, a_day: INTEGER)
         -- Create a time according to this day, time 00:00:00.
         -- Date is assumed to be a local date.
   make_date_time (a_year, a_month, a_day, an_hour, a_minute, a_second: INTEGER)
         -- Date is assumed to be a local date.
         -- We assume daylight saving time setting in effect is
         -- available from system.
   make_from_now
         -- Make value equal to current unix time.
         -- Afterwards call to_local or to_utc to turn individual
         -- fields in local time or in utc time.
   make_from_unix_time (a_value: INTEGER)
         -- a_value is a time_t value.
         -- Afterwards call to_local or to_utc to turn individual
         -- fields in local time or in utc time.
   make_time (an_hour, a_minute, a_second: INTEGER)
         -- Time is assumed to be a local time.
         -- We assume daylight saving time setting in effect is
         -- available from system.
         -- Day will be January 1, minimum_year.
   make_utc_date (a_year, a_month, a_day: INTEGER)
         -- Create a time according to this day, time 00:00:00.
         -- Date is assumed to be in UTC.
   make_utc_date_time (a_year, a_month, a_day, an_hour, a_minute, a_second: INTEGER)
         -- Date is assumed to be in UTC.
         -- Conversion to the unix time is done without taking into
         -- account leap seconds, as according to the specification.
   make_utc_time (an_hour, a_minute, a_second: INTEGER)
         -- Time is assumed to be UTC time at January 1, minimum_year.
         -- We assume daylight saving time setting in effect is
         -- available from system.
feature(s) from STDC_TIME
   -- Initialization
   make_date (a_year, a_month, a_day: INTEGER)
         -- Create a time according to this day, time 00:00:00.
         -- Date is assumed to be a local date.
   make_date_time (a_year, a_month, a_day, an_hour, a_minute, a_second: INTEGER)
         -- Date is assumed to be a local date.
         -- We assume daylight saving time setting in effect is
         -- available from system.
   make_date_time_without_dst (a_year, a_month, a_day, an_hour, a_minute, a_second: INTEGER)
         -- Date is assumed to be a date/time without daylight saving
         -- taken into account, such as a UTC based date/time.
```

make_from_now
-- Make *value* equal to current unix time.
-- Afterwards call *to_local* or *to_utc* to turn individual
-- fields in local time or in utc time.
make_from_unix_time (*a_value*: *INTEGER*)
-- *a_value* is a time_t value.
-- Afterwards call *to_local* or *to_utc* to turn individual
-- fields in local time or in utc time.
make_utc_date (*a_year, a_month, a_day*: *INTEGER*)
-- Create a time according to this day, time 00:00:00.
-- Date is assumed to be in UTC.
make_utc_date_time (*a_year, a_month, a_day, an_hour, a_minute, a_second*: *INTEGER*)
-- Date is assumed to be in UTC.
-- Conversion to the unix time is done without taking into
-- account leap seconds, as according to the specification.
make_utc_time (*an_hour, a_minute, a_second*: *INTEGER*)
-- Time is assumed to be UTC time at January 1, *minimum_year*.
-- We assume daylight saving time setting in effect is
-- available from system.

feature(*s*) **from** *STDC_TIME*
-- Make individual time fields valid
is_local_time: *BOOLEAN*
-- Is time in *local time*?
is_utc_time: *BOOLEAN*
-- Is the time zone UTC?
is_time_zone_known: *BOOLEAN*
-- After a make routine, call either *to_local* or *to_utc*.
to_local
-- Switch time fields to local time based on time in *value*.
to_utc
-- Switch time fields to utc time based on time in *value*.

feature(*s*) **from** *STDC_TIME*
-- Manually set individual time fields
set_date (*a_year, a_month, a_day*: *INTEGER*)
-- Set date part, time remains unchanged, unless daylight
-- savings has to be taken into account.
set_date_time (*a_year, a_month, a_day, an_hour, a_minute, a_second*: *INTEGER*)
-- Set individual time fields. Set *value* based on given
-- fields, assuming that it is a local time.
-- We assume daylight saving time setting in effect (or not)
-- has been set.
set_dst_to_current
-- Let system figure out if daylight saving time is in effect.
set_dst_to_none
-- Daylight saving time is not in effect.
set_dst_in_effect
-- Daylight saving time is in effect.

```
    set_time (an_hour, a_minute, a_second: INTEGER)
        -- Set time part, date remains unchanged unless daylight
        -- savings has to be taken into account.
    to_dos_seconds
        -- Make sure the seconds are divisible by two, a value DOS
        -- and clones like Windows NT like.
feature(s) from STDC_TIME
    -- Individual time fields, need call to to_local or to_utc
    year: INTEGER
    month: INTEGER
    day: INTEGER
        -- Day of the month.
    weekday: INTEGER
        -- Days since Sunday.
    day_of_year: INTEGER
        -- Days since January 1st
    hour: INTEGER
    minute: INTEGER
    second: INTEGER
    is_daylight_savings_in_effect: BOOLEAN
        -- Does the broken down time take into account daylight savings?
    is_daylight_savings_unknown: BOOLEAN
        -- Do we not know if the broken time includes daylight saving?
feature(s) from STDC_TIME
    -- Time as string
    short_weekday_name: STRING
        -- Abbreviated weekday name
    weekday_name: STRING
        -- Full weekday name
    short_month_name: STRING
        -- Abbreviated month name
    month_name: STRING
        -- Full month name
    format (format_str: STRING): STRING
        -- Formatted date/time according to format_str. See
        -- man strftime for details.
    default_format: STRING
        -- Time as string of the form "Mon Apr 17 21:49:20 2000"
    local_date_string: STRING
        -- Date part in format local to current country.
    local_time_string: STRING
        -- Time part in format local to current country.
    rfc_date_string: STRING
        -- RFC 822 style date, i.e. Tue, 15 Nov 1994 08:12:31 GMT.
feature(s) from STDC_TIME
    -- Date calculations
    is_equal (other: like Current): BOOLEAN
```

```
        -- Is other attached to an object considered equal to
        -- current object ?
    infix "-" (other: like Current): like Current
        -- Creates a new time which is the difference between
        -- Current and Other
    infix "<" (other: like Current): BOOLEAN
        -- Is current object less than other?
feature(s) from STDC_TIME
    -- Status
    is_two_digit_year (a_year: INTEGER): BOOLEAN
        -- Is a_year a two digit year that can be handled by
        -- four_digit_year.
    is_valid_date (a_year, a_month, a_day: INTEGER): BOOLEAN
        -- Do a_year, a_month and a_day form a date recognized
        -- by this class?
    is_valid_day (a_year, a_month, a_day: INTEGER): BOOLEAN
        -- Is a_day a valid day given year and month.
    is_valid_time (an_hour, a_minute, a_second: INTEGER): BOOLEAN
        -- Do an_hour, a_minute and a_second form a valid 24
        -- hour clock time?
feature(s) from STDC_TIME
    -- Access
    current_year: INTEGER
        -- Current year.
    four_digit_year (a_year: INTEGER): INTEGER
        -- Return a four digit year given a possibly two digit year.
    hash_code: INTEGER
        -- The hash-code value of Current.
    minimum_year: INTEGER
        -- The minimum year for the current platform.
        -- For POSIX is 1970, for Windows is 1980.
    maximum_year: INTEGER
        -- The maximum Epoch year.
    value: INTEGER
        -- Time in seconds since January 1, 1970.
invariant
    accessing_real_singleton: security_is_real_singleton;
    tm_not_void: tm /= Void;
    tm_has_proper_capacity: tm.capacity >= posix_tm_size;
    value_not_negative: value >= 0;
    my_time_zone_valid: my_time_zone = 0 or else my_time_zone = utc_time_zone or else my_time_zone = local_ti
end of STDC_TIME
```

In this chapter:

- *Short form of ABSTRACT_CURRENT_PROCESS*
- *Short form of ABSTRACT_EXEC_PROCESS*
- *Short form of ABSTRACT_FILE_DESCRIPTOR*
- *Short form of ABSTRACT_FILE_SYSTEM*
- *Short form of ABSTRACT_HOST*
- *Short form of ABSTRACT_IP4_ADDRESS*
- *Short form of ABSTRACT_IP6_ADDRESS*
- *Short form of ABSTRACT_PIPE*
- *Short form of ABSTRACT_SERVICE*
- *Short form of ABSTRACT_STATUS*
- *Short form of ABSTRACT_TCP_CLIENT_SOCKET*
- *Short form of ABSTRACT_TCP_SERVER_SOCKET*

B

Short listing of abstract classes

An abstract class is somewhat above the Standard C classes, and between the features you get when you use a POSIX or Windows class. It is mainly aimed at users who want to write software usable on Unix and Windows, and who do not want to use a POSIX emulator.

You never use an abstract class directly, always use the corresponding effective EPX_XXXX, for which there is a variant in the `src/posix` or `src/windows` directory.

B.1 Short form of ABSTRACT_CURRENT_PROCESS

```
deferred class interface ABSTRACT_CURRENT_PROCESS
feature(s) from STDC_SECURITY_ACCESSOR
   -- The singleton, available to any because its used in preconditions
   security: STDC_SECURITY
      -- Singleton entry point for security.
feature(s) from STDC_BASE
   -- errno
   errno: STDC_ERRNO
      -- Access to the variable that contains the error that occurred.
feature(s) from STDC_CURRENT_PROCESS
   -- My standard input/output/error
   stdin: STDC_TEXT_FILE
   stdout: STDC_TEXT_FILE
   stderr: STDC_TEXT_FILE
feature(s) from STDC_CURRENT_PROCESS
   -- various
   clock: INTEGER
      -- return approximation of processor time used by the
      -- program, or -1 if unknown
feature(s) from STDC_CURRENT_PROCESS
   -- Random numbers
   random: INTEGER
```

```
            -- Returns a pseudo-random integer between 0 and RAND_MAX.
        set_random_seed (a_seed: INTEGER)
            -- Sets a_seed as the seed for a new sequence of
            -- pseudo-random integers to be returned by random. These
            -- sequences are repeatable by calling set_random_seed with
            -- the same seed value. If no seed value is provided, the
            -- random function is automatically seeded with a value of
            -- 1.
    feature(s) from ABSTRACT_PROCESS
        -- Process properties
        pid: INTEGER
            -- The process identifier.
        is_pid_valid: BOOLEAN
            -- current process id is always valid
    feature(s) from ABSTRACT_PROCESS
        -- Signal this process
        terminate
            -- Attempt to gracefully terminate this process.
            require
                valid_pid: is_pid_valid
    feature(s) from ABSTRACT_CURRENT_PROCESS
        -- Every process also has standard file descriptors which might not be compatible with stdin/stdout/stderr (Wind
        fd_stdin: ABSTRACT_FILE_DESCRIPTOR
            ensure
                fd_stdin_not_void: Result /= Void;
                not_owner: not Result.is_owner
        fd_stdout: ABSTRACT_FILE_DESCRIPTOR
            ensure
                fd_stdout_not_void: Result /= Void;
                not_owner: not Result.is_owner
        fd_stderr: ABSTRACT_FILE_DESCRIPTOR
            ensure
                fd_stderr_not_void: Result /= Void;
                not_owner: not Result.is_owner
    invariant
        accessing_real_singleton: security_is_real_singleton;
    end of deferred ABSTRACT_CURRENT_PROCESS
```

B.2 Short form of ABSTRACT_EXEC_PROCESS

```
deferred class interface ABSTRACT_EXEC_PROCESS
feature(s) from ABSTRACT_EXEC_PROCESS
   -- Initialization
   make (a_program: STRING; a_arguments: ARRAY[STRING])
   make_capture_input (a_program: STRING; a_arguments: ARRAY[STRING])
   make_capture_output (a_program: STRING; a_arguments: ARRAY[STRING])
   make_capture_io (a_program: STRING; a_arguments: ARRAY[STRING])
      -- Why not use threedirectional i/o, because youre getting
      -- yourself in great, great trouble anyway.
      -- A bit of advice: call stdin.close before starting to call
      -- stdout.read_string and such...
   make_capture_all (a_program: STRING; a_arguments: ARRAY[STRING])
      -- Threedirectional i/o is a great way to get yourself in trouble.
feature(s) from ABSTRACT_EXEC_PROCESS
   -- (re)set arguments
   has_void_argument (a_arguments: ARRAY[STRING]): BOOLEAN
      -- Is one of the items in a_arguments Void?
   set_arguments (a_arguments: ARRAY[STRING])
feature(s) from ABSTRACT_EXEC_PROCESS
   -- i/o capturing
   capture_input: BOOLEAN
      -- is input captured on execute?
   capture_output: BOOLEAN
      -- is output captured on execute?
   capture_error: BOOLEAN
      -- is error captured on execute?
   set_capture_input (on: BOOLEAN)
   set_capture_output (on: BOOLEAN)
   set_capture_error (on: BOOLEAN)
   fd_stdin: ABSTRACT_FILE_DESCRIPTOR
   fd_stdout: ABSTRACT_FILE_DESCRIPTOR
   fd_stderr: ABSTRACT_FILE_DESCRIPTOR
feature(s) from ABSTRACT_EXEC_PROCESS
   -- Execute
   execute
      -- Executes program_name. After execution, at some point in
      -- time, you have to wait or wait_for for this process to
      -- terminate.
      require
         not_already_started: is_terminated
feature(s) from ABSTRACT_EXEC_PROCESS
   -- Actions that parent may execute
   wait_for (suspend: BOOLEAN)
      -- Wait for this process to terminate. If suspend then we
      -- wait until the information about this process is available,
        -- else we return immediately.
        -- If suspend is False, check the running property to see
        -- if this child is really terminated.
        require
            pid_refers_to_child: is_pid_valid;
            not_terminated: not is_terminated
        ensure
            stdin_closed: is_terminated implies fd_stdin = Void or else not fd_stdin.is_open;
            stdout_closed: is_terminated implies fd_stdout = Void or else not fd_stdout.is_open;
            stderr_closed: is_terminated implies fd_stderr = Void or else not fd_stderr.is_open;
            terminated: suspend implies is_terminated;
            pid_invalid: is_terminated implies not is_pid_valid
feature(s) from ABSTRACT_EXEC_PROCESS
    -- Accessible state
    program_name: STDC_PATH
        -- program to execute
    arguments: ARRAY[STRING]
        -- arguments to pass to program
invariant
    accessing_real_singleton: security_is_real_singleton;
    pid_known_is_not_terminated: is_pid_valid = not is_terminated;
    program_name_not_empty: program_name /= Void and then not program_name.is_empty;
    arguments_not_void: arguments /= Void;
    all_arguments_not_void: not has_void_argument(arguments);
    descriptors_are_owners: (fd_stdin /= Void and then fd_stdin.is_open implies fd_stdin.is_owner) and then (fd_st
end of deferred ABSTRACT_EXEC_PROCESS
```

B.3 Short form of ABSTRACT_FILE_DESCRIPTOR

```
deferred class interface ABSTRACT_FILE_DESCRIPTOR
feature(s) from STDC_SECURITY_ACCESSOR
   -- The singleton, available to any because its used in preconditions
   security: STDC_SECURITY
      -- Singleton entry point for security.
feature(s) from STDC_BASE
   -- errno
   errno: STDC_ERRNO
      -- Access to the variable that contains the error that occurred.
feature(s) from MEMORY
   dispose
      -- Close handle if owner.
feature(s) from KI_OUTPUT_STREAM
   -- Output
   put_character (c: CHARACTER)
      -- Write a character.
   append (an_input_stream: KI_INPUT_STREAM[CHARACTER])
      -- Read items of an_input_stream until the end
      -- of input is reached, and write these items to
      -- current output stream.
      -- append is safe for non-blocking descriptors.
feature(s) from KI_OUTPUT_STREAM
   -- Status report
   is_open_write: BOOLEAN
      -- Can items be written to output stream?
   is_closable_for_writing: BOOLEAN
      -- Can current output stream be closed?
feature(s) from KI_OUTPUT_STREAM
   -- Access
   path: STDC_PATH
      -- Scratch path.
feature(s) from KI_OUTPUT_STREAM
   -- Basic operations
   close_for_writing
      -- Try to close output stream if it is closable. Set
      -- is_open_write to false if operation was successful.
feature(s) from KI_CHARACTER_OUTPUT_STREAM
   -- Output
   put_string (a_string: STRING)
      -- Write a_string to output stream.
   put_integer (i: INTEGER)
      -- Write decimal representation
      -- of i to output stream.
      -- Regexp: 0|(-?[1-9][0-9]*)
   put_boolean (b: BOOLEAN)
```

```
          -- Write "True" to output stream if
          -- b is true, "False" otherwise.
feature(s) from KI_CHARACTER_OUTPUT_STREAM
     -- Basic operations
     flush
          -- Flush buffered data to disk.
feature(s) from EPX_CHARACTER_OUTPUT_STREAM
     -- Output
     last_written: INTEGER
          -- How many bytes were written by last call to write?
          -- Can be less than requested for non-blocking output.
          -- Check last_blocked in that case.
     put_buffer (buf: STDC_BUFFER; offset, nbytes: INTEGER)
          -- More safe version of write in case you have a
          -- STDC_BUFFER object.
     write_buffer (buf: STDC_BUFFER; offset, nbytes: INTEGER)
          -- More safe version of write in case you have a
          -- STDC_BUFFER object.
feature(s) from KI_INPUT_STREAM
     -- Input
     non_blocking_read_character
          -- Read the next item in input stream.
          -- Make the result available in last_item.
     non_blocking_read_to_buffer (a_buffer: KI_BUFFER[CHARACTER]; pos, nb: INTEGER): INTEGER
          -- Fill a_buffer, starting at position pos, with
          -- at most nb items read from input stream.
          -- Return the number of items actually read.
feature(s) from KI_INPUT_STREAM
     -- Status report
     is_closable_for_reading: BOOLEAN
          -- Can current input stream be closed?
     is_open_read: BOOLEAN
          -- Can items be read from input stream?
     is_rewindable: BOOLEAN
          -- Can current input stream be rewound to return input from
          -- the beginning of the stream?
     eof: BOOLEAN
          -- True if end-of-file reached.
          -- Currently Im unsure if detection is reliable.
     valid_unread_character (a_character: CHARACTER): BOOLEAN
          -- Can a_character be put back in input stream?
feature(s) from KI_INPUT_STREAM
     -- Access
     last_character: CHARACTER
          -- Last character read
feature(s) from KI_INPUT_STREAM
     -- Basic operations
```

```
    close_for_reading
        -- Try to close input stream if it is closable. Set
        -- is_open_read to false if operation was successful.
    rewind
        -- Move input positionto the beginning of stream.
feature(s) from KI_CHARACTER_INPUT_STREAM
    -- Input
    non_blocking_read_string (nb: INTEGER)
        -- Read at most nb characters from input stream.
        -- Make the characters that have actually been read
        -- available in last_string.
    non_blocking_read_to_string (a_string: STRING; pos, nb: INTEGER): INTEGER
        -- Fill a_string, starting at position pos, with
        -- at most nb characters read from input stream.
        -- Return the number of characters actually read.
feature(s) from KI_CHARACTER_INPUT_STREAM
    -- Access
    last_string: STRING
        -- Last string read
        -- (Note: this query always return the same object.
        -- Therefore a clone should be used if the result
        -- is to be kept beyond the next call to this feature.
        -- However last_string is not shared between file objects.)
feature(s) from EPX_CHARACTER_INPUT_STREAM
    -- Access
    is_streaming: BOOLEAN
        -- Is data coming through a network stream?
feature(s) from EPX_CHARACTER_INPUT_STREAM
    -- Stream or disk file
    set_streaming (enable: BOOLEAN)
        -- Influence behaviour of certain functions if they should be
        -- optimized for data coming from disk or data coming from
        -- the network. In particular is_streaming implies that a
        -- client application is prepared to handle reads that
        -- return less than the requested number of bytes, but dont
        -- assume that means end-of-file.
feature(s) from EPX_CHARACTER_INPUT_STREAM
    -- Input
    last_read: INTEGER
        -- Last bytes read by read_buffer.
        -- Can be less than requested for non-blocking input.
        -- Check last_blocked in that case.
    read_buffer (buf: STDC_BUFFER; offset, nbytes: INTEGER)
        -- Read data into buf at offset for nbytes bytes.
        -- Number of bytes actually read are available in last_read.
        -- Thisis a more safe version of read in case you have a
        -- STDC_BUFFER object.
```

feature(*s*) **from** *KI_TEXT_INPUT_STREAM*
    -- Input
    read_line
        -- Read characters from input stream until a line separator
        -- or end of file is reached. Make the characters that have
        -- been read available in *last_string* and discard the line
        -- separator characters from the input stream.
        -- Zero characters will be read when non-blocking i/o
        -- is enabled, and *read* would block.
    read_new_line
        -- Read a line separator from input file.
        -- Make the characters making up the recognized
        -- line separator available in *last_string*,
        -- or make *last_string* empty and leave the
        -- input file unchanged if no line separator
        -- was found.
feature(*s*) **from** *KI_TEXT_INPUT_STREAM*
    -- Access
    eol: *STRING*
        -- Line separator
        -- EPX classes do not distinguish between a %R%N or just %N
        -- end-of-line. The platform may though.
feature(*s*) **from** *STDC_HANDLE*
    -- Access
    is_open: *BOOLEAN*
        -- Does *handle* contain an open handle?
    is_owner: *BOOLEAN*
        -- Does this object close the stream on *close* or *dispose*?
        -- Only for resources that are owned, are resource limits checked.
    resource_usage_can_be_increased: *BOOLEAN*
        -- Is it allowed to open another file?
feature(*s*) **from** *STDC_HANDLE*
    -- Influence ownership of the handle. Can help to influence subtile garbage collector problems
    become_owner
        -- This class will own its handle. This is the only function
        -- that actually increases the resource count.
    unown
        -- Resource will not be closed on dispose. Calling close will
        -- be forbidden. This routine may not call any other object,
        -- else it cannot be called from within dispose.
feature(*s*) **from** *STDC_HANDLE*
    -- Close
    close
        -- Close the resource.
    detach
        -- Forget the resource. Resource is not closed.
        -- You cannot read and write anymore.

