CORE FLIGHT SOFTWARE

VERSION DESCRIPTION DOCUMENT

OPERATING SYSTEM ABSTRACTION LAYER (OSAL)

BUILD: 4.2.0

MARCH 4, 2016

SIGNATURES

Submitted by:

X Susanne Strege

Susanne Strege/GSFC-5820
cFS Flight Software Development Lead

SUSANNE STREGE

Digitally signed by SUSANNE STREGE
DN: c=US, o=U.S. Government, ou=NASA, ou=PIV, cn=SUSANNE STREGE, 0.9.2342.19200300.100.1.1=sstrege
Date: 2016.03.04 18:48:11 -05'00'

Approved by:

X Changed 2 days ago by jphickey
I approve the OSAL 4.2.0 VDD as discussed at the 2016-03-01 CCB

Joseph Hickey/GRC-LSS0
cFS Configuration Control Board

X Changed 3 days ago by glimes
I approve of the version as reviewed at 2016-03-01 CCB

Gregory Limes/ARC-TI
cFS Configuration Control Board

X Changed 2 days ago by sduran
I approve the OSAL 4.2.0 VDD discussed at the 03/01/16 CCB.

Steve Duran/JSC-ER611
cFS Configuration Control Board

X Changed 68 minutes ago by dthames
I approve the OSAL 4.2.0 VDD

Chris Thames/LARC-D207
cFS Configuration Control Board

SIGNATURES – CONTINUED

Approved by:

X Changed 3 days ago by stashakk

I accept the updated version as reviewed at 2016-03-01 CCB

Scott Tashakkor/MSFC-ES52
cFS Configuration Control Board

X Changed 0 seconds ago by sstrege

Approving on behalf of Chris Monaco

Approval received via email

Chris Monaco/JHU-APL
cFS Configuration Control Board

1.0 FSW VERSION DESCRIPTION

1.1 PURPOSE AND SUMMARY

The purpose of this build is to refine the core Flight Software (cFS) Operating System Abstraction Layer (OSAL) product. This build provides various bug fixes, as well as, new features and enhancements including:

- Class A safety-critical instantiation of the VxWorks operating system abstraction
- Class A safety-critical black box test updates and added white box test suite
- Network/socket interface abstraction
- Enhanced cmake build system (in addition to classic build)
- PC-RTEMS BSP
- SPARC-VxWorks6.7 BSP
- Unit Test Assert library

There were some minor API changes to this build that may result in compiler warnings with applications/tasks built via previous OSAL releases. These API changes were made to correct and improve the function input parameter types. The changes include:

1. OS_TaskCreate: stack pointer is **NOT** `const`
2. OS_TaskInstallDeleteHandler: use correct function-pointer argument instead of `void *`
3. OS_QueuePut: source data pointer is `const`
4. OS_ExcAttachHandler: context data pointer is `const`
5. All name arguments are now `const char*` instead of `char*` for all API calls
6. OS_SymbolLookup, OS_ModuleInfo: All integers holding memory addresses use the `cpuaddr` type instead of `uint32`

There were several API additions to:

1. Formalize common shutdown routines performed during unit testing
2. Define the network/socket API (implementation of this API will be delivered in a separate release)
3. Support alternate time references for OSAL timers

The transition from OSAL 4.1.1 to 4.2.0 should require minor effort. However, build 4.2.0 does not support backward compatibility with cFE builds 6.4.2 and later and PSP builds 1.2.0 and later. OSAL build 4.2.0 implements the list of Trac tickets listed in Attachment 1.

Functional testing has been completed and baselined on OSAL build 4.2.0 for the pc-linux and VxWorks implementations. Functional test results are included in the release package under:

/src/tests/Results

Black box unit testing was completed and baselined for the pc-linux and vxworks implementations using the test suite updates delivered and integrated by Johnson Space Center (JSC). This test suite was used by JSC's Advanced Exploration Systems (AES) Human Exploration & Operations Mission Directorate for Class A certification of the OSAL VxWorks instantiation currently planned for use on the Orion Program backup flight control system. Black box unit test results are included in the release package under:

/src/unit-tests/Results

White box unit tests were developed for the VxWorks implementation by Johnson Space Center (JSC). This test suite will be used by JSC's Advanced Exploration Systems (AES) Human Exploration & Operations Mission Directorate for Class A certification of CFS on the Orion platform. There are open issues with building this test suite using the "classic" build and possible errors in the tests. This test suite, as delivered in this release, should be used with caution.

In addition to the white box test suite issues, there are some outstanding issues being investigated. Resolutions to these issues may require a new release. The project CCB and community inputs will determine which Trac Tickets to include in the next release.

This distribution contains:

1. The OS Abstraction Layer API Library
2. OS implementations for POSIX, VxWorks6.x, and RTEMS
3. Tests and example applications
4. A directory structure and cmake system (or "classic" build makefiles) to manage it all.

1.2 NEW/CHANGED FUNCTIONALITY IN THIS VERSION

Table 1.2-1 identifies new OSAL functionality that has been implemented and is integrated into this version and the Trac tickets associated with these changes.

