Core Flight System (cFS) Training

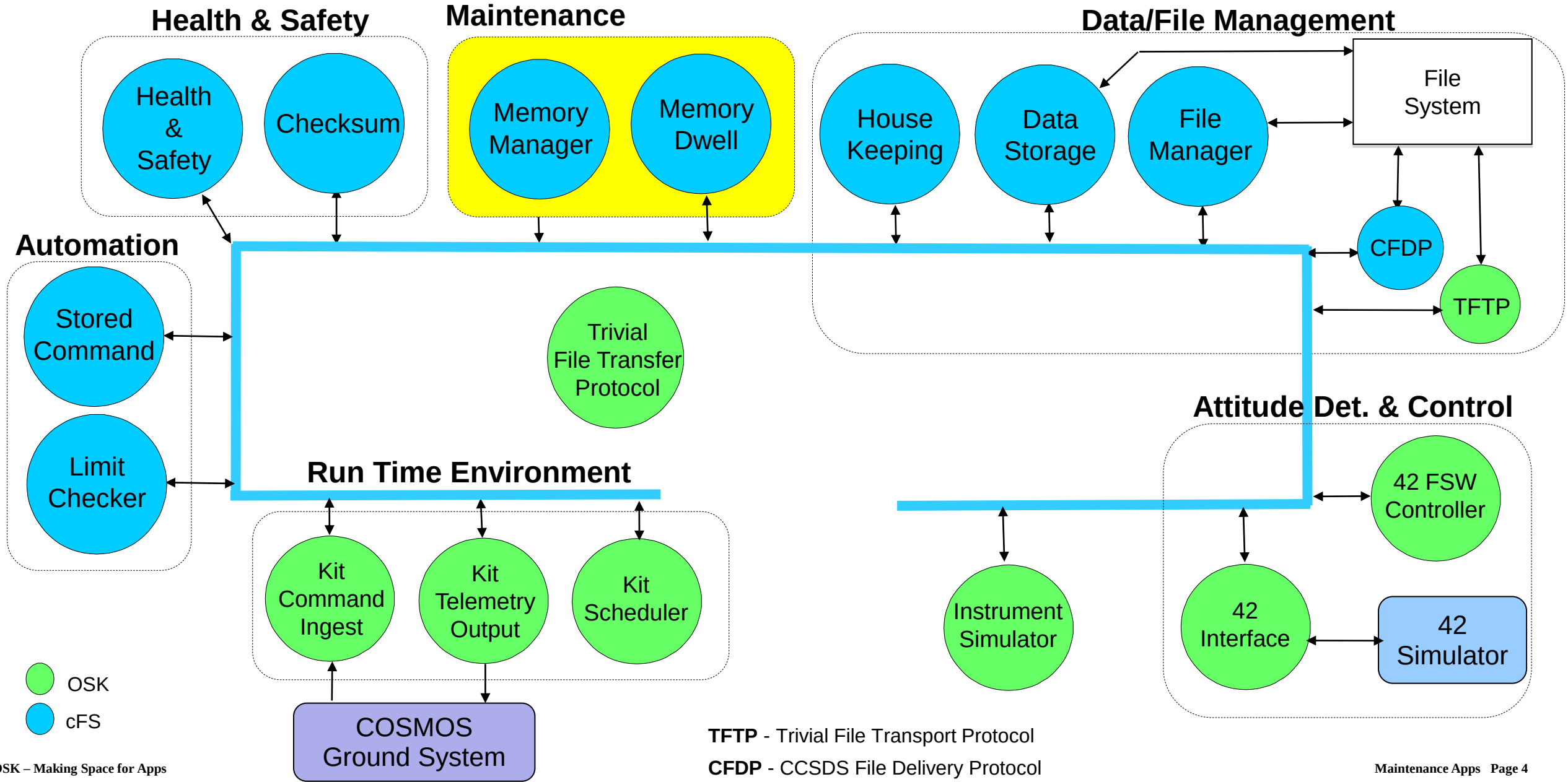
Community Apps: Maintenance

Maintenance App Agenda

- These are draft slides released with OSK v2.8
- An introductory video and demo script was released with v2.8
- The remainder of the slide deck contains slides that were collected from existing material for each app. This material will be matured as each detailed video is created