```
feature(s) from STDC_HANDLE
   -- Resource
   capacity: INTEGER
      -- Number of resources that are in use by handle. For a
      -- file this is 1, for a memory handle, this is the number of
      -- bytes.
   fd: H
      -- Identifier of resource tracked by this class.
feature(s) from EPX_CHARACTER_IO_STREAM
   -- Status report
   is_closable: BOOLEAN
      -- Can current stream be closed for reading and writing?
feature(s) from ABSTRACT_FILE_DESCRIPTOR
   -- Initialization
   open (a_path: STRING; a_flags: INTEGER)
      -- Open given file with access given by flags.
   open_read (a_path: STRING)
      -- Open given file with access given by flags.
   open_write (a_path: STRING)
   open_read_write (a_path: STRING)
   open_truncate (a_path: STRING)
      -- Open file, if it exists, truncate it first.
   create_read_write (a_path: STRING)
      -- Always create a file, existing or not.
      -- Give read/write permissions to user only.
   create_write (a_path: STRING)
      -- Always create a file, existing or not.
      -- Give read/write permissions to user only.
   create_with_mode (a_path: STRING; flags, mode: INTEGER)
      -- Create a file according to flags and with mode access
      -- permissions. Make sure you have th O_CREAT flag in flags
      -- if you really want to create something!
feature(s) from ABSTRACT_FILE_DESCRIPTOR
   -- Special creation
   attach_to_fd (a_fd: INTEGER; a_become_owner: BOOLEAN)
      -- Create file descriptor with value a_fd. File descriptor
      -- will close it when a_become_owner.
   make_as_duplicate (another: ABSTRACT_FILE_DESCRIPTOR)
      -- On creation, create a duplicate from another file descriptor
      -- As normal call, closes its own descriptor first (if open) and
      -- duplicates next.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
   -- Read and write to memory block
   last_blocked: BOOLEAN
      -- Would last call to read or write block?
   read (buf: POINTER; offset, nbytes: INTEGER)
      -- Read data into buf at offset for nbytes bytes.
```

```
        -- The number of bytes actually read, is available in last_read.
    write (buf: POINTER; offset, nbytes: INTEGER)
        -- Write given data from buf at offset, for nbytes
        -- bytes. Number of actually written bytes are in
        -- last_written. last_written can be unequal to nbytes
        -- if i/o is non-blocking or some error has occurred.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- Eiffel like output
    put (a: ANY)
        -- Write any Eiffel object as string using its out value.
    write_character (c: CHARACTER)
        -- Write a character.
    write_string (a_string: STRING)
        -- Write a_string to output stream.
    puts (a_string: STRING)
        -- Write a_string to output stream.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- Buffered input
    read_character
        -- Sets last_character.
        -- If this routine blocks, last_character has the value
        -- %U. Therefore, if non-blocking is enabled, always check
        -- last_blocked to see if the value make sense.
    read_string (nb: INTEGER)
        -- Read at most nb characters from input stream.
        -- Make the characters that have actually been read
        -- available in last_string.
        -- Zero characters will be read when non-blocking i/o
        -- is enabled, and read would block.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- File position
    seek (offset: INTEGER)
        -- Set file position to given absolute offset.
    seek_from_current (offset: INTEGER)
        -- Set file position relative to current position.
    seek_from_end (offset: INTEGER)
        -- Set file position relative to end of file.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- Status report
    is_attached_to_terminal: BOOLEAN
        -- Is the handle associated with character device?
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- Access
    status: ABSTRACT_STATUS
        -- The status for this file descriptor. Cached value,
        -- refreshed only when file reopened.
    value: INTEGER
```

```
          -- The actual file descriptor value.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
     -- non-blocking i/o
     is_blocking_io: BOOLEAN
          -- Is blocking i/o enabled?
          -- Blocking i/o is the default.
          -- If false, calls like read and write will never wait
          -- for input, if there is no input.
     set_blocking_io (enable: BOOLEAN)
          -- Set is_blocking_io.
     supports_nonblocking_io: BOOLEAN
          -- Does this descriptor support non-blocking input/output?
          -- On POSIX systems, any descriptor does.
          -- On Windows sockets and pipes do.
invariant
     open_in_sync: is_open_read or is_open_write implies is_open; -- The reverse is not true, for examples sockets
 -- closed for reading/writing, but still open.
     accessing_real_singleton: security_is_real_singleton;
     capacity_not_negative: capacity >= 0;
     valid_capacity: is_open = (capacity > 0);
     open_implies_handle_assigned: is_open = (fd /= unassigned_value);
     owned_implies_open: is_owner implies is_open;
     owned_implies_handle_assigned: is_owner implies fd /= unassigned_value;
     valid_status: not is_open implies my_status = Void;
     path_not_void: path /= Void;
     line_buffer_index_offset_ok: line_buffer /= Void implies line_buffer_index <= line_buffer.count;
end of deferred ABSTRACT_FILE_DESCRIPTOR
```

B.4 Short form of ABSTRACT_FILE_SYSTEM

```
deferred class interface ABSTRACT_FILE_SYSTEM
feature(s) from STDC_SECURITY_ACCESSOR
   -- The singleton, available to any because its used in preconditions
   security: STDC_SECURITY
      -- Singleton entry point for security.
feature(s) from STDC_BASE
   -- errno
   errno: STDC_ERRNO
      -- Access to the variable that contains the error that occurred.
feature(s) from STDC_FILE_SYSTEM
   -- Path names
   expand_path (a_path: STRING): STDC_PATH
      -- returns a new path
feature(s) from STDC_FILE_SYSTEM
   -- Rename files/directories, remove files/directories
   remove_file (a_path: STRING)
      -- Removes a file from a directory.
      -- For Standard C, its implementation defined what
      -- remove_file does if file is opened by some process
      -- (remove_file fails on Windows for example).
      -- doesnt remove a directory.
   rename_to (current_path, new_path: STRING)
      -- Rename a file or a directory.
      -- new_path should not be an existing path.
feature(s) from STDC_FILE_SYSTEM
   -- Accessibility of files
   is_modifiable (a_path: STRING): BOOLEAN
      -- tests if file is readable and writable by this program
      -- uses real user ID and real group ID instead of effective ones
   is_readable (a_path: STRING): BOOLEAN
      -- Tests if a_path is readable by this program. a_path
      -- can be a file or a directory.
      -- Uses real user ID and real group ID instead of effective
      -- ones.
feature(s) from ABSTRACT_FILE_SYSTEM
   -- Directory access
   change_directory (a_directory: STRING)
      -- Changes the current working directory.
   current_directory: STRING
      -- The current directory.
   make_directory (a_directory: STRING)
      -- Makes a directory, only accessible by owner.
   mkdir (a_directory: STRING)
      -- Makes a directory, only accessible by owner.
   remove_directory (a_directory: STRING)
```

```
            -- Removes an empty directory, see also force_remove_directory
        rmdir (a_directory: STRING)
            -- Removes an empty directory, see also force_remove_directory
        force_remove_directory (a_directory: STRING)
            -- Removes a directory, even when not empty.
            -- I suggest you do not have hard or symbolic links in a_directory...
    feature(s) from ABSTRACT_FILE_SYSTEM
        -- File statistics
        status (a_path: STRING): ABSTRACT_STATUS_PATH
            -- Get information about a file.
            require
                valid_path: a_path /= Void and then not a_path.is_empty;
                existing_file: is_existing(a_path)
            ensure
                status_returned: Result /= Void
        status_may_fail (a_path: STRING): ABSTRACT_STATUS_PATH
            -- Retrieve status information for a_path. a_path may or
            -- may not exist. Check Result.found to see if statistics
            -- were retrieved.
            require
                valid_path: a_path /= Void and then not a_path.is_empty
            ensure
                status_returned: Result /= Void
    feature(s) from ABSTRACT_FILE_SYSTEM
        -- Directory browsing
        browse_directory (a_path: STRING): ABSTRACT_DIRECTORY
            -- Get information about a directory.
            require
                valid_path: a_path /= Void and then not a_path.is_empty;
                path_is_directory: security.error_handling.exceptions_enabled and then status(a_path).is_directory
            ensure
                directory_returned: Result /= Void
    feature(s) from ABSTRACT_FILE_SYSTEM
        -- Accessibility of files
        last_access_result: INTEGER
            -- value of last access test
        is_accessible (a_path: STRING; a_mode: INTEGER): BOOLEAN
            -- Is a_path accessibility using a_mode?
        access (a_path: STRING; a_mode: INTEGER): BOOLEAN
            -- Is a_path accessibility using a_mode?
        is_directory (a_path: STRING): BOOLEAN
            -- Does a_path exists and is it a directory?
        is_existing (a_path: STRING): BOOLEAN
            -- Is a_path an existing file, directory, whatever?
            -- Tests if file does exist, not if it is readable or writable by
            -- this program!
            -- Uses real user ID and real group ID instead of effective ones.
        is_empty (a_path: STRING): BOOLEAN
            -- True if file exists and has a size equal to zero.
        is_executable (a_path: STRING): BOOLEAN
            -- tests if file is executable by this program
        is_regular_file (a_path: STRING): BOOLEAN
            -- Does a_path exists and is it a regular file?
        is_writable (a_path: STRING): BOOLEAN
            -- tests if file is writable by this program
            -- uses real user ID and real group ID instead of effective ones
    feature(s) from ABSTRACT_FILE_SYSTEM
        -- File system properties
        is_case_sensitive: BOOLEAN
            -- is file system case sensitive or not?
            -- This query is dedicated to jwz
        path_separator: CHARACTER
            -- What is the path separator?
    feature(s) from ABSTRACT_FILE_SYSTEM
        -- Path names
        resolved_path_name (a_path: STRING): STRING
            -- Derives from a_path an absolute pathname that names the
            -- same file, whose resolution does not involve ".", "..", or
            -- symbolic links.
        temporary_directory: STRING
            -- The name of the temporary directory.
            -- Name does not end with the directory separator.
            ensure
                directory_returned: Result /= Void;
                directory_exists: is_directory(Result);
                directory_is_writable: is_modifiable(Result);
                last_char_not_separator: Result.item(Result.count) /= path_separator
    feature(s) from ABSTRACT_FILE_SYSTEM
        -- File contents
        file_content_as_string (a_file_name: STRING): STRING
            -- Return contents of a_file_name as a STRING.
    invariant
        accessing_real_singleton: security_is_real_singleton;
    end of deferred ABSTRACT_FILE_SYSTEM
```

B.5 Short form of ABSTRACT_HOST

```
deferred class interface ABSTRACT_HOST
feature(s) from STDC_SECURITY_ACCESSOR
    -- The singleton, available to any because its used in preconditions
    security: STDC_SECURITY
        -- Singleton entry point for security.
feature(s) from STDC_BASE
    -- errno
    errno: STDC_ERRNO
        -- Access to the variable that contains the error that occurred.
feature(s) from ABSTRACT_HOST
    -- Initialization
    make_from_name (a_name: STRING)
        -- Initialize host from name. If name is numerical, the
        -- behaviour is not specified.
    make_from_address (an_address: ABSTRACT_IP_ADDRESS)
        -- Initialize host from ip address an_address.
        -- An attempt is made to resolve the host name using this address.
        -- Status is always found, even when reverse lookup failed.
    make_from_ip4_any
        -- IP address that refers to all local interfaces.
    make_from_ip4_loopback
        -- IP address that refers to the loopback device.
        -- No attempt at resolving is done.
feature(s) from ABSTRACT_HOST
    -- Command
    find_by_address
        -- Attempt to lookup up the host by first ip address in
        -- addresses. Sets found if host could be found.
        -- If found, sets canonical_name, aliases,
        -- address_family, address_length and addresses.
    find_by_name
        -- Attempt to lookup up the host given in name. Sets
        -- found if host could be found.
        -- If found, sets canonical_name, aliases,
        -- address_family, address_length and addresses.
feature(s) from ABSTRACT_HOST
    -- Access
    found: BOOLEAN
        -- Does this class contain a resolved host?
        -- If False, not_found_reason contains the reason.
    name: STRING
        -- Name as given to make_from_name or else equal to
        -- canonical_name.
    not_found_reason: INTEGER
        -- Reason why found is False. Result is a code whose
```

```
            -- interpretation depends on the platform.
        canonical_name: STRING
            -- Official (canonical) name of host.
        aliases: ARRAY[STRING]
            -- Alias names.
        address_family: INTEGER
            -- Host address type: AF_INET or AF_INET6
        address_length: INTEGER
            -- Length of address: 4 or 16.
        addresses: ARRAY[ABSTRACT_IP_ADDRESS]
            -- Array with IPv4 or IPv6 addresses.
    invariant
        accessing_real_singleton: security_is_real_singleton;
        name_void_or_not_empty: name = Void or else not name.is_empty;
        has_canonical_name: found implies name /= Void = (canonical_name /= Void);
        has_at_least_one_ip_address: found = (addresses /= Void and then addresses.count > 0);
        only_non_void_addresses: found implies is_every_address_not_void;
        has_aliases: found = (aliases /= Void);
        valid_length: found implies address_length > 0;
        consistent: addresses /= Void and then addresses.count > 0 implies found;
        my_not_found_reason_valid: found = (my_not_found_reason = 0);
    end of deferred ABSTRACT_HOST
```

B.6 Short form of ABSTRACT_IP4_ADDRESS

```
class interface ABSTRACT_IP4_ADDRESS
feature(s) from STDC_SECURITY_ACCESSOR
    -- The singleton, available to any because its used in preconditions
    security: STDC_SECURITY
        -- Singleton entry point for security.
feature(s) from STDC_BASE
    -- errno
    errno: STDC_ERRNO
        -- Access to the variable that contains the error that occurred.
feature(s) from ABSTRACT_IP_ADDRESS
    -- Initialization
    make_from_pointer (a_ptr: POINTER)
        -- Initialize ip address from 32-bit integer pointed to by a_ptr.
        -- We assume a_ptr points to a value in network byte order.
feature(s) from ABSTRACT_IP_ADDRESS
    -- Status
    is_loopback_address: BOOLEAN
        -- Does this IP address refer to the loopback address?
feature(s) from ABSTRACT_IP_ADDRESS
    -- General ip address features
    address_family: INTEGER
        -- Is it an ip4 or ip6 address.
    address_length: INTEGER
        -- Length of an IPv4 address is 4.
    ptr: POINTER
        -- Pointer to an in_addr or in6_addr structure.
        -- (bytes are in network byte order for in_addr)
feature(s) from ABSTRACT_IP4_ADDRESS
    -- Initialization
    make_from_any
        -- Initialize using the any address (i.e. 0.0.0.0).
    make_from_integer (a_value: INTEGER)
        -- Initialize ip address from 32-bit integer.
    make_from_loopback
        -- Initialize using the loopback address (i.e. 127.0.0.1).
feature(s) from ABSTRACT_IP4_ADDRESS
    -- Access
    value: INTEGER
        -- IPv4 address as 32-bit integer.
        -- Value is in host byte order.
feature(s) from ABSTRACT_IP4_ADDRESS
    -- Change
    set_value (new_value: INTEGER)
        -- Change IP address value to new_value.
feature(s) from ABSTRACT_IP4_ADDRESS
```

```
    -- Output
    out: STRING
        -- Friendly out
invariant
    accessing_real_singleton: security_is_real_singleton;
    buf_not_void: buf /= Void;
    buf_capacity_large_enough: buf.capacity >= abstract_api.posix_in_addr_size;
end of ABSTRACT_IP4_ADDRESS
```

B.7 Short form of ABSTRACT_IP6_ADDRESS

```
deferred class interface ABSTRACT_IP6_ADDRESS
feature(s) from STDC_SECURITY_ACCESSOR
    -- The singleton, available to any because its used in preconditions
    security: STDC_SECURITY
        -- Singleton entry point for security.
feature(s) from STDC_BASE
    -- errno
    errno: STDC_ERRNO
        -- Access to the variable that contains the error that occurred.
feature(s) from ABSTRACT_IP_ADDRESS
    -- Initialization
    make_from_pointer (a_ptr: POINTER)
        -- Initialize ip address from 32-bit integer.
feature(s) from ABSTRACT_IP_ADDRESS
    -- Status
    is_loopback_address: BOOLEAN
        -- Does this IP address refer to the loopback address?
feature(s) from ABSTRACT_IP_ADDRESS
    -- General ip address features
    address_family: INTEGER
        -- Is it an ip4 or ip6 address.
    address_length: INTEGER
        -- Length of an IPv6 address is 16.
    ptr: POINTER
        -- Pointer to an in_addr or in6_addr structure.
        -- (bytes are in network byte order for in_addr)
feature(s) from ABSTRACT_IP6_ADDRESS
    -- Output
    out: STRING
        -- Friendly out
feature(s) from ABSTRACT_IP6_ADDRESS
    -- General ip address features
    scope_id: INTEGER
invariant
    accessing_real_singleton: security_is_real_singleton;
    buf_not_void: buf /= Void;
    buf_capacity_large_enough: buf.capacity >= abstract_api.posix_in6_addr_size;
end of deferred ABSTRACT_IP6_ADDRESS
```

B.8 Short form of ABSTRACT_PIPE

```
deferred class interface ABSTRACT_PIPE
feature(s) from ABSTRACT_PIPE
   -- creation
   make
feature(s) from ABSTRACT_PIPE
   -- pipe operations
   close
feature(s) from ABSTRACT_PIPE
   -- the pipe
   fdout: ABSTRACT_FILE_DESCRIPTOR
   fdin: ABSTRACT_FILE_DESCRIPTOR
invariant
   accessing_real_singleton: security_is_real_singleton;
   valid_pipe: fdin /= Void and fdout /= Void;
end of deferred ABSTRACT_PIPE
```

B.9 Short form of ABSTRACT_SERVICE

```
deferred class interface ABSTRACT_SERVICE
feature(s) from STDC_SECURITY_ACCESSOR
    -- The singleton, available to any because its used in preconditions
    security: STDC_SECURITY
        -- Singleton entry point for security.
feature(s) from STDC_BASE
    -- errno
    errno: STDC_ERRNO
        -- Access to the variable that contains the error that occurred.
feature(s) from ABSTRACT_SERVICE
    -- Initialization
    make_from_name (a_name, a_protocol: STRING)
        -- Find service with a_name and optional a_protocol or raise
        -- exception.
    make_from_port (a_port: INTEGER; a_protocol: STRING)
        -- Initialize service from given a_port.
        -- Make sure to provide a a_protocol if necessary!
feature(s) from ABSTRACT_SERVICE
    -- Access
    port: INTEGER
        -- port number
    name: STRING
        -- official service name
    aliases: ARRAY[STRING]
        -- alias list
    protocol: STRING
        -- protocol to use (udp/tcp)
    protocol_type: INTEGER
        -- SOCK_STREAM or SOCK_DGRAM
invariant
    accessing_real_singleton: security_is_real_singleton;
    name_void_or_not_empty: name = Void or else not name.is_empty;
    valid_port: port >= 0 and port <= 65535;
    valid_protocol: protocol = Void or else protocol.is_empty or else (protocol.is_equal(once_tcp) or protocol.is_equ
    valid_protocol_type: protocol_type = sock_stream or else protocol_type = sock_dgram;
    valid_aliases: aliases /= Void;
end of deferred ABSTRACT_SERVICE
```

B.10 Short form of ABSTRACT_STATUS

```
deferred class interface ABSTRACT_STATUS
feature(s) from ABSTRACT_STATUS
   refresh
      -- refresh the cached information
feature(s) from ABSTRACT_STATUS
   -- stat members
   atime: INTEGER
      -- Unix time of last access.
   access_time: INTEGER
      -- Unix time of last access.
   device_number: INTEGER
      -- ID of device containing the file.
      -- Windows: Drive number of the disk containing the file.
   is_character_special: BOOLEAN
      -- Is this file a character-special file?
   is_directory: BOOLEAN
   is_fifo: BOOLEAN
   is_regular_file: BOOLEAN
   mtime: INTEGER
      -- Unix time of last data modification.
   modification_time: INTEGER
      -- Unix time of last data modification.
   nlink: INTEGER
   number_of_hard_links: INTEGER
   size: INTEGER
      -- Size of file in bytes.
   status_change_time: INTEGER
      -- Unix time of last status change.
      -- For example changing the permission bits will set this time.
feature(s) from ABSTRACT_STATUS
   -- Direct access to the individual stat fields, not recommended
   unix_mode: INTEGER
invariant
   accessing_real_singleton: security_is_real_singleton;
   stat_not_void: stat /= Void and then stat.capacity >= abstract_stat_size;
end of deferred ABSTRACT_STATUS
```

B.11 Short form of ABSTRACT_TCP_CLIENT_SOCKET

```
deferred class interface ABSTRACT_TCP_CLIENT_SOCKET
feature(s) from STDC_SECURITY_ACCESSOR
    -- The singleton, available to any because its used in preconditions
    security: STDC_SECURITY
        -- Singleton entry point for security.
feature(s) from STDC_BASE
    -- errno
    errno: STDC_ERRNO
        -- Access to the variable that contains the error that occurred.
feature(s) from MEMORY
    dispose
        -- Close handle if owner.
feature(s) from KI_OUTPUT_STREAM
    -- Output
    put_character (c: CHARACTER)
        -- Write a character.
    append (an_input_stream: KI_INPUT_STREAM[CHARACTER])
        -- Read items of an_input_stream until the end
        -- of input is reached, and write these items to
        -- current output stream.
        -- append is safe for non-blocking descriptors.
feature(s) from KI_OUTPUT_STREAM
    -- Status report
    is_open_write: BOOLEAN
        -- Can items be written to output stream?
    is_closable_for_writing: BOOLEAN
        -- Can current output stream be closed?
feature(s) from KI_OUTPUT_STREAM
    -- Access
    path: STDC_PATH
        -- Scratch path.
feature(s) from KI_OUTPUT_STREAM
    -- Basic operations
    close_for_writing
        -- Try to close output stream if it is closable. Set
        -- is_open_write to false if operation was successful.
feature(s) from KI_CHARACTER_OUTPUT_STREAM
    -- Output
    put_string (a_string: STRING)
        -- Write a_string to output stream.
    put_integer (i: INTEGER)
        -- Write decimal representation
        -- of i to output stream.
        -- Regexp: 0|(-?[1-9][0-9]*)
    put_boolean (b: BOOLEAN)
```

```
        -- Write "True" to output stream if
        -- b is true, "False" otherwise.
feature(s) from KI_CHARACTER_OUTPUT_STREAM
    -- Basic operations
    flush
        -- Flush buffered data to disk.
feature(s) from EPX_CHARACTER_OUTPUT_STREAM
    -- Output
    last_written: INTEGER
        -- How many bytes were written by last call to write?
        -- Can be less than requested for non-blocking output.
        -- Check last_blocked in that case.
    put_buffer (buf: STDC_BUFFER; offset, nbytes: INTEGER)
        -- More safe version of write in case you have a
        -- STDC_BUFFER object.
    write_buffer (buf: STDC_BUFFER; offset, nbytes: INTEGER)
        -- More safe version of write in case you have a
        -- STDC_BUFFER object.
feature(s) from KI_INPUT_STREAM
    -- Input
    non_blocking_read_character
        -- Read the next item in input stream.
        -- Make the result available in last_item.
    non_blocking_read_to_buffer (a_buffer: KI_BUFFER[CHARACTER]; pos, nb: INTEGER): INTEGER
        -- Fill a_buffer, starting at position pos, with
        -- at most nb items read from input stream.
        -- Return the number of items actually read.
feature(s) from KI_INPUT_STREAM
    -- Status report
    is_closable_for_reading: BOOLEAN
        -- Can current input stream be closed?
    is_open_read: BOOLEAN
        -- Can items be read from input stream?
    is_rewindable: BOOLEAN
        -- Can current input stream be rewound to return input from
        -- the beginning of the stream?
    eof: BOOLEAN
        -- True if end-of-file reached.
        -- Currently Im unsure if detection is reliable.
    valid_unread_character (a_character: CHARACTER): BOOLEAN
        -- Can a_character be put back in input stream?
feature(s) from KI_INPUT_STREAM
    -- Access
    last_character: CHARACTER
        -- Last character read
feature(s) from KI_INPUT_STREAM
    -- Basic operations
```

```
        close_for_reading
                -- Try to close input stream if it is closable. Set
                -- is_open_read to false if operation was successful.
        rewind
                -- Move input positionto the beginning of stream.
feature(s) from KI_CHARACTER_INPUT_STREAM
        -- Input
        non_blocking_read_string (nb: INTEGER)
                -- Read at most nb characters from input stream.
                -- Make the characters that have actually been read
                -- available in last_string.
        non_blocking_read_to_string (a_string: STRING; pos, nb: INTEGER): INTEGER
                -- Fill a_string, starting at position pos, with
                -- at most nb characters read from input stream.
                -- Return the number of characters actually read.
feature(s) from KI_CHARACTER_INPUT_STREAM
        -- Access
        last_string: STRING
                -- Last string read
                -- (Note: this query always return the same object.
                -- Therefore a clone should be used if the result
                -- is to be kept beyond the next call to this feature.
                -- However last_string is not shared between file objects.)
feature(s) from EPX_CHARACTER_INPUT_STREAM
        -- Access
        is_streaming: BOOLEAN
                -- Is data coming through a network stream?
feature(s) from EPX_CHARACTER_INPUT_STREAM
        -- Stream or disk file
        set_streaming (enable: BOOLEAN)
                -- Influence behaviour of certain functions if they should be
                -- optimized for data coming from disk or data coming from
                -- the network. In particular is_streaming implies that a
                -- client application is prepared to handle reads that
                -- return less than the requested number of bytes, but dont
                -- assume that means end-of-file.
feature(s) from EPX_CHARACTER_INPUT_STREAM
        -- Input
        last_read: INTEGER
                -- Last bytes read by read_buffer.
                -- Can be less than requested for non-blocking input.
                -- Check last_blocked in that case.
        read_buffer (buf: STDC_BUFFER; offset, nbytes: INTEGER)
                -- Read data into buf at offset for nbytes bytes.
                -- Number of bytes actually read are available in last_read.
                -- Thisis a more safe version of read in case you have a
                -- STDC_BUFFER object.
```

feature(*s*) **from** *KI_TEXT_INPUT_STREAM*
-- Input
read_line
-- Read characters from input stream until a line separator
-- or end of file is reached. Make the characters that have
-- been read available in *last_string* and discard the line
-- separator characters from the input stream.
-- Zero characters will be read when non-blocking i/o
-- is enabled, and *read* would block.
read_new_line
-- Read a line separator from input file.
-- Make the characters making up the recognized
-- line separator available in *last_string*,
-- or make *last_string* empty and leave the
-- input file unchanged if no line separator
-- was found.
feature(*s*) **from** *KI_TEXT_INPUT_STREAM*
-- Access
eol: *STRING*
-- Line separator
-- EPX classes do not distinguish between a %R%N or just %N
-- end-of-line. The platform may though.
feature(*s*) **from** *STDC_HANDLE*
-- Access
is_open: *BOOLEAN*
-- Does *handle* contain an open handle?
is_owner: *BOOLEAN*
-- Does this object close the stream on *close* or *dispose*?
-- Only for resources that are owned, are resource limits checked.
resource_usage_can_be_increased: *BOOLEAN*
-- Is it allowed to open another file?
feature(*s*) **from** *STDC_HANDLE*
-- Influence ownership of the handle. Can help to influence subtile garbage collector problems
become_owner
-- This class will own its handle. This is the only function
-- that actually increases the resource count.
unown
-- Resource will not be closed on dispose. Calling close will
-- be forbidden. This routine may not call any other object,
-- else it cannot be called from within dispose.
feature(*s*) **from** *STDC_HANDLE*
-- Close
close
-- Close the resource.
detach
-- Forget the resource. Resource is not closed.
-- You cannot read and write anymore.