Table 1.2-1 – New Functionality in this Version

No.	Trac Ticket #	High Level Description of Functionality	Component	Type	Priority
1	#2	Add enhanced cmake build system to OSAL	build	enhancement	major

2	#8	Add OSAL abstraction for network/socket interface. Adds API definition for the following network operations: - Abstraction of OS socket address (sockaddr_in/_in6 for IPv4 and IPv6 at least) - Abstraction of Stream/TCP & Datagram/UDP socket creation - Bind/Listen for TCP - Sendto/Recvfrom for UDP - Abstractions for the basic inet_aton()/inet_ntop() IP-address Note: implementation of this API is forthcoming separately	other	enhancement	major
3	#10	API versioning for OSAL - adds an OS_API_VERSION macro that application code can check	build	enhancement	major
4	#15	Alternate time references for OSAL timers. Adds a new type of OSAL object called a "Time Base". By default a time base can be driven from the local CPU real time clock, which will mimic the current behavior of timers. However the BSP/PSP may create additional time bases and synchronize them to e.g. an external timing interrupt. The same existing "TimerSet" API can be used to set the timers against the alternate time base and it will be transparent to the app.	other	enhancement	minor
5	#28	Do not block synchronous hardware-generated signals in pc-linux BSP. As of OSAL 4.1, all signals are blocked during execution of the OS_Application_Startup() call, then unblocked before entering the wait loop. The issue here is that some signals on Linux are generated by the underlying hardware and the kernel forwards these (synchronously) to the specific thread that was executing on the CPU when the actual signal was triggered. The set of hardware-generated signals: SIGSEGV, SIGILL, SIGBUS, SIGFPE should NOT be blocked at any time by any thread. The hardware-generated signals are now unblocked all the time and the RT signals are blocked all the time in the pc-linux implementation so the user-created threads can use them.	pc-linux	defect	major
6	#29	Add pc-rtems BSP to OSAL	other	enhancement	minor
7	#31	OSAL public/private data structure delineation. Updates to define which data structures of OSAL are intended to act as the "public" interface and which data structures are internal/private to OSAL.	os common	enhancement	minor
8	#32	Adjust usage of feature support macros for glibc/posix. Uses "XOPEN_SOURCE=600" across the entire build. This will enable XPG6 features. XPG6 adds some more realtime features that are likely to be useful to applications such as this, e.g. clock_nanosleep() and some others.	os posix	defect	minor
9	#33	API additions to formalize common routines in OSAL. Adds APIs to formalize normal shutdown procedures and provides the hooks necessary to perform shutdown cleanup.	os common	enhancement	major
10	#37	Implement user-selectable compiler warning switches	build	enhancement	minor
11	#39	Connect compiler warnings to test results parser	bamboo	enhancement	major
12	#57	Import UT-Assert basics into OSAL	unit-test	enhancement	major
13	#63	Add timebase API calls to existing OSALs for API compatibility	os common	defect	minor
14	#110	Add JSC white box unit coverage tests	unit-test	enhancement	major
15	#112	Integrate JSC audit of parameter and return code checking	os vxworks6	defect	major
16	#125	Add SPARC-VxWorks6.7 BSP	unspecified	enhancement	major

Table 1.2-2 identifies changes to OSAL functionality and bug fixes from previously delivered versions and the Trac tickets associated with these changes.

Table 1.2-2 – Changes to Previously Delivered Functionality and Bug Fixes

No.	Trac Ticket #	High Level Description of Functionality/Bug Report	Component	Type	Priority
1	#1	OSAL "common_types.h" is not completely reliable. On some systems (particularly 64-bit) the types defined in OSAL "common_types.h" file do not always match their expected widths.	os common	defect	major
2	#6	Make OSAL tests more autonomous. add a "script-mode" feature to the tests (in the "tests" directory, NOT "unit-test" or "unit-test-coverage") which: • Checks for the expected conditions and maintains an error counter of any tests that did not satisfy expected conditions • Limits the time of execution and exits the test automatically (no CTRL+C) • Returns a non-zero exit code if any errors occurred (for scripting)	build	defect	major
3	#7	Fix warnings in OSAL "unit-test" code	build	defect	major
4	#11	Duplicate OSAL error codes and error string API. osapi-os-filesys.h has its own set of error codes that overlap with and are different from the rest of OSAL error codes. For example, "OS_FS_ERR_NAME_TOO_LONG" is not the same as "OS_ERR_NAME_TOO_LONG". There are several codes that are redefined differently. At a minimum, this is confusing, but it can also cause real bugs if the wrong action is taken due to misinterpreting an error. Cleaned up error codes. FS error codes merged with the rest of OSAL error codes into a single set, with a single implementation of OS_GetErrorName() to get them all.	other	defect	major
5	#13	API prototype changes to address compiler warnings	build	defect	major
6	#16	OS_PEND and OS_CHECK are backwards	os common	enhancement	minor
7	#17	OSAL should use UT framework similar to that of CFE	os common	enhancement	major
8	#19	Fix OSAL build when using RTEMS "pc686" BSP	os rtems	defect	major
9	#20	TSF: osloader OS_ModuleLoad #4. The OS_ModuleLoad test case in the osloader unit test fails during the "test setup" loop. Once the test was augmented to report the failing module name, it was discovered that this was due to not staging the "MODULE%d.so" files to the target running the unit tests.	other	defect	minor
10	#24	pthread_create 3rd arg must be of right type. OS_TaskCreate is handed an OSAL entry point, which is a pointer to a function taking no parameters and returning no return value. It passes it along to pthread_create which is expecting a pointer to a function that takes a single void pointer parameter, and returns a void pointer return value. The code as written triggers a compiler warning.	os posix	defect	major
11	#27	Clean up "-D" compile time macros used in pc-linux build	pc-linux	defect	major
12	#30	Fix OSAL timer test ID usage	unit-test	defect	major
13	#51	Fix simulataneous use of OSAL BSP + CFE PSP compiler flags	build	defect	minor
14	#52	Fix warnings in vxworks support code	os vxworks6	defect	minor
15	#54	OS_QueueCreate failures	os posix	defect	minor