Maintenance
Application Overview

OSK FSW SimSat Applications



TFTP - Trivial File Transport Protocol

CFDP - CCSDS File Delivery Protocol

Maintenance Apps vs cFS Maintainability

- **Memory Dwell (MD)**
  - Provides commands for manipulating directories and files
  - Users obtain information about directories, files and file systems by requesting one time telemetry packets or dumping information to a file
- **Memory Manager (MM)**
  - Combines subsets of multiple source packets from any app into a new packet
  - New packets are generated when an “Output Pkt” command is received

App Overview

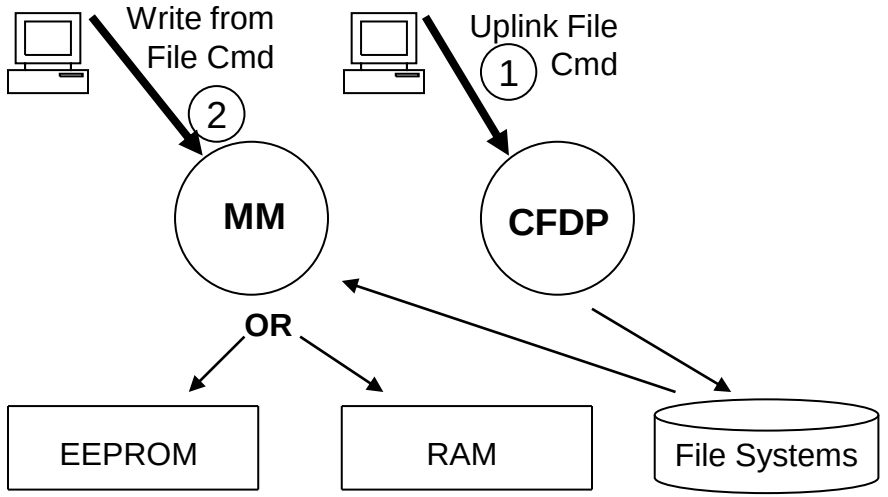
- **Memory Dwell (MD)**
  - Provides commands for manipulating directories and files
  - Users obtain information about directories, files and file systems by requesting one time telemetry packets or dumping information to a file
- **Memory Manager (MM)**
  - Combines subsets of multiple source packets from any app into a new packet
  - New packets are generated when an “Output Pkt” command is received


Operational Scenarios

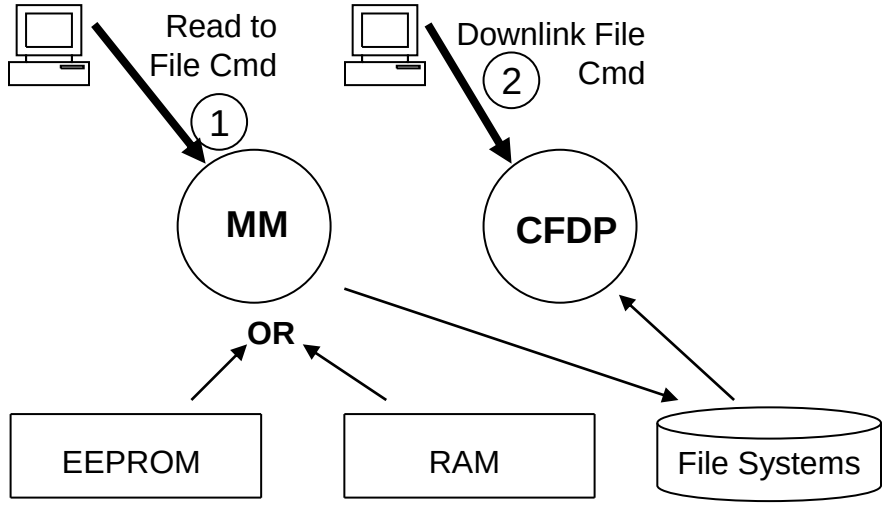
Load-Dump Memory

- **MM Features**
  - Commanded Writes (peek and poke)
  - Commanded Reads via event messages
  - File Reads and Write (show in diagram)
- **Upload to Memory from Ground**
  1. Uplink File using CFDP
  2. Write the data from a file into EEPROM or RAM
- **Download from Memory to Ground**
  1. Read the data from EEPROM or RAM into a file
  2. Downlink File using CFDP