```
feature(s) from STDC_HANDLE
   -- Resource
   capacity: INTEGER
      -- Number of resources that are in use by handle. For a
      -- file this is 1, for a memory handle, this is the number of
      -- bytes.
   fd: H
      -- Identifier of resource tracked by this class.
feature(s) from EPX_CHARACTER_IO_STREAM
   -- Status report
   is_closable: BOOLEAN
      -- Can current stream be closed for reading and writing?
feature(s) from ABSTRACT_FILE_DESCRIPTOR
   -- Initialization
   open (a_path: STRING; a_flags: INTEGER)
      -- Open given file with access given by flags.
   open_read (a_path: STRING)
      -- Open given file with access given by flags.
   open_write (a_path: STRING)
   open_read_write (a_path: STRING)
   open_truncate (a_path: STRING)
      -- Open file, if it exists, truncate it first.
   create_read_write (a_path: STRING)
      -- Always create a file, existing or not.
      -- Give read/write permissions to user only.
   create_write (a_path: STRING)
      -- Always create a file, existing or not.
      -- Give read/write permissions to user only.
   create_with_mode (a_path: STRING; flags, mode: INTEGER)
      -- Create a file according to flags and with mode access
      -- permissions. Make sure you have th O_CREAT flag in flags
      -- if you really want to create something!
feature(s) from ABSTRACT_FILE_DESCRIPTOR
   -- Special creation
   attach_to_socket (a_fd: INTEGER; a_become_owner: BOOLEAN)
      -- Create file descriptor with value a_fd. File descriptor
      -- will close it when a_become_owner.
   make_as_duplicate (another: ABSTRACT_FILE_DESCRIPTOR)
      -- On creation, create a duplicate from another file descriptor
      -- As normal call, closes its own descriptor first (if open) and
      -- duplicates next.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
   -- Read and write to memory block
   last_blocked: BOOLEAN
      -- Would last call to read or write block?
   read (buf: POINTER; offset, nbytes: INTEGER)
      -- Read data into buf at offset for nbytes bytes.
```

```
        -- The number of bytes actually read, is available in last_read.
    write (buf: POINTER; offset, nbytes: INTEGER)
        -- Write given data from buf at offset, for nbytes
        -- bytes. Number of actually written bytes are in
        -- last_written. last_written can be unequal to nbytes
        -- if i/o is non-blocking or some error has occurred.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- Eiffel like output
    put (a: ANY)
        -- Write any Eiffel object as string using its out value.
    write_character (c: CHARACTER)
        -- Write a character.
    write_string (a_string: STRING)
        -- Write a_string to output stream.
    puts (a_string: STRING)
        -- Write a_string to output stream.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- Buffered input
    read_character
        -- Sets last_character.
        -- If this routine blocks, last_character has the value
        -- %U. Therefore, if non-blocking is enabled, always check
        -- last_blocked to see if the value make sense.
    read_string (nb: INTEGER)
        -- Read at most nb characters from input stream.
        -- Make the characters that have actually been read
        -- available in last_string.
        -- Zero characters will be read when non-blocking i/o
        -- is enabled, and read would block.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- File position
    seek (offset: INTEGER)
        -- Set file position to given absolute offset.
    seek_from_current (offset: INTEGER)
        -- Set file position relative to current position.
    seek_from_end (offset: INTEGER)
        -- Set file position relative to end of file.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- Status report
    is_attached_to_terminal: BOOLEAN
        -- Is the handle associated with character device?
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- Access
    status: ABSTRACT_STATUS
        -- The status for this file descriptor. Cached value,
        -- refreshed only when file reopened.
    value: INTEGER
```

```
            -- The actual file descriptor value.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
        -- non-blocking i/o
        is_blocking_io: BOOLEAN
            require
                open: is_open_read
        set_blocking_io (enable: BOOLEAN)
            require
                supports_nonblocking_io: not enable implies supports_nonblocking_io;
                open: is_open
            ensure
                blocking_set: enable = is_blocking_io
        supports_nonblocking_io: BOOLEAN
feature(s) from ABSTRACT_INTERNET_SOCKET
        -- Local and remote addresses
        local_address: ABSTRACT_SOCKET_ADDRESS_IN_BASE
            -- Return address used on this side to talk to remote.
        remote_address: ABSTRACT_SOCKET_ADDRESS_IN_BASE
            -- Return address used at remote side to talk to this side.
feature(s) from ABSTRACT_TCP_SOCKET
        -- Shutdown
        shutdown_read
            -- The read-half of the connection is closed. No more data
            -- can be received on the socket and any data currently in
            -- the socket receive buffer is discarded. The process can no
            -- longer issue any of the read functions on the socket. Any
            -- data received after this call for a TCP socket is
            -- acknowledged and then silently discarded.
        shutdown_read_write
            -- The read-half and write-half of the connection are both
            -- closed. This is equivalent to calling shutdown-read and
            -- shutdown_write.
        shutdown_write
            -- The write-half of the connection is closed. In the case of
            -- TCP, this is called a half-close. Any data currently in
            -- the socket send buffer will be sent, followed by TCPs
            -- normal connection termination sequence. The process can no
            -- longer issue any of the write functions on the socket.
invariant
        open_in_sync: is_open_read or is_open_write implies is_open; -- The reverse is not true, for examples sockets
    -- closed for reading/writing, but still open.
        accessing_real_singleton: security_is_real_singleton;
        capacity_not_negative: capacity >= 0;
        valid_capacity: is_open = (capacity > 0);
        open_implies_handle_assigned: is_open = (fd /= unassigned_value);
        owned_implies_open: is_owner implies is_open;
        owned_implies_handle_assigned: is_owner implies fd /= unassigned_value;
        valid_status: not is_open implies my_status = Void;
        path_not_void: path /= Void;
        line_buffer_index_offset_ok: line_buffer /= Void implies line_buffer_index <= line_buffer.count;
end of deferred ABSTRACT_TCP_CLIENT_SOCKET
```

B.12 Short form of ABSTRACT_TCP_SERVER_SOCKET

```
deferred class interface ABSTRACT_TCP_SERVER_SOCKET
feature(s) from STDC_SECURITY_ACCESSOR
    -- The singleton, available to any because its used in preconditions
    security: STDC_SECURITY
        -- Singleton entry point for security.
feature(s) from STDC_BASE
    -- errno
    errno: STDC_ERRNO
        -- Access to the variable that contains the error that occurred.
feature(s) from MEMORY
    dispose
        -- Close handle if owner.
feature(s) from KI_OUTPUT_STREAM
    -- Output
    put_character (c: CHARACTER)
        -- Write a character.
    append (an_input_stream: KI_INPUT_STREAM[CHARACTER])
        -- Read items of an_input_stream until the end
        -- of input is reached, and write these items to
        -- current output stream.
        -- append is safe for non-blocking descriptors.
feature(s) from KI_OUTPUT_STREAM
    -- Status report
    is_open_write: BOOLEAN
        -- Can items be written to output stream?
    is_closable_for_writing: BOOLEAN
        -- Can current output stream be closed?
feature(s) from KI_OUTPUT_STREAM
    -- Access
    path: STDC_PATH
        -- Scratch path.
feature(s) from KI_OUTPUT_STREAM
    -- Basic operations
    close_for_writing
        -- Try to close output stream if it is closable. Set
        -- is_open_write to false if operation was successful.
feature(s) from KI_CHARACTER_OUTPUT_STREAM
    -- Output
    put_string (a_string: STRING)
        -- Write a_string to output stream.
    put_integer (i: INTEGER)
        -- Write decimal representation
        -- of i to output stream.
        -- Regexp: 0|(-?[1-9][0-9]*)
    put_boolean (b: BOOLEAN)
```

```
        -- Write "True" to output stream if
        -- b is true, "False" otherwise.
feature(s) from KI_CHARACTER_OUTPUT_STREAM
    -- Basic operations
    flush
        -- Flush buffered data to disk.
feature(s) from EPX_CHARACTER_OUTPUT_STREAM
    -- Output
    last_written: INTEGER
        -- How many bytes were written by last call to write?
        -- Can be less than requested for non-blocking output.
        -- Check last_blocked in that case.
    put_buffer (buf: STDC_BUFFER; offset, nbytes: INTEGER)
        -- More safe version of write in case you have a
        -- STDC_BUFFER object.
    write_buffer (buf: STDC_BUFFER; offset, nbytes: INTEGER)
        -- More safe version of write in case you have a
        -- STDC_BUFFER object.
feature(s) from KI_INPUT_STREAM
    -- Input
    non_blocking_read_character
        -- Read the next item in input stream.
        -- Make the result available in last_item.
    non_blocking_read_to_buffer (a_buffer: KI_BUFFER[CHARACTER]; pos, nb: INTEGER): INTEGER
        -- Fill a_buffer, starting at position pos, with
        -- at most nb items read from input stream.
        -- Return the number of items actually read.
feature(s) from KI_INPUT_STREAM
    -- Status report
    is_closable_for_reading: BOOLEAN
        -- Can current input stream be closed?
    is_open_read: BOOLEAN
        -- Can items be read from input stream?
    is_rewindable: BOOLEAN
        -- Can current input stream be rewound to return input from
        -- the beginning of the stream?
    eof: BOOLEAN
        -- True if end-of-file reached.
        -- Currently Im unsure if detection is reliable.
    valid_unread_character (a_character: CHARACTER): BOOLEAN
        -- Can a_character be put back in input stream?
feature(s) from KI_INPUT_STREAM
    -- Access
    last_character: CHARACTER
        -- Last character read
feature(s) from KI_INPUT_STREAM
    -- Basic operations
    close_for_reading
        -- Try to close input stream if it is closable. Set
        -- is_open_read to false if operation was successful.
    rewind
        -- Move input positionto the beginning of stream.
feature(s) from KI_CHARACTER_INPUT_STREAM
    -- Input
    non_blocking_read_string (nb: INTEGER)
        -- Read at most nb characters from input stream.
        -- Make the characters that have actually been read
        -- available in last_string.
    non_blocking_read_to_string (a_string: STRING; pos, nb: INTEGER): INTEGER
        -- Fill a_string, starting at position pos, with
        -- at most nb characters read from input stream.
        -- Return the number of characters actually read.
feature(s) from KI_CHARACTER_INPUT_STREAM
    -- Access
    last_string: STRING
        -- Last string read
        -- (Note: this query always return the same object.
        -- Therefore a clone should be used if the result
        -- is to be kept beyond the next call to this feature.
        -- However last_string is not shared between file objects.)
feature(s) from EPX_CHARACTER_INPUT_STREAM
    -- Access
    is_streaming: BOOLEAN
        -- Is data coming through a network stream?
feature(s) from EPX_CHARACTER_INPUT_STREAM
    -- Stream or disk file
    set_streaming (enable: BOOLEAN)
        -- Influence behaviour of certain functions if they should be
        -- optimized for data coming from disk or data coming from
        -- the network. In particular is_streaming implies that a
        -- client application is prepared to handle reads that
        -- return less than the requested number of bytes, but dont
        -- assume that means end-of-file.
feature(s) from EPX_CHARACTER_INPUT_STREAM
    -- Input
    last_read: INTEGER
        -- Last bytes read by read_buffer.
        -- Can be less than requested for non-blocking input.
        -- Check last_blocked in that case.
    read_buffer (buf: STDC_BUFFER; offset, nbytes: INTEGER)
        -- Read data into buf at offset for nbytes bytes.
        -- Number of bytes actually read are available in last_read.
        -- Thisis a more safe version of read in case you have a
        -- STDC_BUFFER object.
```

feature(*s*) **from** *KI_TEXT_INPUT_STREAM*
  -- Input
  read_line
    -- Read characters from input stream until a line separator
    -- or end of file is reached. Make the characters that have
    -- been read available in *last_string* and discard the line
    -- separator characters from the input stream.
    -- Zero characters will be read when non-blocking i/o
    -- is enabled, and *read* would block.
  read_new_line
    -- Read a line separator from input file.
    -- Make the characters making up the recognized
    -- line separator available in *last_string*,
    -- or make *last_string* empty and leave the
    -- input file unchanged if no line separator
    -- was found.
feature(*s*) **from** *KI_TEXT_INPUT_STREAM*
  -- Access
  eol: *STRING*
    -- Line separator
    -- EPX classes do not distinguish between a %R%N or just %N
    -- end-of-line. The platform may though.
feature(*s*) **from** *STDC_HANDLE*
  -- Access
  is_open: *BOOLEAN*
    -- Does *handle* contain an open handle?
  is_owner: *BOOLEAN*
    -- Does this object close the stream on *close* or *dispose*?
    -- Only for resources that are owned, are resource limits checked.
  resource_usage_can_be_increased: *BOOLEAN*
    -- Is it allowed to open another file?
feature(*s*) **from** *STDC_HANDLE*
  -- Influence ownership of the handle. Can help to influence subtile garbage collector problems
  become_owner
    -- This class will own its handle. This is the only function
    -- that actually increases the resource count.
  unown
    -- Resource will not be closed on dispose. Calling close will
    -- be forbidden. This routine may not call any other object,
    -- else it cannot be called from within dispose.
feature(*s*) **from** *STDC_HANDLE*
  -- Close
  close
    -- Close the resource.
  detach
    -- Forget the resource. Resource is not closed.
    -- You cannot read and write anymore.

```
feature(s) from STDC_HANDLE
   -- Resource
   capacity: INTEGER
      -- Number of resources that are in use by handle. For a
      -- file this is 1, for a memory handle, this is the number of
      -- bytes.
   fd: H
      -- Identifier of resource tracked by this class.
feature(s) from EPX_CHARACTER_IO_STREAM
   -- Status report
   is_closable: BOOLEAN
      -- Can current stream be closed for reading and writing?
feature(s) from ABSTRACT_FILE_DESCRIPTOR
   -- Initialization
   open (a_path: STRING; a_flags: INTEGER)
      -- Open given file with access given by flags.
   open_read (a_path: STRING)
      -- Open given file with access given by flags.
   open_write (a_path: STRING)
   open_read_write (a_path: STRING)
   open_truncate (a_path: STRING)
      -- Open file, if it exists, truncate it first.
   create_read_write (a_path: STRING)
      -- Always create a file, existing or not.
      -- Give read/write permissions to user only.
   create_write (a_path: STRING)
      -- Always create a file, existing or not.
      -- Give read/write permissions to user only.
   create_with_mode (a_path: STRING; flags, mode: INTEGER)
      -- Create a file according to flags and with mode access
      -- permissions. Make sure you have th O_CREAT flag in flags
      -- if you really want to create something!
feature(s) from ABSTRACT_FILE_DESCRIPTOR
   -- Special creation
   attach_to_socket (a_fd: INTEGER; a_become_owner: BOOLEAN)
      -- Create file descriptor with value a_fd. File descriptor
      -- will close it when a_become_owner.
   make_as_duplicate (another: ABSTRACT_FILE_DESCRIPTOR)
      -- On creation, create a duplicate from another file descriptor
      -- As normal call, closes its own descriptor first (if open) and
      -- duplicates next.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
   -- Read and write to memory block
   last_blocked: BOOLEAN
      -- Would last call to read or write block?
   read (buf: POINTER; offset, nbytes: INTEGER)
      -- Read data into buf at offset for nbytes bytes.
```

```
            -- The number of bytes actually read, is available in last_read.
        write (buf: POINTER; offset, nbytes: INTEGER)
            -- Write given data from buf at offset, for nbytes
            -- bytes. Number of actually written bytes are in
            -- last_written. last_written can be unequal to nbytes
            -- if i/o is non-blocking or some error has occurred.
    feature(s) from ABSTRACT_FILE_DESCRIPTOR
        -- Eiffel like output
        put (a: ANY)
            -- Write any Eiffel object as string using its out value.
        write_character (c: CHARACTER)
            -- Write a character.
        write_string (a_string: STRING)
            -- Write a_string to output stream.
        puts (a_string: STRING)
            -- Write a_string to output stream.
    feature(s) from ABSTRACT_FILE_DESCRIPTOR
        -- Buffered input
        read_character
            -- Sets last_character.
            -- If this routine blocks, last_character has the value
            -- %U. Therefore, if non-blocking is enabled, always check
            -- last_blocked to see if the value make sense.
        read_string (nb: INTEGER)
            -- Read at most nb characters from input stream.
            -- Make the characters that have actually been read
            -- available in last_string.
            -- Zero characters will be read when non-blocking i/o
            -- is enabled, and read would block.
    feature(s) from ABSTRACT_FILE_DESCRIPTOR
        -- File position
        seek (offset: INTEGER)
            -- Set file position to given absolute offset.
        seek_from_current (offset: INTEGER)
            -- Set file position relative to current position.
        seek_from_end (offset: INTEGER)
            -- Set file position relative to end of file.
    feature(s) from ABSTRACT_FILE_DESCRIPTOR
        -- Status report
        is_attached_to_terminal: BOOLEAN
            -- Is the handle associated with character device?
    feature(s) from ABSTRACT_FILE_DESCRIPTOR
        -- Access
        status: ABSTRACT_STATUS
            -- The status for this file descriptor. Cached value,
            -- refreshed only when file reopened.
        value: INTEGER
            -- The actual file descriptor value.
    feature(s) from ABSTRACT_FILE_DESCRIPTOR
        -- non-blocking i/o
        is_blocking_io: BOOLEAN
            require
                open: is_open_read
        set_blocking_io (enable: BOOLEAN)
            require
                supports_nonblocking_io: not enable implies supports_nonblocking_io;
                open: is_open
            ensure
                blocking_set: enable = is_blocking_io
        supports_nonblocking_io: BOOLEAN
    feature(s) from ABSTRACT_INTERNET_SOCKET
        -- Local and remote addresses
        local_address: ABSTRACT_SOCKET_ADDRESS_IN_BASE
            -- Return address used on this side to talk to remote.
        remote_address: ABSTRACT_SOCKET_ADDRESS_IN_BASE
            -- Return address used at remote side to talk to this side.
    feature(s) from ABSTRACT_TCP_SOCKET
        -- Shutdown
        shutdown_read
            -- The read-half of the connection is closed. No more data
            -- can be received on the socket and any data currently in
            -- the socket receive buffer is discarded. The process can no
            -- longer issue any of the read functions on the socket. Any
            -- data received after this call for a TCP socket is
            -- acknowledged and then silently discarded.
        shutdown_read_write
            -- The read-half and write-half of the connection are both
            -- closed. This is equivalent to calling shutdown-read and
            -- shutdown_write.
        shutdown_write
            -- The write-half of the connection is closed. In the case of
            -- TCP, this is called a half-close. Any data currently in
            -- the socket send buffer will be sent, followed by TCPs
            -- normal connection termination sequence. The process can no
            -- longer issue any of the write functions on the socket.
    feature(s) from ABSTRACT_TCP_SERVER_SOCKET
        -- Accept
        accept: ABSTRACT_TCP_SOCKET
            -- Return the next completed connection from the front of the
            -- completed connection queue. If there are no completed
            -- connections, the process is put to sleep.
            -- If the socket is non-blocking, Void will be returned and
            -- the process is not put to sleep..
        last_client_address: ABSTRACT_SOCKET_ADDRESS_IN_BASE
```

```
            -- Address of last client accepted by accept.
invariant
    open_in_sync: is_open_read or is_open_write implies is_open; -- The reverse is not true, for examples sockets
  -- closed for reading/writing, but still open.
    accessing_real_singleton: security_is_real_singleton;
    capacity_not_negative: capacity >= 0;
    valid_capacity: is_open = (capacity > 0);
    open_implies_handle_assigned: is_open = (fd /= unassigned_value);
    owned_implies_open: is_owner implies is_open;
    owned_implies_handle_assigned: is_owner implies fd /= unassigned_value;
    valid_status: not is_open implies my_status = Void;
    path_not_void: path /= Void;
    line_buffer_index_offset_ok: line_buffer /= Void implies line_buffer_index <= line_buffer.count;
    client_socket_address_not_void: client_socket_address /= Void;
end of deferred ABSTRACT_TCP_SERVER_SOCKET
```

In this chapter:

C

Short (flat) listing of POSIX classes

C.1 Short form of POSIX_ASYNC_IO_REQUEST

```
class interface POSIX_ASYNC_IO_REQUEST
creation
   make (a_fd: POSIX_FILE_DESCRIPTOR)
feature(s) from POSIX_ASYNC_IO_REQUEST
   -- creation
   make (a_fd: POSIX_FILE_DESCRIPTOR)
feature(s) from POSIX_ASYNC_IO_REQUEST
   -- request properties
   raw_pointer: POINTER
      -- Location for read or written data, usually buffer is a
      -- better idea.
   count: INTEGER
      -- number of bytes to read/write
   offset: INTEGER
```

```
          -- file offset
feature(s) from POSIX_ASYNC_IO_REQUEST
     -- set request properties
     set_buffer (a_buffer: STDC_BUFFER)
          -- set memory location to read/write from.
     set_count (a_count: INTEGER)
          -- set number of bytes to read/write
     set_offset (a_offset: INTEGER)
     set_raw_pointer (a_pointer: POINTER)
          -- set memory location to read/write from. Make sure you have
          -- called set_count first!
feature(s) from POSIX_ASYNC_IO_REQUEST
     -- basic read/write requests
     read
          -- execute async read request
     write
          -- execute async write request
feature(s) from POSIX_ASYNC_IO_REQUEST
     -- Eiffel friendly reads and writes
     last_string: STRING
          -- attempt to return buffer as an Eiffel string
          -- buffer should have a terminating byte!
     read_string
     write_string (text: STRING)
feature(s) from POSIX_ASYNC_IO_REQUEST
     -- other operations
     cancel_failed: BOOLEAN
          -- set by cancel, True if cancel request failed, probably
          -- because operation was already performed
     cancel
          -- cancel request
     synchronize
          -- force all i/o operations queued for the file descriptor
          -- associated with this request to the synchronous state.
          -- Function returns when the request has been initiated or
          -- queued to the file or device (even when the data cannot be
          -- synchronized immediately)
     synchronize_data
          -- force all i/o operations queued for the file descriptor
          -- associated with this request to the synchronous state.
          -- Function returns when the request has been initiated or
          -- queued to the file or device (even when the data cannot be
          -- synchronized immediately)
     wait_for
          -- suspend process, until request completed
feature(s) from POSIX_ASYNC_IO_REQUEST
     -- state
```