16	#58	Combine "pc-linux" and "pc-linux-ut" OSAL BSPs	unit-test	defect	major
17	#61	osnetwork.c API discrepancies. The VxWorks osnetwork.c code differs from the OSAL Library API documentation under some conditions. Even more interesting, the posix and rtems OSAL implementations behave differently from the VxWorks as well. Added coverage tests for osnetwork.c. Switching default osnetwork.c coverage build to 32-bit. Return OS_ERR_NOT_IMPLEMENTED for the "no network" case for VxWorks OSAL only.	os vxworks6	defect	minor
18	#62	Incorrect Logic in OS_TimespecToUsec	os vxworks6	defect	major
19	#67	OS_open: 644 is not 0644	unspecified	defect	minor
20	#68	OS_open and file-api-test in conflict	os posix	defect	minor
21	#70	Enhancements to OSAL UT stub code in osapi_stubs	unit-test	enhancement	major
22	#73	Fix "utbsp.h" not found failure when building on some platforms	build	defect	major
23	#74	classic build broken	build	defect	major
24	#82	vxworks osapi.c OS_GetErrorName() missing codes	os vxworks6	defect	minor
25	#88	vxworks OS_fsBlocksFree() misreports	os vxworks6	defect	major
26	#90	vxworks OS_FS_GetErrorName() errors	os vxworks6	defect	major
27	#102	GCOV results for "osapi" obscured.	build	defect	major
28	#111	JSC updated API tests	unit-test	enhancement	major
29	#113	JSC: Replace "UNINITIALIZED" macro with "OS_UNINITIALIZED"	os vxworks6	defect	major
30	#115	JSC: Audit vxworks6 global table protections	os vxworks6	defect	major
31	#116	JSC: use fixed width types	os vxworks6	defect	major
32	#117	JSC: change fpu get/set mask function return code	os vxworks6	defect	major
33	#118	JSC: general code cleanup: • Make all if/then/else as compound statements • Add "void" to functions that do not take parameters • Add final "else" to all "else if" constructs • Make sure all cases in switch stametents have break • Add explicit casting where the compiler may emit warnings • Add "static" and "extern" keywords where needed	os vxworks6	defect	major
34	#119	Fix up pointer subtraction (do not cast to integers)	unspecified	defect	major
35	#124	resolve "-m32" OSAL classic build issues	build	defect	major
36	#128	usleep is obsolete	unit-test	defect	minor
37	#130	strncpy may not '\0'-terminate	os posix	defect	minor
38	#131	unreachable flow control. cppcheck messages: src/tests/osal-core-test/osal-core-test.h:347: style: Consecutive return, break, continue, goto or throw statements are unnecessary. Removed duplicate return statement	unit-test	defect	minor

39	#132	Removed tests for unsigned variable less than zero cppcheck messages: src/os/posix/osapi.c:2273: style: Checking if unsigned variable 'sem_initial_value' is less than zero. src/os/posix/osapi.c:2794: style: Checking if unsigned variable 'sem_initial_value' is less than zero.	os posix	defect	major
40	#135	repeated assignments without using the value (real code edition). cppcheck messages: src/os/vxworks6/osnetwork.c:73: performance: Variable 'retval' is reassigned a value before the old one has been used. src/os/vxworks6/ostimer.c:331: style: Variable 'status' is assigned a value that is never used.	os vxworks6	defect	major
41	#136	Same expression on both sides of a binary operator. cppcheck message: src/unit-tests/osfile-test/ut_osfile_fileio_test.c:2537: style: Same expression on both sides of '||'.	unit-test	defect	minor
42	#137	Uninitialized variables. cppcheck messages: src/unit-tests/oscore-test/ut_oscore_queue_test.c:552: error: Uninitialized variable: queue_data_out src/unit-tests/oscore-test/ut_oscore_queue_test.c:562: error: Uninitialized variable: queue_data_out	unit-test	defect	minor
43	#144	ostimer unit test needs non-zero stack size parameter passed to OS_TaskCreate	unspecified	defect	major
44	#147	posix impl should use timer_t instead of uint32 for host_timerid	unspecified	defect	major
45	#150	timer-test.c has a difficult time obtaining a count for timer4 due to the start delay.	unspecified	defect	major
46	#151	vxWorks OSAL implementation needs OS_IdleLoop and OS_Application_Shutdown	unspecified	defect	major
47	#152	ostimer unit test uses OS_IdleLoop but not OS_Application_Shutdown	unspecified	defect	major
48	#156	Allow C99 code in OSAL.	build	defect	major