Upload to Memory from Ground

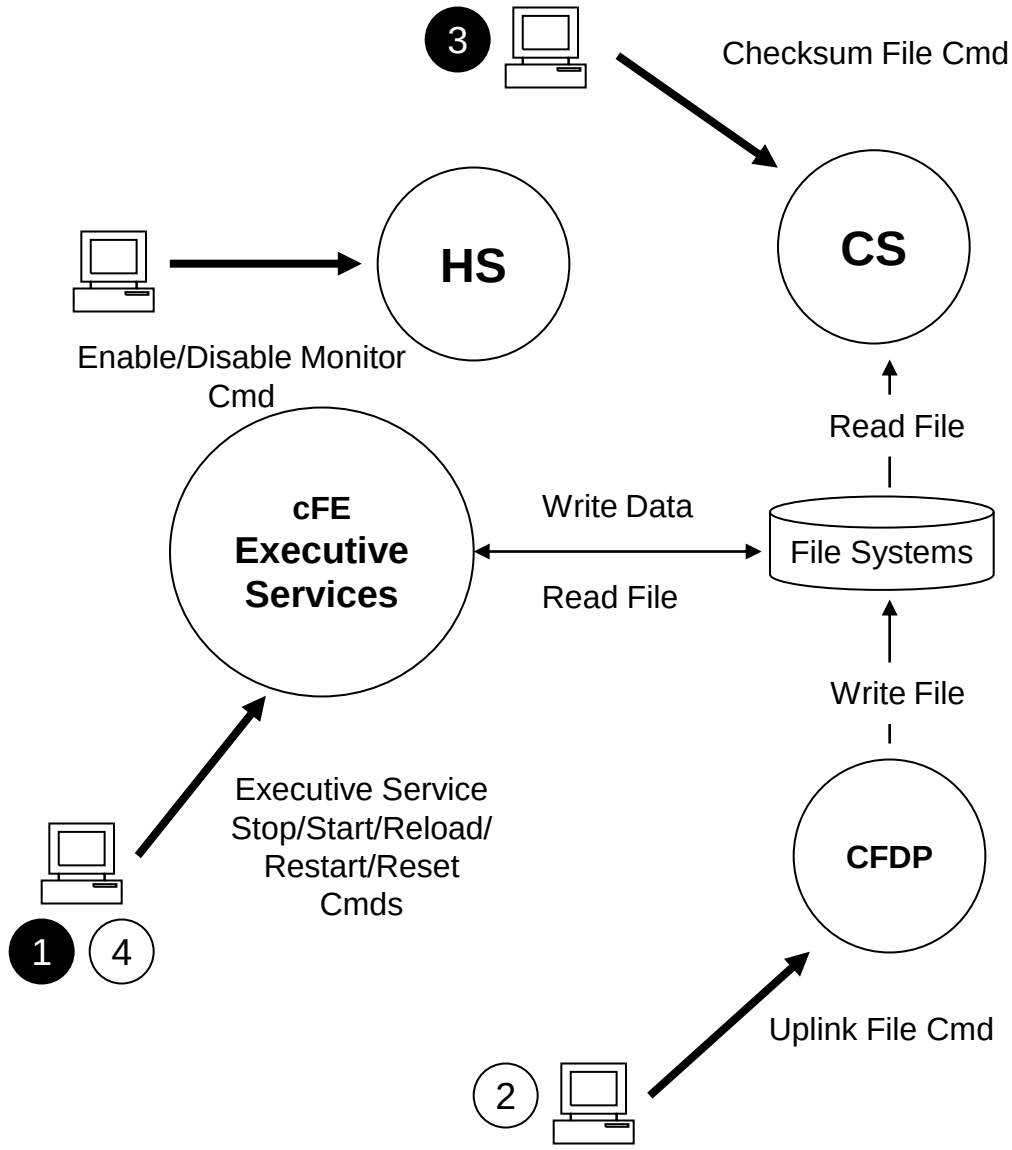


Download from Memory to Ground



Load and Execute Application Updates

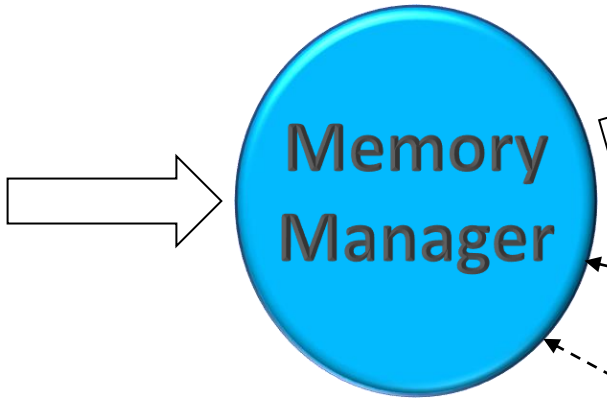
1) **Send Executive Service command to stop application**
2) **Uplink file – file containing code update(s) is written to File System**
3) **Checksum the file**
4) **Send Executive Service commands to:**
   - Reload application
   - Start application
   - Restart application
   - Perform Processor reset



● - Optional Step

MM & MD Demo Context

Commands
- Peek memory location
- Poke memory location
- Dump memory to event
- Dump memory to file
- Load memory from file
- Fill memory with pattern

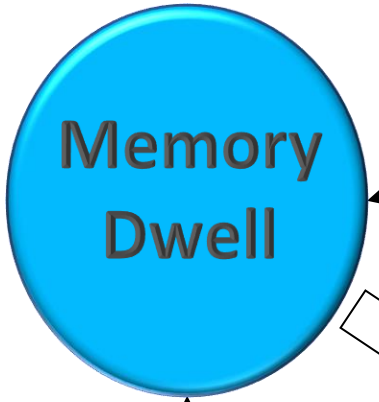


Telemetry
- Housekeeping
- Events messages with data

Memory Dump/Load Files

MM_AppData + 256

Commands
- Jam dwell table
- Set dwell table signature
- Start dwell



Telemetry
- Housekeeping
- Dwell packet

MM Application Data Structure

mm_app.c

```
/*************************
** MM global data
*************************/
MM_AppData_t MM_AppData;
```

mm_app.h

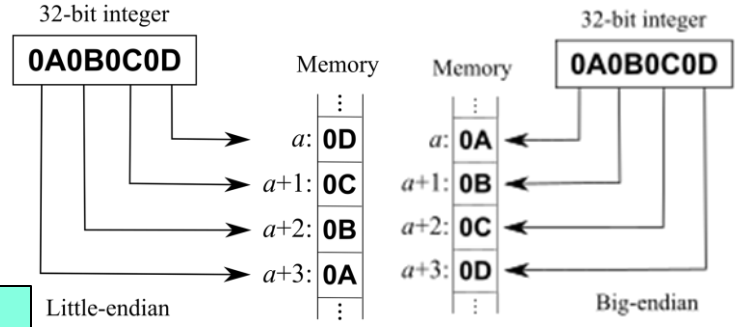
```
typedef struct
{
    MM_HkPacket_t       HkPacket;       /**< \brief Housekeeping telemetry packet  */

    CFE_SB_MsgPtr_t     MsgPtr;         /**< \brief Pointer to command message     */
    CFE_SB_PipeId_t     CmdPipe;        /**< \brief Command pipe ID                */

    uint32              RunStatus;      /**< \brief Application run status         */

    char                PipeName[16];   /**< \brief Command pipe name              */
