# CFS Housekeeping Application (HK) User's Guide

* [CFS Housekeeping Introduction](#AAAAAAAAFE)
* [CFS Housekeeping Overview](#AAAAAAAAFF)
* [CFS Housekeeping Operation](#AAAAAAAAFH)
* [CFS Housekeeping Requirements](#AAAAAAAAFN)
* [CFS Housekeeping Deployment Guide](#AAAAAAAAFO)
* [CFS Housekeeping Table Definitions](#AAAAAAAAFS)
* [CFS Housekeeping Constraints](#AAAAAAAAFW)
* [CFS Housekeeping Frequently Asked Questions](#AAAAAAAAGF)
* [CFS Housekeeping Configuration Parameters](#AAAAAAAAFP)
* [CFS Housekeeping Event Message Reference](#AAAAAAAABY)

# CFS Housekeeping Introduction

#### Scope

This document provides a complete specification for the commands and telemetry associated with the CFS Housekeeping (HK) application software. The document is intended primarily for users of the software (operations personal, test engineers, and maintenance personnel). The last section of the document, the deployment guide section, is intended for mission developers when deploying and configuring the FM application software for a mission flight software build environment.

#### Applicable Documents

|  |  |
| --- | --- |
| **Document ID** | **Document Title** |
| TBD | CFS Housekeeping Application Requirements Document |
| TBD | CFS Housekeeping Heritage Analysis Document |
| TBD | CFS Housekeeping Design Document |
| TBD | *Need Reference to OSAL Document* |

#### Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| API | Application Programming Interface |
| ATP | Absolute Time Processor |
| ATS | Absolute Time tagged command Sequence |
| CCSDS | Consultative Committee for Space Data Systems |
| C&DH | Command and Data Handling |
| CFE | Core Flight Executive |
| CFS | Core Flight System |
| CI | Command Ingest |
| Cmd | Command |
| CPU | Central Processing Unit |
| EDAC | Error Detection and Correction |
| FDS | Flight Data System |
| FM | File Manager |
| FSW | Flight Software |
| GN&C | Guidance Navigation & Control |
| GSFC | Goddard Space Flight Center |
| HK | Housekeeping |
| HW, H/W | Hardware |
| ICD | Interface Control Document |
| ISR | Interrupt Service Routine |
| OS | Operating System |
| OSAL | Operating System Abstraction Layer |
| Pkts | Packets |
| RAM | Random-Access Memory |
| RTOS | Real Time Operating System |
| RTP | Relative Time Processor |
| RTS | Relative Time tagged command Sequence |
| SB | Software Bus Service |
| SBC | Single Board Computer |
| SC | Stored Commands task |
| SW, S/W | Software |
| TBD | To Be Determined |
| TBL | Table |
| TLM | Telemetry |
| UTC | Universal time code |

# CFS Housekeeping Overview

The Housekeeping (HK) component of the Core Flight System (CFS) is responsible for building and sending combined telemetry messages (from individual system applications) to the software bus for routing. Combining messages is performed in order to minimize downlink telemetry bandwidth. Combining certain data from multiple messages into one message eliminates the message headers that would be required if each message was sent individually. Combined messages are also useful for organizing certain types of data. This application may be used for data types other than housekeeping telemetry. HK provides the capability to generate multiple combined packets (a.k.a. output packets) so that data can be sent at different rates (e.g. a fast, medium and slow packet).

The HK Application is a passive application and is completely packet driven. There are no timers used to wake up the application. The housekeeping request commands are not sent to the system apps by HK. A common system design uses the Scheduler (SCH) Application to send the housekeeping requests to the system apps as well as the send-output-message-x command to HK.

A key component that is part of the HK application is the 'copy table'. The copy table has a configurable number of entries. Each entry specifies a source (via msgid and offset into the packet), destination (via msgid and offset) and a number of bytes to copy.

#### HK Design Overview

The HK Application has a single Software Bus pipe and wakes up only when a message is received on the pipe (named HK\_CMD\_PIPE). The HK\_CMD\_PIPE receives commands and input messages. If the message received is not a command, the code extracts the message ID from the packet and begins scanning the copy table for a matching Message ID. Once a match is found, the data is copied from input message to the output message specified in the copy table entry. The code then continues searching for another entry in the table with the same Message ID. The number of destinations for input message data is limited only by the number of entries in the table. Data from an input message may be copied to many different output packets and may also be copied to more than one area in the same output packet. The number of output packets is also limited only by the number of entries in the table. The number of entries in the table is a platform configuration parameter.

The output messages are sent to the software bus when HK receives the 'Send Output Message x' command. Where 'x' is a parameter in the command that specifies the message ID of the output message to send. If a piece of data is missing in the output message at the time the 'Send Output Message x' command is received, the HK app will send a debug event (which is filtered by default) and increment the 'missing data counter' in telemetry.

# CFS Housekeeping Operation

The operational interface of the HK application consists of two commands, two tables and a few telemetry points. The following items detail the operational interface.

#### 1. The application version number

The application version number is displayed in the initialization event and the no-op event. Both events are 'informational' type and are unfiltered by default.

#### 2. Loading the copy table

The copy table is loaded by way of a file. The file must be present when the application starts up. Otherwise, the app will terminate after initialization. The location of the file is specified by the platform configuration parameter named [HK\_COPY\_TABLE\_FILENAME](#AAAAAAAAAT). The value of this define specifies the full path as well as the filename.

After initialization, a new copy table may be loaded at any time. The steps to load a new copy table are:

* Transfer the table file to the on-board file system
* Send the cFE Table load command
* Send the cFE Table validate command
* Send the cFE Table activate command

The HK application will clean up the items used for the old table (such as SB subscriptions) before updating and processing the new table.

#### 3. Sending the No-op Command

To verify connectivity with the Housekeeping application, the ground may send an [HK No-op command](#AAAAAAAABH). If the packet length field in the command is set to the value expected by the HK app, then the command counter will increment and a no-op informational event message will be sent. This no-op event will show the version number of the HK application.

#### 4. Sending the reset counters command

The [reset counters command](#AAAAAAAABI)  will reset only the telemetry points listed here:

a. [Command counter](#AAAAAAAAFJ)

b. [Command error counter](#AAAAAAAAFK)

c. [Combined packets sent counter](#AAAAAAAAFL)

d. [Missing Data counter](#AAAAAAAAFM)

#### 5. Monitoring the command counter

The [command counter](#AAAAAAAAFJ)  will increment only when the HK application receives a valid [HK No-op command](#AAAAAAAABH).

#### 6. Monitoring the command error counter

The [Command error counter](#AAAAAAAAFK)  will increment under the following conditions:

a. Invalid command code

b. Unexpected packet length field for ['Send Combined Tlm'](#AAAAAAAAAG)  command

c. Unexpected packet length field for ['Send HK Housekeeping'](#AAAAAAAAAF)  command

d. Unexpected packet length field for ['HK No-op'](#AAAAAAAABH) command

e. Unexpected packet length field for ['Reset Counters'](#AAAAAAAABI) command

#### 7. Monitoring the 'Combined Packets Sent" counter

Each time a ['Send Combined Tlm'](#AAAAAAAAAG)  command is received without error the [Combined packets sent counter](#AAAAAAAAFL)  will increment.

#### 8. Monitoring the 'Missing Data" counter

Each time a ['Send Combined Tlm'](#AAAAAAAAAG)  command is received without error, the HK app begins checking the 'Data Present' flag (located in the run-time table) for each of the data sections that make up the output packet. If a data portion is missing, HK sends a debug event (which is filtered by default) and increments the ['Missing Data'](#AAAAAAAAFM)  counter. The event will display the Message ID of the input message that would normally provide the missing portion.

**NOTE:** HK will report only one missing data portion in any output message. The ['Missing Data'](#AAAAAAAAFM)  counter will advance by one count (at most) and send one event (at most) for each ['Send Combined Tlm'](#AAAAAAAAAG)  command.

The HK app will not zero-out or alter the missing data section(s) in any way. The missing data values will match the last 'good' section received.

#### 9. Using the Memory Pool handle to get mempool stats

The HK memory pool is used to allocate the memory needed to store the output packets. Each time a new copy table is processed, the memory for the output packets is dynamically allocated from the memory pool. The memory pool handle is sent down in the [HK housekeeping telemetry packet](#AAAAAAAADP). It is possible to get statistics from the cFE ES application on the memory pool used by this application. The [cFE ES command](#AAAAAAAAFI)  used to get statistics will need this memory pool handle as a command parameter.

Next: [CFS Housekeeping Requirements](#AAAAAAAAFN)

Prev: [CFS Housekeeping Overview](#AAAAAAAAFF)

# CFS Housekeeping Requirements

#### Subsystem Requirements:

**CFS-100** The CFS shall provide the capability to combine housekeeping data from system applications into output messages

#### Basic Command Requirements:

**HK1000** Upon receipt of a No-Op command, HK shall increment the HK Valid Command Counter and generate an event message.

**HK1001** Upon receipt of a Reset command, HK shall reset the following housekeeping variables to a value of zero:

a) HK Valid Command Counter

b) HK Command Rejected Counter

c) Number of Output Messages Sent

d) Missing Data Counter

**HK1002** For all HK commands, if the length contained in the message header is not equal to the expected length, HK shall reject the command.

**HK1003** If HK accepts any command as valid, HK shall execute the command, increment the HK Valid Command Counter and issue an event message

**HK1004** If HK rejects any command, HK shall abort the command execution, increment the HK Command Rejected Counter and issue an error event message

#### Operational Requirements:

**HK2000** HK shall collect flight software housekeeping data from table-specified input messages.

**HK2001** HK shall output table-defined messages, at the scheduled rate, by combining input message data starting at the table-defined offset and table-defined number of bytes to the table-defined offset in the output message.

**HK2001.1** Upon a table update, HK shall update the output message formats specified in the table during normal execution.

**HK2001.2** If HK does not receive a message from an application, HK shall use all values associated with last received message for that application in the combined message for that telemetry collection period.

**HK2001.3** If HK does not receive a message from an application, HK app shall increment the missing data counter and send an event specifying the message ID for the missing message

**HK2001.5** If the input message offset + bytes for any input message specified in the HK table is greater than the received message length then HK shall use the last received data associated with that message and issue an event message

#### Status Reporting Requirements:

**HK3000** HK shall generate a housekeeping message containing the following:

a) Valid Command Counter

b) Command Rejected Counter

c) Number of Output Messages Sent

d) Missing Data Counter

#### Initialization Requirements:

**HK4000** Upon initialization of the HK Application, HK shall initialize the following data to zero:

a) Valid Command Counter

b) Command Rejected Counter

c) Number of Output Messages Sent

d) Missing Data Counter

# CFS Housekeeping Deployment Guide

Follow the general guidelines below for platform deployment of the Housekeeping app.

There are two message IDs that must be included in the CFS Scheduler Table: [HK\_SEND\_HK\_MID](#AAAAAAAAAF) is sent out at the housekeeping request interval. The housekeeping app must send its housekeeping data to itself like any other app. [HK\_SEND\_COMBINED\_PKT\_MID](#AAAAAAAAAG) is sent out at the desired rate for each combined packet. UP to four combined packets are available.

The HK app can build up to four combined packets based on the definitions supplied in the HK copy table. The Scheduler Table uses each telemetry IDs ([HK\_COMBINED\_PKT1\_MID](#AAAAAAAAAI) , [HK\_COMBINED\_PKT2\_MID](#AAAAAAAAAJ) , [HK\_COMBINED\_PKT3\_MID](#AAAAAAAAAK) , and [HK\_COMBINED\_PKT4\_MID](#AAAAAAAAAL) ) as a valid parameter in the [HK\_SEND\_COMBINED\_PKT\_MID](#AAAAAAAAAG) table entries.

The ES app uses the HK performance ID, [HK\_APPMAIN\_PERF\_ID](#AAAAAAAAAC) , to keep track of the performance of the HK app.

The platform configuration file [hk\_platform\_cfg.h](#AAAAAAAAAM) contains parameters that can be adjusted to specific platforms. The defined parameters (and their default values) are:

[HK\_PIPE\_DEPTH](#AAAAAAAAAN) defines the depth of the HK app command pipe. The default value is 40. When modifying this value, take into account not only the expected number of ground commands but the number of scheduler table entries for HK. These entries will include requests to send combined telemetry packets as well as requests to send HK housekeeping telemetry packet to itself.