1.3 MISSING PLANNED FEATURES AND KNOWN PROBLEMS

Table 1.3-1 identifies the functions and known discrepancies that are absent from OSAL Version 4.2.0

Information on currently open Trac tickets is available at https://babelfish.arc.nasa.gov/trac/cfs_osal. Note that this is a restricted website that requires a server account. Additional Trac tickets may have been submitted after preparation of this VDD. An OSAL Trac ticket report containing a listing of open tickets is available on request for customers who do not have access to the babelfish server. Please contact Susanne Strege, susie.strege@nasa.gov.

Table 1.3-1 – Functions Absent from this Release

No.	Trac Ticket #	Description	Component	Status	Planned Delivery	Type	Priority
1	#3	Document available BSPs in trac wiki	wiki	accepted	Not Determined	defect	minor
2	#4	Document available OSs in trac wiki	wiki	accepted	Not Determined	defect	minor

3	#5	Refactor common code between VxWorks/Posix/Rtems into OSAL shared layer	os common	work complete	Not Determined	enhancement	major
4	#9	Add free-run tick counter API to OSAL	os common	review	Not Determined	enhancement	major
5	#12	Make file system API work more like the rest of OSAL	other	work complete	Not Determined	enhancement	major
6	#14	More lenient operation when "SIMULATION" compile-time directive is defined	os posix	new	Not Determined	enhancement	major
7	#21	OSAL PPC VxWorks "test runner"	unit-test	accepted	Not Determined	enhancement	major
8	#35	Bogus usage of strncpy in unit tests	unit-test	on hold	Not Determined	defect	major
9	#38	Add Xenomai OSAL	xenomai	new	Not Determined	enhancement	minor
10	#40	Enforce Strict ASCII	other	new	Not Determined	defect	minor
11	#41	Backtrace-tracking feature for debugging OSAL mutexes	os posix	new	Not Determined	enhancement	minor
12	#42	OSAL: Consider Allowing Root Task (caller of OS_API_Init) to Register and Use OSAL Services (GSFC DCR 21564)	other	new	Not Determined	enhancement	trivial
13	#43	OSAL: OS_EOF Macro is Not Defined (GSFC DCR 22719)	os common	new	Not Determined	enhancement	trivial
14	#44	Posix - optionally disable use of some realtime features for debugging	os posix	work complete	osal-next	enhancement	major
15	#45	POSIX - Consider using "SCHED_RR" instead of "SCHED_FIFO" for realtime threads	os posix	work complete	osal-next	enhancement	major
16	#46	Consider Adding a Timed Wait Function to the Mutex API (GSFC 22628)	os common	new	Not Determined	enhancement	major
17	#47	OSAL Library API Document Cut and Paste Errors	docs	new	Not Determined	defect	trivial
18	#48	Update RTEMS OS_IntAttachHandler for the PPC (GSFC #22161)	os rtems	new	Not Determined	enhancement	minor
19	#49	Add VxWorks RTP/Memory Protected Port (GSFC DCR 18626)	os vxworks6	new	Not Determined	enhancement	minor
20	#50	Add user-space message queue library to the OSAL (GSFC DCR 22160)	os common	new	Not Determined	enhancement	minor
21	#53	OS_check_name_length portability	os posix	in work	Not Determined	defect	minor
22	#56	OS_TaskDelete fails if the task (pthread) has already terminated on its own	os posix	assigned	Not Determined	defect	major
23	#64	divide osconfig.h three ways	other	new	Not Determined	defect	minor
24	#65	OS_TimerCreate() Unterminated String	os vxworks6	review	osal-next	defect	major
25	#71	posix ostimer.c functions not using semaphore	os posix	new	Not Determined	defect	major
26	#72	rtems ostimer.c functions not using semaphore	os rtems	new	Not Determined	defect	major
27	#79	Make compiles with --std=c99 work	build	in work	Not Determined	task	minor
28	#81	vxworks osapi.c OS_Milli2Ticks() problems	os vxworks6	review	osal-next	defect	minor
29	#85	vxworks osfilesys.c functions not thread-safe	os vxworks6	assigned	Not Determined	defect	major
30	#93	OS_rename() doesn't first check if a file is in use	os vxworks6	new	Not Determined	defect	minor
31	#95	osfilesys.c mixed return types	os vxworks6	new	Not Determined	defect	minor