    uint16              PipeDepth;      /**< \brief Command pipe message depth     */

    uint8               LimitHK;        /**< \brief Houskeeping messages limit     */
    uint8               LimitCmd;       /**< \brief Command messages limit         */

    uint8               CmdCounter;     /**< \brief MM Application Command Counter       */
    uint8               ErrCounter;     /**< \brief MM Application Command Error Counter */
    uint8               LastAction;     /**< \brief Last command action executed         */
    uint8               MemType;        /**< \brief Memory type for last command         */
    uint32              Address;        /**< \brief Fully resolved address used for last
                                             command                                     */
    uint32              DataValue;      /**< \brief Last command data value -- may be
                                             fill pattern or peek/poke value             */
    uint32              BytesProcessed; /**< \brief Bytes processed for last command     */

    char                FileName[OS_MAX_PATH_LEN];   /**< \brief Name of the data file
                                                          used for last command,
                                                          where applicable          */

    uint32          LoadBuffer[MM_MAX_LOAD_DATA_SEG / 4];  /**< \brief Load file i/o buffer */
    uint32          DumpBuffer[MM_MAX_DUMP_DATA_SEG / 4];  /**< \brief Dump file i/o buffer */
    uint32          FillBuffer[MM_MAX_FILL_DATA_SEG / 4];  /**< \brief Fill memory buffer   */

} MM_AppData_t;
```

