



750 Naples Street • San Francisco, CA 94112 • (415) 584-6360 • <http://www.pumpkininc.com>

# RM-ICC11 Reference Manual

## ***Salvo Compiler Reference Manual – ImageCraft ICC11***

---



## Introduction

This manual is intended for Salvo users who are targeting the Motorola M68HC11 single-chip microcontroller with ImageCraft's (<http://www.imagecraft.com/>) ICC11 C compiler.

## Related Documents

The following Salvo documents should be used in conjunction with this manual when building Salvo applications with ImageCraft's ICC11 C compiler:

*Salvo User Manual*  
*Application Note AN-19*

## Example Projects

Example Salvo projects for use with ImageCraft's ICC11 C compiler and the ImageCraft IDE can be found in the:

```
\salvo\ex\ex1\syst  
\salvo\tut\tu1\syst  
\salvo\tut\tu2\syst  
\salvo\tut\tu3\syst  
\salvo\tut\tu4\syst  
\salvo\tut\tu5\syst  
\salvo\tut\tu6\syst
```

directories of every Salvo for Motorola M68HCxx distribution.

## Features

Table 1 illustrates important features of Salvo's port to ImageCraft's ICC11 C compiler.

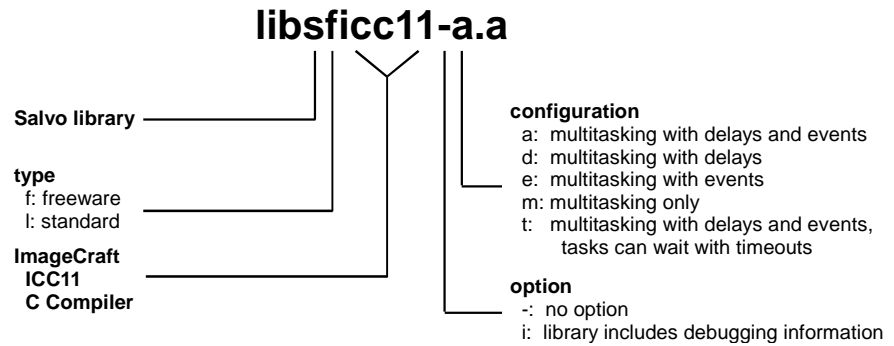
general	
available distributions	Salvo Lite, LE & Pro for Motorola M68HCxx
additional distributions	Salvo tiny & SE for Motorola M68HCxx & ICC11
supported targets	all M68HC11 variants
header file(s)	porticc11.h
other target-specific file(s)	porticc11.s
project subdirectory name(s)	SYST
salvocfg.h	
compiler auto-detected?	no
libraries	
\salvo\lib subdirectory	icc11
context switching	
method	function-based via OSDispatch()
_OSLabel() required?	no
size of auto variables and function parameters in tasks	total size must not exceed 254 8-bit bytes
interrupts	
controlled via	I bit in Condition Code Register
interrupt status preserved in critical sections?	yes
method used	saved on stack via OSEnterCritical()
nesting limit	unlimited
alternate methods possible?	yes <sup>1</sup>
debugging	
source-level debugging with Pro library builds?	yes
compiler	
bitfield packing support?	no
printf() / %p support?	yes / yes
va_arg() support?	yes

**Table 1: Features of Salvo Port to ImageCraft's ICC11 C Compiler**

## Libraries

### Nomenclature

The Salvo libraries for ImageCraft's ICC11 C compiler follow the naming convention shown in Figure 1.



**Figure 1: Salvo Library Nomenclature – ImageCraft's  
 ICC11 C Compiler**

## Type

Salvo Lite distributions contain *freeware* libraries. All other Salvo distributions contain *standard* libraries. See the *Libraries* chapter of the *Salvo User Manual* for more information on library types.

## Target

No target-specific identifiers are required.

## Option

Salvo Pro users can select between two sets of libraries – standard libraries, and standard libraries incorporating source-level debugging information. The latter have been built with ImageCraft's ICC11 C compiler's `+g` command-line option. This adds source-level debugging information to the libraries, making them ideal for source-level debugging and stepping in the ICC11 debugger. To use these libraries, simply select one that includes the debugging information (e.g. `libslicc11it.a`) instead of one without (e.g. `libslicc11-t.a`) in your ICC11 project.

## Configuration

Different library configurations are provided for different Salvo distributions and to enable the user to minimize the Salvo kernel's footprint. See the *Libraries* chapter of the *Salvo User Manual* for more information on library configurations.