32	#97	vxworks osapi.c utility task doesn't exit	os vxworks6	new	Not Determined	enhancement	minor
33	#98	Simplify Function Pointer Manipulations	other	new	Not Determined	task	minor
34	#99	Posix message queues leak.	os posix	new	Not Determined	defect	minor
35	#100	Standardize Version Numbering (in OSAL)	other	new	Not Determined	defect	major
36	#105	Overwriting unused values in variables	cppcheck	new	Not Determined	defect	minor
37	#114	JSC: add static initializers to all local variables	os vxworks6	work complete	osal-next	defect	major
38	#120	Support Insertion/Integration of Third Party/Bridge Libraries	unspecified	new	Not Determined	defect	major
39	#121	OSAL API Documentation Should Be Doxygen Based	unspecified	new	Not Determined	defect	major
40	#122	Expand cppcheck application	cppcheck	in work	Not Determined	enhancement	major
41	#123	cppcheck a vxworks build	cppcheck	in work	Not Determined	task	major
42	#126	Reconcile diffs between unit test makefiles and JSC UT makefiles	unspecified	new	osal-next	enhancement	major
43	#127	May need -rdynamic	unspecified	review	Not Determined	defect	major
44	#129	struct/union member never used	unit-test	in work	Not Determined	defect	minor
45	#133	readdir is not reentrant	os posix	on hold	Not Determined	defect	major
46	#134	repeated assignments without using the value (unit test edition)	unit-test	in work	Not Determined	defect	minor
47	#138	function declaration is not a prototype	unit-test	work complete	osal-next	defect	minor
48	#139	ostimer OS_TimerSet may fail when testing on a VM	unit-test	new	Not Determined	enhancement	trivial
49	#141	Cleanup Relative Paths Used in Makefiles	unspecified	new	Not Determined	enhancement	minor
50	#142	UT assert library has uninitialized "UtTestDataBase" global variable	unit-test	new	Not Determined	enhancement	minor
51	#143	stack_size parameter should be checked for reasonable value in OS_TaskCreate vxWorks implementation	unspecified	new	osal-next	defect	major
52	#145	OS_API_Init() should be called before any OSAL calls are used in the unit tests	unit-test	new	osal-next	defect	
53	#148	OS_API_Init() does not correct cleanup resources on error	unspecified	new	Not Determined	defect	major
54	#149	vxWorks OSAL implementation should use static initialization where possible	unspecified	new	Not Determined	enhancement	major
55	#153	utlist can use malloc(0) and memcpy of 0 size data	unspecified	new	Not Determined	defect	major
56	#157	Clear BSS on App Restart	other	new	Not Determined	enhancement	minor

1.4 TESTED PLATFORMS AND SUPPORTED OS

OSAL version 4.2.0 includes the following OS implementations:

- POSIX
- RTEMS

- VxWorks 6.x

Table 1.4-1 identifies the platforms and development tools used to verify OSAL version 4.2.0. Functional testing of OSAL 4.2.0 has been done in a PC/Linux environment, Qemu PPC750 simulator running VxWorks 6.9, and Gaisler TSIM2 LEON3 simulator running RTEMS 4.11 and 4.12. Black box unit testing has been done in a PC/Linux environment and SPARC Leon3 ut699, running VxWorks 6.7.1.

Table 1.4-1 – Tested Platforms and Verification

Test Type	Platform Information	Results Location
Functional	VxWorks 6.9 Qemu PPC750 simulator	/src/tests/Results/pc-linux
Functional	Linux wirbelwind 4.3.3-gentoo x86_64 Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz GenuineIntel GNU/Linux gentoo linux/64bit built with gcc-5.3 and glibc 2.21-r1	/src/tests/Results/vxworks
Functional	RTEMS 4.11 and 4.12 Gaisler TSIM2 LEON3 simulator	/src/tests/Results/rtems
Black Box Unit Test	Linux wirbelwind 4.3.3-gentoo x86_64 Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz GenuineIntel GNU/Linux gentoo linux/64bit built with gcc-5.3 and glibc 2.21-r1	/src/unit-tests/Results/pc-linux
Black Box Unit Test	VxWorks 6.7.1 SPARC Leon3 ut699 development board SPARC BSP 1.0.13 sparc-wrs-vxworks-4.1-1.0.12	/src/unit-tests/Results/vxworks

2.0 DELIVERED PRODUCTS

Table 2-1 identifies the products relevant to this release. The version or date of the release products and where the product can be located are also provided. Changes from the previous version are identified.

Table 2-1 – Delivered Products and their Locations

Software Element	Changed with this Version?	New Version or Date	Location
Executable for this release	Yes	4.2.0	N/A. Executables are not delivered for the OSAL
Installation Procedures & Special Instructions	Yes	1/31/16	See OSAL Configuration Guide in /doc babelfish.arc.nasa.gov (in git system master branch) and http://sourceforge.net/projects/osal/
Source Code of this release	Yes	4.2.0	babelfish.arc.nasa.gov (in git system master branch) and http://sourceforge.net/projects/osal/
Build Plan	No	N/A	None
Annotated S/W Detailed Design Docs	No	N/A	None
Ground System Scripts developed by FSB	No	N/A	See functional tests in /src/tests babelfish.arc.nasa.gov (in git system master branch) and http://sourceforge.net/projects/osal/
Simulator and Test Data Generator Software	N/A	N/A	None
Executable - Ground Tools associated with FSW (tools to build OSAL implementation)	Yes	N/A	See cmake build scripts babelfish.arc.nasa.gov (in git system master branch) and http://sourceforge.net/projects/osal/
Source Code - Ground Tools associated with FSW (tools to build stored command loads, etc.)	Yes	N/A	None
Unit Test Procedures	Yes	4.2.0	See black box tests in /src/unit_tests and white box tests in /src/unit-test-coverage babelfish.arc.nasa.gov (in git system master branch) and http://sourceforge.net/projects/osal/
Unit Test Data	No	N/A	None
Unit Test Results	Yes	4.2.0	Tlserver3.gsfc.nasa.gov (in MKS CM system)
FSW Make Files	Yes	4.2.0	See “classic” build makefiles babelfish.arc.nasa.gov (in git system master branch) and http://sourceforge.net/projects/osal/
Linker & Compiler Configuration Files	No	N/A	babelfish.arc.nasa.gov (in git system master branch) and http://sourceforge.net/projects/osal/