256-byte offset** is 52 bytes into LoadBuffer

**256 decimal = 0x0100

Endian Refresher

- MD's dwell table jammed to address 32-but integers
- MM memory dump displayed as an array of bytes



Demo displays on a little-endian machine

MM MEMORY DUMP

NUMBER OF BYTES	16
CRC	15859
MEMORY TYPE	1
SPARE_8	0
SPARE_16	0
DATA BYTE 0	4
DATA BYTE 1	3
DATA BYTE 2	2
DATA BYTE 3	1
DATA BYTE 4	8
DATA BYTE 5	7
DATA BYTE 6	6
DATA BYTE 7	5

MD DWELL_PKT_SCREEN

Memory Dwell Table 1

Status

Stream ID 0x0891 Seq Cnt 49408 Tbl ID 1

Description Memory Management Demo

Addr Cnt 4 Byte Cnt 16 Interval 4

Values

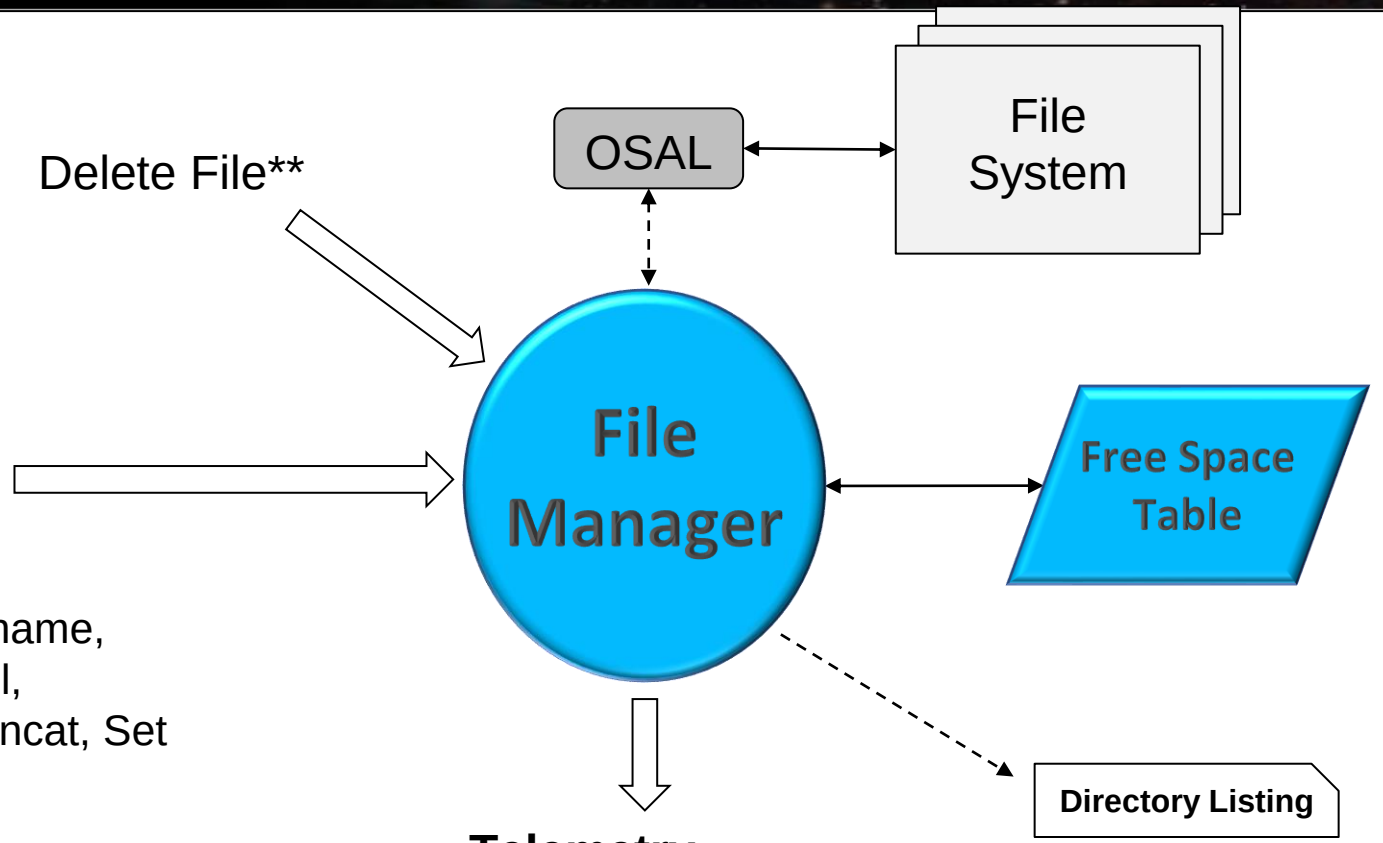
Index	Byte 0	Byte 1	Byte 2	Byte 3
0	01	02	03	04
1	05	06	07	08
2	09	0A	0B	0C
3	0D	0E	0F	10