[HK\_COPY\_TABLE\_ENTRIES](#AAAAAAAAAP) defines the size of the HK Copy Table. The default value is 128. When modifying this value, the system engineer should consider the number of combined packets to be used as well as the number of apps whose housekeeping telemetry will be included in each combined packet.

[HK\_NUM\_BYTES\_IN\_MEM\_POOL](#AAAAAAAAAQ) defines the amount of memory dynamically allocated for the combined packets. The default value is 6144 (6KB) but must be large enough to accommodate the packet sizes as well as overhead required for the memory pool management.

[HK\_RUNTIME\_TABLE\_NAME](#AAAAAAAAAS) is the name of the HK Run-time table. The default value is "RunTimeTable".

[HK\_COPY\_TABLE\_FILENAME](#AAAAAAAAAT) is the name of the HK Copy Table. The default value is "/cf/hk\_cpy\_tbl.tbl". Note that the path is specified in the filename definition.

Housekeeping app configuration parameter details are also available at [CFS Housekeeping Configuration Parameters](#AAAAAAAAFP)

# CFS Housekeeping Table Definitions

The Housekeeping Application uses two tables. A load-dump table referred to as the "copy table" and a dump-only table referred to as the "run-time table". Each table has the same number of entries, defined by the configuration parameter [HK\_COPY\_TABLE\_ENTRIES](#AAAAAAAAAP). Each run-time table entry has a direct correspondence to the same copy table entry. During normal operation, the fields in the run-time table change periodically while the copy table fields are static. Two tables were chosen so that checksumming can be executed on the more-static copy table.

**HK Copy Table Validation** - The HK copy table currently has an empty validation call-back function that always returns success. At the time of development, a validation process that would apply to all projects was not perceived.

**HK Copy Table Entries** - Each entry in the copy table has five fields and can be translated to the following phrase:

*Copy A bytes from input message B, byte-offset C to output message Y, byte-offset Z.*

The structure format of a single copy table entry is as follows:

typedef struct

{

CFE\_SB\_MsgId\_t InputMid;

uint16 InputOffset;

CFE\_SB\_MsgId\_t OutputMid;

uint16 OutputOffset;

uint16 NumBytes;

} hk\_copy\_table\_entry\_t;

**HK Run-time Table Entries** - Each entry in the run-time table holds the address of the output packet and has variables that tell whether the input message ID has been subscribed to and whether or not the data from this entry is present in the output packet (since the last time the output packet was sent).

The structure format of a single run-time table entry is as follows:

typedef struct

{

CFE\_SB\_MsgPtr\_t OutputPktAddr;

uint8 InputMidSubscribed;

uint8 DataPresent;

} hk\_runtime\_tbl\_entry\_t;

**HK Copy Table Updates** - The copy table is polled for changes every housekeeping request. When an update is pending, HK cleans-up the SB subscriptions and frees the memory (allocated for output messages) from the old table. After the old table cleanup is complete, the new table is updated, then processed. The processing involves subscribing to input messages, allocating memory for the output messages and initializing runtime variables.

# CFS Housekeeping Configuration Parameters

The following are configuration parameters used to configure the CFS Housekeeping Application either for each platform or for a mission as a whole.

##### Global [HK\_COPY\_TABLE\_ENTRIES](#AAAAAAAAAP)

Maximum Number of HK Copy Table Entries

##### Global [HK\_COPY\_TABLE\_FILENAME](#AAAAAAAAAT)

HK Copy Table Filename

##### Global [HK\_COPY\_TABLE\_NAME](#AAAAAAAAAR)

Name of the HK Copy Table

##### Global [HK\_DISCARD\_INCOMPLETE\_COMBO](#AAAAAAAAAO)

Discard Incomplete Combo Packets

##### Global [HK\_MISSION\_REV](#AAAAAAAAAU)

Mission specific version number for HK application

##### Global [HK\_NUM\_BYTES\_IN\_MEM\_POOL](#AAAAAAAAAQ)

Number of bytes in the HK Memory Pool

##### Global [HK\_PIPE\_DEPTH](#AAAAAAAAAN)

Application Pipe Depth

##### Global [HK\_RUNTIME\_TABLE\_NAME](#AAAAAAAAAS)

Name of the HK Run-time Table

# CFS Housekeeping Constraints

The Housekeeping Application needs to find a valid table at the location specified by the configuration parameter [HK\_COPY\_TABLE\_FILENAME](#AAAAAAAAAT) in order to startup. Otherwise the application will send an error event or syslog message then terminate. See [CFS Housekeeping Table Definitions](#AAAAAAAAFS) for more detail.

The HK application must successfully initialize in order to startup. The general initialization steps are defined in [CFS Housekeeping Frequently Asked Questions](#AAAAAAAAGF) of this document.

The valid settings for the [HK\_COPY\_TABLE\_ENTRIES](#AAAAAAAAAP) configuration parameter are 1 to 8192.

# CFS Housekeeping Frequently Asked Questions

**(Q) Does the protocol for collecting telemetry use a single housekeeping request command for all apps or a unique housekeeping request command for each app?**

*It is unknown to the HK application because HK does not send the housekeeping request command(s). These commands are usually sent by the scheduler. However, each CFS application is capable of receiving a unique housekeeping request command. The message ID of this command is specified in the XX\_msgids.h header file using the format of XX\_SEND\_HK\_CMD for the #define. The 'XX' refers to the application acronym. Having unique defines for each application gives the system engineer the option of having individual housekeeping request commands for each app or using a single command for all apps. In the latter case, all the defines would be set to the same value.*

**(Q) What is the basic flow of the application?**

*The HK application uses a common application format. When the app starts, initialization is performed, then the app enters an infinite loop waiting for commands from the software bus.*

**(Q) In general, what is done during initialization?**

*During initialization, HK does the typical cFE initialization steps (create pipes, register events etc) then processes the copy table. Processing the copy table involves registering the copy table and run-time (dump-only) table, loading the copy table, subscribing to input messages defined in the copy table, allocating memory for the output messages and initializing the runtime variables. If any error is encountered during initialization, the application will send an event or syslog message, then terminate.*

**(Q) How does the app react if the file to load the copy table (during initialization) is not found?**

*HK will send an error event or a syslog message, then terminate.*

**(Q) At what frequency is the housekeeping request command sent to the system applications?**

*The HK application does not send the housekeeping request command to the system applications. The frequency of the housekeeping request commands is typically specified in the scheduler table and sent by the scheduler application.*

**(Q) What factor controls the input message timing?**

*In a common implementation when HK is used to combine housekeeping data, the input message timing is based on the period of the housekeeping request command(s). The housekeeping request command(s) are not sent by the HK application. Rather, it is typically sent by the scheduler application.*

**(Q) What determines the timing of the combined output messages?**

*The output messages are sent (via the Software Bus) only on command. A common implementation would have the scheduler sending HK the 'Send Output Message x' command periodically. Where 'x' is a parameter in the command that specifies the message ID of the output message to send.*

**(Q) How is the format of the combined output messages defined?**

*The output message format is defined by the entries in the copy table. If a format change to one or more output messages is needed, a new table must be loaded.*

# HK Event Message Cross Reference

##### Global [HK\_ACCESSING\_PAST\_PACKET\_END\_EID](#AAAAAAAACE)

'HK table definition exceeds packet length. MID:0x%08X, Length:%d, Count:%d'

##### Global [HK\_CANT\_SUBSCRIBE\_TO\_SB\_PKT\_EID](#AAAAAAAACG)

'HK Processing New Table:SB\_Subscribe for Mid 0x%08X returned 0x%04X'

##### Global [HK\_CC\_ERR\_EID](#AAAAAAAACA)

'Cmd Msg with Invalid command code Rcvd – ID = 0x%08X, CC = %d'

##### Global [HK\_CMD\_LEN\_ERR\_EID](#AAAAAAAACB)

'Cmd Msg with Bad length Rcvd: ID = 0x%X, CC = %d, Exp Len = %d, Len = %d'

##### Global [HK\_CPTBL\_GADR\_ERR\_EID](#AAAAAAAACV)

'Error Getting Adr for Cpy Tbl,RC=0x%08X'

##### Global [HK\_CPTBL\_LD\_ERR\_EID](#AAAAAAAACS)

'Error Loading Copy Table,RC=0x%08X'

##### Global [HK\_CPTBL\_MNG\_ERR\_EID](#AAAAAAAACT)

'Error from TBL Manage call for Copy Table,RC=0x%08X'

##### Global [HK\_CPTBL\_REG\_ERR\_EID](#AAAAAAAACQ)

'Error Registering Copy Table,RC=0x%08X'

##### Global [HK\_CR\_PIPE\_ERR\_EID](#AAAAAAAACL)

'Error Creating SB Pipe,RC=0x%08X'

##### Global [HK\_CR\_POOL\_ERR\_EID](#AAAAAAAACP)

'Error Creating Memory Pool,RC=0x%08X'

##### Global [HK\_INIT\_EID](#AAAAAAAABZ)

'HK Initialized. Version %d.%d.%d.%d'

##### Global [HK\_MEM\_POOL\_FREE\_FAILED\_EID](#AAAAAAAACH)

'HK TearDown Old Table: ES\_PutPoolBuf for pkt:0x%08X returned 0x%04X'

##### Global [HK\_MEM\_POOL\_MALLOC\_FAILED\_EID](#AAAAAAAACF)

'HK Processing New Table: ES\_GetPoolBuf for size %d returned 0x%04X'

##### Global [HK\_MSG\_LEN\_ERR\_EID](#AAAAAAAACZ)

'Msg with Bad length Rcvd: ID = 0x%08X, Exp Len = %d, Len = %d'

##### Global [HK\_NEWCPYTBL\_HK\_FAILED\_EID](#AAAAAAAADH)

'Process New Copy Table Failed, status = 0x%08X'

##### Global [HK\_NEWCPYTBL\_INIT\_FAILED\_EID](#AAAAAAAADI)

'Process New Copy Table Failed, status = 0x%08X'

##### Global [HK\_NOOP\_CMD\_EID](#AAAAAAAACC)

'HK No-op command, Version %d.%d.%d.%d'

##### Global [HK\_NULL\_POINTER\_NEWCPY\_ERR\_EID](#AAAAAAAADF)

'Null pointer detected in new copy tbl processing: CpyTbl = 0x%08X, RtTbl = 0x%08X'

##### Global [HK\_NULL\_POINTER\_TEARCPY\_ERR\_EID](#AAAAAAAADG)

'Null pointer detected in copy tbl tear down: CpyTbl = 0x%08X, RtTbl = 0x%08X'

##### Global [HK\_OUTPKT\_MISSING\_DATA\_EID](#AAAAAAAACK)

'Combined Packet 0x%08X missing data from Input Pkt 0x%08X'

##### Global [HK\_RCV\_MSG\_ERR\_EID](#AAAAAAAACX)

'HK\_APP Exiting due to CFE\_SB\_RcvMsg error 0x%08X'

##### Global [HK\_RESET\_CNTRS\_CMD\_EID](#AAAAAAAACD)

'HK Reset Counters command received'

##### Global [HK\_RTTBL\_GADR\_ERR\_EID](#AAAAAAAACW)

'Error Getting Adr for Runtime Table,RC=0x%08X'

##### Global [HK\_RTTBL\_MNG\_ERR\_EID](#AAAAAAAACU)

'Error from TBL Manage call for Runtime Table,RC=0x%08X'

##### Global [HK\_RTTBL\_REG\_ERR\_EID](#AAAAAAAACR)

'Error Registering Runtime Table,RC=0x%08X'

##### Global [HK\_SUB\_CMB\_ERR\_EID](#AAAAAAAACM)

'Error Subscribing to HK Snd Cmb Pkt, MID=0x%08X, RC=0x%08X'

##### Global [HK\_SUB\_CMD\_ERR\_EID](#AAAAAAAACO)

'Error Subscribing to HK Gnd Cmds, MID=0x%08X, RC=0x%08X'

##### Global [HK\_SUB\_REQ\_ERR\_EID](#AAAAAAAACN)

'Error Subscribing to HK Request, MID=0x%08X, RC=0x%08X'