3.0 INSTALLATION PROCEDURES

Table 3-1 identifies the nominal Installation Procedure(s) for this release onto the intended target system. The procedure version identifier, the date of the procedure and where it can be located are also provided. In addition, the readme file that is included with the release provides a set of "Getting Started" instructions.

Table 3-1 Installation Procedure(s)

Destination (Target System)	Filename	Version and Date	Location
Procedure is generic for each OS and CPU	OSAL-Configuration-guide.pdf	4.2.0	babelfish.arc.nasa.gov (in git system master branch) and http://sourceforge.net/projects/osal/ in the /doc directory

4.0 CONFIGURATION SUMMARY AND VERSION IDENTIFICATION

OSAL Build 4.2.0 can be found on babelfish.arc.nasa.gov and is provided as open source on sourceforge.net:

http://sourceforge.net/projects/osal/

OSAL version information is documented in the following source file: /src/os/inc/osapi-version.h.

5.0 RELEASE HISTORY

Table 5.0-1 provide the release notes from previous OSAL releases.

Table 5.0-1 Release History Notes

Version	Release Date	Release Notes
4.1.1	April 28, 2014	This release fixes two issues: 1. The posix port OS_QueueCreate (posix message queue version) was hardcoding the queue depth rather than using the passed in parameter. 2. A bug was introduced in 4.1.0 in OS_open that caused a zero length file to be created when a file is opened with the OS_READ_ONLY flag and it does not exist. The original change was intended to create a new file if one did not exist, but only if the file was opened as READ/WRITE or WRITE ONLY.
4.1.0	January 31, 2014	This release contains one new function, and one slightly altered function: The new function OS_GetFsInfo returns information about the file systems including: • Number of mounted/mapped volumes • Maximum number of mounted/mapped volumes • Number of open files • Maximum number of open files OS_QueueGet has a slightly modified behavior. The size passed is the size of the buffer that the message is supposed to be copied into. If the size of the buffer passed in is smaller than the maximum size of the message specified when the queue was created, then the call will return an OS_INVALID_SIZE error. This will prevent buffer overflow errors. Previous versions of OS_QueueGet required the size of the buffer to match the exact size of the Queue and message being received. This works fine for a system with fixed size messages, but will not work for a system that uses variable sized messages. This release adds a suite of unit tests developed primarily by Tam Ngo of NASA/JSC. The tests run on Linux and use gcov to provide code coverage information. This release fixes a number of bugs and adds a number of improvements: • Fixed issues reported by static analysis tool

		• Implemented signal mask improvements on POSIX port • Improved OS_TaskDelay on POSIX port • Added additional parameter checks in osloader ports • Fixed the priority and scheduler selection on the POSIX port (priorities were being ignored) • Removed error printfs in POSIX port • Fixed incorrect table reference in CountSemCreate POSIX port • Fixed minor bugs in POSIX osfilesys and osfileapi functions • Fixed POSIX OS_TimerCreate • Fixed divide by zero bug in OS_Milli2Ticks • Improved POSIX mutex lock by masking signals • Improved Queue handling by allowing variable size messages and preventing buffer overflows on RTEMS queue receive • Updated common_types.h include file to include ARM and x86_64 • Added C++ extern "C" keywords in include files
4.0.0	January 16, 2013	This release contains no new API functionality. This release focuses on improving documentation and fixing bugs. This release removes support for OS X and Cygwin as OSAL targets. Support for OS X and Cygwin was out of date and incomplete. Due to limited resources and the ease of deploying linux virtual machines on OS X and Windows, it was decided to focus on Linux. Documentation fixes: cleaned up call restrictions, return codes, and flags that were documented but not in the code. The "apps" directory has been removed and replaced with "examples" and "tests". New tests have been added to test the semaphores. Expect additional tests in future releases. Overhaul of the binary and counting semaphores on all 3 ports. The posix port now uses pthread condition variables and mutexes for a more robust implementation. The vxworks and rtems ports use the native binary and counting semaphores and no longer try to maintain counters in the OSAL. The result is a faster and more robust implementation. Fixed incorrect comments in vxWorks OS_TaskCreate function header Removed unused variable in RTEMS port

