



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

# RM-ICC430 Reference Manual

## ***Salvo Compiler Reference Manual – ImageCraft ICC430***

---



# Salvo™

The RTOS that runs in tiny places.™

## Introduction

This manual is intended for Salvo users who are targeting TI's MSP430 ultra-low-power single-chip microcontroller with ImageCraft's (<http://www.imagecraft.com/>) ICC430 C compiler.

## Related Documents

The following Salvo documents should be used in conjunction with this manual when building Salvo applications with ImageCraft's ICC430 C compiler:

*Salvo User Manual*  
*Application Note AN-20*

## Example Projects

Example Salvo projects for use with ImageCraft's ICC430 C compiler and the ImageCraft IDE can be found in the:

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

directories of every Salvo for TI's MSP430 distribution.

## Features

Table 1 illustrates important features of Salvo's port to ImageCraft's ICC430 C compiler.

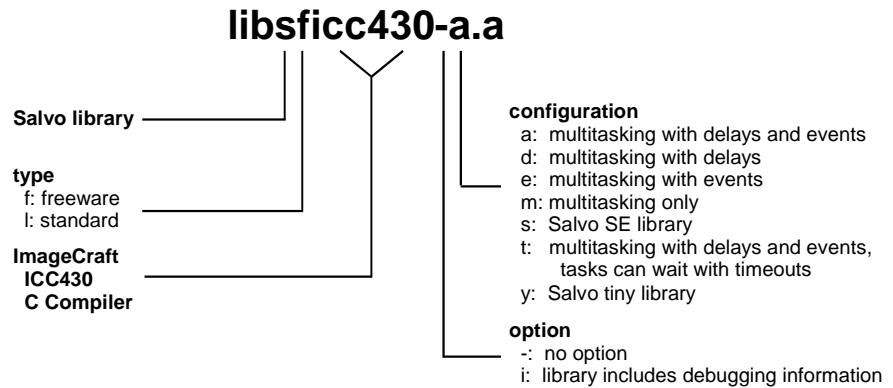
general	
available distributions	Salvo Lite, LE & Pro for TI's MSP430
additional distributions	Salvo tiny & SE for TI's MSP430 & ICC430
supported targets	entire MSP430 family
header file(s)	porticc430.h
other target-specific file(s)	porticc430.s
project subdirectory name(s)	SYSS
salvocfg.h	
compiler auto-detected?	yes <sup>1</sup>
libraries	
\\salvo\\lib subdirectory	icc430
context switching	
method	function-based via OSDispatch() & OSCtxSw()
_OSLabel() required?	no
size of auto variables and function parameters in tasks	total size must not exceed 255 8-bit bytes
interrupts	
controlled via	GIE bit
interrupt status preserved in critical sections?	yes
method used	saved on stack via #pragma monitor
nesting limit	unlimited
alternate methods possible?	yes <sup>2</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 ICC430 C Compiler**

## Libraries

### Nomenclature

The Salvo libraries for ImageCraft's ICC430 C compiler follow the naming convention shown in Figure 1.



**Figure 1: Salvo Library Nomenclature – ImageCraft's ICC430 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 ICC430 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 ICC430 debugger. To use these libraries, simply select one that includes the debugging information (e.g. `libslicc430it.a`) instead of one without (e.g. `libslicc430-t.a`) in your ICC430 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 ICC430 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 ICC430 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 ICC430 C compiler. Each Salvo for TI's MSP430 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 TI's MSP430 distributions targeting the MSP430F149.

---

**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 OSUSE_LIBRARY           TRUE
#define OSLIBRARY_TYPE         OSF
#define OSLIBRARY_CONFIG       OSA
```

**Listing 1: Example salvocfg.h for Library Build Using  
libslicc430-a.a**

## Salvo tiny Library Build

```
#define OSUSE_LIBRARY           TRUE
#define OSLIBRARY_TYPE         OSL
#define OSLIBRARY_CONFIG       OSY
```

**Listing 2: Example salvocfg.h for Library Build Using  
libslicc430-y.a**

## Salvo SE Library Build

```
#define OSUSE_LIBRARY           TRUE
#define OSLIBRARY_TYPE         OSL
#define OSLIBRARY_CONFIG       OSS
```

**Listing 3: Example salvocfg.h for Library Build Using  
libslicc430-s.a**

## Salvo LE & Pro Library Build

```
#define OSUSE_LIBRARY           TRUE
#define OSLIBRARY_TYPE         OSL
#define OSLIBRARY_CONFIG       OSA
```

**Listing 4: Example salvocfg.h for Library Build Using  
libslicc430-a.a or libslicc430ia.a**

## Salvo Pro Source-Code Build

```
#define OSENABLE_IDLE_HOOK     TRUE
#define OSENABLE_SEMAPHORES   TRUE
#define OSEVENTS                1
#define OSTASKS                  3
```

**Listing 5: Example salvocfg.h for Source-Code Build**

## Performance

### Memory Usage

tutorial memory usage <sup>3</sup>	total ROM <sup>4</sup>	total RAM <sup>5</sup>
tu1lite	380	22
tu2lite	560	22
tu3lite	602	24
tu4lite	1152	34
tu5lite	1666	50
tu6lite	1798 <sup>6</sup>	52 <sup>7</sup>
tu6pro	1638 <sup>8</sup>	48 <sup>9</sup>

**Table 3: ROM and RAM requirements for Salvo Applications built with ImageCraft's ICC430 C Compiler**

## Special Considerations

### Library Locations

ImageCraft's ICC430 C compiler expects libraries to be in `\icc\lib`. Therefore the Salvo installer places its libraries for ICC430 in both `\salvo\lib\icc430` and `\icc\lib`.

<sup>1</sup> This is done automatically through the `__IMAGECRAFT__` and `_MSP430` symbols defined by the compiler.

<sup>2</sup> Since control of the GIE bit is intimately associated with the RETI instruction and the compiler's `#pragma monitor`, alternate methods are generally not recommended.

<sup>3</sup> Salvo v3.2.0-b with ICC430 v6.03.

<sup>4</sup> In bytes. Does not include interrupt vectors.

<sup>5</sup> In bytes. Does not include RAM allocated to the stack.

<sup>6</sup> Includes 2 bytes from the `idata` section.

<sup>7</sup> Includes 2 bytes from the `data` section.

<sup>8</sup> Includes 2 bytes from the `idata` section.

<sup>9</sup> Includes 2 bytes from the `data` section.