##### Global [HK\_UNEXPECTED\_DUMPTOBUFFER\_RET\_EID](#AAAAAAAADE)

'Unexpected CFE\_TBL\_DumpToBuffer return (0x%08X) for Runtime Table'

##### Global [HK\_UNEXPECTED\_GETADDR\_RET\_EID](#AAAAAAAADA)

'Unexpected CFE\_TBL\_GetAddress return (0x%08X) for Copy Table'

##### Global [HK\_UNEXPECTED\_GETSTAT2\_RET\_EID](#AAAAAAAACY)

'Unexpected CFE\_TBL\_GetStatus return (0x%08X) for Runtime Table'

##### Global [HK\_UNEXPECTED\_GETSTAT\_RET\_EID](#AAAAAAAACI)

'Unexpected CFE\_TBL\_GetStatus return (0x%08X) for Copy Table'

##### Global [HK\_UNEXPECTED\_RELADDR\_RET\_EID](#AAAAAAAADC)

'Unexpected CFE\_TBL\_ReleaseAddress return (0x%08X) for Copy Table'

##### Global [HK\_UNEXPECTED\_TBLUPD\_RET\_EID](#AAAAAAAADB)

'Unexpected CFE\_TBL\_Update return (0x%08X) for Copy Table'

##### Global [HK\_UNEXPECTED\_TBLVLD\_RET\_EID](#AAAAAAAADD)

'Unexpected CFE\_TBL\_Validate return (0x%08X) for Copy Table'

##### Global [HK\_UNKNOWN\_COMBINED\_PACKET\_EID](#AAAAAAAACJ)

'Combined HK Packet 0x%08X is not found in current HK Copy Table'

# HK Telemetry Mnemonic Cross Reference

##### Global [HK\_HkPacket\_t::CmdCounter](#AAAAAAAAFJ)

##### Global [HK\_HkPacket\_t::CombinedPacketsSent](#AAAAAAAAFL)

##### Global [HK\_HkPacket\_t::ErrCounter](#AAAAAAAAFK)

##### Global [HK\_HkPacket\_t::MemPoolHandle](#AAAAAAAAHU)

##### Global [HK\_HkPacket\_t::MissingDataCtr](#AAAAAAAAFM)

##### Global [HK\_HkPacket\_t::Padding](#AAAAAAAAHX)

# Data Structure Documentation

## HK\_AppData\_t Struct Reference

HK global data structure.

#include <hk\_app.h>

### Data Fields

* [HK\_HkPacket\_t](#AAAAAAAADP) [HkPacket](#AAAAAAAAHY)

*HK Housekeeping Packet.*

* CFE\_SB\_PipeId\_t [CmdPipe](#AAAAAAAAHZ)

*Pipe Id for HK command pipe.*

* uint8 [CmdCounter](#AAAAAAAAIA)

*Number of valid commands received.*

* uint8 [ErrCounter](#AAAAAAAAIB)

*Number of invalid commands received.*

* uint16 [MissingDataCtr](#AAAAAAAAIC)

*Number of times missing data was detected.*

* uint16 [CombinedPacketsSent](#AAAAAAAAID)

*Count of combined output msgs sent.*

* CFE\_ES\_MemHandle\_t [MemPoolHandle](#AAAAAAAAIE)

*HK mempool handle for output pkts.*

* uint32 [RunStatus](#AAAAAAAAIF)

*HK App run status.*

* CFE\_TBL\_Handle\_t [CopyTableHandle](#AAAAAAAAIG)

*Copy Table handle.*

* CFE\_TBL\_Handle\_t [RuntimeTableHandle](#AAAAAAAAIH)

*Run-time table handle.*

* [hk\_copy\_table\_entry\_t](#AAAAAAAADS) \* [CopyTablePtr](#AAAAAAAAII)

*Ptr to copy table entry.*

* [hk\_runtime\_tbl\_entry\_t](#AAAAAAAADT) \* [RuntimeTablePtr](#AAAAAAAAIJ)

*Ptr to run-time table entry.*

* uint8 [MemPoolBuffer](#AAAAAAAAIK) [[HK\_NUM\_BYTES\_IN\_MEM\_POOL](#AAAAAAAAAQ)]

*HK mempool buffer.*

### Detailed Description

HK global data structure.

### Field Documentation

#### uint8 HK\_AppData\_t::CmdCounter

Number of valid commands received.

#### CFE\_SB\_PipeId\_t HK\_AppData\_t::CmdPipe

Pipe Id for HK command pipe.

#### uint16 HK\_AppData\_t::CombinedPacketsSent

Count of combined output msgs sent.

#### CFE\_TBL\_Handle\_t HK\_AppData\_t::CopyTableHandle

Copy Table handle.

#### [hk\_copy\_table\_entry\_t](#AAAAAAAADS)\* HK\_AppData\_t::CopyTablePtr

Ptr to copy table entry.

#### uint8 HK\_AppData\_t::ErrCounter

Number of invalid commands received.

#### [HK\_HkPacket\_t](#AAAAAAAADP) HK\_AppData\_t::HkPacket

HK Housekeeping Packet.

#### uint8 HK\_AppData\_t::MemPoolBuffer[[HK\_NUM\_BYTES\_IN\_MEM\_POOL](#AAAAAAAAAQ)]

HK mempool buffer.

#### CFE\_ES\_MemHandle\_t HK\_AppData\_t::MemPoolHandle

HK mempool handle for output pkts.

#### uint16 HK\_AppData\_t::MissingDataCtr

Number of times missing data was detected.

#### uint32 HK\_AppData\_t::RunStatus

HK App run status.

#### CFE\_TBL\_Handle\_t HK\_AppData\_t::RuntimeTableHandle

Run-time table handle.

#### [hk\_runtime\_tbl\_entry\_t](#AAAAAAAADT)\* HK\_AppData\_t::RuntimeTablePtr

Ptr to run-time table entry.

#### The documentation for this struct was generated from the following file:

* fsw/src/[hk\_app.h](#AAAAAAAABK)

## hk\_copy\_table\_entry\_t Struct Reference

HK Copy Table Entry Format.

#include <hk\_tbldefs.h>

### Data Fields

* CFE\_SB\_MsgId\_t [InputMid](#AAAAAAAAIL)

*MsgId of the input packet.*

* uint16 [InputOffset](#AAAAAAAAIM)

*ByteOffset into the input pkt where copy will begin.*

* CFE\_SB\_MsgId\_t [OutputMid](#AAAAAAAAIN)

*MsgId of the output packet.*

* uint16 [OutputOffset](#AAAAAAAAIO)

*ByteOffset into the output pkt where data will be placed.*

* uint16 [NumBytes](#AAAAAAAAIP)

*Number of data bytes to copy from input to output pkt.*

### Detailed Description

HK Copy Table Entry Format.

### Field Documentation

#### CFE\_SB\_MsgId\_t hk\_copy\_table\_entry\_t::InputMid

MsgId of the input packet.

#### uint16 hk\_copy\_table\_entry\_t::InputOffset

ByteOffset into the input pkt where copy will begin.

#### uint16 hk\_copy\_table\_entry\_t::NumBytes

Number of data bytes to copy from input to output pkt.

#### CFE\_SB\_MsgId\_t hk\_copy\_table\_entry\_t::OutputMid

MsgId of the output packet.

#### uint16 hk\_copy\_table\_entry\_t::OutputOffset

ByteOffset into the output pkt where data will be placed.

#### The documentation for this struct was generated from the following file:

* fsw/src/[hk\_tbldefs.h](#AAAAAAAADR)

## HK\_HkPacket\_t Struct Reference

#include <hk\_msg.h>

### Data Fields

* CFE\_MSG\_TelemetryHeader\_t [TlmHeader](#AAAAAAAAIQ)

*cFE Software Bus Telemetry Message Header*

* uint8 [CmdCounter](#AAAAAAAAFJ)

*Count of valid commands received.*

* uint8 [ErrCounter](#AAAAAAAAFK)

*Count of invalid commands received.*

* uint16 [Padding](#AAAAAAAAHX)

*Padding to force 32 bit alignment.*

* uint16 [CombinedPacketsSent](#AAAAAAAAFL)

*Count of combined tlm pkts sent.*

* uint16 [MissingDataCtr](#AAAAAAAAFM)

*Number of times missing data was detected.*

* CFE\_ES\_MemHandle\_t [MemPoolHandle](#AAAAAAAAHU)

*Memory pool handle used to get mempool diags.*

### Detailed Description

##### Purpose:

HK Application housekeeping Packet

### Field Documentation

#### uint8 HK\_HkPacket\_t::CmdCounter

Count of valid commands received.

#### uint16 HK\_HkPacket\_t::CombinedPacketsSent

Count of combined tlm pkts sent.

#### uint8 HK\_HkPacket\_t::ErrCounter

Count of invalid commands received.

#### CFE\_ES\_MemHandle\_t HK\_HkPacket\_t::MemPoolHandle

Memory pool handle used to get mempool diags.

#### uint16 HK\_HkPacket\_t::MissingDataCtr

Number of times missing data was detected.

#### uint16 HK\_HkPacket\_t::Padding

Padding to force 32 bit alignment.

#### CFE\_MSG\_TelemetryHeader\_t HK\_HkPacket\_t::TlmHeader

cFE Software Bus Telemetry Message Header

## HK\_NoArgCmd\_t Struct Reference

No-Operation command packet structure.

#include <hk\_msg.h>

### Data Fields

* CFE\_MSG\_CommandHeader\_t [CmdHeader](#AAAAAAAAIR)

*cFE SB cmd hdr*

### Detailed Description

No-Operation command packet structure.

For command details see [HK\_NOOP\_CC](#AAAAAAAABH), [HK\_RESET\_CC](#AAAAAAAABI)

### Field Documentation

#### CFE\_MSG\_CommandHeader\_t HK\_NoArgCmd\_t::CmdHeader

cFE SB cmd hdr

## hk\_runtime\_tbl\_entry\_t Struct Reference

HK Run-time Table Entry Format.

#include <hk\_tbldefs.h>

### Data Fields

* CFE\_SB\_Buffer\_t \* [OutputPktAddr](#AAAAAAAAIS)

*Addr of output packet.*

* uint8 [InputMidSubscribed](#AAAAAAAAIT)

*Indicates if input MID has been subscribed to.*

* uint8 [DataPresent](#AAAAAAAAIU)

*Indicates if the data associated with the entry is present.*

### Detailed Description

HK Run-time Table Entry Format.

### Field Documentation

#### uint8 hk\_runtime\_tbl\_entry\_t::DataPresent

Indicates if the data associated with the entry is present.

#### uint8 hk\_runtime\_tbl\_entry\_t::InputMidSubscribed

Indicates if input MID has been subscribed to.

#### CFE\_SB\_Buffer\_t\* hk\_runtime\_tbl\_entry\_t::OutputPktAddr

Addr of output packet.

## HK\_Send\_Out\_Msg\_t Struct Reference

Send Combined Output Message Command.

#include <hk\_msg.h>

### Data Fields

* CFE\_MSG\_CommandHeader\_t [Hdr](#AAAAAAAAIV)

*cFE Software Bus Command Message Header #CFE\_MSG\_CommandHeader\_t*

* CFE\_SB\_MsgId\_t [OutMsgToSend](#AAAAAAAAIW)

*MsgId #CFE\_SB\_MsgId\_t of combined tlm pkt to send.*

### Detailed Description

Send Combined Output Message Command.

This structure contains the format of the command used to inform HK to send the specified combined output message

### Field Documentation

#### CFE\_MSG\_CommandHeader\_t HK\_Send\_Out\_Msg\_t::Hdr

cFE Software Bus Command Message Header #CFE\_MSG\_CommandHeader\_t

#### CFE\_SB\_MsgId\_t HK\_Send\_Out\_Msg\_t::OutMsgToSend

MsgId #CFE\_SB\_MsgId\_t of combined tlm pkt to send.