		Fixed define in OS_API_Init in posix port Fixed timer structure initialization in posix port Fixed use of size_copied parameter in RTEMS OS_QueueGet Fixed use of access and mode parameters for OS_open and OS_creat in all ports Fixed mutex protection in OS_TaskCreate Fixed OS_FDGetinfo to use correct return codes Removed second "close" call in OS_close functions Close file descriptor in vxworks OS_unmount to remove memory/fd leak Use posix statfs on vxworks OS_fsBytesFree instead of FIONFREE64 ioctl Fixed OS_mv in rtems and vxworks to work across volumes and be consistant
3.5.0	April 18, 2012	Incorporated suggestions from RTEMS port code walkthrough. Mostly Cosmetic changes, but there were a few semaphore fixes. Inhibit output from OS_printf if called from an ISR (RTEMS only) Added OS_printf_enable and OS_printf_disable API Added OS_USED macro to common_types.h for the GNU "used" attribute Fixed error in OSAL API Document for OS_QueueGet
3.4.1	January 17, 2012	Quick fix: The OS X port had a compilation error: - added -m32 to OS X link rule
3.4.0	December 5, 2011	Added OS_rewinddir API Removed OS_MEM_TABLE_SIZE from osconfig.h -- no longer used Changed the RTEMS volatile/ram disk from NVRAM disk to the regular RTEMS RAM disk for efficiency Completed the implementation of the RTEMS shell command API. It works with RTEMS 4.10+ to execute a shell command and return the results. Improved the error handling in some of the example programs Protected internal data structures in Counting Semaphore APIs in all host OSs Fixed OS_creat in RTEMS where it was not overwriting an existing file
3.3.0	May 31, 2011	Added an API to close a file given the original filename/path

		Added an API to close all files opened by the OSAL Changed OS_stat to not look at the length of a directory segment as a file (length restriction) Added permissions to vxworks6/OS_creat so it will work on an NFS volume In vxworks6, replaced xbdBlkDev calls with "sync" versions to allow the xbd volumes to be created without a hard-coded delay after the call. Implemented symbol table dump function on vxworks6 and RTEMS (RTEMS using the GSFC static loader) Removed the -fvolatile compiler option from the PPC vxWorks makefiles. This is no longer needed for vxWorks 6.x Added semaphore protection around the file system functions in RTEMS. RTEMS does not provide protection in it's high level file system calls Fixed RTEMS OS_cp error Fixed vxworks6 OS_BinSemTimedWait - It was incrementing the incorrect counter. Improved posix message queue and semaphore pends. Now pends that were interrupted by a signal are continued. Improved posix message queue port to create unique message queue names for each process. This allows multiple OSAL apps to run on one machine Simplified posix file system path mapping. Now the path mapping does not try to create or delete directories on the running system. The OSAL path to host path translation is a simple 1 to 1 mapping. For example: OSAL path "/cf/apps" can translate to "/media/compactFlash0/apps". The OSAL will not try to create "ramdev0" etc. Cleaned up the documentation, code comments and OSAL code with regard to return codes. The return codes are consistent with the API guide and each port conforms to the documentation much better. There are still a few instances where functions are not implemented on one of the ports.
3.2.0	November 15, 2010	Various bug fixes in the RTEMS port. There were left over internal posix mutexes and a couple of cut and paste errors with the internal muteness. Added a new API: OS_FileOpenCheck Removed special symbols from source code (the copyright symbol). This was causing some debuggers and editors trouble. Updated some of the make rules for RTEMS 4.10

3.1.0	March 10, 2010	Removed the "arch" directory which had the porting layer for the OSAL. This has been simplified and turned into the "bsp" directory. This is where the OSAL port to a particular board/OS is done. For example, under the old "arch" directory structure we had: src/arch/ppc/mac/osx and src/arch/x86/mac/osx. These ports were nearly identical, yet they had a bunch of code that has to be maintained and tested. The new "bsp" directory structure has "bsp/mac-osx" which can be used to make the OSAL run on an intel mac. It could be used for a PPC mac with a few changes. Overall, another move for simplicity and ease of maintenance. Consolidated the "osx" and "linux" ports into "posix". We had considered dropping OS X, but it is close enough to warrant a single "posix" port with a few "ifdefs" to make it work. This removed over 3k lines of code from the OSAL. Verified Cygwin operation using Cygwin version 1.7.1. Cygwin 1.7.1 works almost identical to Linux for the OSAL. Earlier versions of Cygwin are not supported. Removed all POSIX code from the "rtems" port. This makes the rtems port more consistent and, in my opinion cleaner. Added support for the CEXP and a static loader in RTEMS. Neither one are included, but it is possible to do dynamic loading in RTEMS. Eventually RTEMS will have its own native dynamic loader. Added support for creating a RAM disk in RTEMS using the NVRAM disk device and the RFS file system. These are new features in RTEMS 4.10. Added an API to return the free bytes in a file system (required a 64 bit data type) Various fixes for warnings Various bug fixes Future releases: - Still would like a Win32 port. Preferably using the MinGW32 compiler. - Need to make sure the OSAL works correctly on 64 bit OSs.
3.0.0		Removed the hardware API. Now the OSAPI is more focused on the Operating System abstraction and not trying to abstract hardware. One of the main reasons for doing this was that the hardware platforms were just not being maintained and updated as they should. For NASA, we split these functions out of the OSAL and incorporated them into our cFE platform support package (where they were being copied anyway). The end result is that the OSAL project is just trying to do one thing: abstract the RTOS. Various bug fixes
2.12.0	September 5, 2008	Finished Memory Range API Progress on the Loader and Symbol API: The vxWorks, linux, and RTEMS APIs are complete with the exception of the dump symbol table API. This will probably never work on the linux/OSX/Cygwin ports. The RTEMS Loader and Symbol API