```
    buffer: STDC_BUFFER
            -- buffer where data that is being read/write comes from,
            -- unless set_pointer has been called
    fd: POSIX_FILE_DESCRIPTOR
    is_pending: BOOLEAN
            -- True if io request is still pending
    return_status: INTEGER
            -- return status of asynchronous i/o operation, equal to what
            -- the synchronous read, write of fsync would have returned
invariant
    accessing_real_singleton: security_is_real_singleton;
    valid_aiocb: aiocb /= Void;
    synced_buffer_and_raw_pointer: buffer /= Void implies buffer.ptr = raw_pointer;
end of POSIX_ASYNC_IO_REQUEST
```

C.2 Short form of POSIX_BASE

```
class interface POSIX_BASE
invariant
    accessing_real_singleton: security_is_real_singleton;
end of POSIX_BASE
```

C.3 Short form of POSIX_CHILD_PROCESS

```
deferred class interface POSIX_CHILD_PROCESS
feature(s) from POSIX_CHILD_PROCESS
    -- Childs pid
    pid: INTEGER
        -- The process identifier.
    is_pid_valid: BOOLEAN
        -- return True if this object refers to a child process, so
        -- it has an id
feature(s) from POSIX_CHILD_PROCESS
    -- Actions that parent may execute
    wait_for (suspend: BOOLEAN)
        -- Wait for this process to terminate. If suspend then we
        -- wait until the information about this process is available,
        -- else we return immediately.
        -- If suspend is False, check the running property to see
        -- if this child is really terminated.
invariant
    accessing_real_singleton: security_is_real_singleton;
    pid_known_is_not_terminated: is_pid_valid = not is_terminated;
end of deferred POSIX_CHILD_PROCESS
```

C.4 Short form of POSIX_CONSTANTS

```
class interface POSIX_CONSTANTS
feature(s) from STDC_CONSTANTS
    -- Error codes
    edom: INTEGER
        -- Math argument out of domain of function
    erange: INTEGER
        -- Math result not representable
    emfile: INTEGER
        -- Too many open files
feature(s) from STDC_CONSTANTS
    -- Standard streams
    stream_stdin: POINTER
    stream_stdout: POINTER
    stream_stderr: POINTER
feature(s) from STDC_CONSTANTS
    -- Special characters
    const_eof: INTEGER
        -- signals EOF
feature(s) from STDC_CONSTANTS
    -- I/O buffering
    iofbf: INTEGER
        -- full buffering
    iolbf: INTEGER
        -- line buffering
    ionbf: INTEGER
        -- no buffering
feature(s) from STDC_CONSTANTS
    -- file positioning
    seek_set: INTEGER
    seek_cur: INTEGER
    seek_end: INTEGER
feature(s) from STDC_CONSTANTS
    -- Signal related constants
    sig_dfl: POINTER
    sig_err: POINTER
    sig_ign: POINTER
feature(s) from STDC_CONSTANTS
    -- Signals
    sigabrt: INTEGER
    sigfpe: INTEGER
        -- erroneous arithmetic operation, such as zero divide or an
        -- operation resulting in overflow
    sigill: INTEGER
        -- illegal instruction
    sigint: INTEGER
```

-- receipt of an interactive attention signal
sigsegv: *INTEGER*
-- invalid access to storage
sigterm: *INTEGER*
feature(*s*) **from** *STDC_CONSTANTS*
-- random numbers
rand_max: *INTEGER*
-- maximum value returned by the *random* function
feature(*s*) **from** *STDC_CONSTANTS*
-- category constants
lc_ctype: *INTEGER*
lc_numeric: *INTEGER*
lc_time: *INTEGER*
lc_collate: *INTEGER*
lc_monetary: *INTEGER*
lc_all: *INTEGER*
feature(*s*) **from** *STDC_CONSTANTS*
-- various
clocks_per_sec: *INTEGER*
feature(*s*) **from** *STDC_CONSTANTS*
-- exit codes
exit_failure: *INTEGER*
-- exit status when something has gone wrong
exit_success: *INTEGER*
-- exit status upon success
feature(*s*) **from** *POSIX_CONSTANTS*
-- error codes
eagain: *INTEGER*
ewouldblock: *INTEGER*
ebadf: *INTEGER*
eexist: *INTEGER*
einprogress: *INTEGER*
eintr: *INTEGER*
enoent: *INTEGER*
-- A file or directory does not exist
enospc: *INTEGER*
-- There is no free space remaining on the device
enosys: *INTEGER*
feature(*s*) **from** *POSIX_CONSTANTS*
-- standard file numbers
stderr_fileno: *INTEGER*
stdin_fileno: *INTEGER*
stdout_fileno: *INTEGER*
feature(*s*) **from** *POSIX_CONSTANTS*
-- posix open symbolic constants
o_append: *INTEGER*
-- Set the file offset to the end-of-file prior to each write

```
    o_creat: INTEGER
        -- If the file does not exist, allow it to be created. This
        -- flag indicates that the mode argument is present in the
        -- call to open.
    o_dsync: INTEGER
        -- Write according to synchronized i/o data integrity completion
    o_excl: INTEGER
        -- Open fails if the file already exists
    o_exclusive: INTEGER
        -- Open fails if the file already exists
    o_noctty: INTEGER
        -- prevents terminal from becoming the controlling terminal
        -- for this process
    o_nonblock: INTEGER
        -- Do not wait for device or file to be ready or available
    o_rdonly: INTEGER
        -- Open for reading only
    o_rdwr: INTEGER
        -- Open fo reading and writing
    o_rsync: INTEGER
        -- Synchronized read i/o operations
    o_sync: INTEGER
        -- Write according to synchronized i/o file integrity completion
    o_trunc: INTEGER
        -- Use only on ordinary files opened for writing. It causes
        -- the file to be truncated to zero length.
    o_wronly: INTEGER
        -- Open for writing only
feature(s) from POSIX_CONSTANTS
    -- posix permission symbolic constants
    s_irusr: INTEGER
    s_iread: INTEGER
    s_iwusr: INTEGER
    s_iwrite: INTEGER
    s_ixusr: INTEGER
    s_iexec: INTEGER
    s_irgrp: INTEGER
    s_iwgrp: INTEGER
    s_ixgrp: INTEGER
    s_iroth: INTEGER
    s_iwoth: INTEGER
    s_ixoth: INTEGER
    s_isuid: INTEGER
    s_isgid: INTEGER
feature(s) from POSIX_CONSTANTS
    -- Posix accessibility constants
    f_ok: INTEGER
```

```
   r_ok: INTEGER
   w_ok: INTEGER
   x_ok: INTEGER
feature(s) from POSIX_CONSTANTS
   -- Posix signal constants
   sa_nocldstop: INTEGER
   sighup: INTEGER
      -- hangup detected on controlling terminal or death of
      -- controlling process
   signal_hangup: INTEGER
      -- hangup detected on controlling terminal or death of
      -- controlling process
   sigalrm: INTEGER
      -- Timeout signal, such as initiated by the alarm() function
      -- or see POSIX_TIMED_COMMAND
   signal_alarm: INTEGER
      -- Timeout signal, such as initiated by the alarm() function
      -- or see POSIX_TIMED_COMMAND
   sigchld: INTEGER
      -- Child process terminated or stopped
   signal_child: INTEGER
      -- Child process terminated or stopped
   sigkill: INTEGER
      -- Termination signal (cannot be caught or ignored)
   signal_kill: INTEGER
      -- Termination signal (cannot be caught or ignored)
   sigpipe: INTEGER
      -- Write on a pipe with no readers
   signal_pipe: INTEGER
      -- Write on a pipe with no readers
   sigquit: INTEGER
      -- Interactive termination signal
   signal_quit: INTEGER
      -- Interactive termination signal
   sigcont: INTEGER
      -- Continue if stopped
   signal_continue: INTEGER
      -- Continue if stopped
   sigstop: INTEGER
      -- Stop signal, cannot be caught or ignored
   signal_stop: INTEGER
      -- Stop signal, cannot be caught or ignored
   sigtstp: INTEGER
      -- Interactive stop signal
   signal_interactive_stop: INTEGER
      -- Interactive stop signal
   sigttin: INTEGER
```

```
            -- Read from control terminal attempted by a member of a
            -- background process group
        signal_terminal_in: INTEGER
            -- Read from control terminal attempted by a member of a
            -- background process group
        sigttou: INTEGER
            -- Write to control terminal attempted by a member of a
            -- background process group
        signal_terminal_out: INTEGER
            -- Write to control terminal attempted by a member of a
            -- background process group
    feature(s) from POSIX_CONSTANTS
        -- sigprocmask how values
        sig_block: INTEGER
        sig_unblock: INTEGER
        sig_setmask: INTEGER
    feature(s) from POSIX_CONSTANTS
        -- Posix pathconf constants
        pc_name_max: INTEGER
            -- The maximum length of a filename for this directory
    feature(s) from POSIX_CONSTANTS
        -- terminal i/o local mode flags
        isig: INTEGER
        icanon: INTEGER
        echo: INTEGER
            -- If set, input characters are echoed back to the terminal
        echoe: INTEGER
        echok: INTEGER
        echonl: INTEGER
        noflsh: INTEGER
        tostop: INTEGER
        iexten: INTEGER
    feature(s) from POSIX_CONSTANTS
        -- set terminal settings options
        tcsanow: INTEGER
        tcsadrain: INTEGER
        tcsaflush: INTEGER
    feature(s) from POSIX_CONSTANTS
        -- Semaphore constants
        sem_value_max: INTEGER
            -- Valid maximum initial value for a semaphore.
    feature(s) from POSIX_CONSTANTS
        -- terminal baud rates
        b0: INTEGER
        b50: INTEGER
        b75: INTEGER
        b110: INTEGER
        b134: INTEGER
        b150: INTEGER
        b200: INTEGER
        b300: INTEGER
        b600: INTEGER
        b1200: INTEGER
        b1800: INTEGER
        b2400: INTEGER
        b4800: INTEGER
        b9600: INTEGER
        b19200: INTEGER
        b38400: INTEGER
        b57600: INTEGER
        b115200: INTEGER
        b230400: INTEGER
    feature(s) from POSIX_CONSTANTS
        -- terminal i/o control mode constants
        csize: INTEGER
        cs5: INTEGER
        cs6: INTEGER
        cs7: INTEGER
        cs8: INTEGER
        cstopb: INTEGER
        cread: INTEGER
        parenb: INTEGER
        parodd: INTEGER
        hupcl: INTEGER
        clocal: INTEGER
    feature(s) from POSIX_CONSTANTS
        -- terminal i/o input control flags
        ignbrk: INTEGER
        brkint: INTEGER
        ignpar: INTEGER
        parmrk: INTEGER
        inpck: INTEGER
        istrip: INTEGER
        inlcr: INTEGER
        igncr: INTEGER
        icrnl: INTEGER
        ixon: INTEGER
        ixoff: INTEGER
    feature(s) from POSIX_CONSTANTS
        -- category constants
        lc_messages: INTEGER
    feature(s) from POSIX_CONSTANTS
        -- pathname variable values
        max_input: INTEGER
        -- Minimum number of bytes for which space will be available
        -- in a terminal input queue; therefore, the maximum number
        -- of bytes a portable application may required to be typed
        -- as input before eading them
    name_max: INTEGER
        -- Maximum number of bytes in a file name
    path_max: INTEGER
        -- Maximum number of bytes in a pathname
    pipe_buf: INTEGER
        -- Maximum number of bytes that can be written atomically
        -- when writing to a pipe.
feature(s) from POSIX_CONSTANTS
    -- invariant values
    ssize_max: INTEGER
        -- The maximum value that can be stored in an object of type ssize_t
end of POSIX_CONSTANTS
```

C.5 Short form of POSIX_CURRENT_PROCESS

```
class interface POSIX_CURRENT_PROCESS
feature(s) from STDC_CURRENT_PROCESS
   -- My standard input/output/error
   stdin: POSIX_TEXT_FILE
   stdout: POSIX_TEXT_FILE
   stderr: POSIX_TEXT_FILE
feature(s) from STDC_CURRENT_PROCESS
   -- various
   clock: INTEGER
      -- return approximation of processor time used by the
      -- program, or -1 if unknown
feature(s) from STDC_CURRENT_PROCESS
   -- Random numbers
   random: INTEGER
      -- Returns a pseudo-random integer between 0 and RAND_MAX.
   set_random_seed (a_seed: INTEGER)
      -- Sets a_seed as the seed for a new sequence of
      -- pseudo-random integers to be returned by random. These
      -- sequences are repeatable by calling set_random_seed with
      -- the same seed value. If no seed value is provided, the
      -- random function is automatically seeded with a value of
      -- 1.
feature(s) from ABSTRACT_CURRENT_PROCESS
   -- process properties
   pid: INTEGER
      -- The process identifier.
   is_pid_valid: BOOLEAN
      -- current process id is always valid
feature(s) from ABSTRACT_CURRENT_PROCESS
   -- Every process also has standard file descriptors which might not be compatible with stdin/stdout/stderr (Wind
   fd_stdin: POSIX_FILE_DESCRIPTOR
   fd_stdout: POSIX_FILE_DESCRIPTOR
   fd_stderr: POSIX_FILE_DESCRIPTOR
feature(s) from STDC_SECURITY_ACCESSOR
   -- The singleton, available to any because its used in preconditions
   security: STDC_SECURITY
      -- Singleton entry point for security.
feature(s) from STDC_BASE
   -- errno
   errno: STDC_ERRNO
      -- Access to the variable that contains the error that occurred.
feature(s) from ABSTRACT_PROCESS
   -- Signal this process
   terminate
      -- attempt to gracefully terminate this process
```

```
feature(s) from POSIX_PROCESS
   -- signal this process
   kill (a_signal_code: INTEGER)
      -- Send signal signal_code to the process
feature(s) from POSIX_CURRENT_PROCESS
   -- POSIX locale specifics
   set_native_messages
      -- Select native language as the language in which messages
      -- are displayed
invariant
   accessing_real_singleton: security_is_real_singleton;
end of POSIX_CURRENT_PROCESS
```

C.6 Short form of POSIX_DAEMON

```
deferred class interface POSIX_DAEMON
feature(s) from POSIX_DAEMON
    -- Daemon specific actions
    detach
        -- detach from command-line, not very useful if you want to
        -- spawn multiple daemons, but you can always pass daemons to
        -- the fork routine yourself.
    after_fork
        -- Code thanks to W. Richard Stevens.
        -- If you are started from inetd, youre in big trouble
        -- already and getting deeper in the mud. For inetd there will
        -- be another method to call, perhaps init_inetd or so.
invariant
    accessing_real_singleton: security_is_real_singleton;
    pid_known_is_not_terminated: is_child_pid_valid = not is_terminated;
end of deferred POSIX_DAEMON
```

C.7 Short form of POSIX_DIRECTORY

```
class interface POSIX_DIRECTORY
creation
    make (a_directory_name: STRING)
        -- Initialize for browsing a_directory_name.
invariant
    accessing_real_singleton: security_is_real_singleton;
    directory_name_not_empty: directory_name /= Void and then not directory_name.is_empty;
    my_status_tracks_item: my_status /= Void implies my_status.path.is_equal(full_name);
end of POSIX_DIRECTORY
```

C.8 Short form of POSIX_EXEC_PROCESS

```
class interface POSIX_EXEC_PROCESS
creation
   make (a_program: STRING; a_arguments: ARRAY[STRING])
   make_capture_input (a_program: STRING; a_arguments: ARRAY[STRING])
   make_capture_output (a_program: STRING; a_arguments: ARRAY[STRING])
   make_capture_io (a_program: STRING; a_arguments: ARRAY[STRING])
      -- Why not use threedirectional i/o, because youre getting
      -- yourself in great, great trouble anyway.
      -- A bit of advice: call stdin.close before starting to call
      -- stdout.read_string and such...
   make_capture_all (a_program: STRING; a_arguments: ARRAY[STRING])
      -- Threedirectional i/o is a great way to get yourself in trouble.
feature(s) from STDC_CHILD_PROCESS
   -- Termination info
   is_terminated: BOOLEAN
      -- Is child not running any more?
   exit_code: INTEGER
      -- Low-order 8 bits of call to _exit or exit for this process.
feature(s) from ABSTRACT_CHILD_PROCESS
   -- Actions that parent may execute
   wait_for (suspend: BOOLEAN)
      -- Wait for this process to terminate. If suspend then we
      -- wait until the information about this process is available,
      -- else we return immediately.
      -- If suspend is False, check the running property to see
      -- if this child is really terminated.
feature(s) from STDC_CURRENT_PROCESS
   -- My standard input/output/error
   child_stdin: POSIX_TEXT_FILE
   child_stdout: POSIX_TEXT_FILE
   child_stderr: POSIX_TEXT_FILE
feature(s) from STDC_CURRENT_PROCESS
   -- various
   clock: INTEGER
      -- return approximation of processor time used by the
      -- program, or -1 if unknown
feature(s) from STDC_CURRENT_PROCESS
   -- Random numbers
   random: INTEGER
      -- Returns a pseudo-random integer between 0 and RAND_MAX.
   set_random_seed (a_seed: INTEGER)
      -- Sets a_seed as the seed for a new sequence of
      -- pseudo-random integers to be returned by random. These
      -- sequences are repeatable by calling set_random_seed with
      -- the same seed value. If no seed value is provided, the
```

```
        -- random function is automatically seeded with a value of
        -- 1.
feature(s) from ABSTRACT_CURRENT_PROCESS
    -- process properties
    child_pid: INTEGER
        -- The process identifier.
    is_child_pid_valid: BOOLEAN
        -- return True if this object refers to a child process, so
        -- it has an id
feature(s) from ABSTRACT_CURRENT_PROCESS
    -- Every process also has standard file descriptors which might not be compatible with stdin/stdout/stderr (Win
    child_fd_stin: POSIX_FILE_DESCRIPTOR
    child_fd_stdout: POSIX_FILE_DESCRIPTOR
    child_fd_sterr: POSIX_FILE_DESCRIPTOR
feature(s) from STDC_SECURITY_ACCESSOR
    -- The singleton, available to any because its used in preconditions
    security: STDC_SECURITY
        -- Singleton entry point for security.
feature(s) from STDC_BASE
    -- errno
    errno: STDC_ERRNO
        -- Access to the variable that contains the error that occurred.
feature(s) from ABSTRACT_PROCESS
    -- Signal this process
    terminate
        -- attempt to gracefully terminate this process
feature(s) from POSIX_PROCESS
    -- signal this process
    kill (a_signal_code: INTEGER)
        -- Send signal signal_code to the process
feature(s) from POSIX_CURRENT_PROCESS
    -- POSIX locale specifics
    set_native_messages
        -- Select native language as the language in which messages
        -- are displayed
feature(s) from ABSTRACT_EXEC_PROCESS
    -- Initialization
    make (a_program: STRING; a_arguments: ARRAY[STRING])
    make_capture_input (a_program: STRING; a_arguments: ARRAY[STRING])
    make_capture_output (a_program: STRING; a_arguments: ARRAY[STRING])
    make_capture_io (a_program: STRING; a_arguments: ARRAY[STRING])
        -- Why not use threedirectional i/o, because youre getting
        -- yourself in great, great trouble anyway.
        -- A bit of advice: call stdin.close before starting to call
        -- stdout.read_string and such...
    make_capture_all (a_program: STRING; a_arguments: ARRAY[STRING])
        -- Threedirectional i/o is a great way to get yourself in trouble.
```

```
feature(s) from ABSTRACT_EXEC_PROCESS
   -- (re)set arguments
   has_void_argument (a_arguments: ARRAY[STRING]): BOOLEAN
      -- Is one of the items in a_arguments Void?
   set_arguments (a_arguments: ARRAY[STRING])
feature(s) from ABSTRACT_EXEC_PROCESS
   -- i/o capturing
   capture_input: BOOLEAN
      -- is input captured on execute?
   capture_output: BOOLEAN
      -- is output captured on execute?
   capture_error: BOOLEAN
      -- is error captured on execute?
   set_capture_input (on: BOOLEAN)
   set_capture_output (on: BOOLEAN)
   set_capture_error (on: BOOLEAN)
   fd_stdin: POSIX_FILE_DESCRIPTOR
   fd_stdout: POSIX_FILE_DESCRIPTOR
   fd_stderr: POSIX_FILE_DESCRIPTOR
feature(s) from ABSTRACT_EXEC_PROCESS
   -- Execute
   execute
      -- Executes program_name
      -- dont forget to wait for this process to terminate
feature(s) from ABSTRACT_EXEC_PROCESS
   -- Accessible state
   program_name: STDC_PATH
      -- program to execute
   arguments: ARRAY[STRING]
      -- arguments to pass to program
feature(s) from POSIX_FORK_ROOT
   -- deferred routines
   after_fork
      -- chance for code to do something before the main execute
      -- mainly here for POSIX_DAEMON.
feature(s) from POSIX_FORK_ROOT
   -- termination info
   is_terminated_normally: BOOLEAN
      -- Has this process been terminated normally?
   is_exited: BOOLEAN
      -- Has this process been terminated normally?
   is_signalled: BOOLEAN
      -- Child process was terminated due to receipt of a signal
      -- that was not caught.
   signal_code: INTEGER
      -- Signal of process terminated abnormally or was stopped.
invariant
```

```
    accessing_real_singleton: security_is_real_singleton;
    pid_known_is_not_terminated: is_child_pid_valid = not is_terminated;
    program_name_not_empty: program_name /= Void and then not program_name.is_empty;
    arguments_not_void: arguments /= Void;
    all_arguments_not_void: not has_void_argument(arguments);
    descriptors_are_owners: (fd_stdin /= Void and then fd_stdin.is_open implies fd_stdin.is_owner) and then (fd_st
    streams_are_not_owner: (stdin /= Void implies not stdin.is_owner) and then (stdout /= Void implies not stdout
end of POSIX_EXEC_PROCESS
```

C.9 Short form of POSIX_FILE

```
deferred class interface POSIX_FILE
feature(s) from POSIX_FILE
   -- special makes
   make_from_file_descriptor (a_file_descriptor: ABSTRACT_FILE_DESCRIPTOR; a_mode: STRING)
      -- Open a stream from a given file descriptor.
      -- The stream will become leading so when the file
      -- descriptor is closed, it will not close, you have to close
      -- the stream to close the file descriptor.
invariant
   accessing_real_singleton: security_is_real_singleton;
   open_in_sync: is_open_read or is_open_write implies is_open; -- The reverse is not true, for examples sockets
 -- closed for reading/writing, but still open.
   capacity_not_negative: capacity >= 0;
   valid_capacity: is_open = (capacity > 0);
   open_implies_handle_assigned: is_open = (stream /= unassigned_value);
   owned_implies_open: is_owner implies is_open;
   owned_implies_handle_assigned: is_owner implies stream /= unassigned_value;
   last_string_valid: last_string /= Void;
   gets_buf_valid: gets_buf /= Void;
end of deferred POSIX_FILE
```

C.10 Short form of POSIX_FILE_DESCRIPTOR

