PUMPKIN™
REAL-TIME SOFTWARE
750 Naples Street • San Francisco, CA 94112 • (415) 584-6360 • http://www.pumpkininc.com

Release Notes

Salvo v2.2.0

Introduction

The 2.2.0 release of Salvo marks a substantial performance improvement in several areas, including runtime performance, the system timer, interaction with interrupts, call…return stack depth management, inclusion of standard libraries, improved freeware libraries and support for new targets and new compilers.

We felt that now was a good time to implement small but powerful changes that affect end-user applications.

Note This release marks a substantial change to the source code and will require minor modifications to *all* Salvo applications & projects. Please see *Migrating from Earlier Versions*, below, for a complete list of the changes necessary for a successful 2.2.0 build.

New Names

The freeware version of Salvo is now called Salvo Lite. The standard, full version is called Salvo.

New Support

Targets

- Microchip baseline PICmicros (e.g. 12C509) now supported.
- Microchip PIC18 family PICmicros (e.g. PIC18C452) now supported.

Compilers

- HI-TECH PICC-18 v8.01 PL1 and higher is now fully certified for use with Salvo.
- Microchip MPLAB-C18 support is in alpha testing stage.

Libraries

- Salvo now ships with a full suite of freeware and standard libraries for PIC12, PIC16, PIC17 and PIC18 PICmicro devices.
- Libraries for 80x86 C console applications built with Metrowerks CodeWarrior are included.

Demos

- `salvo\demo\d4` runs on PIC16C77+PICDEM-2, PIC18C452+PICDEM2, and PIC16F877+MPLAB-ICD. A simple three-task multitasking application with behavior that illuminates what task priorities can do …

Templates

- Salvo now includes templates with simple, fill-in-the-blanks that you can cut-and-paste into your own applications. They are in `salvo\template`.

New Features

- The timer module has been completely revamped, including removal of the SuperTimer (no longer necessary), major simplifications to `OSTimer()` leading to smaller ISRs, and overall new timer methodology for better performance.
- A new event type, event flags (`eFlag`), has been added.
- Event creation and signaling services (e.g. `OSCreateMsg()` and `OSSignalBinSem()`, respectively) now consumes only a single stack level, enabling event creation and signaling from interrupts with minimal call…return stack usage.

- Salvo objects (e.g. tasks and events) are now referenced by pointer, not index. This has many positive ramifications.
- `OSTimer()` can now be inlined into your ISR for all sorts of performance benefits. See `OSUSE_INLINE_OSTIMER` for more information.
- `OSSched()` can now be inlined and thereby reduce the call…return stack level that Salvo tasks run at. See `OSUSE_INLINE_OSSCHED` for more information. This feature required that `OSSched()` be moved into its own source code module, `salvo\source\sched.c`.
- Now using the C preprocessor's `#error` directive to alert user to problems with configuration parameters.
- The interface to the precompiled (i.e. standard and freeware) libraries is much simpler. Where applicable, all three event-signaling variants (signaling supported from background, from foreground, or from anywhere) of each library now exist.
- New services: `OSClrEFlag()`, `OSCreateEFlag()`, `OSDestroyTask()`, `OSMsgQEmpty()`, `OSSetPrioTask()`,`OSGetPrio()`, `OSGetPrioTask()`,`OSGetState()`, `OSGetStateTask()`,`OSReadBinSem()`, `OSReadEFlag()`, `OSReadMsg()`, `OSReadMsgQ()`,`OSReadSem()`, `OSSetEFlag()`, `OSStopTask()`, `OSTryBinSem()`, `OSTryMsg()`, `OSTryMsgQ()`, `OSTrySem()`, `OS_WaitEFlag()`.
- New configuration options: `OSCALL_OSCREATEEVENT`, `OSCALL_OSGETPRIOTASK`, `OSCALL_OSGETSTATETASK`, `OSCALL_OSMSGQEMPTY`, `OSCALL_OSRETURNEVENT`, `OSCALL_OSSIGNALEVENT`, `OSCOMBINE_EVENT_SERVICES`, `OSDISABLE_TASK_PRIORITIES`, `OSENABLE_EVENT_READING`, `OSENABLE_EVENT_TRYING`, `OSENABLE_SCHEDULER_HOOK`, `OSLIBRARY_CONFIG`, `OSLIBRARY_TYPE`, `OSLIBRARY_VARIANT`, `OSLOC_ALL`, `OSUSE_LIBRARY`.
- The tutorial now includes projects that use the precompiled libraries in addition to ones that use the full version's source code modules.
- The precompiled libraries now contain all of the Salvo source modules – there is no longer a need to compile `chk.c` and `mem.c` separately.