Open
Sat Kit
cFS
Memory Dwell
https://github.com/nasa/MD

File Manager Objectives

- **Provide an interface for managing files and directories**
  - Primary interface is the ground but receiving commands from other apps must also be taken into consideration
- **Meet the specific needs of the spacecraft file management environment**
  - Once a spacecraft is operational, directory structures are typically fairly static
    - The file system ‘clients’ are based on the spacecraft’s interfaces (data producers/consumers) that don’t change
  - Deterministic and efficient file system performance is often required
  - Working over a command and telemetry space link limits real-time file system management
  - File transfer protocols like CCSDS File Delivery Protocol (CFDP) offer options to automatically delete a file once a file transfer has completed successfully
- **Operate in the cFS environment**
  - All directory and file commands use the cFS’ Operating System Abstraction Layer (OSAL) to access the file system
    - Consistent behavior depends on the underlying OS
  - Executive Service’s shell command provides a ‘backdoor’

File Manager Context



Commands

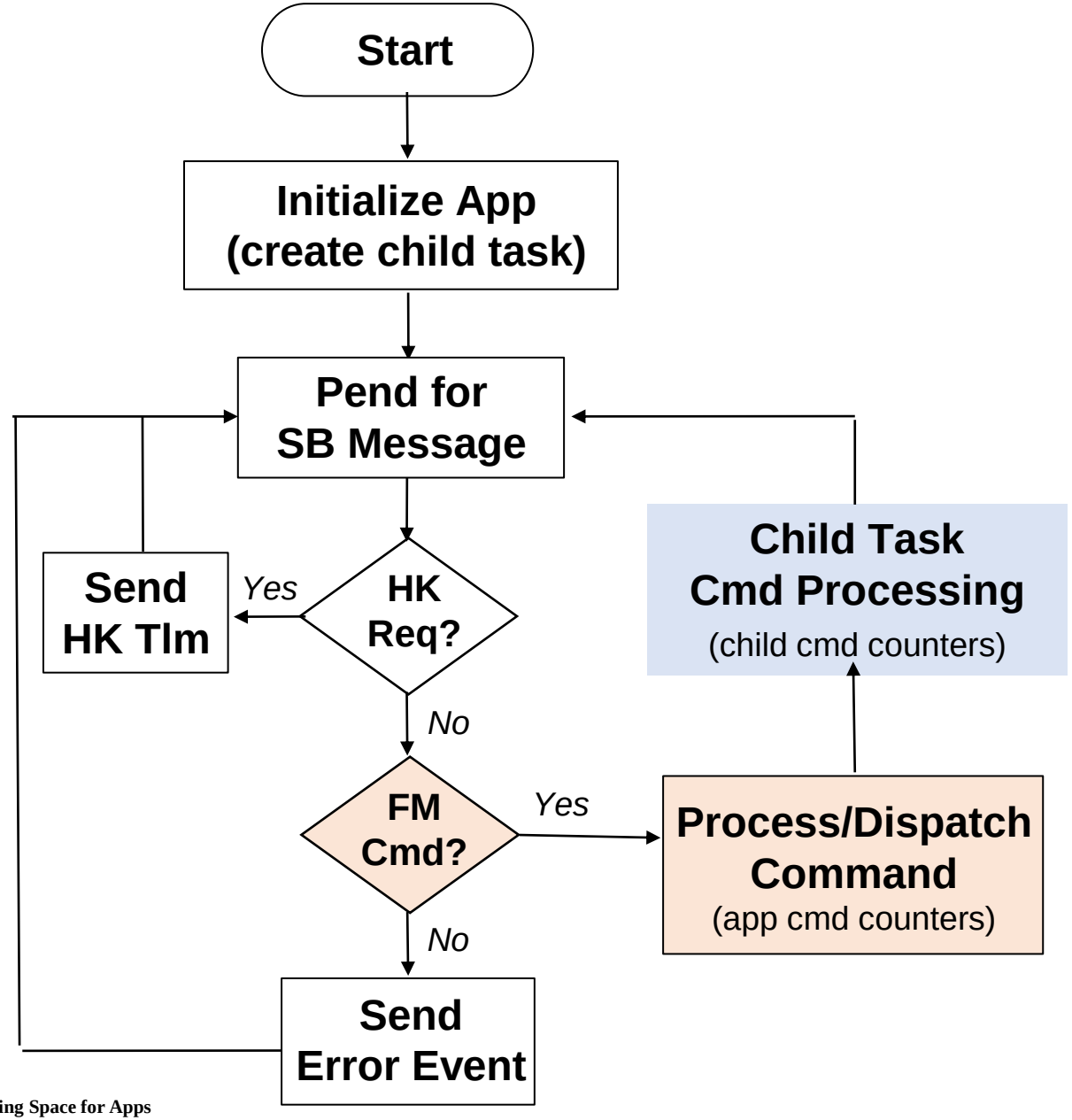
- Directories
  - Create, Delete
- Files
  - Copy, Move, Rename, Delete, Delete All, Decompress, Concat, Set Permissions
- Status
  - Send File Info Tlm Pkt
  - Send Open File Tlm Pkt
  - Write Dir to File
  - Send Dir Tlm Pkt
  - Send Free Space Tlm Pkt

Telemetry

- Housekeeping
- File Info
- Open Files
- Directory Listing
- File Sys Free Space