```
class interface POSIX_FILE_DESCRIPTOR
creation
    open (a_path: STRING; a_flags: INTEGER)
        -- Open given file with access given by flags.
    open_read (a_path: STRING)
        -- Open given file with access given by flags.
    open_write (a_path: STRING)
    open_read_write (a_path: STRING)
    open_truncate (a_path: STRING)
        -- Open file, if it exists, truncate it first.
    create_read_write (a_path: STRING)
        -- Always create a file, existing or not.
        -- Give read/write permissions to user only.
    create_write (a_path: STRING)
        -- Always create a file, existing or not.
        -- Give read/write permissions to user only.
    create_with_mode (a_path: STRING; flags, mode: INTEGER)
        -- Create a file according to flags and with mode access
        -- permissions. Make sure you have th O_CREAT flag in flags
        -- if you really want to create something!
    make_as_duplicate (another: ABSTRACT_FILE_DESCRIPTOR)
        -- On creation, create a duplicate from another file descriptor
        -- As normal call, closes its own descriptor first (if open) and
        -- duplicates next.
    make_from_file (file: STDC_FILE)
        -- Create file descriptor from given stream
        -- The stream is leading, so this file descriptor will
        -- never close itself, unless it is made an owner.
    attach_to_fd (a_fd: INTEGER; a_become_owner: BOOLEAN)
        -- Create file descriptor with value a_fd. File descriptor
        -- will close it when a_become_owner.
feature(s) from MEMORY
    dispose
        -- Close handle if owner.
feature(s) from KI_OUTPUT_STREAM
    -- Output
    put_character (c: CHARACTER)
        -- Write a character.
    append (an_input_stream: KI_INPUT_STREAM[CHARACTER])
        -- Read items of an_input_stream until the end
        -- of input is reached, and write these items to
        -- current output stream.
        -- append is safe for non-blocking descriptors.
feature(s) from KI_OUTPUT_STREAM
    -- Status report
    is_open_write: BOOLEAN
        -- Can items be written to output stream?
    is_closable_for_writing: BOOLEAN
        -- Can current output stream be closed?
feature(s) from KI_OUTPUT_STREAM
    -- Access
    path: STDC_PATH
        -- Scratch path.
feature(s) from KI_OUTPUT_STREAM
    -- Basic operations
    close_for_writing
        -- Try to close output stream if it is closable. Set
        -- is_open_write to false if operation was successful.
feature(s) from KI_CHARACTER_OUTPUT_STREAM
    -- Output
    put_string (a_string: STRING)
        -- Write a_string to output stream.
    put_integer (i: INTEGER)
        -- Write decimal representation
        -- of i to output stream.
        -- Regexp: 0|(-?[1-9][0-9]*)
    put_boolean (b: BOOLEAN)
        -- Write "True" to output stream if
        -- b is true, "False" otherwise.
feature(s) from KI_CHARACTER_OUTPUT_STREAM
    -- Basic operations
    flush
        -- Flush buffered data to disk.
feature(s) from EPX_CHARACTER_OUTPUT_STREAM
    -- Output
    last_written: INTEGER
        -- How many bytes were written by last call to write?
        -- Can be less than requested for non-blocking output.
        -- Check last_blocked in that case.
    put_buffer (buf: STDC_BUFFER; offset, nbytes: INTEGER)
        -- More safe version of write in case you have a
        -- STDC_BUFFER object.
    write_buffer (buf: STDC_BUFFER; offset, nbytes: INTEGER)
        -- More safe version of write in case you have a
        -- STDC_BUFFER object.
feature(s) from KI_INPUT_STREAM
    -- Input
    non_blocking_read_character
        -- Read the next item in input stream.
        -- Make the result available in last_item.
    non_blocking_read_to_buffer (a_buffer: KI_BUFFER[CHARACTER]; pos, nb: INTEGER): INTEGER
        -- Fill a_buffer, starting at position pos, with
```

```
          -- at most nb items read from input stream.
          -- Return the number of items actually read.
feature(s) from KI_INPUT_STREAM
     -- Status report
     is_closable_for_reading: BOOLEAN
          -- Can current input stream be closed?
     is_open_read: BOOLEAN
          -- Can items be read from input stream?
     is_rewindable: BOOLEAN
          -- Can current input stream be rewound to return input from
          -- the beginning of the stream?
     eof: BOOLEAN
          -- True if end-of-file reached.
          -- Currently Im unsure if detection is reliable.
     valid_unread_character (a_character: CHARACTER): BOOLEAN
          -- Can a_character be put back in input stream?
feature(s) from KI_INPUT_STREAM
     -- Access
     last_character: CHARACTER
          -- Last character read
feature(s) from KI_INPUT_STREAM
     -- Basic operations
     close_for_reading
          -- Try to close input stream if it is closable. Set
          -- is_open_read to false if operation was successful.
     rewind
          -- Move input positionto the beginning of stream.
feature(s) from KI_CHARACTER_INPUT_STREAM
     -- Input
     non_blocking_read_string (nb: INTEGER)
          -- Read at most nb characters from input stream.
          -- Make the characters that have actually been read
          -- available in last_string.
     non_blocking_read_to_string (a_string: STRING; pos, nb: INTEGER): INTEGER
          -- Fill a_string, starting at position pos, with
          -- at most nb characters read from input stream.
          -- Return the number of characters actually read.
feature(s) from KI_CHARACTER_INPUT_STREAM
     -- Access
     last_string: STRING
          -- Last string read
          -- (Note: this query always return the same object.
          -- Therefore a clone should be used if the result
          -- is to be kept beyond the next call to this feature.
          -- However last_string is not shared between file objects.)
feature(s) from EPX_CHARACTER_INPUT_STREAM
     -- Access
```

```
    is_streaming: BOOLEAN
        -- Is data coming through a network stream?
feature(s) from EPX_CHARACTER_INPUT_STREAM
    -- Stream or disk file
    set_streaming (enable: BOOLEAN)
        -- Influence behaviour of certain functions if they should be
        -- optimized for data coming from disk or data coming from
        -- the network. In particular is_streaming implies that a
        -- client application is prepared to handle reads that
        -- return less than the requested number of bytes, but dont
        -- assume that means end-of-file.
feature(s) from EPX_CHARACTER_INPUT_STREAM
    -- Input
    last_read: INTEGER
        -- Last bytes read by read_buffer.
        -- Can be less than requested for non-blocking input.
        -- Check last_blocked in that case.
    read_buffer (buf: STDC_BUFFER; offset, nbytes: INTEGER)
        -- Read data into buf at offset for nbytes bytes.
        -- Number of bytes actually read are available in last_read.
        -- Thisis a more safe version of read in case you have a
        -- STDC_BUFFER object.
feature(s) from KI_TEXT_INPUT_STREAM
    -- Input
    read_line
        -- Read characters from input stream until a line separator
        -- or end of file is reached. Make the characters that have
        -- been read available in last_string and discard the line
        -- separator characters from the input stream.
        -- Zero characters will be read when non-blocking i/o
        -- is enabled, and read would block.
    read_new_line
        -- Read a line separator from input file.
        -- Make the characters making up the recognized
        -- line separator available in last_string,
        -- or make last_string empty and leave the
        -- input file unchanged if no line separator
        -- was found.
feature(s) from KI_TEXT_INPUT_STREAM
    -- Access
    eol: STRING
        -- Line separator
        -- EPX classes do not distinguish between a %R%N or just %N
        -- end-of-line. The platform may though.
feature(s) from STDC_HANDLE
    -- Access
    is_open: BOOLEAN
```

```
        -- Does handle contain an open handle?
    is_owner: BOOLEAN
        -- Does this object close the stream on close or dispose?
        -- Only for resources that are owned, are resource limits checked.
    resource_usage_can_be_increased: BOOLEAN
        -- Is it allowed to open another file?
feature(s) from STDC_HANDLE
    -- Influence ownership of the handle. Can help to influence subtile garbage collector problems
    become_owner
        -- This class will own its handle. This is the only function
        -- that actually increases the resource count.
    unown
        -- Resource will not be closed on dispose. Calling close will
        -- be forbidden. This routine may not call any other object,
        -- else it cannot be called from within dispose.
feature(s) from STDC_HANDLE
    -- Close
    close
        -- Close the resource.
    detach
        -- Forget the resource. Resource is not closed.
        -- You cannot read and write anymore.
feature(s) from STDC_HANDLE
    -- Resource
    capacity: INTEGER
        -- Number of resources that are in use by handle. For a
        -- file this is 1, for a memory handle, this is the number of
        -- bytes.
    fd: H
        -- Identifier of resource tracked by this class.
feature(s) from EPX_CHARACTER_IO_STREAM
    -- Status report
    is_closable: BOOLEAN
        -- Can current stream be closed for reading and writing?
feature(s) from STDC_SECURITY_ACCESSOR
    -- The singleton, available to any because its used in preconditions
    security: STDC_SECURITY
        -- Singleton entry point for security.
feature(s) from STDC_BASE
    -- errno
    errno: STDC_ERRNO
        -- Access to the variable that contains the error that occurred.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- Initialization
    open (a_path: STRING; a_flags: INTEGER)
        -- Open given file with access given by flags.
    open_read (a_path: STRING)
```

```
          -- Open given file with access given by flags.
     open_write (a_path: STRING)
     open_read_write (a_path: STRING)
     open_truncate (a_path: STRING)
          -- Open file, if it exists, truncate it first.
     create_read_write (a_path: STRING)
          -- Always create a file, existing or not.
          -- Give read/write permissions to user only.
     create_write (a_path: STRING)
          -- Always create a file, existing or not.
          -- Give read/write permissions to user only.
     create_with_mode (a_path: STRING; flags, mode: INTEGER)
          -- Create a file according to flags and with mode access
          -- permissions. Make sure you have th O_CREAT flag in flags
          -- if you really want to create something!
feature(s) from ABSTRACT_FILE_DESCRIPTOR
     -- Special creation
     attach_to_fd (a_fd: INTEGER; a_become_owner: BOOLEAN)
          -- Create file descriptor with value a_fd. File descriptor
          -- will close it when a_become_owner.
     make_as_duplicate (another: ABSTRACT_FILE_DESCRIPTOR)
          -- On creation, create a duplicate from another file descriptor
          -- As normal call, closes its own descriptor first (if open) and
          -- duplicates next.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
     -- Read and write to memory block
     last_blocked: BOOLEAN
          -- Would last call to read or write block?
     read (buf: POINTER; offset, nbytes: INTEGER)
          -- Read data into buf at offset for nbytes bytes.
          -- The number of bytes actually read, is available in last_read.
     write (buf: POINTER; offset, nbytes: INTEGER)
          -- Write given data from buf at offset, for nbytes
          -- bytes. Number of actually written bytes are in
          -- last_written. last_written can be unequal to nbytes
          -- if i/o is non-blocking or some error has occurred.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
     -- Eiffel like output
     put (a: ANY)
          -- Write any Eiffel object as string using its out value.
     write_character (c: CHARACTER)
          -- Write a character.
     write_string (a_string: STRING)
          -- Write a_string to output stream.
     puts (a_string: STRING)
          -- Write a_string to output stream.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
```

```
    -- Buffered input
    read_character
        -- Sets last_character.
        -- If this routine blocks, last_character has the value
        -- %U. Therefore, if non-blocking is enabled, always check
        -- last_blocked to see if the value make sense.
    read_string (nb: INTEGER)
        -- Read at most nb characters from input stream.
        -- Make the characters that have actually been read
        -- available in last_string.
        -- Zero characters will be read when non-blocking i/o
        -- is enabled, and read would block.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- File position
    seek (offset: INTEGER)
        -- Set file position to given absolute offset.
    seek_from_current (offset: INTEGER)
        -- Set file position relative to current position.
    seek_from_end (offset: INTEGER)
        -- Set file position relative to end of file.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- Status report
    is_attached_to_terminal: BOOLEAN
        -- Is the handle associated with character device?
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- Access
    status: POSIX_STATUS
        -- The status for this file descriptor. Cached value,
        -- refreshed only when file reopened.
    value: INTEGER
        -- The actual file descriptor value.
feature(s) from ABSTRACT_FILE_DESCRIPTOR
    -- non-blocking i/o
    is_blocking_io: BOOLEAN
        -- Is blocking i/o enabled (default)?
    set_blocking_io (enable: BOOLEAN)
        -- Set is_blocking_io.
    supports_nonblocking_io: BOOLEAN
        -- Does this descriptor support non-blocking input/output?
        -- On POSIX systems, any descriptor does.
        -- On Windows sockets and pipes do.
feature(s) from POSIX_FILE_DESCRIPTOR
    -- Initialization
    make_from_file (file: STDC_FILE)
        -- Create file descriptor from given stream
        -- The stream is leading, so this file descriptor will
        -- never close itself, unless it is made an owner.
```

```
feature(s) from POSIX_FILE_DESCRIPTOR
    -- Close
    close_on_execute
        -- Close this descriptor when forking.
feature(s) from POSIX_FILE_DESCRIPTOR
    -- Synchronisation
    supports_file_synchronization: BOOLEAN
        -- Do we support synchronization?
    supports_data_synchronization: BOOLEAN
        -- Do we support synchronization of data without metadata?
    synchronize
        -- Synchronize the state of a file (includes synchronize_data).
    synchronize_data
        -- Synchronize the data of a file. Cheaper than
        -- synchronize, but not always supported.
feature(s) from POSIX_FILE_DESCRIPTOR
    -- Locking
    get_lock (lock_to_test: POSIX_LOCK): POSIX_LOCK
        -- Gets lock information, returns True if a lock is set on
        -- the region in a_lock. a_lock is overwritten with that lock.
    set_lock_failed: BOOLEAN
        -- Did set_lock obtain a lock?
    attempt_lock (a_lock: POSIX_LOCK)
        -- Attempt to set lock, if not possible, set
        -- set_lock_failed.
    set_lock (a_lock: POSIX_LOCK)
        -- Attempt to set lock, wait if necessary.
feature(s) from POSIX_FILE_DESCRIPTOR
    -- Access
    file_descriptor_flags: INTEGER
        -- All file descriptor bits associated with this handle.
    terminal: POSIX_TERMIOS
        -- Terminal settings.
    ttyname: STRING
        -- Terminal path name, or empty if this file descriptor does
        -- not refer to a terminal
invariant
    accessing_real_singleton: security_is_real_singleton;
    open_in_sync: is_open_read or is_open_write implies is_open; -- The reverse is not true, for examples sockets
 -- closed for reading/writing, but still open.
    capacity_not_negative: capacity >= 0;
    valid_capacity: is_open = (capacity > 0);
    open_implies_handle_assigned: is_open = (fd /= unassigned_value);
    owned_implies_open: is_owner implies is_open;
    owned_implies_handle_assigned: is_owner implies fd /= unassigned_value;
    valid_status: not is_open implies my_status = Void;
    path_not_void: path /= Void;
    line_buffer_index_offset_ok: line_buffer /= Void implies line_buffer_index <= line_buffer.count;
end of POSIX_FILE_DESCRIPTOR
```

C.11 Short form of POSIX_FILE_SYSTEM

```
class interface POSIX_FILE_SYSTEM
feature(s) from STDC_SECURITY_ACCESSOR
    -- The singleton, available to any because its used in preconditions
    security: STDC_SECURITY
        -- Singleton entry point for security.
feature(s) from STDC_BASE
    -- errno
    errno: STDC_ERRNO
        -- Access to the variable that contains the error that occurred.
feature(s) from STDC_FILE_SYSTEM
    -- Path names
    expand_path (a_path: STRING): STDC_PATH
        -- returns a new path
feature(s) from STDC_FILE_SYSTEM
    -- Rename files/directories, remove files/directories
    remove_file (a_path: STRING)
        -- calls unlink when a_path is a file, or rmdir when
        -- a_path is a directory.
        -- error when file could not be removed (and it exists)
    rename_to (current_path, new_path: STRING)
        -- Rename a file or a directory.
        -- new_path should not be an existing path.
feature(s) from STDC_FILE_SYSTEM
    -- Accessibility of files
    is_modifiable (a_path: STRING): BOOLEAN
        -- tests if file is readable and writable by this program
        -- uses real user ID and real group ID instead of effective ones
    is_readable (a_path: STRING): BOOLEAN
        -- Tests if a_path is readable by this program. a_path
        -- can be a file or a directory.
        -- Uses real user ID and real group ID instead of effective
        -- ones.
feature(s) from ABSTRACT_FILE_SYSTEM
    -- Directory access
    change_directory (a_directory: STRING)
        -- Changes the current working directory.
    current_directory: STRING
        -- The current directory.
    make_directory (a_directory: STRING)
        -- Makes a directory, only accessible by owner.
    mkdir (a_directory: STRING)
        -- Makes a directory, only accessible by owner.
    remove_directory (a_directory: STRING)
        -- Removes an empty directory, does not fail if directory
        -- does not exist
    rmdir (a_directory: STRING)
        -- Removes an empty directory, does not fail if directory
        -- does not exist
    force_remove_directory (a_directory: STRING)
        -- Removes a directory, even when not empty.
        -- I suggest you do not have hard or symbolic links in a_directory...
feature(s) from ABSTRACT_FILE_SYSTEM
    -- File statistics
    status (a_path: STRING): POSIX_STATUS_PATH
        -- Gets information about a file
    status_may_fail (a_path: STRING): ABSTRACT_STATUS_PATH
        -- Retrieve status information for a_path. a_path may or
        -- may not exist. Check Result.found to see if statistics
        -- were retrieved.
feature(s) from ABSTRACT_FILE_SYSTEM
    -- Directory browsing
    browse_directory (a_path: STRING): POSIX_DIRECTORY
        -- Get information about a directory.
feature(s) from ABSTRACT_FILE_SYSTEM
    -- Accessibility of files
    last_access_result: INTEGER
        -- value of last access test
    is_accessible (a_path: STRING; a_mode: INTEGER): BOOLEAN
        -- Is a_path accessibility using a_mode?
    access (a_path: STRING; a_mode: INTEGER): BOOLEAN
        -- Is a_path accessibility using a_mode?
    is_directory (a_path: STRING): BOOLEAN
        -- Does a_path exists and is it a directory?
    is_existing (a_path: STRING): BOOLEAN
        -- Is a_path an existing file, directory, whatever?
        -- Tests if file does exist, not if it is readable or writable by
        -- this program!
        -- Uses real user ID and real group ID instead of effective ones.
    is_empty (a_path: STRING): BOOLEAN
        -- True if file exists and has a size equal to zero.
    is_executable (a_path: STRING): BOOLEAN
        -- tests if file is executable by this program
    is_regular_file (a_path: STRING): BOOLEAN
        -- Does a_path exists and is it a regular file?
    is_writable (a_path: STRING): BOOLEAN
        -- tests if file is writable by this program
        -- uses real user ID and real group ID instead of effective ones
feature(s) from ABSTRACT_FILE_SYSTEM
    -- File system properties
    is_case_sensitive: BOOLEAN
        -- is file system case sensitive or not?
    path_separator: CHARACTER
```

```
      -- What is the path separator?
feature(s) from ABSTRACT_FILE_SYSTEM
   -- Path names
   resolved_path_name (a_path: STRING): STRING
      -- Derives from a_path an absolute pathname that names the
      -- same file, whose resolution does not involve ".", "..", or
      -- symbolic links.
   temporary_directory: STRING
      -- the temporary directory
feature(s) from ABSTRACT_FILE_SYSTEM
   -- File contents
   file_content_as_string (a_file_name: STRING): STRING
      -- Return contents of a_file_name as a STRING.
feature(s) from POSIX_FILE_SYSTEM
   -- read/write permissions
   chmod (a_path: STRING; a_mode: INTEGER)
      -- Changes file mode.
   change_mode (a_path: STRING; a_mode: INTEGER)
      -- Changes file mode.
   permissions (a_path: STRING): POSIX_PERMISSIONS
      -- return the permissions object (a new one every time!) for
      -- the given file
   set_read_only (a_path: STRING)
      -- Make given file read_only
   set_writable (a_path: STRING)
      -- Make given file read_only
feature(s) from POSIX_FILE_SYSTEM
   -- file times
   touch (a_path: STRING)
      -- Sets the modification and access times of a_path to the
      -- current time of day.
      -- File is created if it does not exist.
   utime (a_path: STRING; access_time, modification_time: POSIX_TIME)
      -- Sets file access and modification times
feature(s) from POSIX_FILE_SYSTEM
   -- further directory access
   link (existing, new: STRING)
      -- Creates a hard link to a file
   unlink (a_path: STRING)
      -- Removes a directory entry, should be a file, not a directory.
      -- its not an error if path does not exist, but all other
      -- errors are reported
feature(s) from POSIX_FILE_SYSTEM
   -- mkfifo
   create_fifo (a_path: STRING; a_mode: INTEGER)
      -- Creates a FIFO special file.
   mkfifo (a_path: STRING; a_mode: INTEGER)
```

```
        -- Creates a FIFO special file.
feature(s) from POSIX_FILE_SYSTEM
    -- Shared memory
    unlink_shared_memory_object (name: STRING)
        -- Remove a shared memory object.
invariant
    accessing_real_singleton: security_is_real_singleton;
end of POSIX_FILE_SYSTEM
```

C.12 Short form of POSIX_FORK_ROOT

```
deferred class interface POSIX_FORK_ROOT
feature(s) from STDC_CHILD_PROCESS
   -- Termination info
   is_terminated: BOOLEAN
      -- Is child not running any more?
   exit_code: INTEGER
      -- Low-order 8 bits of call to _exit or exit for this process.
feature(s) from ABSTRACT_CHILD_PROCESS
   -- Actions that parent may execute
   wait_for (suspend: BOOLEAN)
      -- Wait for this process to terminate. If suspend then we
      -- wait until the information about this process is available,
      -- else we return immediately.
      -- If suspend is False, check the running property to see
      -- if this child is really terminated.
feature(s) from STDC_CURRENT_PROCESS
   -- My standard input/output/error
   stdin: POSIX_TEXT_FILE
   stdout: POSIX_TEXT_FILE
   stderr: POSIX_TEXT_FILE
feature(s) from STDC_CURRENT_PROCESS
   -- various
   clock: INTEGER
      -- return approximation of processor time used by the
      -- program, or -1 if unknown
feature(s) from STDC_CURRENT_PROCESS
   -- Random numbers
   random: INTEGER
      -- Returns a pseudo-random integer between 0 and RAND_MAX.
   set_random_seed (a_seed: INTEGER)
      -- Sets a_seed as the seed for a new sequence of
      -- pseudo-random integers to be returned by random. These
      -- sequences are repeatable by calling set_random_seed with
      -- the same seed value. If no seed value is provided, the
      -- random function is automatically seeded with a value of
      -- 1.
feature(s) from ABSTRACT_CURRENT_PROCESS
   -- process properties
   child_pid: INTEGER
      -- The process identifier.
   is_child_pid_valid: BOOLEAN
      -- return True if this object refers to a child process, so
      -- it has an id
feature(s) from ABSTRACT_CURRENT_PROCESS
   -- Every process also has standard file descriptors which might not be compatible with stdin/stdout/stderr (Win
   fd_stdin: POSIX_FILE_DESCRIPTOR
   fd_stdout: POSIX_FILE_DESCRIPTOR
   fd_stderr: POSIX_FILE_DESCRIPTOR
feature(s) from STDC_SECURITY_ACCESSOR
   -- The singleton, available to any because its used in preconditions
   security: STDC_SECURITY
      -- Singleton entry point for security.
feature(s) from STDC_BASE
   -- errno
   errno: STDC_ERRNO
      -- Access to the variable that contains the error that occurred.
feature(s) from ABSTRACT_PROCESS
   -- Signal this process
   terminate
      -- attempt to gracefully terminate this process
feature(s) from POSIX_PROCESS
   -- signal this process
   kill (a_signal_code: INTEGER)
      -- Send signal signal_code to the process
feature(s) from POSIX_CURRENT_PROCESS
   -- POSIX locale specifics
   set_native_messages
      -- Select native language as the language in which messages
      -- are displayed
feature(s) from POSIX_FORK_ROOT
   -- deferred routines
   after_fork
      -- chance for code to do something before the main execute
      -- mainly here for POSIX_DAEMON.
   execute
      -- Start if child process.
feature(s) from POSIX_FORK_ROOT
   -- termination info
   is_terminated_normally: BOOLEAN
      -- Has this process been terminated normally?
   is_exited: BOOLEAN
      -- Has this process been terminated normally?
   is_signalled: BOOLEAN
      -- Child process was terminated due to receipt of a signal
      -- that was not caught.
   signal_code: INTEGER
      -- Signal of process terminated abnormally or was stopped.
invariant
   accessing_real_singleton: security_is_real_singleton;
   pid_known_is_not_terminated: is_child_pid_valid = not is_terminated;
end of deferred POSIX_FORK_ROOT
```

C.13 Short form of POSIX_GROUP

```
class interface POSIX_GROUP
creation
    make_from_name (a_name: STRING)
    make_from_gid (a_gid: INTEGER)
feature(s) from POSIX_GROUP
    -- creation
    make_from_name (a_name: STRING)
    make_from_gid (a_gid: INTEGER)
feature(s) from POSIX_GROUP
    -- refresh cache
    refresh
        -- refresh cache with latest info from user database
feature(s) from POSIX_GROUP
    -- queries
    name: STRING
        -- group name
    gid: INTEGER
        -- ID number
invariant
    accessing_real_singleton: security_is_real_singleton;
    valid_group: group /= default_pointer;
end of POSIX_GROUP
```

C.14 Short form of POSIX_LOCK