# File Documentation

## docs/dox\_src/users\_guide/cfs\_hk.dox File Reference

## fsw/mission\_inc/hk\_perfids.h File Reference

### Macros

* HK Performance ID#define [HK\_APPMAIN\_PERF\_ID](#AAAAAAAAAC)  25

### Macro Definition Documentation

#### #define HK\_APPMAIN\_PERF\_ID  25

## fsw/platform\_inc/hk\_msgids.h File Reference

### Macros

* #define [HK\_CMD\_MID](#AAAAAAAAAE)  0x189A

*HK Ground Commands Message ID.*

* #define [HK\_SEND\_HK\_MID](#AAAAAAAAAF)  0x189B

*HK Send Housekeeping Data Cmd Message ID.*

* #define [HK\_SEND\_COMBINED\_PKT\_MID](#AAAAAAAAAG)  0x189C

*HK Send Combined Pkt Cmd Message ID.*

* #define [HK\_HK\_TLM\_MID](#AAAAAAAAAH)  0x089B

*HK Housekeeping Telemetry Message ID.*

* #define [HK\_COMBINED\_PKT1\_MID](#AAAAAAAAAI)  0x089C

*HK Combined Packet 1 Message ID.*

* #define [HK\_COMBINED\_PKT2\_MID](#AAAAAAAAAJ)  0x089D

*HK Combined Packet 2 Message ID.*

* #define [HK\_COMBINED\_PKT3\_MID](#AAAAAAAAAK)  0x089E

*HK Combined Packet 3 Message ID.*

* #define [HK\_COMBINED\_PKT4\_MID](#AAAAAAAAAL)  0x089F

*HK Combined Packet 4 Message ID.*

### Macro Definition Documentation

#### #define HK\_CMD\_MID  0x189A

HK Ground Commands Message ID.

#### #define HK\_COMBINED\_PKT1\_MID  0x089C

HK Combined Packet 1 Message ID.

#### #define HK\_COMBINED\_PKT2\_MID  0x089D

HK Combined Packet 2 Message ID.

#### #define HK\_COMBINED\_PKT3\_MID  0x089E

HK Combined Packet 3 Message ID.

#### #define HK\_COMBINED\_PKT4\_MID  0x089F

HK Combined Packet 4 Message ID.

#### #define HK\_HK\_TLM\_MID  0x089B

HK Housekeeping Telemetry Message ID.

#### #define HK\_SEND\_COMBINED\_PKT\_MID  0x189C

HK Send Combined Pkt Cmd Message ID.

#### #define HK\_SEND\_HK\_MID  0x189B

HK Send Housekeeping Data Cmd Message ID.

## fsw/platform\_inc/hk\_platform\_cfg.h File Reference

### Macros

* #define [HK\_PIPE\_DEPTH](#AAAAAAAAAN)  40
* #define [HK\_DISCARD\_INCOMPLETE\_COMBO](#AAAAAAAAAO)  0
* #define [HK\_COPY\_TABLE\_ENTRIES](#AAAAAAAAAP)  128
* #define [HK\_NUM\_BYTES\_IN\_MEM\_POOL](#AAAAAAAAAQ)  (6 \* 1024)
* #define [HK\_COPY\_TABLE\_NAME](#AAAAAAAAAR)  "CopyTable"
* #define [HK\_RUNTIME\_TABLE\_NAME](#AAAAAAAAAS)  "RuntimeTable"
* #define [HK\_COPY\_TABLE\_FILENAME](#AAAAAAAAAT)  "/cf/hk\_cpy\_tbl.tbl"
* #define [HK\_MISSION\_REV](#AAAAAAAAAU)  0

### Macro Definition Documentation

#### #define HK\_COPY\_TABLE\_ENTRIES  128

##### Purpose:

Maximum Number of HK Copy Table Entries

##### Description:

Dictates the number of elements in the hk copy table.

##### Limits

The maximum size of this paramater is 8192

#### #define HK\_COPY\_TABLE\_FILENAME  "/cf/hk\_cpy\_tbl.tbl"

##### Purpose:

HK Copy Table Filename

##### Description:

The value of this constant defines the filename of the HK Copy Table

##### Limits

The length of each string, including the NULL terminator cannot exceed the #OS\_MAX\_PATH\_LEN value.

#### #define HK\_COPY\_TABLE\_NAME  "CopyTable"

##### Purpose:

Name of the HK Copy Table

##### Description:

This parameter defines the name of the HK Copy Table.

##### Limits

The Housekeeping app does not place a limit on this parameter

#### #define HK\_DISCARD\_INCOMPLETE\_COMBO  0

##### Purpose:

Discard Incomplete Combo Packets

##### Description:

Dictates whether combo packets that have not had all data contents updated since last requested will be discarded (YES = 1) or sent anyway (NO = 0).

##### Limits

This parameter can be set to 0 or 1 only.

#### #define HK\_MISSION\_REV  0

##### Purpose:

Mission specific version number for HK application

##### Description:

An application version number consists of four parts: major version number, minor version number, revision number and mission specific revision number. The mission specific revision number is defined here and the other parts are defined in "hk\_version.h".

##### Limits:

Must be defined as a numeric value that is greater than or equal to zero.

#### #define HK\_NUM\_BYTES\_IN\_MEM\_POOL  (6 \* 1024)

##### Purpose:

Number of bytes in the HK Memory Pool

##### Description:

The HK memory pool contains the memory needed for the output packets. The output packets are dynamically allocated from this pool when the HK copy table is initially processed or loaded with new data.

##### Limits

The Housekeeping app does not place a limit on this parameter, but there is an overhead cost in the memory pool. The value must be larger than what is needed.

#### #define HK\_PIPE\_DEPTH  40

##### Purpose:

Application Pipe Depth

##### Description:

Dictates the pipe depth of the hk command pipe.

##### Limits

The minimum size of this paramater is 1 The maximum size dictated by cFE platform configuration parameter CFE\_SB\_MAX\_PIPE\_DEPTH

#### #define HK\_RUNTIME\_TABLE\_NAME  "RuntimeTable"

##### Purpose:

Name of the HK Run-time Table

##### Description:

This parameter defines the name of the HK Run-time Table.

##### Limits

The Housekeeping app does not place a limit on this parameter

## fsw/src/hk\_app.c File Reference

#include "hk\_app.h"

#include "hk\_events.h"

#include "hk\_msgids.h"

#include "hk\_perfids.h"

#include "hk\_verify.h"

#include "hk\_version.h"

#include "hk\_platform\_cfg.h"

#include "hk\_utils.h"

#include <string.h>

### Functions

* void [HK\_AppMain](#AAAAAAAAAW) (void)

*CFS Housekeeping (HK) application entry point.*

* int32 [HK\_AppInit](#AAAAAAAAAX) (void)

*Initialize the housekeeping application.*

* int32 [HK\_TableInit](#AAAAAAAAAY) (void)

*Initialize the Copy Table and the Runtime Table.*

* void [HK\_AppPipe](#AAAAAAAAAZ) (const CFE\_SB\_Buffer\_t \*BufPtr)

*Process a command pipe message.*

* void [HK\_SendCombinedHKCmd](#AAAAAAAABA) (const CFE\_SB\_Buffer\_t \*BufPtr)

*Send Combined Housekeeping Message.*

* void [HK\_HousekeepingCmd](#AAAAAAAABB) (const CFE\_MSG\_CommandHeader\_t \*Msg)

*Process housekeeping request.*

* void [HK\_NoopCmd](#AAAAAAAABC) (const CFE\_SB\_Buffer\_t \*BufPtr)

*Process noop command.*

* void [HK\_ResetCtrsCmd](#AAAAAAAABD) (const CFE\_SB\_Buffer\_t \*BufPtr)

*Process reset counters command.*

* void [HK\_ResetHkData](#AAAAAAAABE) (void)

*Reset housekeeping data.*

### Variables

* [HK\_AppData\_t](#AAAAAAAABF) [HK\_AppData](#AAAAAAAABG)

*HK Housekeeping Packet.*

### Function Documentation

#### int32 HK\_AppInit (void )

Initialize the housekeeping application.

##### Description

Housekeeping application initialization routine. This function performs all the required startup steps to get the application registered with the cFE services so it can begin to receive command messages.

##### Assumptions, External Events, and Notes:

None

##### Returns:

#CFE\_SUCCESS Return codes from #CFE\_EVS\_Register Return codes from #CFE\_SB\_CreatePipe Return codes from #CFE\_SB\_Subscribe

#### void HK\_AppMain (void )

CFS Housekeeping (HK) application entry point.

##### Description

Housekeeping application entry point and main process loop.

##### Assumptions, External Events, and Notes:

None

#### void HK\_AppPipe (const CFE\_SB\_Buffer\_t \* *BufPtr*)

Process a command pipe message.

##### Description

Processes a single software bus command pipe message. Checks the message and command IDs and calls the appropriate routine to handle the command.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A #CFE\_SB\_Buffer\_t\* pointer that references the software bus message |

##### See also:

#CFE\_SB\_RcvMsg

#### void HK\_HousekeepingCmd (const CFE\_MSG\_CommandHeader\_t \* *Msg*)

Process housekeeping request.

##### Description

Processes an on-board housekeeping request message.

##### Assumptions, External Events, and Notes:

This command does not affect the command execution counter, but this command will increment the cmd error counter if an invalid cmd length is detected.

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A #CFE\_MSG\_CommandHeader\_t\* pointer that references the software bus message |

#### void HK\_NoopCmd (const CFE\_SB\_Buffer\_t \* *BufPtr*)

Process noop command.

##### Description

Processes a noop ground command.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A #CFE\_SB\_Buffer\_t\* pointer that references the software bus message |

##### See also:

[HK\_NOOP\_CC](#AAAAAAAABH)

#### void HK\_ResetCtrsCmd (const CFE\_SB\_Buffer\_t \* *BufPtr*)

Process reset counters command.

##### Description

Processes a reset counters ground command which will reset the memory manager commmand error and command execution counters to zero.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A #CFE\_SB\_Buffer\_t\* pointer that references the software bus message |

##### See also:

[HK\_RESET\_CC](#AAAAAAAABI)

#### void HK\_ResetHkData (void )

Reset housekeeping data.

##### Description

Function called in response to a Reset Counters Command. This function will reset the HK housekeeping data.

##### Assumptions, External Events, and Notes:

None

##### See also:

[HK\_RESET\_CC](#AAAAAAAABI)

#### void HK\_SendCombinedHKCmd (const CFE\_SB\_Buffer\_t \* *BufPtr*)

Send Combined Housekeeping Message.

##### Description

Processes the command to send a combined housekeeping message

##### Assumptions, External Events, and Notes:

This command does not affect the command execution counter, but this command will increment the cmd error counter if an invalid cmd length is detected.

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A #CFE\_SB\_Buffer\_t\* pointer that references the software bus message |

#### int32 HK\_TableInit (void )

Initialize the Copy Table and the Runtime Table.

##### Description

Registers the Copy table and Runtime table with cFE Table Services. Also processes the copy table.

##### Assumptions, External Events, and Notes:

None

##### See also:

[HK\_AppInit](#AAAAAAAABJ)

### Variable Documentation

#### [HK\_AppData\_t](#AAAAAAAABF) HK\_AppData

HK Housekeeping Packet.

## fsw/src/hk\_app.h File Reference

#include "cfe.h"

#include "hk\_msgdefs.h"

#include "hk\_msg.h"

#include "hk\_utils.h"

#include "hk\_platform\_cfg.h"

### Data Structures

* struct [HK\_AppData\_t](#AAAAAAAABF)

### *HK global data structure.* Macros

* #define [HK\_PIPE\_NAME](#AAAAAAAABL)  "HK\_CMD\_PIPE"

*Application Pipe Name.*

* #define [HK\_SUCCESS](#AAAAAAAABM)  (0)

*HK return code for success.*

* #define [HK\_ERROR](#AAAAAAAABN)  (-1)

*HK return code for general error.*

* #define [HK\_BAD\_MSG\_LENGTH\_RC](#AAAAAAAABO)  (-2)

*HK return code for unexpected cmd length.*

### Functions

* void [HK\_AppMain](#AAAAAAAABP) (void)

*CFS Housekeeping (HK) application entry point.*

* int32 [HK\_AppInit](#AAAAAAAABJ) (void)

*Initialize the housekeeping application.*

* int32 [HK\_TableInit](#AAAAAAAABQ) (void)

*Initialize the Copy Table and the Runtime Table.*

* void [HK\_AppPipe](#AAAAAAAABR) (const CFE\_SB\_Buffer\_t \*BufPtr)

*Process a command pipe message.*

* void [HK\_SendCombinedHKCmd](#AAAAAAAABS) (const CFE\_SB\_Buffer\_t \*BufPtr)

*Send Combined Housekeeping Message.*

* void [HK\_HousekeepingCmd](#AAAAAAAABT) (const CFE\_MSG\_CommandHeader\_t \*Msg)

*Process housekeeping request.*

* void [HK\_NoopCmd](#AAAAAAAABU) (const CFE\_SB\_Buffer\_t \*BufPtr)

*Process noop command.*

* void [HK\_ResetCtrsCmd](#AAAAAAAABV) (const CFE\_SB\_Buffer\_t \*BufPtr)

*Process reset counters command.*

* void [HK\_ResetHkData](#AAAAAAAABW) (void)

*Reset housekeeping data.*

### Variables

* [HK\_AppData\_t](#AAAAAAAABF) [HK\_AppData](#AAAAAAAABX)

*HK Housekeeping Packet.*

### Macro Definition Documentation

#### #define HK\_BAD\_MSG\_LENGTH\_RC  (-2)

HK return code for unexpected cmd length.