** Onboard command that doesn't affect ground command counters

FM Application Control Flow



Child Task

- Separate thread of execution that shares main app's memory space
- Implements all command functions except
  - Get File System Free Space
  - Set Free Space Table State
  - Send Open File Telemetry Pkt
- Tune performance using build-time configuration parameters

Command Processing

- Main app processing validates command input parameters
  - Increments valid command count if valid inputs and successful command dispatch to child task
- Child task performs command function and increments command counters if successful

FM Demo Directories

~/OpenSatKit-master/cfs/build/exe/cpu1/cf

cFE startup script
App object and table files

demo

Demo work area

adir

afile.txt
osk_tmp_bin.dat

simsat

Reference mission files

rec

Science and engineering data recorder
isim_xx.txt

test

Integration and Functional test work area

Highlighted directory and files used in FM demos

Key Configuration Parameters

Data Structure Definitions	Description	External Dependencies
FM_DIR_LIST_FILE_ENTRIES	Max directory entries written to a file	Table Manger binary file definition
FM_DIR_LIST_PKT_ENTRIES	Max directory entries listed in the directory telemetry packet	Telemetry packet definition
FM_TABLE_ENTRY_COUNT	Max FreeSpace table entries	Table Manger binary file definition
OS_MAX_PATH_LEN	OSAL definition used in multiple full path filename definitions	C&T pkt and binary file definitions
OS_MAX_NUM_OPEN_FILES	OSAL definition used in max open file telemetry listing	Telemetry packet definition