```
class interface POSIX_LOCK
creation
    make
feature(s) from POSIX_LOCK
    -- creation
    make
feature(s) from POSIX_LOCK
    -- members
    allow_read: BOOLEAN
        -- This is a read lock
    allow_all: BOOLEAN
        -- No lock or used to remove a lock
    allow_none: BOOLEAN
        -- This is a write lock
    start: INTEGER
    length: INTEGER
    pid: INTEGER
feature(s) from POSIX_LOCK
    -- settable members
    set_allow_read
        -- this is a read or shared lock
    set_allow_all
        -- to remove a lock
    set_allow_none
        -- this is a write or exclusive lock
    set_seek_start
        -- start is measured from the beginning of the file
    set_seek_current
        -- start is measured from the current position
    set_seek_end
        -- start is measured from the end of the file
    set_start (a_start: INTEGER)
        -- set relative offset in bytes
    set_length (a_length: INTEGER)
        -- number of bytes to lock
invariant
    accessing_real_singleton: security_is_real_singleton;
    valid_buf: buf /= Void;
    lock_type_known: allow_all or else allow_none or else allow_read;
end of POSIX_LOCK
```

C.15 Short form of POSIX_MEMORY_MAP

```
class interface POSIX_MEMORY_MAP
creation
    make (a_fd: POSIX_FILE_DESCRIPTOR; a_offset, a_size: INTEGER; a_base: POINTER; a_prot, a_flags: INTE
        -- Raw interface to mmap.
        -- This function can fail on certain system (Linux for
        -- example) if a_offset is not a multiple of PAGE_SIZE.
    make_private (a_fd: POSIX_FILE_DESCRIPTOR; a_offset, a_size: INTEGER)
        -- Make the given file descriptor. a_fd should have been opened
        -- with read/write access.
        -- This is a mapping where changes are private.
        -- a_offset denotes the offset from a_fd.
        -- This function can fail on certain system (Linux for
        -- example) if a_offset is not a multiple of PAGE_SIZE.
    make_shared (a_fd: POSIX_FILE_DESCRIPTOR; a_offset, a_size: INTEGER)
        -- Make the given file descriptor. a_fd should have been opened
        -- with read/write access.
        -- This is a mapping where changes are shared, i.e. the
        -- a_offset denotes the offset from a_fd.
        -- underlying object is also changed.
        -- This function can fail on certain system (Linux for
        -- example) if a_offset is not a multiple of PAGE_SIZE.
feature(s) from POSIX_MEMORY_MAP
    -- Initialization
    make (a_fd: POSIX_FILE_DESCRIPTOR; a_offset, a_size: INTEGER; a_base: POINTER; a_prot, a_flags: INTE
        -- Raw interface to mmap.
        -- This function can fail on certain system (Linux for
        -- example) if a_offset is not a multiple of PAGE_SIZE.
    make_private (a_fd: POSIX_FILE_DESCRIPTOR; a_offset, a_size: INTEGER)
        -- Make the given file descriptor. a_fd should have been opened
        -- with read/write access.
        -- This is a mapping where changes are private.
        -- a_offset denotes the offset from a_fd.
        -- This function can fail on certain system (Linux for
        -- example) if a_offset is not a multiple of PAGE_SIZE.
    make_shared (a_fd: POSIX_FILE_DESCRIPTOR; a_offset, a_size: INTEGER)
        -- Make the given file descriptor. a_fd should have been opened
        -- with read/write access.
        -- This is a mapping where changes are shared, i.e. the
        -- a_offset denotes the offset from a_fd.
        -- underlying object is also changed.
        -- This function can fail on certain system (Linux for
        -- example) if a_offset is not a multiple of PAGE_SIZE.
feature(s) from POSIX_MEMORY_MAP
    -- Cleanup
    dispose
```

```
        -- Close handle if owner.
feature(s) from POSIX_MEMORY_MAP
    -- Unmap
    close
        -- Remove the mapping.
feature(s) from POSIX_MEMORY_MAP
    -- State
    offset: INTEGER
        -- Offset from file.
    fd: POSIX_FILE_DESCRIPTOR
        -- The file that is mapped.
invariant
    accessing_real_singleton: security_is_real_singleton;
    capacity_not_negative: capacity >= 0;
    valid_capacity: is_allocated = (capacity > 0);
    open_implies_handle_assigned: is_allocated = (ptr /= unassigned_value);
    owned_implies_open: is_owner implies is_allocated;
    owned_implies_handle_assigned: is_owner implies ptr /= unassigned_value;
    size_positive: is_open implies capacity > 0;
    ptr_valid: is_open implies ptr /= default_pointer and not is_open implies ptr = default_pointer;
    offset_not_negative: offset >= 0;
end of POSIX_MEMORY_MAP
```

C.16 Short form of POSIX_PERMISSIONS

```
deferred class interface POSIX_PERMISSIONS
feature(s) from POSIX_PERMISSIONS
   apply
      -- make permissions changes (if any) permanent
   refresh
      -- synchronize with permission changes possibly made on disk
feature(s) from POSIX_PERMISSIONS
   -- query mode
   allow_anyone_execute: BOOLEAN
      -- anyone allowed to execute the file?
   allow_anyone_read: BOOLEAN
      -- anyone allowed to read the file?
   allow_anyone_read_write: BOOLEAN
      -- anyone allowed to read and write the file?
   allow_anyone_write: BOOLEAN
      -- anyone allowed to write the file?
   allow_group_execute: BOOLEAN
      -- process with a group ID that matches the files group
      -- allowed to execute the file?
   allow_group_read: BOOLEAN
      -- process with a group ID that matches the files group
      -- allowed to read the file?
   allow_group_read_write: BOOLEAN
      -- process with a group ID that matches the files group
      -- allowed to read the file?
   allow_group_write: BOOLEAN
      -- process with a group ID that matches the files group
      -- allowed to write the file?
   allow_owner_execute: BOOLEAN
      -- owner allowed to execute the file
   allow_read: BOOLEAN
   allow_owner_read: BOOLEAN
   allow_read_write: BOOLEAN
   allow_owner_read_write: BOOLEAN
   allow_write: BOOLEAN
   allow_owner_write: BOOLEAN
   is_set_group_id: BOOLEAN
      -- group ID set on execution?
   is_set_gid: BOOLEAN
      -- group ID set on execution?
   is_set_user_id: BOOLEAN
      -- user ID set on execution?
   is_set_uid: BOOLEAN
      -- user ID set on execution?
feature(s) from POSIX_PERMISSIONS
```

```
    -- set permissions
    set_allow_anyone_execute (allow: BOOLEAN)
        -- give anyone execute permission
    set_allow_anyone_read (allow: BOOLEAN)
        -- give anyone read permission
    set_allow_anyone_read_write (allow: BOOLEAN)
        -- give anyone read and write permissions
    set_allow_anyone_write (allow: BOOLEAN)
        -- give anyone write permission
    set_allow_group_execute (allow: BOOLEAN)
        -- give group execute permission
    set_allow_group_read (allow: BOOLEAN)
        -- give group read permission
    set_allow_group_read_write (allow: BOOLEAN)
        -- give group read and write permission
    set_allow_group_write (allow: BOOLEAN)
        -- give group write permission
    set_allow_owner_execute (allow: BOOLEAN)
        -- give owner execute permission
    set_allow_read (allow: BOOLEAN)
        -- give read permission
    set_allow_owner_read (allow: BOOLEAN)
        -- give read permission
    set_allow_read_write (allow: BOOLEAN)
        -- give read/write permission
    set_allow_write (allow: BOOLEAN)
        -- give write permission
    set_allow_owner_write (allow: BOOLEAN)
        -- give write permission
feature(s) from POSIX_PERMISSIONS
    -- direct access to Unix fields
    uid: INTEGER
        -- id of object owner, always 0 on NT
    owner_id: INTEGER
        -- id of object owner, always 0 on NT
    gid: INTEGER
        -- id of group, always 0 on NT
    group_id: INTEGER
        -- id of group, always 0 on NT
    mode: INTEGER
        -- the bit coded Unix mode field
feature(s) from POSIX_PERMISSIONS
    -- set owner and group
    set_owner_id (a_owner_id: INTEGER)
        -- change the owner
    set_group_id (a_group_id: INTEGER)
        -- change the group
```

```
invariant
    accessing_real_singleton: security_is_real_singleton;
end of deferred POSIX_PERMISSIONS
```

C.17 Short form of POSIX_PIPE

```
class interface POSIX_PIPE
creation
    make
        -- Create pipe
feature(s) from POSIX_PIPE
    -- the pipe
    fdin: POSIX_FILE_DESCRIPTOR
    fdout: POSIX_FILE_DESCRIPTOR
invariant
    accessing_real_singleton: security_is_real_singleton;
    valid_pipe: fdin /= Void and fdout /= Void;
end of POSIX_PIPE
```

C.18 Short form of POSIX_SEMAPHORE

```
class interface POSIX_SEMAPHORE
feature(s) from POSIX_SEMAPHORE
    -- commands
    attempt_acquire
        -- Lock the semaphore only if it is not locked. If it is locked
        -- by some process, this command returns immediately and the
        -- semaphore is not locked
    acquire
        -- lock the semaphore
    release
        -- unlock the semaphore
feature(s) from POSIX_SEMAPHORE
    -- queries
    is_initialized: BOOLEAN
        -- True if semaphore is initialized/opened/created
    is_locked: BOOLEAN
        -- True if this process has locked the semaphore
    supports_semaphores: BOOLEAN
        -- True if semaphores are supported
        -- most systems support unnamed semaphores, but still return False here
    value: INTEGER
        -- value of semaphore if not locked.
        -- Value is <= 0 if this semaphore is locked.
invariant
    accessing_real_singleton: security_is_real_singleton;
    sem_value_valid: sem_value /= Void;
end of POSIX_SEMAPHORE
```

C.19 Short form of POSIX_SIGNAL

```
class interface POSIX_SIGNAL
creation
   make (a_value: INTEGER)
feature(s) from POSIX_SIGNAL
   -- Initialization
   make (a_value: INTEGER)
feature(s) from POSIX_SIGNAL
   -- Set signal properties, make effective with apply
   apply
      -- Make changes effective.
   set_child_stop (stop: BOOLEAN)
      -- Generate SIGCHLD when children stop.
   set_default_action
      -- Install signal-specific default action when apply is called.
   set_ignore_action
      -- Ignore signal when apply is called..
   set_handler (a_handler: STDC_SIGNAL_HANDLER)
      -- Install ones own signal handler when apply is called.
   set_mask (a_mask: POSIX_SIGNAL_SET)
feature(s) from POSIX_SIGNAL
   -- signal functions
   raise_in (a_pid: INTEGER)
      -- Raise the signal in the given process.
feature(s) from POSIX_SIGNAL
   -- Signal state
   child_stop: BOOLEAN
      -- generate SIGCHLD when children stop
   handler: POINTER
      -- pointer to function which catches this signal
   is_defaulted: BOOLEAN
      -- signal is handled by its specific default action
   is_ignored: BOOLEAN
      -- signal is ignored
   is_ignorable: BOOLEAN
      -- True if this signal is ignorable, either it is so by
      -- default or it may be set so.
   mask: POSIX_SIGNAL_SET
   refresh
      -- get latest state for this signal
invariant
   accessing_real_singleton: security_is_real_singleton;
   accessing_real_singleton: signal_switch_is_real_singleton;
   valid_signal_value: value >= 1;
   has_memory: sigaction /= Void;
end of POSIX_SIGNAL
```

C.20 Short form of POSIX_SIGNAL_SET

```
class interface POSIX_SIGNAL_SET
creation
    make_empty
        -- make an initially empty signal set
    make_full
        -- make a set where all signals are enabled
    make_pending
        -- this signal set will be the set of signals that are blocked
        -- and pending
feature(s) from POSIX_SIGNAL_SET
    -- creation, make a set
    make_empty
        -- make an initially empty signal set
    make_full
        -- make a set where all signals are enabled
    make_pending
        -- this signal set will be the set of signals that are blocked
        -- and pending
feature(s) from POSIX_SIGNAL_SET
    -- change a set
    extend (signo: INTEGER)
        -- add signal to set
    put (signo: INTEGER)
        -- add signal to set
    prune (signo: INTEGER)
        -- remove the signal from the set
    wipe_out
        -- remove all items
feature(s) from POSIX_SIGNAL_SET
    -- commands to do something with set
    add_to_blocked_signals
        -- Add the signals to the set of blocked signals
    remove_from_blocked_signals
        -- Remove the signals from the set of blocked signals
    set_blocked_signals
        -- Set the set of blocked signals to this set
    suspend
        -- Suspend process, until delivery of a signal whose action
        -- is either to execute a signal-catching function or to
        -- terminate the process
feature(s) from POSIX_SIGNAL_SET
    -- queries
    has (signo: INTEGER): BOOLEAN
        -- is signal signo in the set
invariant
```

```
    accessing_real_singleton: security_is_real_singleton;
    have_set: set /= Void;
end of POSIX_SIGNAL_SET
```

C.21 Short form of POSIX_STATUS

```
deferred class interface POSIX_STATUS
feature(s) from POSIX_STATUS
    -- stat members
    is_block_special: BOOLEAN
        -- True if block-special file
    ino: INTEGER
    inode: INTEGER
    permissions: POSIX_PERMISSIONS
        -- file permissions
        ensure
            valid_result: Result /= Void
feature(s) from POSIX_STATUS
    -- direct access to the unix fields, not recommended
    unix_gid: INTEGER
    unix_uid: INTEGER
invariant
    accessing_real_singleton: security_is_real_singleton;
    stat_not_void: stat /= Void and then stat.capacity >= abstract_stat_size;
end of deferred POSIX_STATUS
```

C.22 Short form of POSIX_SYSTEM

```
class interface POSIX_SYSTEM
feature(s) from POSIX_SYSTEM
    -- Sysconf queries, run-time determined
    child_max: INTEGER
        -- The number of simultaneous processes per real user ID.
    clock_ticks: INTEGER
        -- The number of clock ticks per second.
    has_job_control: BOOLEAN
        -- Job control functions are supported.
    has_saved_ids: BOOLEAN
        -- Each process has a saved set-user-ID and a saved set-group-ID.
    ngroups_max: INTEGER
        -- The number of simultaneous supplementary group IDs.
    page_size: INTEGER
        -- granularity in bytes of memory mapping and process memory locking.
    posix_version: INTEGER
        -- Indicates the 4-digit year and 2-digit month that the
        -- standard was approved.
feature(s) from POSIX_SYSTEM
    -- Compile-time determined queries
    supports_asynchronous_io: BOOLEAN
        -- True if the message passing API is supported.
    supports_file_synchronization: BOOLEAN
        -- True if file synchronization is supported.
    supports_memory_mapped_files: BOOLEAN
        -- True if memory mapped files are supported.
    supports_memory_locking: BOOLEAN
        -- True if memory locking is supported.
    supports_memlock_range: BOOLEAN
        -- True if memory range locking is supported.
    supports_memory_protection: BOOLEAN
        -- True if memory protection is supported.
    supports_message_passing: BOOLEAN
        -- True if the message passing API is supported.
    supports_priority_scheduling: BOOLEAN
        -- True if priority scheduling is supported.
    supports_semaphores: BOOLEAN
        -- True if semaphores are supported.
    supports_shared_memory_objects: BOOLEAN
        -- True if shared memory objects are supported.
    supports_synchronized_io: BOOLEAN
        -- True if synchronized io is supported.
    supports_timers: BOOLEAN
        -- True if timers are supported.
    supports_threads: BOOLEAN
```

```
            -- True if thread are supported.
invariant
    accessing_real_singleton: security_is_real_singleton;
end of POSIX_SYSTEM
```

C.23 Short form of POSIX_TERMIOS

```
class interface POSIX_TERMIOS
creation
    make (a_fd: POSIX_FILE_DESCRIPTOR)
feature(s) from POSIX_TERMIOS
    -- Access, raw individual fields
    iflag: INTEGER
        -- Input mode flags
    oflag: INTEGER
        -- output mode flags
    cflag: INTEGER
        -- control mode flags
    lflag: INTEGER
        -- local mode flags
feature(s) from POSIX_TERMIOS
    -- More friendly settings
    is_input_echoed: BOOLEAN
        -- are input characters echoed back to the terminal?
    is_receiving: BOOLEAN
        -- If false, no characters are received
    set_echo_input (enable: BOOLEAN)
    set_echo_new_line (enable: BOOLEAN)
    set_input_control (enable: BOOLEAN)
        -- enable start/stop input control
    set_receive (enable: BOOLEAN)
feature(s) from POSIX_TERMIOS
    -- line control functions
    flush_input
        -- Discards all data that has been received but not read.
    drain
        -- Wait for all output to be transmitted to the terminal.
    send_break
        -- sends a break to the terminal
feature(s) from POSIX_TERMIOS
    -- Get/set baudrates as symbols
    input_speed: INTEGER
        -- The terminal input baud rate as symbolic value.
    output_speed: INTEGER
        -- The terminal output baud rate as symbolic value.
    set_input_speed (new_rate: INTEGER)
        -- Set terminal input baud rate, new_rate is one of the
        -- BXXXX constants
    set_output_speed (new_rate: INTEGER)
        -- Set terminal output baud rate, new_rate is one of the
        -- BXXXX constants
feature(s) from POSIX_TERMIOS
```

```
    -- symbol to baud rate conversions
    speed_to_baud_rate (symbol: INTEGER): INTEGER
        -- Given a baud rate symbol, the real baud rate is returned.
feature(s) from POSIX_TERMIOS
    -- Apply/refresh state
    apply_now
        -- Change occurs immediately.
    apply_drain
        -- Change occurs after all output written to fd has been
        -- transmitted. This function should be used when changing
        -- parameters that affect output.
    apply_flush
        -- Change occurs after all output written to fd has been
        -- transmitted. All input that has been received but not
        -- read, is discarded before the change is made.
    refresh
        -- Get terminal settings currently in effect.
feature(s) from POSIX_TERMIOS
    -- Access
    fd: POSIX_FILE_DESCRIPTOR
        -- The file descriptor for these terminal settings.
invariant
    accessing_real_singleton: security_is_real_singleton;
    valid_attr: attr /= Void and then attr.capacity = posix_termios_size;
    valid_fd: fd /= Void;
end of POSIX_TERMIOS
```

C.24 Short form of POSIX_TIMED_COMMAND

```
deferred class interface POSIX_TIMED_COMMAND
feature(s) from POSIX_TIMED_COMMAND
   -- Initialization
   make (a_seconds: INTEGER)
feature(s) from POSIX_TIMED_COMMAND
   -- Execution
   execute: BOOLEAN
      -- Did do_execute complete its task within seconds seconds?
feature(s) from POSIX_TIMED_COMMAND
   -- Access
   is_signal_alarm_handled: BOOLEAN
      -- Does the signal SIGNAL_ALARM cause an Eiffel exception?
feature(s) from POSIX_TIMED_COMMAND
   -- State
   remaining_seconds: INTEGER
      -- number of seconds left in previous request
   seconds: INTEGER
      -- the number of seconds available to execute the command
   set_seconds (a_seconds: INTEGER)
invariant
   accessing_real_singleton: security_is_real_singleton;
   valid_seconds: seconds >= 1;
end of deferred POSIX_TIMED_COMMAND
```

C.25 Short form of POSIX_USER

```
class interface POSIX_USER
creation
    make_from_name (a_name: STRING)
    make_from_uid (a_uid: INTEGER)
feature(s) from POSIX_USER
    -- creation
    make_from_name (a_name: STRING)
    make_from_uid (a_uid: INTEGER)
feature(s) from POSIX_USER
    -- Base commands
    refresh
        -- Refresh cache with latest info from user database.
feature(s) from POSIX_USER
    -- Access
    name: STRING
        -- login name
    uid: INTEGER
        -- ID number
    gid: INTEGER
        -- group ID number
    home_directory: STRING
        -- initial working directory
    shell: STRING
        -- initial user program
invariant
    accessing_real_singleton: security_is_real_singleton;
    valid_passwd: passwd /= default_pointer;
end of POSIX_USER
```

C.26 Short form of POSIX_USER_DATABASE

```
class interface POSIX_USER_DATABASE
feature(s) from POSIX_USER_DATABASE
   -- Access
   is_existing_uid (uid: INTEGER): BOOLEAN
      -- Returns True if this uid exists in /etc/passwd
      -- (or through NIS or whatever mechanisms that might be in use)
   is_existing_login (login: STRING): BOOLEAN
      -- Returns True if this login exists in /etc/passwd
      -- (or through NIS or whatever mechanisms that might be in use)
invariant
   accessing_real_singleton: security_is_real_singleton;
end of POSIX_USER_DATABASE
```

In this chapter:

- *Short form of SUS_CONSTANTS*
- *Short form of SUS_ENV_VAR*
- *Short form of SUS_FILE_SYSTEM*
- *Short form of SUS_HOST*
- *Short form of SUS_SERVICE*
- *Short form of SUS_SOCKET_ADDRESS*
- *Short form of SUS_SYSLOG*
- *Short form of SUS_TCP_SOCKET*

D

Short (flat) listing of Single Unix Specification classes

Classes in this appendix are based on the Single Unix Specification. They inherit from the POSIX classes. Inherited features are not shown.

D.1 Short form of SUS_CONSTANTS

```
class interface SUS_CONSTANTS
feature(s) from SUS_CONSTANTS
    -- Syslog facility codes
    log_kern: INTEGER
        -- kernel messages
    log_user: INTEGER
        -- random user-level messages
    log_mail: INTEGER
        -- mail system
    log_daemon: INTEGER
        -- system daemons
    log_auth: INTEGER
        -- security/authorization messages
    log_lpr: INTEGER
        -- line printer subsystem
    log_news: INTEGER
        -- network news subsystem
    log_uucp: INTEGER
        -- UUCP subsystem
    log_cron: INTEGER
        -- clock daemon
    log_local0: INTEGER
        -- Reserved for local use
    log_local1: INTEGER
        -- Reserved for local use
    log_local2: INTEGER
        -- Reserved for local use
    log_local3: INTEGER
```

```
          -- Reserved for local use
     log_local4: INTEGER
          -- Reserved for local use
     log_local5: INTEGER
          -- Reserved for local use
     log_local6: INTEGER
          -- Reserved for local use
     log_local7: INTEGER
          -- Reserved for local use
feature(s) from SUS_CONSTANTS
     -- Syslog open options
     log_pid: INTEGER
          -- log the pid with each message
     log_cons: INTEGER
          -- log on the console if errors in sending
     log_odelay: INTEGER
          -- delay open until first syslog() (default)
     log_ndelay: INTEGER
          -- dont delay open
feature(s) from SUS_CONSTANTS
     -- Syslog priorities
     log_emerg: INTEGER
     log_alert: INTEGER
     log_crit: INTEGER
     log_err: INTEGER
     log_warning: INTEGER
     log_notice: INTEGER
     log_info: INTEGER
     log_debug: INTEGER
feature(s) from SUS_CONSTANTS
     -- Socket kinds
     sock_dgram: INTEGER
          -- Connectionless, unreliable datagrams of fixed maximum length.
     sock_packet: INTEGER
          -- Linux specific way of getting packets at the dev level.
          -- For writing rarp and other similar things on the user
          -- level.
     sock_raw: INTEGER
          -- Raw protocol interface.
     sock_seqpacket: INTEGER
          -- Sequenced, reliable, connection-based, datagrams of fixed
          -- maximum length.
     sock_stream: INTEGER
          -- Sequenced, reliable, connection-based byte streams.
feature(s) from SUS_CONSTANTS
     -- Protocol families
     af_inet: INTEGER
```