#### #define HK\_ERROR  (-1)

HK return code for general error.

#### #define HK\_PIPE\_NAME  "HK\_CMD\_PIPE"

Application Pipe Name.

#### #define HK\_SUCCESS  (0)

HK return code for success.

### Function Documentation

#### int32 HK\_AppInit (void )

Initialize the housekeeping application.

##### Description

Housekeeping application initialization routine. This function performs all the required startup steps to get the application registered with the cFE services so it can begin to receive command messages.

##### Assumptions, External Events, and Notes:

None

##### Returns:

#CFE\_SUCCESS Return codes from #CFE\_EVS\_Register Return codes from #CFE\_SB\_CreatePipe Return codes from #CFE\_SB\_Subscribe

#### void HK\_AppMain (void )

CFS Housekeeping (HK) application entry point.

##### Description

Housekeeping application entry point and main process loop.

##### Assumptions, External Events, and Notes:

None

#### void HK\_AppPipe (const CFE\_SB\_Buffer\_t \* *BufPtr*)

Process a command pipe message.

##### Description

Processes a single software bus command pipe message. Checks the message and command IDs and calls the appropriate routine to handle the command.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A #CFE\_SB\_Buffer\_t\* pointer that references the software bus message |

##### See also:

#CFE\_SB\_RcvMsg

#### void HK\_HousekeepingCmd (const CFE\_MSG\_CommandHeader\_t \* *Msg*)

Process housekeeping request.

##### Description

Processes an on-board housekeeping request message.

##### Assumptions, External Events, and Notes:

This command does not affect the command execution counter, but this command will increment the cmd error counter if an invalid cmd length is detected.

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A #CFE\_MSG\_CommandHeader\_t\* pointer that references the software bus message |

#### void HK\_NoopCmd (const CFE\_SB\_Buffer\_t \* *BufPtr*)

Process noop command.

##### Description

Processes a noop ground command.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A #CFE\_SB\_Buffer\_t\* pointer that references the software bus message |

##### See also:

[HK\_NOOP\_CC](#AAAAAAAABH)

#### void HK\_ResetCtrsCmd (const CFE\_SB\_Buffer\_t \* *BufPtr*)

Process reset counters command.

##### Description

Processes a reset counters ground command which will reset the memory manager commmand error and command execution counters to zero.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A #CFE\_SB\_Buffer\_t\* pointer that references the software bus message |

##### See also:

[HK\_RESET\_CC](#AAAAAAAABI)

#### void HK\_ResetHkData (void )

Reset housekeeping data.

##### Description

Function called in response to a Reset Counters Command. This function will reset the HK housekeeping data.

##### Assumptions, External Events, and Notes:

None

##### See also:

[HK\_RESET\_CC](#AAAAAAAABI)

#### void HK\_SendCombinedHKCmd (const CFE\_SB\_Buffer\_t \* *BufPtr*)

Send Combined Housekeeping Message.

##### Description

Processes the command to send a combined housekeeping message

##### Assumptions, External Events, and Notes:

This command does not affect the command execution counter, but this command will increment the cmd error counter if an invalid cmd length is detected.

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A #CFE\_SB\_Buffer\_t\* pointer that references the software bus message |

#### int32 HK\_TableInit (void )

Initialize the Copy Table and the Runtime Table.

##### Description

Registers the Copy table and Runtime table with cFE Table Services. Also processes the copy table.

##### Assumptions, External Events, and Notes:

None

##### See also:

[HK\_AppInit](#AAAAAAAABJ)

### Variable Documentation

#### HK\_AppData\_t HK\_AppData

HK Housekeeping Packet.

## fsw/src/hk\_events.h File Reference

### Macros

* #define [HK\_INIT\_EID](#AAAAAAAABZ)  1

*'HK Initialized. Version %d.%d.%d.%d'*

* #define [HK\_CC\_ERR\_EID](#AAAAAAAACA)  2

*'Cmd Msg with Invalid command code Rcvd – ID = 0x%08X, CC = %d'*

* #define [HK\_CMD\_LEN\_ERR\_EID](#AAAAAAAACB)  3

*'Cmd Msg with Bad length Rcvd: ID = 0x%X, CC = %d, Exp Len = %d, Len = %d'*

* #define [HK\_NOOP\_CMD\_EID](#AAAAAAAACC)  4

*'HK No-op command, Version %d.%d.%d.%d'*

* #define [HK\_RESET\_CNTRS\_CMD\_EID](#AAAAAAAACD)  5

*'HK Reset Counters command received'*

* #define [HK\_ACCESSING\_PAST\_PACKET\_END\_EID](#AAAAAAAACE)  6

*'HK table definition exceeds packet length. MID:0x%08X, Length:%d, Count:%d'*

* #define [HK\_MEM\_POOL\_MALLOC\_FAILED\_EID](#AAAAAAAACF)  7

*'HK Processing New Table: ES\_GetPoolBuf for size %d returned 0x%04X'*

* #define [HK\_CANT\_SUBSCRIBE\_TO\_SB\_PKT\_EID](#AAAAAAAACG)  8

*'HK Processing New Table:SB\_Subscribe for Mid 0x%08X returned 0x%04X'*

* #define [HK\_MEM\_POOL\_FREE\_FAILED\_EID](#AAAAAAAACH)  9

*'HK TearDown Old Table: ES\_PutPoolBuf for pkt:0x%08X returned 0x%04X'*

* #define [HK\_UNEXPECTED\_GETSTAT\_RET\_EID](#AAAAAAAACI)  10

*'Unexpected CFE\_TBL\_GetStatus return (0x%08X) for Copy Table'*

* #define [HK\_UNKNOWN\_COMBINED\_PACKET\_EID](#AAAAAAAACJ)  11

*'Combined HK Packet 0x%08X is not found in current HK Copy Table'*

* #define [HK\_OUTPKT\_MISSING\_DATA\_EID](#AAAAAAAACK)  12

*'Combined Packet 0x%08X missing data from Input Pkt 0x%08X'*

* #define [HK\_CR\_PIPE\_ERR\_EID](#AAAAAAAACL)  14

*'Error Creating SB Pipe,RC=0x%08X'*

* #define [HK\_SUB\_CMB\_ERR\_EID](#AAAAAAAACM)  15

*'Error Subscribing to HK Snd Cmb Pkt, MID=0x%08X, RC=0x%08X'*

* #define [HK\_SUB\_REQ\_ERR\_EID](#AAAAAAAACN)  16

*'Error Subscribing to HK Request, MID=0x%08X, RC=0x%08X'*

* #define [HK\_SUB\_CMD\_ERR\_EID](#AAAAAAAACO)  17

*'Error Subscribing to HK Gnd Cmds, MID=0x%08X, RC=0x%08X'*

* #define [HK\_CR\_POOL\_ERR\_EID](#AAAAAAAACP)  18

*'Error Creating Memory Pool,RC=0x%08X'*

* #define [HK\_CPTBL\_REG\_ERR\_EID](#AAAAAAAACQ)  19

*'Error Registering Copy Table,RC=0x%08X'*

* #define [HK\_RTTBL\_REG\_ERR\_EID](#AAAAAAAACR)  20

*'Error Registering Runtime Table,RC=0x%08X'*

* #define [HK\_CPTBL\_LD\_ERR\_EID](#AAAAAAAACS)  21

*'Error Loading Copy Table,RC=0x%08X'*

* #define [HK\_CPTBL\_MNG\_ERR\_EID](#AAAAAAAACT)  22

*'Error from TBL Manage call for Copy Table,RC=0x%08X'*

* #define [HK\_RTTBL\_MNG\_ERR\_EID](#AAAAAAAACU)  23

*'Error from TBL Manage call for Runtime Table,RC=0x%08X'*

* #define [HK\_CPTBL\_GADR\_ERR\_EID](#AAAAAAAACV)  24

*'Error Getting Adr for Cpy Tbl,RC=0x%08X'*

* #define [HK\_RTTBL\_GADR\_ERR\_EID](#AAAAAAAACW)  25

*'Error Getting Adr for Runtime Table,RC=0x%08X'*

* #define [HK\_RCV\_MSG\_ERR\_EID](#AAAAAAAACX)  26

*'HK\_APP Exiting due to CFE\_SB\_RcvMsg error 0x%08X'*

* #define [HK\_UNEXPECTED\_GETSTAT2\_RET\_EID](#AAAAAAAACY)  27

*'Unexpected CFE\_TBL\_GetStatus return (0x%08X) for Runtime Table'*

* #define [HK\_MSG\_LEN\_ERR\_EID](#AAAAAAAACZ)  28

*'Msg with Bad length Rcvd: ID = 0x%08X, Exp Len = %d, Len = %d'*

* #define [HK\_UNEXPECTED\_GETADDR\_RET\_EID](#AAAAAAAADA)  29

*'Unexpected CFE\_TBL\_GetAddress return (0x%08X) for Copy Table'*

* #define [HK\_UNEXPECTED\_TBLUPD\_RET\_EID](#AAAAAAAADB)  30

*'Unexpected CFE\_TBL\_Update return (0x%08X) for Copy Table'*

* #define [HK\_UNEXPECTED\_RELADDR\_RET\_EID](#AAAAAAAADC)  31

*'Unexpected CFE\_TBL\_ReleaseAddress return (0x%08X) for Copy Table'*

* #define [HK\_UNEXPECTED\_TBLVLD\_RET\_EID](#AAAAAAAADD)  32

*'Unexpected CFE\_TBL\_Validate return (0x%08X) for Copy Table'*

* #define [HK\_UNEXPECTED\_DUMPTOBUFFER\_RET\_EID](#AAAAAAAADE)  33

*'Unexpected CFE\_TBL\_DumpToBuffer return (0x%08X) for Runtime Table'*

* #define [HK\_NULL\_POINTER\_NEWCPY\_ERR\_EID](#AAAAAAAADF)  34

*'Null pointer detected in new copy tbl processing: CpyTbl = 0x%08X, RtTbl = 0x%08X'*

* #define [HK\_NULL\_POINTER\_TEARCPY\_ERR\_EID](#AAAAAAAADG)  35

*'Null pointer detected in copy tbl tear down: CpyTbl = 0x%08X, RtTbl = 0x%08X'*

* #define [HK\_NEWCPYTBL\_HK\_FAILED\_EID](#AAAAAAAADH)  36

*'Process New Copy Table Failed, status = 0x%08X'*

* #define [HK\_NEWCPYTBL\_INIT\_FAILED\_EID](#AAAAAAAADI)  37

*'Process New Copy Table Failed, status = 0x%08X'*

### Macro Definition Documentation

#### #define HK\_ACCESSING\_PAST\_PACKET\_END\_EID  6

'HK table definition exceeds packet length. MID:0x%08X, Length:%d, Count:%d'

##### Event Message:

'HK table definition exceeds packet length. MID:0x%08X, Length:%d, Count:%d'

##### Type: ERROR

##### Cause:

This event message is issued when the received input packet is not large enough to accommodate every entry in the copy table. The count indicates the total number of copy table entries that reference past the end of the input packet.

#### #define HK\_CANT\_SUBSCRIBE\_TO\_SB\_PKT\_EID  8

'HK Processing New Table:SB\_Subscribe for Mid 0x%08X returned 0x%04X'

##### Event Message:

'HK Processing New Table:SB\_Subscribe for Mid 0x%08X returned 0x%04X'

##### Type: ERROR

##### Cause:

This event message is issued when the HK application receives an error while subscribing to the input messages.