Child Task Definitions	Description
FM_CHILD_TASK_PRIORITY	Execution priority for child task
FM_CHILD_QUEUE_DEPTH	Max number of commands that can be queued to child task
FM_CHILD_FILE_BLOCK_SIZE	Size of each block read/written from/to files
FM_CHILD_FILE_LOOP_COUNT	Number of file blocks read/written between task sleeps
FM_CHILD_FILE_SLEEP_MS	Duration of child task's sleep between file block reads/writes
FM_CHILD_STAT_SLEEP_FILECOUNT	Number of file status inquires between task sleeps
FM_CHILD_STAT_SLEEP_MS	Duration of child task's sleep between file status inquiries

FM Commands (1 of 2)

Command	Description
Noop	Increments the Command Accepted Counter and sends a debug event message
Reset Command Counters	Initializes the following FM counters to 0: Command Rejected Counter, Command Accepted Counter
File Copy	Copies the command-specified file to the command-specified destination file or directory
File Move	Moves the command-specified file to the command-specified destination file or directory
Rename File	Renames the command-specified file to the command-specified file
Delete File	Deletes the command-specified file, if and only if, the file is closed
Delete All Files	Deletes all files in the command-specified directory, if and only if, the files are closed.
Decompress File	Decompresses the command-specified file creating the command-specified destination file
Concatenate Files	Concatenates the command-specified source files creating the command-specified destination file
File Information	Creates and sends a software bus message containing the file size, last modification time, and file status (Open, Closed) of a given file, if and only if, the file exists

FM Commands (2 of 2)

Command	Description
List Open Files	Creates and sends a software bus message containing the number of open files, the name/path of each open file, and application identifier associated with each open file
Create Directory	Creates the command-specified directory
Delete Directory	Removes the command-specified directory, if and only if, the command-specified directory is empty
Directory Listing via File	Writes to a file the complete listing of the command-specified directory
Directory Listing via Message	Creates and sends a software bus message containing the contents of a directory (up to <PLATFORM_DEFINED> filenames, starting at the command-specified offset)

File Information Telemetry Message

Telemetry Point	Description
FileStatus	Status indicating whether the file is Open or Closed
CRC_Computed	Flag indicating if a CRC was computed on the command specified file
<OPTIONAL> CRC	Computed CRC of file contents
FileSize	Size of file in bytes
LastModifiedTime	System time the file was last modified
Filename	Echo of command specified filename

- CRC ground tool provided

FM Open File Listing Telemetry Message

Telemetry Point	Description
NumOpenFiles	Number of open files in the FSW system
FileNames[1..n] where n = <PLATFORM_DEFINED> FM_MAX_OPEN_FILE_LIST_MSG_FILES	Names of open files in the FSW system
AppNames[1..n] where n = <PLATFORM_DEFINED> FM_MAX_OPEN_FILE_LIST_MSG_FILES	Names of applications that have files open in the FSW system

Directory Listing Telemetry Message

Telemetry Point	Description
DirSize	Directory size in bytes
DirOffset	Echo of command specified directory offset
TotalFiles	Total number of files contained in the command specified directory
FileSizes[1..n] where n = <PLATFORM_DEFINED> FM_MaxDirListMsgFiles	Sizes of the files contained within the command-specified directory starting at the command specified offset
FileLastModTimes[1..n] where n = <PLATFORM_DEFINED> FM_MaxDirListMsgFiles	Last modification times of the files contained within the command-specified directory starting at the command specified offset
DirName	Echo of command specified directory name
FileNames[1..n] where n = <PLATFORM_DEFINED> FM_MaxDirListMsgFiles	Names of files contained within the command-specified directory starting at the command-specified offset

Directory Listing File

- **File Format**
  - Binary
- **File Content**
  - cFE file header
    - Header length
    - Spacecraft ID
    - Processor ID
    - Application ID
    - Creation Time (seconds and subseconds)
    - File Description
  - Echo of command-specified directory name
  - Directory size in bytes
  - Total number of files in the directory
  - For each file contained in the directory:
    - File Name
    - File Size
    - Last Modification Time

Open Files Telemetry Message

Telemetry Point	Description
CommandCounter	Number of rejected commands
CommandErrCounter	Number of accepted commands
NumOpenFiles	Number of open files in the entire FSW system
BlockSize[1..n]	Block size of drive n
NumBlocks[1..n] where n = <MISSION_DEFINED> FMMaxNumDevices	Number of available blocks on drive n

Memory Manager

https://github.com/nasa/MM