```
			-- Internet domain sockets for use with IPv4 addresses.
		af_inet6: INTEGER
			-- Internet domain sockets for use with IPv6 addresses.
		af_unix: INTEGER
			-- UNIX domain sockets.
		af_unspec: INTEGER
			-- Unspecified.
	feature(s) from SUS_CONSTANTS
		-- Shutdown options
		shut_rd: INTEGER
			-- No more receptions.
		shut_rdwr: INTEGER
			-- No more receptions or transmissions.
		shut_wr: INTEGER
			-- No more transmissions.
	feature(s) from SUS_CONSTANTS
		-- h_errno values
		try_again: INTEGER
			-- Non-Authoritative Host not found, or SERVERFAIL.
		no_recovery: INTEGER
			-- Non recoverable errors, FORMERR, REFUSED, NOTIMP.
		no_data: INTEGER
			-- Valid name, no data record of requested type.
		no_address: INTEGER
			-- No address, look for MX record. Equal to NO_DATA.
	feature(s) from SUS_CONSTANTS
		-- Lengths of string forms of ip addresses
		inet_addrstrlen: INTEGER
			-- Length of an IPv4 string.
		inet6_addrstrlen: INTEGER
			-- Length of an IPv6 string.
	feature(s) from SUS_CONSTANTS
		-- Other constants
		somaxconn: INTEGER
			-- Maximum backlog.
	feature(s) from SUS_CONSTANTS
		-- Socket options level
		sol_socket: INTEGER
	feature(s) from SUS_CONSTANTS
		-- SOL_SOCKET option names
		so_reuseaddr: INTEGER
	feature(s) from SUS_CONSTANTS
		-- Special IPv4 addresses
		inaddr_any: INTEGER
			-- 0.0.0.0
		inaddr_broadcast: INTEGER
			-- 255.255.255.255
```

```
    inaddr_loopback: INTEGER
        -- 127.0.0.1
end of SUS_CONSTANTS
```

D.2 Short form of SUS_ENV_VAR

```
class interface SUS_ENV_VAR
creation
    make (a_name: STRING)
feature(s) from SUS_ENV_VAR
    -- Commands
    set_value (new_value: STRING)
invariant
    accessing_real_singleton: security_is_real_singleton;
end of SUS_ENV_VAR
```

D.3 Short form of SUS_FILE_SYSTEM

```
class interface SUS_FILE_SYSTEM
feature(s) from SUS_FILE_SYSTEM
    -- File statistics
    status (a_path: STRING): SUS_STATUS_PATH
        -- Return information about path.
    symbolic_link_status (a_path: STRING): SUS_STATUS
        -- Return information about path, but if it is a symbolic
        -- link, about the symbolic link instead of the referenced path
feature(s) from SUS_FILE_SYSTEM
    -- Symbolic links
    create_symbolic_link (old_path, new_path: STRING)
        -- Creates a symbolic link
    symlink (old_path, new_path: STRING)
        -- Creates a symbolic link
feature(s) from SUS_FILE_SYSTEM
    -- File system properties
    resolved_path_name (a_path: STRING): STRING
        -- Derives from a_path an absolute pathname that names the
        -- same file, whose resolution does not involve ".", "..", or
        -- symbolic links.
invariant
    accessing_real_singleton: security_is_real_singleton;
end of SUS_FILE_SYSTEM
```

D.4 Short form of SUS_HOST

```
class interface SUS_HOST
creation
    make_from_name (a_name: STRING)
        -- Initialize host from name. If name is numerical, the
        -- behaviour is not specified.
    make_from_address (an_address: ABSTRACT_IP_ADDRESS)
        -- Initialize host from ip address an_address.
        -- An attempt is made to resolve the host name using this address.
        -- Status is always found, even when reverse lookup failed.
invariant
    accessing_real_singleton: security_is_real_singleton;
    name_void_or_not_empty: name = Void or else not name.is_empty;
    has_canonical_name: found implies name /= Void = (canonical_name /= Void);
    has_at_least_one_ip_address: found = (addresses /= Void and then addresses.count > 0);
    only_non_void_addresses: found implies is_every_address_not_void;
    has_aliases: found = (aliases /= Void);
    valid_length: found implies address_length > 0;
    consistent: addresses /= Void and then addresses.count > 0 implies found;
    my_not_found_reason_valid: found = (my_not_found_reason = 0);
end of SUS_HOST
```

D.5 Short form of SUS_SERVICE

```
class interface SUS_SERVICE
creation
    make_from_name (a_name, a_protocol: STRING)
        -- Find service with a_name and optional a_protocol or raise
        -- exception.
    make_from_port (a_port: INTEGER; a_protocol: STRING)
        -- Initialize service from given a_port.
        -- Make sure to provide a a_protocol if necessary!
invariant
    accessing_real_singleton: security_is_real_singleton;
    name_void_or_not_empty: name = Void or else not name.is_empty;
    valid_port: port >= 0 and port <= 65535;
    valid_protocol: protocol = Void or else protocol.is_empty or else (protocol.is_equal(once_tcp) or protocol.is_equ
    valid_protocol_type: protocol_type = sock_stream or else protocol_type = sock_dgram;
    valid_aliases: aliases /= Void;
end of SUS_SERVICE
```

D.6 Short form of SUS_SOCKET_ADDRESS

```
class interface SUS_SOCKET_ADDRESS
"Use EPX_HOST_PORT instead."
end of SUS_SOCKET_ADDRESS
```

D.7 Short form of SUS_SYSLOG

```
class interface SUS_SYSLOG
feature(s) from SUS_SYSLOG
   -- open and close
   open (a_identification: STRING; a_format, a_facility: INTEGER)
      -- start logging with the given identification
   close
      -- stop logging
feature(s) from SUS_SYSLOG
   -- Write log messages, will auto-open if not is_open
   emergency (msg: STRING)
      -- the system is unusable
   alert (msg: STRING)
      -- action must be taken immediately
   critical (msg: STRING)
      -- critical conditions
   error (msg: STRING)
      -- error conditions
   warning (msg: STRING)
      -- warning conditions
   notice (msg: STRING)
      -- normal but significant condition
   info (msg: STRING)
      -- informational
   debug_dump (msg: STRING)
      -- Debug-level messages.
feature(s) from SUS_SYSLOG
   -- state
   identification: STRING
   format: INTEGER
   facility: INTEGER
   is_open: BOOLEAN
invariant
   accessing_real_singleton: security_is_real_singleton;
   remain_single: Current = singleton;
   have_identification: is_open implies identification /= Void and then not identification.is_empty;
end of SUS_SYSLOG
```

D.8 Short form of SUS_TCP_SOCKET

```
class interface SUS_TCP_SOCKET
creation
    attach_to_socket (a_fd: INTEGER; a_become_owner: BOOLEAN)
        -- Create file descriptor with value a_fd. File descriptor
        -- will close it when a_become_owner.
invariant
    open_in_sync: is_open_read or is_open_write implies is_open; -- The reverse is not true, for examples sockets
 -- closed for reading/writing, but still open.
    accessing_real_singleton: security_is_real_singleton;
    capacity_not_negative: capacity >= 0;
    valid_capacity: is_open = (capacity > 0);
    open_implies_handle_assigned: is_open = (fd /= unassigned_value);
    owned_implies_open: is_owner implies is_open;
    owned_implies_handle_assigned: is_owner implies fd /= unassigned_value;
    valid_status: not is_open implies my_status = Void;
    path_not_void: path /= Void;
    line_buffer_index_offset_ok: line_buffer /= Void implies line_buffer_index <= line_buffer.count;
    unassigned_value_is_error_value: unassigned_value = -1;
end of SUS_TCP_SOCKET
```

In this chapter:

- *Short form of EPX_CGI*
- *Short form of EPX_SOAP_WRITER*
- *Short form of EPX_URI*
- *Short form of EPX_XML_WRITER*
- *Short form of EPX_XHTML_WRITER*

E
Short (flat) listing of Standard C bonus classes

Classes in this appendix are based on Standard C only.

E.1 Short form of EPX_CGI

```
deferred class interface EPX_CGI
feature(s) from EPX_CGI
   -- Output
   execute
      -- To be implemented by child.
feature(s) from EPX_CGI
   -- Debug support
   dump_input
      -- Write cgi input to $TMPDIR/cgi_input.
      -- First line contains the content header, is not actually in input!
feature(s) from EPX_CGI
   -- Standard variables
   auth_type: STRING
      -- type of authentication used
   content_type: STRING
      -- MIME type of data when invoked with POST method
   content_length: INTEGER
      -- length, in bytes, of data when invoked with POST method
   gateway_interface: STRING
      -- Name and version of the gateway, for example CGI/1.1
   http_accept: STRING
      -- Contents of the Accept header line sent by the client
   http_cookie: STRING
      -- All cookies sent by the client in the form of key=value,
      -- semi-colon separated.
   http_referer: STRING
      -- Contents of the Referer header line.
   http_user_agent: STRING
      -- Name of the client program that is making the request.
```

```
    path_info: STRING
        -- Extra path information as it was passed to the server in
        -- the query URL
    path_translated: STRING
        -- Extra path information translated to a final, usable
        -- form. The Web document root is prepended to the query
        -- path, and any other path translations are executed.
    query_string: STRING
        -- The input when invoked with the GET method.
    remote_addr: STRING
        -- IP address of the client that made the request
    remote_address: STRING
        -- IP address of the client that made the request
    remote_host: STRING
        -- name of the remote computer that made the request
    remote_ident: STRING
        -- user name as given by the ident protocol
    remote_user: STRING
        -- name of the remote user that made the request
    request_method: STRING
        -- name of the method used to invoke the CGI
        -- application. Valid values are GET and POST
    script_name: STRING
        -- name of script that was invoked
    server_name: STRING
        -- domain name of the computer that is running the server software
    server_port: INTEGER
        -- TCP port number on which the server that invoked the CGI
        -- application is operating
    server_protocol: STRING
        -- name of the protocol that the server is using and the
        -- version of that protocol. The protocol name and version
        -- are separated by a forward slash with no spaces, for
        -- instance HTTP/1.0
    server_software: STRING
        -- name of the server that is handling the request
feature(s) from EPX_CGI
    -- CGI headers
    content_text_html
    content_text_plain
    finish_header
        -- Finish the header by emitting an empty line.
        -- If cookies have been set, they are written as well.
    location (a_url: STRING)
        -- Redirect to a_url by emitting a Location header.
feature(s) from EPX_CGI
    -- Cookies
    cookies: DS_HASH_TABLE[EPX_HTTP_COOKIE,STRING]
        -- Cookies that will be returned to the browser.
    set_cookie (a_cookie: EPX_HTTP_COOKIE)
        -- Add a new cookie that will be send to the browser then
        -- context_text_html is called.
feature(s) from EPX_CGI
    -- Server push, multipart header
    content_multipart_x_mixed_replace (boundary: STRING)
        -- Initiate server push.
    content_next_part
        -- Write boundary so next part of multipart msg can be written.
    content_multipart_end
        -- Write boundary of multipart.
    is_multipart_message: BOOLEAN
        -- Are we writing server push, multipart output?
feature(s) from EPX_CGI
    -- Form input
    has_input: BOOLEAN
        -- Is input passed to cgi program?
    has_key (key: STRING): BOOLEAN
        -- Is key passed as parameter/form-data?
    is_meta_char (c: CHARACTER): BOOLEAN
        -- Is c a commonly used meta character?
    meta_chars: STRING
        -- Commonly used meta characters.
        -- Check if this list is correct...
    raw_value (key: STRING): STRING
        -- Returns value for key.
        -- if key does not exist, the empty string is returned.
    remove_meta_chars (s: STRING)
        -- If s contains meta characters, theyre removed.
    value (key: STRING): STRING
        -- Returns safe value for key, meta characters are removed.
invariant
    -- lower_a_code_definition: lower_a_code = (a).code
    -- Same thing for all other codes.
    -- (see "note" in indexing clause.)
    accessing_real_singleton: security_is_real_singleton;
    my_xml_not_void: my_xml /= Void;
    same_size: attributes.count = values.count;
    has_tag_stack: tags /= Void;
    comparing_references_is_not_good_enough: tags.equality_tester /= Void;
    fragment_has_no_header: is_fragment implies is_header_written;
    values_not_void: values /= Void;
    attributes_not_void: attributes /= Void;
    every_attribute_has_a_value: attributes.count = values.count;
end of deferred EPX_CGI
```

E.2 Short form of EPX_SOAP_WRITER

```
class interface EPX_SOAP_WRITER
creation
    make
        -- Create an XML document with initial capacity of 1024 characters.
    make_with_capacity (a_capacity: INTEGER)
        -- Create an XML document with initial capacity of
        -- a_capacity characters.
feature(s) from EPX_SOAP_WRITER
    -- SOAP specific calls
    start_envelope
    stop_envelope
    start_header
    stop_header
    start_body
    stop_body
feature(s) from EPX_SOAP_WRITER
    -- SOAP header attributes
    set_must_understand (value: BOOLEAN)
        -- Set the SOAP-Env:mustUnderstand attribute to value.
feature(s) from EPX_SOAP_WRITER
    -- Queries if tags started
    is_envelope_started: BOOLEAN
    is_header_started: BOOLEAN
    is_body_started: BOOLEAN
feature(s) from EPX_SOAP_WRITER
    -- SOAP tags
    soap_env_body: STRING
    soap_env_envelope: STRING
    soap_env_header: STRING
feature(s) from EPX_SOAP_WRITER
    -- SOAP name space
    soap_env: STRING
    soap_name_space: STRING
invariant
    -- lower_a_code_definition: lower_a_code = (a).code
    -- Same thing for all other codes.
    -- (see "note" in indexing clause.)
    accessing_real_singleton: security_is_real_singleton;
    my_xml_not_void: my_xml /= Void;
    same_size: attributes.count = values.count;
    has_tag_stack: tags /= Void;
    comparing_references_is_not_good_enough: tags.equality_tester /= Void;
    fragment_has_no_header: is_fragment implies is_header_written;
    values_not_void: values /= Void;
    attributes_not_void: attributes /= Void;
    every_attribute_has_a_value: attributes.count = values.count;
end of EPX_SOAP_WRITER
```

E.3 Short form of EPX_URI

```
class interface EPX_URI
creation
    make (a_reference: STRING)
        -- Create an absolute or relative URI.
    make_resolve (base: EPX_URI; a_reference: STRING)
        -- If a_reference is a partial URI, it is resolved using
        -- base.
        -- The path component in a_reference will not contain
        -- relative components like ".." if a_reference is not absolute.
feature(s) from EPX_URI
    -- Initialization.
    make (a_reference: STRING)
        -- Create an absolute or relative URI.
    make_resolve (base: EPX_URI; a_reference: STRING)
        -- If a_reference is a partial URI, it is resolved using
        -- base.
        -- The path component in a_reference will not contain
        -- relative components like ".." if a_reference is not absolute.
feature(s) from EPX_URI
    -- Status
    is_absolute: BOOLEAN
        -- Does this URI have a scheme?
    is_path_resolved: BOOLEAN
        -- Does path not contain relative components like ".."?
    is_relative: BOOLEAN
        -- Is this a relative URI?
        -- A relative uri is a URI without scheme.
    has_absolute_path: BOOLEAN
        -- Has this URI a path and does this path start with a slash?
feature(s) from EPX_URI
    -- Encoding
    uri_encoding: EPX_URL_ENCODING
        -- Encoding/decoding routines and tests.
feature(s) from EPX_URI
    -- Most generic URI components
    full_reference: STRING
        -- The entire thing.
    scheme: STRING
        -- Scheme used, like "http" or "ftp", anything before the :.
    scheme_specific_part: STRING
        -- Interpretation depends on scheme, everything after the :
        -- and before the ?
feature(s) from EPX_URI
    -- If URI has a hierarchical relationships within the namespace
    authority: STRING
```

```
        -- Authority part of scheme_specific_part, usually a host name.
        -- It can be more complex however like: <userinfo>@<host>:<port>.
        -- Use parse_authority to split authority in these
        -- components if that is applicable for the protocol.
    path: STRING
        -- Path in scheme_specific_part, consisting of names
        -- separated by slashes.
    query: STRING
        -- Anything after the ? if present, else Void
    fragment: STRING
        -- The part after the # if present, else Void
feature(s) from EPX_URI
    -- If authority is <userinfo>@<host>:<port>
    user_info: STRING
        -- Usually a user name.
    host: STRING
        -- hostname or IP4 address. IP6 addresses are explicitly not
        -- supported by RFC 2396
    port: INTEGER
        -- TCP port, 0 if no port present.
    is_server_authority: BOOLEAN
        -- True if authority can be parsed as:
        -- [ userinfo @ ] host [ : port ]
        -- and port, if present, is an integer.
    parse_authority (default_port: INTEGER)
        -- Assume authority can be parsed as:
        -- [ userinfo @ ] host [ : port ].
        -- If assumption is untrue, you get a nice exception...
        -- default_port is 0 means no default.
feature(s) from EPX_URI
    -- Set url components
    add_key_value (key, value: STRING)
        -- Add a key=value pair to query. value is adding in
        -- escaped form.
    set_path (a_path: STRING)
        -- Set path.
    set_query (a_query: STRING)
        -- Set query.
    unescape_components
        -- Unescape the path, host and user_info components.
invariant
    scheme_void_or_not_empty: scheme = Void or else not scheme.is_empty;
    scheme_is_valid: scheme /= Void implies uri_encoding.is_valid_scheme(scheme);
    either_absolute_or_relative: is_absolute xor is_relative;
    full_reference_not_empty: full_reference /= Void and then not full_reference.is_empty;
    full_reference_is_valid: not uri_encoding.has_excluded_characters(full_reference); -- Im really unsure if these co
 -- Constraints on elements of a parsed URI.
```

```
    valid_authority: authority = Void or else not authority.is_empty;
    path_void_or_not_empty: path = Void or else not path.is_empty;
    valid_path: path /= Void implies not (path.has(’?’) or path.has(’#’));
    query_void_or_not_empty: query = Void or else not query.is_empty;
    valid_query: query = Void or else not query.has(’#’);
    fragment_void_or_not_empty: fragment = Void or else not fragment.is_empty;
    vaid_fragment: fragment = Void or else not fragment.has(’#’); -- Contraints on parsed authority
    user_info_occurs_in_authority: user_info /= Void implies authority.substring_index(user_info,1) /= 0;
    host_occurs_in_authority: host /= Void implies authority.substring_index(host,1) /= 0;
    valid_port: port >= 0 and port <= 65535;
end of EPX_URI
```

E.4 Short form of EPX_XML_WRITER

```
class interface EPX_XML_WRITER
creation
    make
        -- Create an XML document with initial capacity of 1024 characters.
    make_with_capacity (a_capacity: INTEGER)
        -- Create an XML document with initial capacity of
        -- a_capacity characters.
    make_fragment
        -- Create an XML fragment (document without header) with
        -- initial capacity of 1024 characters.
    make_fragment_with_capacity (a_capacity: INTEGER)
        -- Create an XML fragment (document without header) with
        -- initial capacity of a_capacity characters.
feature(s) from EPX_XML_WRITER
    -- Initialization
    make
        -- Create an XML document with initial capacity of 1024 characters.
    make_fragment
        -- Create an XML fragment (document without header) with
        -- initial capacity of 1024 characters.
    make_with_capacity (a_capacity: INTEGER)
        -- Create an XML document with initial capacity of
        -- a_capacity characters.
    make_fragment_with_capacity (a_capacity: INTEGER)
        -- Create an XML fragment (document without header) with
        -- initial capacity of a_capacity characters.
feature(s) from EPX_XML_WRITER
    -- Status
    is_a_parent (tag: STRING): BOOLEAN
        -- Is tag the current element, or is it a parent of the
        -- current tag at some point?
    is_element_with_data: BOOLEAN
        -- Has data been added to this element or in case this
        -- element has not yet been written, has data been added to
        -- its parents element?
    is_fragment: BOOLEAN
        -- Is the XML document being created a fragment?
    is_header_written: BOOLEAN
        -- Is the XML header is written or is this a fragment that
        -- does not need a header?
    is_ns_started (a_name_space, a_tag: STRING): BOOLEAN
        -- Is name_space:tag the current element?
    is_started (a_tag: STRING): BOOLEAN
        -- Is tag the current element?
    is_tag_started: BOOLEAN
```

```
          -- Is there an unclosed element?
feature(s) from EPX_XML_WRITER
     -- Access
     unfinished_xml: STRING
          -- The xml in progress
     as_string: STRING
          -- The result as plain STRING
     as_uc_string: UC_STRING
          -- The result as Unicode string, i.e. UC_STRING
feature(s) from EPX_XML_WRITER
     -- Influence state
     clear
          -- Start fresh.
feature(s) from EPX_XML_WRITER
     -- Commands that expand xml
     add_header (encoding: STRING)
          -- Add the XML header, document is encoded in
          -- encoding. Making sure this encoding is followed, is the
          -- responsibility of the client.
     add_header_iso_8859_1_encoding
          -- Document is iso-8859-1 encoded.
     add_header_utf_8_encoding
          -- Document is utf8 encoded.
     add_data (data: STRING)
          -- Write data in the current tag.
          -- Invalid characters like < or > are quoted.
          -- Use add_raw if you dont want quoting.
     puts (data: STRING)
          -- Write data in the current tag.
          -- Invalid characters like < or > are quoted.
          -- Use add_raw if you dont want quoting.
     add_entity (an_entity_name: STRING)
          -- Write entity name as element data.
     add_raw (raw_data: STRING)
          -- Write data straight in the current tag, meta characters
          -- are not quoted, control characters are not checked, etc.
     add_system_doctype (root_tag, system_id: STRING)
          -- Add a <!DOCTYPE element.
          -- Only allowed when no tags have been written.
     add_tag (tag, data: STRING)
          -- Shortcut for add_tag, add_data and stop_tag.
     add_ns_tag (name_space, tag, data: STRING)
          -- Shortcut for add_ns_tag, add_data and stop_tag.
     get_attribute (attribute: STRING): STRING
          -- Get contents of attribute attribute for
          -- current tag. attribute may include a name space.
          -- Returns Void if attribute doesnt exist
```

```
    put (a: ANY)
        -- Write data within the current tag.
    put_new_line
        -- Add a new line in the current tag.
    set_attribute (attribute, value: STRING)
        -- Set an attribute of the current tag.
        -- attribute must be name-space less, else use set_ns_attribute.
        -- value may not contain an entity reference.
        -- As the attribute is not immediately written, make sure
        -- attribute and value do not change (ie are cloned or
        -- immutable).
    set_a_name_space (a_prefix, a_uri: STRING)
        -- Define a name space.
        -- As the attribute is not immediately written, make sure
        -- a_prefix and a_uri do not change (ie are cloned or
        -- immutable).
    set_default_name_space (uri: STRING)
        -- Set the default name space.
    set_ns_attribute (name_space, attribute, value: STRING)
        -- Set an attribute of the current tag.  value may not
        -- contain an entity reference. name_space is the optional
        -- prefix to be used, not the actual URI.
        -- As the attribute is not immediately written, make sure
        -- name_space, attribute and value do not change (ie
        -- are cloned or immutable).
    start_ns_tag (name_space, tag: STRING)
        -- Start a new tag in the given name_space. name_space is
        -- a prefix only, not the actual URI. If name_space is Void
        -- or empty, the tag will not get a prefix.
        -- As the tag is not immediately written, be sure that tag
        -- does not change (ie is cloned or immutable) if
        -- name_space is Void or empty.
    start_tag (tag: STRING)
        -- Start a new tag.
        -- As the tag is not immediately written, make sure tag
        -- does not change (ie is cloned or immutable).
    stop_tag
        -- Stop last started tag.
feature(s) from EPX_XML_WRITER
    -- Quote unsafe characters
    replace_content_meta_characters (s: STRING)
        -- Replace all characters in s that have a special meaning in
        -- XML. These characters are < and & and the sequence "]]>".
        -- This routine is slow when data is actually a UC_STRING
        -- and is very large. Moving bytes to the right to insert the
        -- quoting characters takes up a very long time.
feature(s) from EPX_XML_WRITER
```