#### #define HK\_CC\_ERR\_EID  2

'Cmd Msg with Invalid command code Rcvd – ID = 0x%08X, CC = %d'

##### Event Message:

'Cmd Msg with Invalid command code Rcvd – ID = 0x%08X, CC = %d'

##### Type: ERROR

##### Cause:

This event message is issued when the HK application receives an unexpected command code

#### #define HK\_CMD\_LEN\_ERR\_EID  3

'Cmd Msg with Bad length Rcvd: ID = 0x%X, CC = %d, Exp Len = %d, Len = %d'

##### Event Message:

'Cmd Msg with Bad length Rcvd: ID = 0x%X, CC = %d, Exp Len = %d, Len = %d'

##### Type: ERROR

##### Cause:

This event message is issued when the actual command length does not equal the expected command length

#### #define HK\_CPTBL\_GADR\_ERR\_EID  24

'Error Getting Adr for Cpy Tbl,RC=0x%08X'

##### Event Message:

'Error Getting Adr for Cpy Tbl,RC=0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_TBL\_GetAddress for the copy table returns a value other than CFE\_TBL\_INFO\_UPDATED

#### #define HK\_CPTBL\_LD\_ERR\_EID  21

'Error Loading Copy Table,RC=0x%08X'

##### Event Message:

'Error Loading Copy Table,RC=0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_TBL\_Load for the copy table returns a value other than CFE\_SUCCESS

#### #define HK\_CPTBL\_MNG\_ERR\_EID  22

'Error from TBL Manage call for Copy Table,RC=0x%08X'

##### Event Message:

'Error from TBL Manage call for Copy Table,RC=0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_TBL\_Manage for the copy table returns a value other than CFE\_SUCCESS

#### #define HK\_CPTBL\_REG\_ERR\_EID  19

'Error Registering Copy Table,RC=0x%08X'

##### Event Message:

'Error Registering Copy Table,RC=0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_TBL\_Register for the copy table returns a value other than CFE\_SUCCESS

#### #define HK\_CR\_PIPE\_ERR\_EID  14

'Error Creating SB Pipe,RC=0x%08X'

##### Event Message:

'Error Creating SB Pipe,RC=0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_SB\_CreatePipe during HK initialization returns a value other than CFE\_SUCCESS

#### #define HK\_CR\_POOL\_ERR\_EID  18

'Error Creating Memory Pool,RC=0x%08X'

##### Event Message:

'Error Creating Memory Pool,RC=0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_ES\_PoolCreate during HK initialization returns a value other than CFE\_SUCCESS

#### #define HK\_INIT\_EID  1

'HK Initialized. Version %d.%d.%d.%d'

##### Event Message:

'HK Initialized. Version %d.%d.%d.%d'

##### Type: INFORMATION

##### Cause:

This event message is issued when the Housekeeping App completes its initialization.

The Version fields contain the [HK\_MAJOR\_VERSION](#AAAAAAAADJ), [HK\_MINOR\_VERSION](#AAAAAAAADK), [HK\_REVISION](#AAAAAAAADL), and [HK\_MISSION\_REV](#AAAAAAAAAU) version identifiers.

#### #define HK\_MEM\_POOL\_FREE\_FAILED\_EID  9

'HK TearDown Old Table: ES\_PutPoolBuf for pkt:0x%08X returned 0x%04X'

##### Event Message:

'HK TearDown Old Table: ES\_PutPoolBuf for pkt:0x%08X returned 0x%04X'

##### Type: ERROR

##### Cause:

This event message is issued when the HK application receives an error while attempting to free the memory for an output message

#### #define HK\_MEM\_POOL\_MALLOC\_FAILED\_EID  7

'HK Processing New Table: ES\_GetPoolBuf for size %d returned 0x%04X'

##### Event Message:

'HK Processing New Table: ES\_GetPoolBuf for size %d returned 0x%04X'

##### Type: ERROR

##### Cause:

This event message is issued when the HK application receives an error when requesting memory from the ES memory pool

#### #define HK\_MSG\_LEN\_ERR\_EID  28

'Msg with Bad length Rcvd: ID = 0x%08X, Exp Len = %d, Len = %d'

##### Event Message:

'Msg with Bad length Rcvd: ID = 0x%08X, Exp Len = %d, Len = %d'

##### Type: ERROR

##### Cause:

This event is issued when an internal message which has a length that is inconsistant with the expected length for its message id.

#### #define HK\_NEWCPYTBL\_HK\_FAILED\_EID  36

'Process New Copy Table Failed, status = 0x%08X'

##### Event Message:

'Process New Copy Table Failed, status = 0x%08X'

##### Type: CRITICAL

##### Cause:

This event is issued when an error occurs while processing a new copy table during housekeeping processing. The probable reason for a failure is that a NULL pointer was used as an argument in the called routine (a status value of 0xFFFFFFFF indicates a NULL pointer was detected). In this case (as opposed to during initialization), the event is critical.

#### #define HK\_NEWCPYTBL\_INIT\_FAILED\_EID  37

'Process New Copy Table Failed, status = 0x%08X'

##### Event Message:

'Process New Copy Table Failed, status = 0x%08X'

##### Type: ERROR

##### Cause:

This event is issued when an error occurs while processing a new copy table during app initialization. The probable reason for a failure is that a NULL pointer was used as an argument in the called routine (a status value of 0xFFFFFFFF indicates a NULL pointer was detected). In this case (as opposed to during housekeeping processing), the event is an error, but not critical.

#### #define HK\_NOOP\_CMD\_EID  4

'HK No-op command, Version %d.%d.%d.%d'

##### Event Message:

'HK No-op command, Version %d.%d.%d.%d'

##### Type: INFORMATION

##### Cause:

This event message is issued when the HK application successfully receives a [No-op command.](#AAAAAAAABH)  The command is used primarily as an indicator that the HK application can receive commands and generate telemetry.

The Version fields contain the [HK\_MAJOR\_VERSION](#AAAAAAAADJ), [HK\_MINOR\_VERSION](#AAAAAAAADK), [HK\_REVISION](#AAAAAAAADL), and [HK\_MISSION\_REV](#AAAAAAAAAU) version identifiers.

#### #define HK\_NULL\_POINTER\_NEWCPY\_ERR\_EID  34

'Null pointer detected in new copy tbl processing: CpyTbl = 0x%08X, RtTbl = 0x%08X'

##### Event Message:

'Null pointer detected in new copy tbl processing: CpyTbl = 0x%08X, RtTbl = 0x%08X'

##### Type: ERROR

##### Cause:

This event is issued when at least one of the input arguments for processing a new copy table is NULL.

#### #define HK\_NULL\_POINTER\_TEARCPY\_ERR\_EID  35

'Null pointer detected in copy tbl tear down: CpyTbl = 0x%08X, RtTbl = 0x%08X'

##### Event Message:

'Null pointer detected in copy tbl tear down: CpyTbl = 0x%08X, RtTbl = 0x%08X'

##### Type: ERROR

##### Cause:

This event is issued when at least one of the input arguments for tearing down an old copy table is NULL.

#### #define HK\_OUTPKT\_MISSING\_DATA\_EID  12

'Combined Packet 0x%08X missing data from Input Pkt 0x%08X'

##### Event Message:

'Combined Packet 0x%08X missing data from Input Pkt 0x%08X'

##### Type: DEBUG

##### Cause:

This event message is issued when at least one section of data is missing in an output message.

#### #define HK\_RCV\_MSG\_ERR\_EID  26

'HK\_APP Exiting due to CFE\_SB\_RcvMsg error 0x%08X'

##### Event Message:

'HK\_APP Exiting due to CFE\_SB\_RcvMsg error 0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_SB\_RcvMsg returns a value other than CFE\_SUCCESS in the main loop

#### #define HK\_RESET\_CNTRS\_CMD\_EID  5

'HK Reset Counters command received'

##### Event Message:

'HK Reset Counters command received'

##### Type: DEBUG

##### Cause:

This event message is issued when the HK application receives a Reset Counters command.

#### #define HK\_RTTBL\_GADR\_ERR\_EID  25

'Error Getting Adr for Runtime Table,RC=0x%08X'

##### Event Message:

'Error Getting Adr for Runtime Table,RC=0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_TBL\_GetAddress for the runtime table returns a value other than CFE\_SUCCESS

#### #define HK\_RTTBL\_MNG\_ERR\_EID  23

'Error from TBL Manage call for Runtime Table,RC=0x%08X'

##### Event Message:

'Error from TBL Manage call for Runtime Table,RC=0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_TBL\_Manage for the runtime table returns a value other than CFE\_SUCCESS

#### #define HK\_RTTBL\_REG\_ERR\_EID  20

'Error Registering Runtime Table,RC=0x%08X'

##### Event Message:

'Error Registering Runtime Table,RC=0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_TBL\_Register for the runtime table returns a value other than CFE\_SUCCESS

#### #define HK\_SUB\_CMB\_ERR\_EID  15

'Error Subscribing to HK Snd Cmb Pkt, MID=0x%08X, RC=0x%08X'

##### Event Message:

'Error Subscribing to HK Snd Cmb Pkt, MID=0x%08X, RC=0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_SB\_Subscribe for the [HK\_SEND\_COMBINED\_PKT\_MID](#AAAAAAAAAG), during HK initialization returns a value other than CFE\_SUCCESS

#### #define HK\_SUB\_CMD\_ERR\_EID  17

'Error Subscribing to HK Gnd Cmds, MID=0x%08X, RC=0x%08X'

##### Event Message:

'Error Subscribing to HK Gnd Cmds, MID=0x%08X, RC=0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_SB\_Subscribe for the [HK\_CMD\_MID](#AAAAAAAAAE), during HK initialization returns a value other than CFE\_SUCCESS

#### #define HK\_SUB\_REQ\_ERR\_EID  16

'Error Subscribing to HK Request, MID=0x%08X, RC=0x%08X'

##### Event Message:

'Error Subscribing to HK Request, MID=0x%08X, RC=0x%08X'

##### Type: ERROR

##### Cause:

This event message is issued when the call to CFE\_SB\_Subscribe for the [HK\_SEND\_HK\_MID](#AAAAAAAAAF), during HK initialization returns a value other than CFE\_SUCCESS

#### #define HK\_UNEXPECTED\_DUMPTOBUFFER\_RET\_EID  33

'Unexpected CFE\_TBL\_DumpToBuffer return (0x%08X) for Runtime Table'

##### Event Message:

'Unexpected CFE\_TBL\_DumpToBuffer return (0x%08X) for Runtime Table'

##### Type: CRITICAL

##### Cause:

This event message is issued when the HK application receives an unexpected return value when calling the CFE\_TBL\_DumpToBuffer API for the runtime table during the check of the table status. This is a critical error (due to unexpected behavior of a cFE service) and will cause the HK app to exit.

#### #define HK\_UNEXPECTED\_GETADDR\_RET\_EID  29

'Unexpected CFE\_TBL\_GetAddress return (0x%08X) for Copy Table'

##### Event Message:

'Unexpected CFE\_TBL\_GetAddress return (0x%08X) for Copy Table'

##### Type: CRITICAL

##### Cause:

This event message is issued when the HK application receives an unexpected return value when calling the CFE\_TBL\_GetAddress API for the copy table during the check of the table status. This is a critical error (due to unexpected behavior of a cFE service) and will cause the HK app to exit.

#### #define HK\_UNEXPECTED\_GETSTAT2\_RET\_EID  27

'Unexpected CFE\_TBL\_GetStatus return (0x%08X) for Runtime Table'

##### Event Message:

'Unexpected CFE\_TBL\_GetStatus return (0x%08X) for Runtime Table'

##### Type: ERROR

##### Cause:

This event message is issued when the HK application receives an unexpected return value when calling the CFE\_TBL\_GetStatus API for the dump table

#### #define HK\_UNEXPECTED\_GETSTAT\_RET\_EID  10

'Unexpected CFE\_TBL\_GetStatus return (0x%08X) for Copy Table'

##### Event Message:

'Unexpected CFE\_TBL\_GetStatus return (0x%08X) for Copy Table'

##### Type: CRITICAL

##### Cause:

This event message is issued when the HK application receives an unexpected return value when calling the CFE\_TBL\_GetStatus API for the copy table

#### #define HK\_UNEXPECTED\_RELADDR\_RET\_EID  31

'Unexpected CFE\_TBL\_ReleaseAddress return (0x%08X) for Copy Table'

##### Event Message:

'Unexpected CFE\_TBL\_ReleaseAddress return (0x%08X) for Copy Table'

##### Type: CRITICAL

##### Cause:

This event message is issued when the HK application receives an unexpected return value when calling the CFE\_TBL\_ReleaseAddress API for the copy table during the check of the table status. This is a critical error (due to unexpected behavior of a cFE service) and will cause the HK app to exit.