- Added `OSLOC_ALL` to set all `OSLOC_XYZ` configuration parameters at once. Also added `OSLOC_CTCB`, `OSLOC_GLSTAT`, `OSLOG_SIGQ`.
- A (text) version string is now available in `OSversion`.
- `OSRpt()` was substantially simplified and cleaned up.
- In the full version of Salvo, code size for `OSCreateXyz()`, `OSSignalXyz()` and `OS_WaitXyz()` has been reduced when multiple event types are enabled. In the precompiled libraries, these services are completely independent and therefore larger. `OSCOMBINE_EVENT_SERVICES` is used to control this feature.
- Task priorities can be disabled for code size (ROM) savings via `OSDISABLE_TASK_PRIORITIES`.
- The configuration options used to generate each library have been custom-tailored to maximize the library's usefulness on its target.

Bug Fixes

- Fixed problems involving `OSFROM_ANYWHERE` for `OSCALL_OSCREATEEVENT` and `OSCALL_OSSIGNAL-EVENT` configuration options – supported services can now be called from mainline code and from interrupts. A new source code module, `salvo\include\multcall.h`, is used to manage multiple call graph issues that these configuration options create with certain compilers.
- The missing call…return stack level of Salvo tasks has now been fixed. Correct results are displayed when `OSENABLE_STACK_CHECKING` is `TRUE`.
- The 2.1 installers did not overwrite existing, newer files. This would cause problems if the full version was installed over the patched demo version. 2.2.0 and later installers replace all files, regardless of date.
- Fixed `OSSignalXyz()` bug where eventID was not being range-tested. eventIDs are no longer used as arguments in 2.2.0, however …

Source Code Changes

- Include guards are now "`_IG`" instead of "`_INCLUDES`".
- IDs are now numbered 1 through maximum, instead of 0 through maximum-1.

- `OSRtnNumTasks()`, `OSRtnNumEvents()` and `OSRtnIdleTaskNum()` have been deleted.
- New macros to obtain Salvo object pointers: `OSECBP()`, `OSEFCBP()`, `OSMQCBP()`, `OSTCBP()`.
- All instances of `NULL` have been replaced with `0`.
- Unnecessary typecasts have been removed.
- Context-switcher portability has been improved.
- All Salvo pathnames are now fully lower-case.
- `OSdlyQP` is now called `OSdelayQP`.
- Changed parenthesis style to be like K&R.
- `qdel.c` and `qins.c` reorganized for legibility and maintenance.
- New source code modules: `multcall.h`, `eflag.c`, `eid.c`, `initecb.c`, `inittask.c`, `inittcb.c`, `sched.c`, `tid.c`, `ver.c`.
- The SuperTimer and circular queues are obsolete / no longer supported and have been removed from the source code.
- Event creation and signaling services (e.g. `OSCreateBinSem()` and `OSSignalMsgQ()`) are now implemented as macros or functions, depending on the value of `OSCOMBINE_EVENT_SERVICES`. Use `OSCALL_OSCREATE_EVENT` and `OSCALL_OSSIGNALEVENT` configuration macros for interrupt control. Added `OSCOMBINE_EVENT_SERVICES` to select between ones that use common code and variable argument lists (the default) and ones that are independent, with duplicated code. The former is better for source code users, the latter is better for freeware library generation.
- The default value for all `OSENABLE_XYZ` configuration options is now `FALSE`.
- All `OSGetXyz() event` services have been renamed `OSReadXyz()`.
- Qualified types are now `OSgltypeXyz`.
- Timeouts are no longer logged as warnings. They are still counted when `OSGATHER_STATISTICS` and `OSENABLE_TIMEOUTS` are `TRUE`.

User Manual Changes

- *Release Notes* and *Libraries* chapter added.
- *Tutorial* chapter reorganized and parts rewritten.
- *Application Notes* and *Assembly Guides* now listed in *Appendices*.
- Many other small changes.

Other Document Changes

- Updated Application Notes, include major revisions to *AN-1* and *AN-4*.

Installer Changes

- Detects previous installation (if any) and offers to back it up.
- Readme and Release Notes now in Adobe PDF (`.pdf`) format.
- Prompts user to register software when installation is complete.
- Program groups for documentation and web links.
- App Notes included.

Known Problems

- Array mode, originally scheduled for the 2.2.0 release, will appear in a later release.
- *Performance* chapter is not up-to-date.
- If the user chooses to backup the existing Salvo installation to a new folder during installation, and then does a Custom install, some Windows program group items will remain. Solution: delete them manually.

Migrating from Earlier Versions

Applications compiled under Salvo 2.1 *will not* compile successfully or function properly under 2.2.0. When migrating from Salvo 2.2.0 or earlier, please observe the following:

- Upgrade only by running the 2.2.0 Installer. The installer will automatically put all the new and changed files into your Salvo directories.
- Change all references to Salvo objects (usually as parameters to user services) to pointers. The easiest way to do this is to use `OSECBP()` and similar macros.
- "Bump up" all IDs by one. E.g. where `#define TASKA 0` was correct under 2.1, now use `#define TASKA_P OSTCBP(1)`.