```
    -- Comments
    add_comment (a_comment: STRING)
        -- Add a comment.
    start_comment
        -- Write the XML comment start tag.
    stop_comment
        -- Stop a started XML comment.
invariant
    -- lower_a_code_definition: lower_a_code = (a).code
    -- Same thing for all other codes.
    -- (see "note" in indexing clause.)
    accessing_real_singleton: security_is_real_singleton;
    my_xml_not_void: my_xml /= Void;
    same_size: attributes.count = values.count;
    has_tag_stack: tags /= Void;
    comparing_references_is_not_good_enough: tags.equality_tester /= Void;
    fragment_has_no_header: is_fragment implies is_header_written;
    values_not_void: values /= Void;
    attributes_not_void: attributes /= Void;
    every_attribute_has_a_value: attributes.count = values.count;
end of EPX_XML_WRITER
```

E.5 Short form of EPX_XHTML_WRITER

```
class interface EPX_XHTML_WRITER
creation
   make
      -- Create an XML document with initial capacity of 1024 characters.
   make_with_capacity (a_capacity: INTEGER)
      -- Create an XML document with initial capacity of
      -- a_capacity characters.
   make_fragment
      -- Create an XML fragment (document without header) with
      -- initial capacity of 1024 characters.
   make_fragment_with_capacity (a_capacity: INTEGER)
      -- Create an XML fragment (document without header) with
      -- initial capacity of a_capacity characters.
feature(s) from EPX_XHTML_WRITER
   -- overrule some xml stuff
   new_line_after_closing_tag (a_tag: STRING)
      -- Outputs a new line, called when a_tag is closed
      -- can be overridden to start a new line only occasionally
      -- For XHTML documents a new line is treated as a single
      -- space, so it can influence layout.
   new_line_before_starting_tag (a_tag: STRING)
      -- Outputs a new line, called when a_tag is about to begin.
feature(s) from EPX_XHTML_WRITER
   -- doctype
   doctype
      -- Default doctype is doctype_strict.
   doctype_frameset
      -- Output will be frame-based.
   doctype_strict
      -- Output will be strict XHTML in the ISO-8859-1 encoding.
   doctype_transitional
      -- Output will be transitional XHTML.
feature(s) from EPX_XHTML_WRITER
   -- Set well-known attribute
   set_id (a_id: STRING)
      -- Set the id attribute.
   set_xhtml_name_space
      -- Add the XHTML name space to the current tag.
feature(s) from EPX_XHTML_WRITER
   -- Page
   b_html
   e_html
feature(s) from EPX_XHTML_WRITER
   -- Header
   meta_refresh_other (time: INTEGER; url: STRING)
```

```
    b_head
    e_head
    title (a_text: STRING)
feature(s) from EPX_XHTML_WRITER
    -- Body
    b_body
    e_body
feature(s) from EPX_XHTML_WRITER
    -- Section headers
    h1 (header_text: STRING)
    h2 (header_text: STRING)
feature(s) from EPX_XHTML_WRITER
    -- Paragraph
    br
        -- break.
    br_clear_all
        -- Add break and flush all floats.
    b_p
    e_p
    p (par: STRING)
feature(s) from EPX_XHTML_WRITER
    -- Layout
    b_tt
        -- teletype writer font
    e_tt
feature(s) from EPX_XHTML_WRITER
    -- Quotes
    b_blockquote
    e_blockquote
    blockquote (a_quote: STRING)
feature(s) from EPX_XHTML_WRITER
    -- Link
    b_a (href: STRING)
    e_a
    a (href, s: STRING)
feature(s) from EPX_XHTML_WRITER
    -- Rules
    hr
        -- horizontal rule
feature(s) from EPX_XHTML_WRITER
    -- White space
    nbsp
        -- Add a non breaking white space.
feature(s) from EPX_XHTML_WRITER
    -- Verbatim
    b_pre
    e_pre
```

```
feature(s) from EPX_XHTML_WRITER
   -- Tables
   b_table
      -- Begin a table.
   e_table
      -- End a table.
   b_tr
      -- Begin a row.
   e_tr
      -- End a row.
   td (a_content: STRING)
      -- Add cell with optional contents.
   b_td
      -- Begin a column.
   e_td
      -- End a column.
   th (a_title: STRING)
      -- Add a header cell.
   b_th
      -- Begin a table header cell.
   e_th
      -- Add a table header cell.
feature(s) from EPX_XHTML_WRITER
   -- Forms
   b_form (method, action: STRING)
   b_form_get (action: STRING)
   b_form_post (action: STRING)
   e_form
   b_input (type, name: STRING)
   e_input
   hidden (name, value: STRING)
   b_button_submit (name, value: STRING)
   e_button_submit
   button_submit (name, value: STRING)
      -- Submit button.
   b_button_reset
   e_button_reset
   button_reset
   b_checkbox (name, value: STRING)
   e_checkbox
   label (a_label, a_for: STRING)
      -- Emit label tag a_label for a control with id a_for.
   b_radio (name, value: STRING)
   e_radio
   b_select (name: STRING)
   e_select
   b_option
```

```
    e_option
    option (text: STRING)
    selected_option (choice: STRING)
    b_textarea (name: STRING)
        -- Begin multiline input control.
    e_textarea
        -- End multiline input control.
    input_text (name: STRING; size: INTEGER; value: STRING)
        -- Single line input.
    b_input_text (name: STRING; size: INTEGER; value: STRING)
        -- Single line input.
    e_input_text
        -- End single line input.
    input_password (name: STRING; size: INTEGER; value: STRING)
        -- Single line password input.
feature(s) from EPX_XHTML_WRITER
    -- CSS style sheet support
    b_style
        -- Start inline style.
    e_style
    set_class (name: STRING)
        -- set attribute class
    set_style (an_inline_style: STRING)
        -- Set the style attribute.
    style_sheet (a_location, a_description, a_media: STRING)
        -- Put in a link to refer to an external style sheet on disk.
        -- a_media is the intended destination medium for style
        -- information. It may be a single media descriptor or a
        -- comma-separated list. The default value for this attribute
        -- is "screen".
    alternate_style_sheet (a_location, a_description, a_media: STRING)
        -- Put in a link to refer to an alternative style sheet.
        -- a_media is the intended destination medium for style
        -- information. It may be a single media descriptor or a
        -- comma-separated list. The default value for this attribute
        -- is "screen".
feature(s) from EPX_XHTML_WRITER
    -- Link
    link (a_href, a_forward_link_types, a_backward_link_types, a_content_type, a_title, a_media: STRING)
        -- Add a <link> element. This is used for document relationships.
feature(s) from EPX_XHTML_WRITER
    -- Divisions
    b_div
    e_div
feature(s) from EPX_XHTML_WRITER
    -- HTML tag names
    once_a: STRING
    once_blockquote: STRING
    once_body: STRING
    once_br: STRING
    once_div: STRING
    once_form: STRING
    once_h1: STRING
    once_h2: STRING
    once_h3: STRING
    once_head: STRING
    once_html: STRING
    once_input: STRING
    once_label: STRING
    once_link: STRING
    once_meta: STRING
    once_option: STRING
    once_p: STRING
    once_pre: STRING
    once_select: STRING
    once_table: STRING
    once_td: STRING
    once_textarea: STRING
    once_tr: STRING
    once_title: STRING
feature(s) from EPX_XHTML_WRITER
    -- Attribute values
    once_selected: STRING
    once_submit: STRING
    once_text: STRING
invariant
    -- lower_a_code_definition: lower_a_code = (a).code
    -- Same thing for all other codes.
    -- (see "note" in indexing clause.)
    accessing_real_singleton: security_is_real_singleton;
    my_xml_not_void: my_xml /= Void;
    same_size: attributes.count = values.count;
    has_tag_stack: tags /= Void;
    comparing_references_is_not_good_enough: tags.equality_tester /= Void;
    fragment_has_no_header: is_fragment implies is_header_written;
    values_not_void: values /= Void;
    attributes_not_void: attributes /= Void;
    every_attribute_has_a_value: attributes.count = values.count;
end of EPX_XHTML_WRITER
```

F

Short (flat) listing of network protocol bonus classes

In this chapter:

- *Short form of EPX_HOST_PORT*
- *Short form of EPX_HTTP_10_CLIENT*
- *Short form of EPX_IMAP4_CLIENT*
- *Short form of ULM_LOGGING*

Classes in this appendix build upon the abstract layer and generally need network access.

F.1 Short form of EPX_HOST_PORT

```
class interface EPX_HOST_PORT
creation
    make (a_host: EPX_HOST; a_service: EPX_SERVICE)
        -- Initialize socket for resolved host, using its first ip
        -- address.
feature(s) from STDC_SECURITY_ACCESSOR
    -- The singleton, available to any because its used in preconditions
    security: STDC_SECURITY
        -- Singleton entry point for security.
feature(s) from STDC_BASE
    -- errno
    errno: STDC_ERRNO
        -- Access to the variable that contains the error that occurred.
feature(s) from EPX_HOST_PORT
    -- Access
    host: EPX_HOST
        -- Resolved host name.
    service: EPX_SERVICE
        -- Port and protocol (udp/tcp) type.
    socket_address: ABSTRACT_SOCKET_ADDRESS_IN_BASE
        -- The socket address struct to be used by connect.
feature(s) from EPX_HOST_PORT
    -- Fill socket structure, so ptr returns something valid
    set_address (item: INTEGER)
        -- Use the ip address at item of host as the socket
        -- address.
invariant
    accessing_real_singleton: security_is_real_singleton;
    host_resolved: host /= Void and then host.found;
```

```
		has_service: service /= Void;
		socket_address_not_void: socket_address /= Void;
		address_type_matches: host.address_family = socket_address.address_family;
		port_matches: service.port = socket_address.port;
	end of EPX_HOST_PORT
```

F.2 Short form of EPX_HTTP_10_CLIENT

```
class interface EPX_HTTP_10_CLIENT
creation
   make (host_name: STRING)
      -- Prepare for request to host_name.
   make_from_port (host_name: STRING; port: INTEGER)
      -- Prepare for request.
      -- Use port is 0 to use the default port (80).
   make_from_host (a_host: EPX_HOST)
      -- Prepare for request to resolved a_host. If port is 0,
      -- the default port is taken, else the port can be overruled.
   make_from_host_and_port (a_host: EPX_HOST; port: INTEGER)
      -- Prepare for request to a_host. If port is 0, the
      -- default port is taken, else the port can be overruled.
feature(s) from EPX_HTTP_10_CLIENT
   -- Client http version
   client_version: STRING
      -- Clients version of the http protocol
feature(s) from EPX_HTTP_10_CLIENT
   -- Requests
   delete (a_request_uri: STRING)
   get (a_request_uri: STRING)
      -- Send GET request to server.
   head (a_request_uri: STRING)
      -- Send HEAD request to server.
      -- a_request_uri should not include http: and the host name, only
      -- the page that is requested. Any query and fragment parts are ok.
   options (a_request_uri: STRING)
      -- Get server options. a_request_uri is required when the
      -- request is being made to a proxy.
   post (a_request_uri: STRING; a_post_data: EPX_MIME_PART)
feature(s) from EPX_HTTP_10_CLIENT
   -- Fields that are send with a request if set
   accept: STRING
      -- What kind of output can the client handle?
      -- Examples are:
      --   Accept: text/plain; q=0.5, text/html,
      --           text/x-dvi; q=0.8, text/x-c
   user_agent: STRING
      -- Identification of client program.
      -- Common examples are:
      --   Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
      --   Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.0.0) Gecko/20020529
      --   Microsoft Internet Explorer
   set_accept (value: STRING)
      -- Set the media types which are acceptable for the response.
   set_user_agent (value: STRING)
      -- Set the client identification.
feature(s) from EPX_HTTP_10_CLIENT
   -- Response
   body: EPX_MIME_BODY_TEXT
      -- Return body as text, if applicable, else Void.
   fields: DS_HASH_TABLE[EPX_MIME_FIELD,STRING]
      -- Header fields of response.
   is_response_ok: BOOLEAN
      -- Does the returned response_code indicate success?
   part: EPX_MIME_PART
      -- The entire parsed MIME message
   read_response
      -- Read entire resonse, parse while reading.
   response_code: INTEGER
   response_phrase: STRING
   server_version: STRING
      -- Set by read_response.
feature(s) from EPX_HTTP_10_CLIENT
   -- Individual response fields, Void if not available
   location: STRING
invariant
   accessing_real_singleton: security_is_real_singleton;
   host_found: host /= Void and then host.found;
   have_address: sa /= Void;
end of EPX_HTTP_10_CLIENT
```

F.3 Short form of EPX_IMAP4_CLIENT

```
class interface EPX_IMAP4_CLIENT
creation
    make (a_host: STRING)
        -- Initialize client and try to open connection to imap server.
        -- Check is_open if could connect to server.
        -- If not, a_host might not be resolvable.
feature(s) from STDC_SECURITY_ACCESSOR
    -- The singleton, available to any because its used in preconditions
    security: STDC_SECURITY
        -- Singleton entry point for security.
feature(s) from STDC_BASE
    -- errno
    errno: STDC_ERRNO
        -- Access to the variable that contains the error that occurred.
feature(s) from EPX_IMAP4_CLIENT
    -- Open/close
    open
        -- Open connection to an imap server.
    close
        -- Close connection to imap server.
feature(s) from EPX_IMAP4_CLIENT
    -- Access
    response: EPX_IMAP4_RESPONSE
        -- Responses received by server.
    state: EPX_IMAP4_STATE
        -- Current state, one of four.
feature(s) from EPX_IMAP4_CLIENT
    -- Status
    is_open: BOOLEAN
        -- Is client connected to IMAP server?
feature(s) from EPX_IMAP4_CLIENT
    -- Not-authenticated state commands
    login (a_user_name, a_password: STRING)
        -- Login to the IMAP server using a_user_name and
        -- a_password. If login successful, then state will be
        -- set to Authenticated_state. If login was unsuccessful,
        -- see login_failure_reason for a human readable error message.
    noop
        -- Since any command can return a status update as untagged
        -- data, the NOOP command can be used as a periodic poll for
        -- new messages or message status updates during a period of
        -- inactivity. The NOOP command can also be used to reset
        -- any inactivity autologout timer on the server.
        -- A noop can be issued in any state.
feature(s) from EPX_IMAP4_CLIENT
```

```
    -- Authenticated state commands
    create_mailbox (a_mailbox_name: STRING)
        -- The CREATE command creates a mailbox with the given name.
        -- An OK response is returned only if a new mailbox with that
        -- name has been created.  It is an error to attempt to
        -- create INBOX or a mailbox with a name that refers to an
        -- extant mailbox.
    delete_mailbox (a_mailbox_name: STRING)
        -- The DELETE command permanently removes the mailbox with
        -- the given name.
    examine (a_mailbox_name: STRING)
        -- The EXAMINE command is identical to SELECT and returns the
        -- same output; however, the selected mailbox is identified
        -- as read-only. No changes to the permanent state of the
        -- mailbox, including per-user state, are permitted.
    get_delimiter
        -- Make sure response.delimiter has the correct value.
    list_all
        -- list_all returns the complete set of all names available
        -- to the client.
    list_subscribed
        -- list_subscribed returns the complete set of names that
        -- the user has declared as being "active" or "subscribed".
    select_mailbox (a_mailbox_name: STRING)
        -- The SELECT command selects a mailbox so that messages in
        -- the mailbox can be accessed.
feature(s) from EPX_IMAP4_CLIENT
    -- Selected state commands
    check_mailbox
        -- The CHECK command requests a checkpoint of the currently
        -- selected mailbox. A checkpoint refers to any
        -- implementation-dependent housekeeping associated with the
        -- mailbox (e.g. resolving the servers in-memory state of
        -- the mailbox with the state on its disk) that is not
        -- normally executed as part of each command. A checkpoint
        -- MAY take a non-instantaneous amount of real time to
        -- complete. If a server implementation has no such
        -- housekeeping considerations, CHECK is equivalent to NOOP.
        -- There is no guarantee that an EXISTS untagged response
        -- will happen as a result of CHECK.  NOOP, not CHECK, SHOULD
        -- be used for new mail polling.
    close_mailbox
        -- This command permanently removes from the currently
        -- selected mailbox all messages that have the \Deleted flag
        -- set, and returns to authenticated state from selected
        -- state.
    copy_message (sequence_number: INTEGER; to_mailbox_name: STRING)
```

```
            -- Copy message with sequence_number sequence_number to the
            -- mailbox to_mailbox_name.
        delete_message (sequence_number: INTEGER)
            -- Delete message with sequence_number sequence_number from
            -- the current mailbox.
        expunge
            -- The EXPUNGE command permanently removes all messages that
            -- have the \Deleted flag set from the currently selected
            -- mailbox.
        fetch (a_set: STRING; a_format: STRING)
            -- Fetch messages described by a_set in format described by
            -- a_format. Data is stored into a new
            -- response.current_message object.
        fetch_body (sequence_number: INTEGER)
            -- Fetch message body, return raw RFC822 body in
            -- last_body.
        fetch_header (sequence_number: INTEGER)
            -- Fetch message header, return raw RFC822 header in
            -- last_header.
        fetch_message (sequence_number: INTEGER)
            -- Fetch message, return raw RFC822 message in response.message.
        fetch_size (sequence_number: INTEGER)
            -- Fetch message, return raw RFC822 size in response.message_size.
        logout
            -- Inform the server that the client is done with the
            -- connection.
        mark_unseen (sequence_number: INTEGER)
            -- Remove the \Seen flag from the given message.
            -- It does not update current_message.flags as it runs
            -- silently.
    feature(s) from EPX_IMAP4_CLIENT
        -- Selected state queries
        is_valid_sequence_number (a_number: INTEGER): BOOLEAN
            -- Is a_number a valid sequence number for current_mailbox?
        is_valid_mailbox_name (a_name: STRING): BOOLEAN
            -- Is a_mailbox_name a valid mailbox name?
            -- It should not be empty, and it should not have the double
            -- quote character in its name.
    invariant
        accessing_real_singleton: security_is_real_singleton;
        host_name_not_empty: host_name /= Void and then not host_name.is_empty;
        state_not_void: state /= Void;
        closed_implies_unauthenticated: not is_open implies state.is_not_authenticated;
        authenticated_implies_open: not state.is_not_authenticated implies is_open;
        response_not_void: response /= Void;
        selected_state_has_current_mailbox: state.is_selected implies response.current_mailbox /= Void;
        unselected_state_has_no_current_mailbox: not state.is_selected implies response.current_mailbox = Void;
    end of EPX_IMAP4_CLIENT
```

F.4 Short form of ULM_LOGGING

This class depends on Standard C only. It is the EPX_LOG_HANDLER that is platform specific. e-POSIX provides implementations of this class for Unix through syslog and for Windows through the NT event log.

```
class interface ULM_LOGGING
creation
    make (a_handler: ULM_LOG_HANDLER; a_program_name: STRING)
        -- Start logging for program. The host name is derived from
        -- an OS specific call through a_handler.
feature(s) from ULM_LOGGING
    -- Log methods
    log_error (level: INTEGER; subsystem: STRING; error_number: INTEGER; error_message: STRING)
        -- Useful for logging errors.
    log_event (level: INTEGER; subsystem: STRING; fields: ARRAY[ULM_FIELD])
        -- Log event, consisting of one or more fields. It is the
        -- responsibility of the client to make sure the values are
        -- proper for each field.
        -- This function adds any ULM required field if not present.
        -- subsystem, if present is appended with a dot to
        -- program and written in the "PROG" field.
        -- DATE is logged in GMT.
    log_single_field (level: INTEGER; subsystem, field_name, value: STRING)
        -- Log value for field_name. value will be properly
        -- quoted if necessary. value should be in the proper
        -- format for field_name.
        -- This function adds any ULM required field.
        -- subsystem, if present is appended with a dot to
        -- program and written in the "PROG" field.
        -- in the "PROG" field.
        -- DATE is logged in GMT.
    log_message (level: INTEGER; subsystem, value: STRING)
        -- Log a simple message with the MSG field.
        -- This function adds any ULM required field.
        -- subsystem, if present is appended with a dot to
        -- program and written in the "PROG" field.
        -- DATE is logged in GMT.
feature(s) from ULM_LOGGING
    -- Queries
    is_valid_field_name (field_name: STRING): BOOLEAN
        -- Returns True if field_name is valid according to ULM spec.
        -- Basically it should consist of one or more letters and have
        -- no spaces.
    is_valid_partial_field_list (fields: ARRAY[ULM_FIELD]): BOOLEAN
        -- Contains True if fields contains at least one item, and
        -- if every item in fields is not Void and if fieldsdoes
        -- not contain a duplicate field and if fields does not
        -- contain the LVL field.
feature(s) from ULM_LOGGING
    -- Standard field names
    lvl: STRING
        -- Importance and category of the ULM.
    host: STRING
        -- Name of software component which issues the ULM.
    prog: STRING
        -- Name of the software component which issued the ULM.
    date: STRING
        -- Instantaneous date of the event.
    lang: STRING
        -- Language used for text fields. Default is english (EN).
    dur: STRING
        -- Indicates duration (in seconds) of the event.
    ps: STRING
        -- Process id which issued the ULM.
    id: STRING
        -- System reference to the concerned document.
    src_ip: STRING
        -- The IP number of the source host.
    src_fqdn: STRING
        -- Fully qualified Domain Name for the source host.
    src_name: STRING
        -- Generic name qualifying the source.
    src_port: STRING
        -- Port number for TCP, UDP or other protocol.
    src_usr: STRING
        -- User name or user id.
    src_mail: STRING
        -- Email address.
    dst_ip: STRING
        -- The IP number of the destination host.
    dst_fqdn: STRING
        -- Fully qualified Domain Name for the destination host.
    dst_name: STRING
        -- Generic name qualifying the destination.
    dst_port: STRING
        -- Port number for TCP, UDP or other protocol.
    dst_usr: STRING
        -- User name or user id.
    dst_mail: STRING
        -- Email address.
    rel_ip: STRING
        -- The IP number of the proxy/relayer host.
    rel_fqdn: STRING
        -- Fully qualified Domain Name for the proxy/relayer host.
    rel_name: STRING
        -- Generic name qualifying the proxy/relayer.
    rel_port: STRING
        -- Port number for TCP, UDP or other protocol.
    rel_usr: STRING
        -- User name or user id.
    rel_mail: STRING
        -- Email address.
    vol: STRING
        -- Volume (number of bytes) sent and received from the source
        -- point of view.
    vol_sent: STRING
        -- Volume (number of bytes) sent from the source point of view.
    vol_rcvd: STRING
        -- Volume (number of bytes) received from the source point of view.
    cnt: STRING
        -- Count (of articles, files, events) sent and received from
        -- the source point of view.
    cnt_sent: STRING
        -- Count (of articles, files, events) sent from the source
        -- point of view.
    cnt_rcvd: STRING
        -- Count (of articles, files, events) received from the
        -- source point of view.
    prog_file: STRING
        -- Name of the program source file from which the ULM was generated.
    stat: STRING
        -- State or status of the designed process. Possible values
        -- for this field may include "Failure", "Success", "Start",
        -- "End".
    tty: STRING
        -- Users physical connection to the host.
    doc: STRING
        -- Name of accessed document like the path of an ftp file,
        -- the name of a newsgroup, or the non-host part of an URL.
    prot: STRING
        -- Protocol used.
    cmd: STRING
        -- Issued command.
    msg: STRING
        -- The only field which should contain arbitrary data.
feature(s) from ULM_LOGGING
    -- Public state
    host_name: STRING
        -- Name of the host which issues the ULM.
    program_name: STRING
        -- Name of the software component which issues the ULM.
```

```
invariant
    log_level_text_lower_index_ok: log_level_text.lower = emergency;
    log_level_text_upper_index_ok: log_level_text.upper = debugging;
    accessing_real_singleton: security_is_real_singleton;
    handler_not_void: handler /= Void;
    host_name_not_empty: host /= Void and then not host.is_empty;
    program_name_not_empty: program_name /= Void and then not program_name.is_empty;
    have_my_date: my_date /= Void;
    have_my_host: my_host /= Void;
    have_my_prog: my_prog /= Void;
    have_my_lvl: my_lvl /= Void;
end of ULM_LOGGING
```