## Build Settings

Salvo's libraries for ImageCraft's ICC11 C compiler are built using the default settings outlined in the *Libraries* chapter of the *Salvo User Manual*. Target-specific settings and overrides are listed in Table 2.

compiled limits	
max. number of tasks	3
max. number of events	5
max. number of event flags	1
max. number of message queues	1
target-specific settings	
delay sizes	8 bits
watchdog timer	cleared in <code>OSSched()</code> . Watchdog timer configuration is unchanged
system tick counter	available, 32 bits

**Table 2: Build Settings and Overrides for Salvo Libraries for ImageCraft's ICC11 C Compiler**

---

**Note** The compiled limits for tasks, events, etc. in Salvo libraries can be overridden to be less (all Salvo distributions) or more (all Salvo distributions except Salvo Lite) than the library default. See the *Libraries* chapter of the *Salvo User Manual* for more information.

---

## Available Libraries

There are 17 Salvo libraries for ImageCraft's ICC11 C compiler. Each Salvo for Motorola M68HCxx distribution contains the Salvo libraries of the lesser distributions beneath it.

## salvocfg.h Examples

Below are examples of `salvocfg.h` project configuration files for different Salvo for Motorola M68HCxx distributions targeting the M68HC11.

---

**Note** When overriding the default number of tasks, events, etc. in a Salvo library build, `OSTASKS` and `OSEVENTS` (respectively) *must also be defined* in the project's `salvocfg.h`. If left undefined, the default values (see Table 2) will be used.

---

## Salvo Lite Library Build

```
#define OSCOMPILER          OSIMAGECRAFT
#define OSTARGET            OSM68HC11
#define OSUSE_LIBRARY      TRUE
#define OSLIBRARY_TYPE     OSF
#define OSLIBRARY_CONFIG   OSA
```

**Listing 1: Example salvocfg.h for Library Build Using libsficc11-a.a**

## Salvo LE & Pro Library Build

```
#define OSCOMPILER          OSIMAGECRAFT
#define OSTARGET            OSM68HC11
#define OSUSE_LIBRARY      TRUE
#define OSLIBRARY_TYPE     OSL
#define OSLIBRARY_CONFIG   OSA
```

**Listing 2: Example salvocfg.h for Library Build Using libslicc11-a.a or libslicc11ia.a**

## Salvo Pro Source-Code Build

```
#define OSCOMPILER          OSIMAGECRAFT
#define OSTARGET            OSM68HC11
#define OSENABLE_IDLING_HOOK TRUE
#define OSENABLE_SEMAPHORES TRUE
#define OSEVENTS           1
#define OSTASKS            3
```

**Listing 3: Example salvocfg.h for Source-Code Build**

## Performance

### Memory Usage

tutorial memory usage <sup>2</sup>	total ROM <sup>3</sup>	total RAM <sup>4</sup>
tu1lite	868	24
tu2lite	1087	24
tu3lite	1141	26
tu4lite	1783	35
tu5lite	3230	47
tu6lite	3384 <sup>5</sup>	49 <sup>6</sup>
tu6pro	1921 <sup>7</sup>	45 <sup>8</sup>

**Table 3: ROM and RAM requirements for Salvo Applications built with ImageCraft's ICC11 C Compiler**

## Special Considerations

### Library Locations

ImageCraft's ICC11 C compiler expects libraries to be in `\icc\lib`. Therefore the Salvo installer places its libraries for ICC11 in both `\salvo\lib\icc11` and `\icc\lib`.

- 
- <sup>1</sup> Future versions of ICC11 may include a `monitor` pragma.
  - <sup>2</sup> Salvo v3.2.0-b with ICC11 v6.04. The unusual disparity in ROM sizes between `lite` (library) and `pro` (source-code) builds is due to the behavior of the ICC11 linker and the organization of the ICC11 library functions. Many included functions (e.g. `__ludiv`) are not called by the application.
  - <sup>3</sup> In bytes. Does not include interrupt vectors.
  - <sup>4</sup> In bytes. Does not include RAM allocated to the stack.
  - <sup>5</sup> Includes 2 bytes from the `idata` section.
  - <sup>6</sup> Includes 2 bytes from the `data` section.
  - <sup>7</sup> Includes 2 bytes from the `idata` section.
  - <sup>8</sup> Includes 2 bytes from the `data` section.