#### #define HK\_UNEXPECTED\_TBLUPD\_RET\_EID  30

'Unexpected CFE\_TBL\_Update return (0x%08X) for Copy Table'

##### Event Message:

'Unexpected CFE\_TBL\_Update return (0x%08X) for Copy Table'

##### Type: CRITICAL

##### Cause:

This event message is issued when the HK application receives an unexpected return value when calling the CFE\_TBL\_Update API for the copy table during the check of the table status. This is a critical error (due to unexpected behavior of a cFE service) and will cause the HK app to exit.

#### #define HK\_UNEXPECTED\_TBLVLD\_RET\_EID  32

'Unexpected CFE\_TBL\_Validate return (0x%08X) for Copy Table'

##### Event Message:

'Unexpected CFE\_TBL\_Validate return (0x%08X) for Copy Table'

##### Type: ERROR

##### Cause:

This event message is issued when the HK application receives an unexpected return value when calling the CFE\_TBL\_Validate API for the copy table during the check of the table status.

#### #define HK\_UNKNOWN\_COMBINED\_PACKET\_EID  11

'Combined HK Packet 0x%08X is not found in current HK Copy Table'

##### Event Message:

'Combined HK Packet 0x%08X is not found in current HK Copy Table'

##### Type: ERROR

##### Cause:

This event message is issued when the HK application receives a command to send a combined output message that is not listed in the copy table.

## fsw/src/hk\_msg.h File Reference

#include "cfe.h"

### Data Structures

* struct [HK\_Send\_Out\_Msg\_t](#AAAAAAAADN)
* *Send Combined Output Message Command.* struct [HK\_NoArgCmd\_t](#AAAAAAAADO)
* *No-Operation command packet structure.* struct [HK\_HkPacket\_t](#AAAAAAAADP)

## fsw/src/hk\_msgdefs.h File Reference

#include "cfe.h"

### Macros

* #define [HK\_NOOP\_CC](#AAAAAAAABH)  0
* #define [HK\_RESET\_CC](#AAAAAAAABI)  1

### Macro Definition Documentation

#### #define HK\_NOOP\_CC  0

##### Purpose:

Housekeeping No-Op

##### Description

This command will increment the command execution counter and send an event containing the version number of the application

##### Command Structure

#CFE\_MSG\_CommandHeader\_t

##### Command Verification

Successful execution of this command may be verified with the following telemetry:

* - command execution counter will increment
* The [HK\_NOOP\_CMD\_EID](#AAAAAAAACC) informational event message will be generated

##### Error Conditions

There are no error conditions for this command. If the Housekeeping app receives the command, the event is sent (although it may be filtered by EVS) and the counter is incremented unconditionally.

##### Criticality

None

##### See also:

#### #define HK\_RESET\_CC  1

##### Purpose:

Housekeeping Reset Counters

##### Description

This command resets the following counters within the HK housekeeping telemetry:

* Command Execution Counter ()
* Command Error Counter ()
* Combined Packets Sent Counter ()
* Missing Data Counter ()

##### Command Structure

#CFE\_MSG\_CommandHeader\_t

##### Command Verification

Successful execution of this command may be verified with the following telemetry:

* - command execution counter will be reset
* - command error counter will be reset
* - combined packets sent counter will be reset
* - missing data counter will be reset
* The [HK\_RESET\_CNTRS\_CMD\_EID](#AAAAAAAACD) informational event message will be generated

##### Error Conditions

There are no error conditions for this command. If the Housekeeping App receives the command, the event is sent (although it may be filtered by EVS) and the counter is incremented unconditionally.

##### Criticality

This command is not inherently dangerous. However, it is possible for ground systems and on-board safing procedures to be designed such that they react to changes in the counter values that are reset by this command.

##### See also:

## fsw/src/hk\_tbldefs.h File Reference

#include "cfe.h"

### Data Structures

* struct [hk\_copy\_table\_entry\_t](#AAAAAAAADS)
* *HK Copy Table Entry Format.* struct [hk\_runtime\_tbl\_entry\_t](#AAAAAAAADT)

*HK Run-time Table Entry Format.*

## fsw/src/hk\_utils.c File Reference

#include "cfe.h"

#include "hk\_utils.h"

#include "hk\_app.h"

#include "hk\_events.h"

#include <string.h>

### Functions

* void [HK\_ProcessIncomingHkData](#AAAAAAAADV) (const CFE\_SB\_Buffer\_t \*BufPtr)

*Process incoming housekeeping data message.*

* int32 [HK\_ValidateHkCopyTable](#AAAAAAAADW) (void \*TblPtr)

*Validate Housekeeping Copy Table.*

* int32 [HK\_ProcessNewCopyTable](#AAAAAAAADX) ([hk\_copy\_table\_entry\_t](#AAAAAAAADS) \*CpyTblPtr, [hk\_runtime\_tbl\_entry\_t](#AAAAAAAADT) \*RtTblPtr)

*Process New Copy Table.*

* int32 [HK\_TearDownOldCopyTable](#AAAAAAAADY) ([hk\_copy\_table\_entry\_t](#AAAAAAAADS) \*CpyTblPtr, [hk\_runtime\_tbl\_entry\_t](#AAAAAAAADT) \*RtTblPtr)

*Tear Down Old Copy Table.*

* void [HK\_SendCombinedHkPacket](#AAAAAAAADZ) (CFE\_SB\_MsgId\_t WhichMidToSend)

*Send combined output message.*

* int32 [HK\_CheckStatusOfTables](#AAAAAAAAEA) (void)

*HK\_CheckStatusOfTables.*

* int32 [HK\_CheckStatusOfCopyTable](#AAAAAAAAEB) (void)

*HK\_CheckStatusOfCopyTable.*

* int32 [HK\_CheckStatusOfDumpTable](#AAAAAAAAEC) (void)

*HK\_CheckStatusOfDumpTable.*

* int32 [HK\_CheckForMissingData](#AAAAAAAAED) (CFE\_SB\_MsgId\_t OutPktToCheck, CFE\_SB\_MsgId\_t \*MissingInputMid)

*Check for Missing Data.*

* void [HK\_SetFlagsToNotPresent](#AAAAAAAAEE) (CFE\_SB\_MsgId\_t OutPkt)

*Set Data Present Flags to 'Not Present'.*

* int32 [HK\_VerifyCmdLength](#AAAAAAAAEF) (const CFE\_SB\_Buffer\_t \*BufPtr, size\_t ExpectedLength)

*Verify length of HK commands.*

### Function Documentation

#### int32 HK\_CheckForMissingData (CFE\_SB\_MsgId\_t *OutPktToCheck*, CFE\_SB\_MsgId\_t \* *MissingInputMid*)

Check for Missing Data.

##### Description

This routine checks for missing data for the given output message. It returns [HK\_MISSING\_DATA\_DETECTED](#AAAAAAAAEG) at the first piece of data that is not present. The missing Input MsgId is sent back to the caller through the given pointer named MissingInputMid.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *OutPktToCheck* | MsgId of the combined output message to check |
| in | *MissingInputMid* | A pointer to the caller provided MsgId variable |
| out | *\*MissingInputMid* | The value of the missing input MsgId |

##### Returns:

[HK\_MISSING\_DATA\_DETECTED](#AAAAAAAAEG) Output Msg has missing data. [HK\_NO\_MISSING\_DATA](#AAAAAAAAEH) Output Msg has no missing data.

##### See also:

#### int32 HK\_CheckStatusOfCopyTable (void )

HK\_CheckStatusOfCopyTable.

##### Description

This is a high level routine that controls the actions taken by HK when a copy table update is detected

##### Assumptions, External Events, and Notes:

None

##### Returns:

[HK\_SUCCESS](#AAAAAAAABM) HK return code for success. [HK\_ERROR](#AAAAAAAABN) HK return code for general error.

##### See also:

#### int32 HK\_CheckStatusOfDumpTable (void )

HK\_CheckStatusOfDumpTable.

##### Description

This is a high level routine that controls the actions taken by HK when a runtime table dump is pending

##### Assumptions, External Events, and Notes:

None

##### Returns:

[HK\_SUCCESS](#AAAAAAAABM) HK return code for success. [HK\_ERROR](#AAAAAAAABN) HK return code for general error.

##### See also:

#### int32 HK\_CheckStatusOfTables (void )

HK\_CheckStatusOfTables.

##### Description

This is a high level routine that controls the actions taken by HK when a copy table update is detected or a runtime table dump is pending

##### Assumptions, External Events, and Notes:

None

##### Returns:

[HK\_SUCCESS](#AAAAAAAABM) HK return code for success. [HK\_ERROR](#AAAAAAAABN) HK return code for general error.

##### See also:

#### void HK\_ProcessIncomingHkData (const CFE\_SB\_Buffer\_t \* *BufPtr*)

Process incoming housekeeping data message.

##### Description

This routine examines each entry in the table and determines whether its field comprises a legal entry. If so, a portion of the input packet is copied to the appropriate combined output packet.

##### Assumptions, External Events, and Notes:

Currently the combined telemetry packets are not initialized after they are sent so values will repeat if no housekeeping update is received.

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A pointer to the input message. |

##### See also:

[HK\_AppPipe](#AAAAAAAABR)

#### int32 HK\_ProcessNewCopyTable ([hk\_copy\_table\_entry\_t](#AAAAAAAADS) \* *CpyTblPtr*, [hk\_runtime\_tbl\_entry\_t](#AAAAAAAADT) \* *RtTblPtr*)

Process New Copy Table.

##### Description

Upon the arrival of a new HK Copy Table, this routine will handle whatever is necessary to make this new data functional.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *CpyTblPtr* | A pointer to the first entry in the new copy table. |
| in | *RtTblPtr* | A pointer to the first entry in the run-time table. |

##### Returns:

CFE\_SUCCESS if the function succeeds HK\_NULL\_POINTER\_DETECTED if at least one input argument was NULL

##### See also:

#### void HK\_SendCombinedHkPacket (CFE\_SB\_MsgId\_t *WhichMidToSend*)

Send combined output message.

##### Description

This routine searches for the combined HK that contains the specified MID. Once found, the packet is sent. If not found, an event is generated. Also sets the data pieces for this output pkt

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *WhichMidToSend* | - the MsgId of the combined output message to send |

##### See also:

#### void HK\_SetFlagsToNotPresent (CFE\_SB\_MsgId\_t *OutPkt*)

Set Data Present Flags to 'Not Present'.

##### Description

This routine will set the data present flags to data-not-present for given combined output message

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *OutPkt* | The MsgId whose data present flags will be set. |

##### See also:

#### int32 HK\_TearDownOldCopyTable ([hk\_copy\_table\_entry\_t](#AAAAAAAADS) \* *CpyTblPtr*, [hk\_runtime\_tbl\_entry\_t](#AAAAAAAADT) \* *RtTblPtr*)

Tear Down Old Copy Table.

##### Description

This routine does what is necessary in order to remove the table data from the system.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *CpyTblPtr* | A pointer to the first entry in the copy table. |
| in | *RtTblPtr* | A pointer to the first entry in the run-time table. |

##### Returns:

CFE\_SUCCESS if the function succeeds HK\_NULL\_POINTER\_DETECTED if at least one input argument was NULL

##### See also:

#### int32 HK\_ValidateHkCopyTable (void \* *TblPtr*)

Validate Housekeeping Copy Table.

##### Description

This routine is called from CFE\_TBL\_Register. It determines whether the data contained in the new table is acceptable.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *TblPtr* | A pointer to the new table data. |

##### Returns:

Zero indicates acceptable table, non-zero indicates unacceptable table

##### See also:

[HK\_TableInit](#AAAAAAAABQ)

#### int32 HK\_VerifyCmdLength (const CFE\_SB\_Buffer\_t \* *BufPtr*, size\_t *ExpectedLength*)

Verify length of HK commands.