		had to be left incomplete for this release due to time constraints. This will be finished in version 2.13 (hopefully before the end of 2008). The RTEMS port will rely on the CEXP dynamic loader. Broke apart the osapiarch.c file into osmemeeprom.c, osmemport.c, osmemram.c, osmemrange.c and osmemutils.c. This was done to make the differences between platforms easier to deal with. Also, since all of this code is generic on the existing platforms, I created a src/arch/common directory with only one copy of these files. If you need to customize one of them, copy the file and put it in the arch/<cpu>/<platform>hal directory. The makefile will pick it up from there first. Created an OS timer API -- This is documented in the API reference. The vxWorks, Linux, and RTEMS timer APIs are complete for this release. The OS X timer API will be complete on the next release, but it will probably not be too pretty :) OS X lacks the POSIX timer API, so the timer code has to be handled with one timer interrupt. Also, cygwin support is unknown at this point. That will be addressed in the next release as well Added a timer test and a memory range test sample to the apps.
2.11.0	February 14, 2008	Update OS X BSP to support 10.5 Fix queue timeout implementation in OS X and Linux in the socket queue implementation. It now uses select instead of a wait loop. Fix bug in OS_open and OS_creat for all ports regarding path length #define used Fixed Application link rule in Cygwin Added POSIX message queue implementation in Linux. Linux can use the message queues rather than UDP sockets. Updated binary and counting semaphore implementations for all ports to not be able to have a semaphore 'give' and increment its value beyond its maximum value Added a way to get the value of a binary or counting semaphore through OS_*SemGetInfo return structure. Added a function for remapping the OSAL priorities to the underlying OS's. The priority levels are now completely abstracted. Added a task delete hook handler for a task to clean up its own non OSAL resources. Fixed task create problem in cygwin Added valid memory range checking API -- The implementation is not complete, and will be in version 2.12 New dynamic load/ symbol table lookup API -- The implementation is not complete, and will be in version 2.12

		All sample programs (example1, test1, and test2) work on Linux, OS X, Cygwin, vxWorks-6.4, and RTEMS
2.10.0	October 25, 2007	A Counting Semaphore API was added. All OS APIs have corresponding delete APIs (delete tasks, queues, semaphores, etc) Many bugs have been fixed. The OSAPI internal data structures are now all gaurded by mutexes. The OS_printf API has been added along with a utility task that buffers the output, rather than dumping it to stdout. New interrupt functions have been added to correspond to vxWorks intLock and intUnlock. Additional file system APIs have been added: OS_initfs, OS_GetPhysDriveName, OS_cp, OS_mv, OS_FDGetInfo, OS_rmfs The File system API now uses its own file descriptor, rather than passing the system file descriptor through. The Makefile/build system has been re-done. Now all of the OSAL code and example programs are built in a separate directory Obsolete OSs and architectures have been removed. The following OSs are supported: Mac OS X, Linux/Cygwin, RTEMS, and vxWorks 6.x The following platforms are supported in this release: Generic PPC/vxWorks 6.4, x86 Linux, x86 and PPC Mac OS X, Coldfire/RTEMS Other Platforms and Architectures should be easy to add (i.e. Sparc/LEON RTEMS, ARM/RTEMS, x86 works, etc.)
2.0.0 – 2.9.0		Nicholas Yanchik, NASA/GSFC, Code 582 - Version 2 API coding - Documentation - Tests and examples - (just about everything in version 2) Jacob Hageman, NASA/GSFC, Code 582 - Testing/updating Linux version to run on Cygwin
1.0.0		Alan Cudmore, NASA/GSFC, Code 582 - Original design and coding

		- POSIX based ports - Directory structure and makefiles J-P Swinski, NASA/GSFC, Code 582 - Coding/vxWorks port Ezra Yeheskeli, NASA/GSFC, RSC - Design and coding - RTEMS Port - Documentation

ACRONYMS

AES……..…….Advance Exploration Systems
API……..…….Application Program Interface
cFE.....Core Flight Executive
C&DH..... Command and Data Handling
cFS……Core Flight Software
CM ..... Configuration Management
CPM ..... CFS Performance Monitor
COTS ..... Commercial Off-The-Shelf
DCR ..... Discrepancy/Change Request
ES ..... Executive Services
ETU..... Engineering Test Unit
FSB..... Flight Software Branch
FSW .....Flight Software
JSC ..... Johnson Space Center
I&T ..... Integration & Test
MMS ..... Magnetospheric Multiscale Mission
OSAL……..……Operating System Abstraction Layer
POSIX .....Portable Operating System Interface
RTOS ..... Real-Time Operating System
SPARC.....Scalable Processor Architecture
TBL……..……Table
T&C..... Telemetry and Command
URL..... Universal Resource Locator
UTF……..….Unit Test Framework
VDD ..... Version Description Document