- `sched.c` must now be added to your projects, because it contains `OSSched()` (formerly in `task.c`). Most projects will also need to add `initecb.c`, `inittask.c` and `inittcb.c`.
- You will need to set `OSENABLE_XYZ` to `TRUE` for each type of event services you are using. For example, if you are using semaphores in your application, add `OSENABLE_SEMAPHORES` to `salvocfg.h` and set it to `TRUE`.
- For targets with banked RAM, we recommend you add `OSLOC_ALL` to your `salvocfg.h` and set it to the value of the majority of your `OSLOC_XYZ` configuration options. This will guarantee that all of Salvo's variables are placed in the bank(s) you specify.
- Remove all references to the SuperTimer from your code. SuperTimer configuration options contained in `salvocfg.h` will be flagged as an error when compiling 2.2.0 projects.
- If you were using any macros to access directly tcb and/or ecb fields (e.g. `OSTCB_PTR()`, see `salvo.h`), these macros have been renamed in 2.2.0 to follow Salvo's naming convention (e.g. `OSTCBP()`). You will also need to "bump" the index used up by 1.
- Projects that used the 2.1 freeware libraries should delete `chk.c` and `mem.c` from their lists of nodes to be built.
- If necessary, change all instances of `OSCALL_OSCREATEBINSEM`, `OSCALL_OSCREATEMSG`, `OSCALL_OSCREATEMSGQ` and `OSCALL_OSCREATESEM` to `OSCALL_OSCREATEEVENT`. Only one `OSCALL_OSCREATEEVENT` is required per `salvocfg.h`. References to these older configuration parameters will be flagged as errors.
- Similarly, if necessary, change all instances of `OSCALL_OSSIGNALBINSEM`, `OSCALL_OSSIGNALMSG`, `OSCALL_OSSIGNALMSGQ` and `OSCALL_OSSIGNALSEM` to `OSCALL_OSSIGNALEVENT`. Only one `OSCALL_OSSIGNALEVENT` is required per `salvocfg.h`. References to these older configuration parameters will be flagged as errors.
- Remove all references to the `OSUSE_CIRCULAR_QUEUES` configuration parameter from your salvocfg.h. It will be flagged as an error when compiling 2.2.0 projects.

- If necessary, change all references to `OSUSE_INSELIGQ_MACRO` in `salvocfg.h` to `OSUSE_INSELIG_MACRO`.
- If necessary, change all references to `OStypeMsgQP` to `OSgltypeMsgQP`.
- If necessary, change all references to `OS_Prio()` to `OS_SetPrio()`.
- If necessary, change all references to `OSPrio()` to `OSSetPrio()`.

See the relevant sections of the Salvo User Manual for more information on the issues mentioned above.

Library Generation

Libraries were generated with the following compilers:

- PIC12, PIC16 and PIC17: HI-TECH PICC v7.87 PL2
- PIC18: HI-TECH PICC-18 v8.01 PL1
- 80x86: Metrowerks CodeWarrior 3.1

How to Contact Pumpkin for Support

Pumpkin provides online Salvo support via the Pumpkin World Wide Web (WWW) site. Files and information are available to all Salvo users via the web site. To access the site, you'll need web access and a browser (e.g. Netscape, Opera, Internet Explorer).

Connecting to Pumpkin's Web Site

Use your web browser to access the Pumpkin web site at

http://www.pumpkininc.com/

Information available on the web site includes

- Latest News
- User Manuals
- Software Downloads & Upgrades
- Application Notes
- Assembly Guides
- Release Notes
- User Forums

What To Provide when Requesting Support

Registered users requesting Salvo technical support should supply:

- The Salvo version number
- The compiler name and version number
- The user's source code snippet(s) in question
- The user's salvocfg.h file
- All other relevant files, details, etc.

Small code sections can be posted directly to the Salvo User Forums – see the on-line posting FAQ on how to use the UBB code tags (`[code]` and `[/code]`) to preserve the code's formatting and make it more legible.

If the need arises to send larger code sections, or even a complete, buildable project, please compress the files and email them directly to Salvo Technical support (see below). Please be sure to provide all necessary files to enable Technical Support to build your Salvo application locally in an attempt to solve your problem. Keep in mind that without the appropriate target system hardware, support in these cases is generally limited to non-runtime problem solving. Technical Support will keep all user code in strictest confidence.

Other Means of Contacting Pumpkin

We **strongly prefer** that you post your technical support question(s) to the Salvo User Forums (available on our website), and we will answer them there. Pumpkin technical support can also be reached directly at support@pumpkininc.com – this is the right address for sending attachments, private information (e.g. serial numbers), etc.

Pumpkin's mailing address and phone and fax numbers are:

Pumpkin, Inc.
750 Naples Street
San Francisco, CA 94112 USA
tel: 415-584-6360
fax: 415-585-7948

Time Zone: GMT–0800 (Pacific Standard Time)