##### Description

Function called when an HK command is received.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A #CFE\_SB\_Buffer\_t\* pointer that references the software bus message |
| in | *ExpectedLength* | The expected lenght of the command |

##### Returns:

[HK\_SUCCESS](#AAAAAAAABM) if actual cmd length is same as expected [HK\_BAD\_MSG\_LENGTH\_RC](#AAAAAAAABO) if actual cmd length is not as expected

##### See also:

## fsw/src/hk\_utils.h File Reference

#include "cfe.h"

#include "hk\_tbldefs.h"

### Macros

* #define [HK\_INPUTMID\_SUBSCRIBED](#AAAAAAAAEJ)  ( 0xFF )

*Input MsgId has been subscribed to.*

* #define [HK\_INPUTMID\_NOT\_SUBSCRIBED](#AAAAAAAAEK)  ( 0 )

*Input MsgId is not subscribed.*

* #define [HK\_DATA\_NOT\_PRESENT](#AAAAAAAAEL)  ( 0 )

*Input MsgId present in output msg.*

* #define [HK\_DATA\_PRESENT](#AAAAAAAAEM)  ( 1 )

*Input MsgId not present.*

* #define [HK\_NO\_MISSING\_DATA](#AAAAAAAAEH)  ( 0 )

*Output Msg has no missing data.*

* #define [HK\_MISSING\_DATA\_DETECTED](#AAAAAAAAEG)  ( 1 )

*Output Msg has missing data.*

* #define [HK\_UNDEFINED\_ENTRY](#AAAAAAAAEN)  ( 0 )

*Undefined table field entry.*

* #define [HK\_NULL\_POINTER\_DETECTED](#AAAAAAAAEO)  ( -1 )

*An input table pointer was NULL.*

### Functions

* void [HK\_ProcessIncomingHkData](#AAAAAAAAEP) (const CFE\_SB\_Buffer\_t \*BufPtr)

*Process incoming housekeeping data message.*

* int32 [HK\_ValidateHkCopyTable](#AAAAAAAAEQ) (void \*TblPtr)

*Validate Housekeeping Copy Table.*

* int32 [HK\_ProcessNewCopyTable](#AAAAAAAAER) ([hk\_copy\_table\_entry\_t](#AAAAAAAADS) \*CpyTblPtr, [hk\_runtime\_tbl\_entry\_t](#AAAAAAAADT) \*RtTblPtr)

*Process New Copy Table.*

* int32 [HK\_TearDownOldCopyTable](#AAAAAAAAES) ([hk\_copy\_table\_entry\_t](#AAAAAAAADS) \*CpyTblPtr, [hk\_runtime\_tbl\_entry\_t](#AAAAAAAADT) \*RtTblPtr)

*Tear Down Old Copy Table.*

* void [HK\_SendCombinedHkPacket](#AAAAAAAAET) (CFE\_SB\_MsgId\_t WhichMidToSend)

*Send combined output message.*

* int32 [HK\_CheckStatusOfTables](#AAAAAAAAEU) (void)

*HK\_CheckStatusOfTables.*

* int32 [HK\_CheckStatusOfCopyTable](#AAAAAAAAEV) (void)

*HK\_CheckStatusOfCopyTable.*

* int32 [HK\_CheckStatusOfDumpTable](#AAAAAAAAEW) (void)

*HK\_CheckStatusOfDumpTable.*

* int32 [HK\_CheckForMissingData](#AAAAAAAAEX) (CFE\_SB\_MsgId\_t OutPktToCheck, CFE\_SB\_MsgId\_t \*MissingInputMid)

*Check for Missing Data.*

* void [HK\_SetFlagsToNotPresent](#AAAAAAAAEY) (CFE\_SB\_MsgId\_t OutPkt)

*Set Data Present Flags to 'Not Present'.*

* int32 [HK\_VerifyCmdLength](#AAAAAAAAEZ) (const CFE\_SB\_Buffer\_t \*BufPtr, size\_t ExpectedLength)

*Verify length of HK commands.*

### Macro Definition Documentation

#### #define HK\_DATA\_NOT\_PRESENT  ( 0 )

Input MsgId present in output msg.

#### #define HK\_DATA\_PRESENT  ( 1 )

Input MsgId not present.

#### #define HK\_INPUTMID\_NOT\_SUBSCRIBED  ( 0 )

Input MsgId is not subscribed.

#### #define HK\_INPUTMID\_SUBSCRIBED  ( 0xFF )

Input MsgId has been subscribed to.

#### #define HK\_MISSING\_DATA\_DETECTED  ( 1 )

Output Msg has missing data.

#### #define HK\_NO\_MISSING\_DATA  ( 0 )

Output Msg has no missing data.

#### #define HK\_NULL\_POINTER\_DETECTED  ( -1 )

An input table pointer was NULL.

#### #define HK\_UNDEFINED\_ENTRY  ( 0 )

Undefined table field entry.

### Function Documentation

#### int32 HK\_CheckForMissingData (CFE\_SB\_MsgId\_t *OutPktToCheck*, CFE\_SB\_MsgId\_t \* *MissingInputMid*)

Check for Missing Data.

##### Description

This routine checks for missing data for the given output message. It returns [HK\_MISSING\_DATA\_DETECTED](#AAAAAAAAEG) at the first piece of data that is not present. The missing Input MsgId is sent back to the caller through the given pointer named MissingInputMid.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *OutPktToCheck* | MsgId of the combined output message to check |
| in | *MissingInputMid* | A pointer to the caller provided MsgId variable |
| out | *\*MissingInputMid* | The value of the missing input MsgId |

##### Returns:

[HK\_MISSING\_DATA\_DETECTED](#AAAAAAAAEG) Output Msg has missing data. [HK\_NO\_MISSING\_DATA](#AAAAAAAAEH) Output Msg has no missing data.

##### See also:

#### int32 HK\_CheckStatusOfCopyTable (void )

HK\_CheckStatusOfCopyTable.

##### Description

This is a high level routine that controls the actions taken by HK when a copy table update is detected

##### Assumptions, External Events, and Notes:

None

##### Returns:

[HK\_SUCCESS](#AAAAAAAABM) HK return code for success. [HK\_ERROR](#AAAAAAAABN) HK return code for general error.

##### See also:

#### int32 HK\_CheckStatusOfDumpTable (void )

HK\_CheckStatusOfDumpTable.

##### Description

This is a high level routine that controls the actions taken by HK when a runtime table dump is pending

##### Assumptions, External Events, and Notes:

None

##### Returns:

[HK\_SUCCESS](#AAAAAAAABM) HK return code for success. [HK\_ERROR](#AAAAAAAABN) HK return code for general error.

##### See also:

#### int32 HK\_CheckStatusOfTables (void )

HK\_CheckStatusOfTables.

##### Description

This is a high level routine that controls the actions taken by HK when a copy table update is detected or a runtime table dump is pending

##### Assumptions, External Events, and Notes:

None

##### Returns:

[HK\_SUCCESS](#AAAAAAAABM) HK return code for success. [HK\_ERROR](#AAAAAAAABN) HK return code for general error.

##### See also:

#### void HK\_ProcessIncomingHkData (const CFE\_SB\_Buffer\_t \* *BufPtr*)

Process incoming housekeeping data message.

##### Description

This routine examines each entry in the table and determines whether its field comprises a legal entry. If so, a portion of the input packet is copied to the appropriate combined output packet.

##### Assumptions, External Events, and Notes:

Currently the combined telemetry packets are not initialized after they are sent so values will repeat if no housekeeping update is received.

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A pointer to the input message. |

##### See also:

[HK\_AppPipe](#AAAAAAAABR)

#### int32 HK\_ProcessNewCopyTable ([hk\_copy\_table\_entry\_t](#AAAAAAAADS) \* *CpyTblPtr*, [hk\_runtime\_tbl\_entry\_t](#AAAAAAAADT) \* *RtTblPtr*)

Process New Copy Table.

##### Description

Upon the arrival of a new HK Copy Table, this routine will handle whatever is necessary to make this new data functional.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *CpyTblPtr* | A pointer to the first entry in the new copy table. |
| in | *RtTblPtr* | A pointer to the first entry in the run-time table. |

##### Returns:

CFE\_SUCCESS if the function succeeds HK\_NULL\_POINTER\_DETECTED if at least one input argument was NULL

##### See also:

#### void HK\_SendCombinedHkPacket (CFE\_SB\_MsgId\_t *WhichMidToSend*)

Send combined output message.

##### Description

This routine searches for the combined HK that contains the specified MID. Once found, the packet is sent. If not found, an event is generated. Also sets the data pieces for this output pkt

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *WhichMidToSend* | - the MsgId of the combined output message to send |

##### See also:

#### void HK\_SetFlagsToNotPresent (CFE\_SB\_MsgId\_t *OutPkt*)

Set Data Present Flags to 'Not Present'.

##### Description

This routine will set the data present flags to data-not-present for given combined output message

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *OutPkt* | The MsgId whose data present flags will be set. |

##### See also:

#### int32 HK\_TearDownOldCopyTable ([hk\_copy\_table\_entry\_t](#AAAAAAAADS) \* *CpyTblPtr*, [hk\_runtime\_tbl\_entry\_t](#AAAAAAAADT) \* *RtTblPtr*)

Tear Down Old Copy Table.

##### Description

This routine does what is necessary in order to remove the table data from the system.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *CpyTblPtr* | A pointer to the first entry in the copy table. |
| in | *RtTblPtr* | A pointer to the first entry in the run-time table. |

##### Returns:

CFE\_SUCCESS if the function succeeds HK\_NULL\_POINTER\_DETECTED if at least one input argument was NULL

##### See also:

#### int32 HK\_ValidateHkCopyTable (void \* *TblPtr*)

Validate Housekeeping Copy Table.

##### Description

This routine is called from CFE\_TBL\_Register. It determines whether the data contained in the new table is acceptable.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *TblPtr* | A pointer to the new table data. |

##### Returns:

Zero indicates acceptable table, non-zero indicates unacceptable table

##### See also:

[HK\_TableInit](#AAAAAAAABQ)

#### int32 HK\_VerifyCmdLength (const CFE\_SB\_Buffer\_t \* *BufPtr*, size\_t *ExpectedLength*)

Verify length of HK commands.

##### Description

Function called when an HK command is received.

##### Assumptions, External Events, and Notes:

None

##### Parameters:

|  |  |  |
| --- | --- | --- |
| in | *BufPtr* | A #CFE\_SB\_Buffer\_t\* pointer that references the software bus message |
| in | *ExpectedLength* | The expected lenght of the command |

##### Returns:

[HK\_SUCCESS](#AAAAAAAABM) if actual cmd length is same as expected [HK\_BAD\_MSG\_LENGTH\_RC](#AAAAAAAABO) if actual cmd length is not as expected

##### See also:

## fsw/src/hk\_verify.h File Reference

#include "cfe.h"

#include "hk\_platform\_cfg.h"

#include "hk\_app.h"

## fsw/src/hk\_version.h File Reference

### Macros

* #define [HK\_MAJOR\_VERSION](#AAAAAAAADJ)  2
* #define [HK\_MINOR\_VERSION](#AAAAAAAADK)  4
* #define [HK\_REVISION](#AAAAAAAADL)  3

### Macro Definition Documentation

#### #define HK\_MAJOR\_VERSION  2

#### #define HK\_MINOR\_VERSION  4

#### #define HK\_REVISION  3

#### 

## fsw/tables/hk\_cpy\_tbl.c File Reference

#include "cfe.h"

#include "cfe\_msgids.h"

#include "hk\_utils.h"

#include "hk\_app.h"

#include "hk\_msgids.h"

#include "hk\_tbldefs.h"

#include "cfe\_tbl\_filedef.h"

### Variables

* [hk\_copy\_table\_entry\_t](#AAAAAAAADS) [HK\_CopyTable](#AAAAAAAAFD) [[HK\_COPY\_TABLE\_ENTRIES](#AAAAAAAAAP)]

### Variable Documentation

#### [hk\_copy\_table\_entry\_t](#AAAAAAAADS) HK\_CopyTable[[HK\_COPY\_TABLE\_ENTRIES](#AAAAAAAAAP)]