**SIL FSW Integration Guide**

# Purpose

The goal of this guide is to walk a FSW developer through the process of integrating a SIL application into the CFS environment.

## Definitions

## App Name

For the purposes of this guide, the app being integrated will be identified by *<XX> and <xx>, where <XX>* is the uppercase identifier and *<xx>* is the lowercase identifier. Anytime *<XX>* is found within this guide it should be replaced by the uppercase identifier for the particular app being integrated, same with *<xx>*.

For example, to integrate an app called AC:

*‘<xx>*\_perfids.h’ should be ‘ac\_perfids.h’

*‘<XX>*\_CMD\_MID’ should be ‘AC\_CMD\_MID’

## Build Directory

When found in a path, <cfs directory> should be replaced with the root of the CFS development directory.

# Steps

1. Create or update *<xx>*\_perfids.h in <cfs directory>/apps/*<xx>*/fsw/mission\_inc with a macro definition, XX\_PERF\_ID. Verify that it is defined as a unique positive, integer value. If updated, make sure this file is copied to directory <cfs directory>/build/mission\_inc during the make process.
2. Create or update xx\_msgids.h in <cfs directory>/apps/*<xx>*/fsw/platform\_inc with the following preprocessor directives before any macro definitions:

#ifdef SIL\_SEND\_HK\_MID

#undef SIL\_SEND\_HK\_MID

#endif

#ifdef SIL\_TBL\_MANAGE\_MID

#undef SIL\_TBL\_MANAGE\_MID

#endif

In order to avoid potential compiler errors regarding redefining any macro defintions.

1. Create or update xx\_msgids.h in <cfs directory>/apps/*<xx>*/fsw/platform\_inc with the following macro definitions:

* 1. *<XX>*\_CMD\_MID
  2. SIL\_SEND\_HK\_MID
  3. SIL\_TBL\_MANAGE\_MID
  4. *<XX>*\_FDC\_MID
  5. *<XX>*\_TICK\_MID
  6. *<XX>*\_HK\_MID
  7. Macros defined in SIL\_MsgSnd[] in cfe\_sl\_interface.h

Define each with a hexadecimal value that conform with CCSDS standards for telemetry and command messages.

1. Include macros definitions for the macros in SIL\_MsgRcv[], listed in cfe\_sl\_interface.h, in *<xx>*\_msgids.h. If necessary, define them with macros from other application message ID header files. However, make sure to include these other header files in *<xx>*\_msgids.h BEFORE the preprocessor directives mentioned in step 2.
2. If *<xx>*\_msgids.h is updated, make sure that this file is copied to directory <cfs directory>/build/<target>/inc during the make process.
3. Create, if necessary, app\_perfids.h in <cfs directory>/apps/*<xx>*/fsw/sil\_inc. Include “*<xx>*\_perfids.h”.
4. Create/update table\_validation.h, as necessary, in <cfs directory>/apps/*<xx>*/fsw/sil\_inc. In table\_validation.h, define the necessary number of table validation function prototypes. Each function prototype should have the following format:

int32 <Table Validation Function Name>(void\* TblPtr);

1. In table\_validation.h, verify that a structure called

CFE\_TBL\_CallbackFuncPtr\_t TableValidationTbl[]

is specifed. Set it equal to a list of function prototype pointers that were defined in step 6 (surrounded by braces). The order of the table validation function pointers in ‘TableValidationTbl[]’ correspond to the ordering of tables in ‘SL\_ParmTable[]’ in csc\_sl\_interface.h. If one of the tables in ‘SL\_ParmTable[]’ does not have/need a table validation function, place a ‘NULL’ in the corresponding entry of TableValidationTbl[] (The table values will always be accepted as valid).

1. Create / update table\_validation.c as necessary in < cfs directory>/apps/*<xx>*/fsw/sil\_inc, which specifies the the table validation functions defined in table\_validation.h.
2. Update ‘*<xx>*tables.mak’ in <cfs directory>/apps/*<xx>*/fsw/for\_build> and verify the following:
   1. PARENTAPP = ‘*<xx>*’
   2. TABLES = <Table source file name in ‘tables’ directory>.tbl … for all table source files in the ‘tables’ directory.

If updated, verify that a copy of ‘*<xx>*tables.mak’ is copied to <cfs directory>/build/cpu1/xx.

1. Update ‘Makefile’ in <cfs directory>/apps/*<xx>*/fsw/for\_build> and verify the following:
   1. APPTARGET = ‘*<xx>*’
   2. ENTRY\_PT = SIL\_MAIN (in cfe\_sl\_interface.h)
   3. OBJS := <source files name in delivered ‘src’ directory>.o … for all sources files in ‘src’ directory
   4. OBJS += <Table source file name in ‘tables’ directory>.o … for all table source files in the ‘tables’ directory.
   5. OBJS += table\_validation.o
   6. OBJS += app\_faultrep.o (if FDC reporting is enabled)
   7. VPATH += $(CFS\_APP\_SRC)/sil/fsw/faultrep (if FDC reporting is enabled)
   8. VPATH += $(CFS\_APP\_SRC)/$(APPTARGET)/fsw/sil\_inc
   9. VPATH += $(CFS\_APP\_SRC)/$(APPTARGET)/fsw/tables
   10. -I$(CFS\_APP\_SRC)/sil/fsw/faultrep \ is added to #INCLUDE\_PATH (if FDC reporting is enabled)

If updated, verify that a copy of ‘Makefile’ is copied to <cfs directory>/build/cpu1/*<xx>*.

1. Update ‘Makefile’ in <cfs directory>/build/<target>.
   1. Add ‘*<xx>*’ to THE\_APPS
   2. Add ‘*<xx>*’ to THE\_TBLS
2. Verify the ‘*<xx>*’ application is added to cfe\_es\_startup.scr in <cfs directory>/build/<target>/exe using the line:

CFE\_APP, /cf/apps/*<xx>*.o, *<xx>*\_AppMain, *<XX>*, <priority>, <stack size>, 0x0, 0;