OpenSatKit (OSK) Quick Start Guide

OpenSatKit – 2.0

Introduction



- The primary goal of OpenSatKit (OSK) is to provide a core Flight System (cFS) development and run time environment that can be used to learn about the cFS and to serve as a starting point for a new project
- In addition to the cFS itself, OSK uses two additional open source applications
 - Ball Aerospace's COSMOS command and control platform for embedded systems
 - NASA Goddard's 42 dynamic simulator
- Each open source package is contained in its own OpenSatKit subdirectory



Approach



- OSK comes with the cFS pre-configured for a fictitious satellite called SimpleSat (SimSat).
 - The cFS can be used for many different types of embedded systems. A spacecraft was
 chosen due to the increased usage of the cFS on CubeSats
- OSK implements extensive COSMOS configurations and customizations so COSMOS can serve as the primary OSK user interface
 - OSK is arranged with the following user progression in mind
 - **1.** Learn the cFS using SimSat to provide a context and working examples
 - 2. Manage and develop applications within the Linux desktop environment
 - a. Add apps by creating new apps or importing from the app library
 - b. Configure runtime app suite
 - 3. Extend OSK
 - a. Deploy the cFS to a target system
 - i. Run benchmarks
 - ii. Use OSK as a ground system for a remote system
 - **b.** Advanced application development and extensions
 - i. External Code Interface (ECI), ROS2 bridge, etc.

Running OpenSatKit (1 of 2)

- Open a terminal window (Ctrl-Alt-t)
- Navigate to the base directory where you installed OSK
- Change directory to cosmos
 - [~] cd opensatkit-master/cosmos
- Start COSMOS
 - [~/OpenSatKit/cosmos]ruby Launcher
 - You'll see a screen similar to the right.
 - Select <OK>
 - This creates the "Launcher" screen shown on the next slide

👌 🗈 🛛 Legal A	greement	2012/2011 22:113
	COSMOS Bell Aerospace & Technologies Corp.	
	COSMOS Version: 4.1.1	
Copyright 20 All Rights Res	17 Ball Aerospace & Technologies Corp. erved.	
COSMOS is fr Public Licens found in the I COSMOS is d the implied w General Publi	ee software: you can redistribute it and/or modify it under the + as published by the Free Software Foundation, version 3 with JCENSE.txt stributed in the hope that it will be useful, but WITHOUT ANY arranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR c License for more details.	e terms of the GNU General n attribution addendums as WARRANTY; without even PURPOSE. See the GNU
By clicking O	C below, you accept these terms.	
Warning: Proj Remove USE Warning: 1 Pro PROJECT con CRC Expecte	ct CRC file updated by user <_MODIFIED from config/data/crc.txt to clear this warning ject CRC checks failed! ig/targets/F42/cmd_tmr/F42_tlmtxt t: 0x83CF3442, CRC Calculated: 0x8F0C99BF	
		ĩ
Ok	Update Project CRCs	Cancel



Running OpenSatKit (2 of 2)





- Each tools on the COSMOS "Launcher" runs as a separate Linux process with a Graphical User Interface (GUI)
- Shaded tool titles indicate the COSMOS tools used by OSK
 - You do not have to invoke these tools directly
 - OSK screens launch COSMOS tools as they are needed to perform a task
 - A backup slide shows a COSMOS architectural view with the data flows between tools
- Select "OpenSatKit" with a single click
 - This launches COSMOS's Command and Telemetry Server, Telemetry Viewer, and displays OSK's main window
 - You can minimize the COSMOS tools, but don't close them
- A picture of OSK's main window follows 2 slides that briefly describe each COSMOS tool

COSMOS Tool Summary (1 of 2)



- Provides a graphical interface for launching each of the tools that make up the COSMOS system
- Custom OSK ICON "cFS Starter Kit" launches OSK's main page

Command and Telemetry Server

- Connects COSMOS to targets for real-time commanding and telemetry processing.
- All real-time COSMOS tools communicate with targets through the Command and Telemetry Server ensuring that all communications are logged.
- Localhost 127.0.0.1 used as cFS connection Targets created

Telemetry Viewer

 Provides a way to organize telemetry points into custom "screens" that allow for the creation of unique and organized views of telemetry data.

Command Sender

- Individually send any FSW command using GUI form
- Raw data files can be used to inject faults
- OSK provides custom menus for common cFS commands

Packet Viewer

- View any telemetry packet with no extra configuration necessary
- OSK provides custom telemetry screens functionally organized

COSMOS Tool Summary (2 of 2)



- Real-time or offline graphing of any FSW telemetry point
- OSK provides convenient access through some of its custom screens

Table Manager

- Edit and display binary files
- OSK provides definitions for most of the cFE binary files and a limited number of cFS application binary files

Script Runner

- Develop and execute test procedures using Ruby Scripts and COSMOS APIs
- OSK provides additional APIs for functions like file transfer and binary file management

Test Runner

- Test framework for organizing, executing, and verifying test scripts
- *Currently* OSK only includes some prototype scripts. The goal is to provide a complete test suite that can be extended by the user.

Main OSK Window



- Three tabs Explore cFS/SimSat, Manage Apps, and Extend OSK provide the top-level organization
- *Explore cFS/SimSat* allows the user to learn the cFS using SimSat
- *Manage Apps* provides tools for adding, removing, and creating apps
- *Extend OSK* is in its infancy, but it's goal is to allow the user to bridge the cFS to other systems and control remote devices



Start the Flight Software (FSW)



• Click <Start cFS> to run the FSW. <Start cFS/42> is used later.

- A new terminal window is created for the Linux process running the cFS
- Enter "osk" when prompted for a password.
- In a few seconds the time box should turn white time with advancing
 - If time doesn't advance click <Send Config Cmd> "ENA_TLM"

CFS_KIT CFS_KIT_SCREEN	⊜ ₪ ⊗
Open Sat Kit	
Explore cFS/SimSat App Developemnt Extend OSK	
System	
Start (FS Start (FS/42 Syste	em Time(secs) 1008351
Stop cFS Stop 42	
Config System Documentation	
Send Config Cmd ENA_TLM COSK Quick Sta	art OSK Users Guide
Scripts	
Run Ops Example Run Integration Test	
Applications	
Runtime Envr Data/File Mgmt	Autonomy
Attitude Det/Ctrl Health Safety	Maintenance
core Flight Executive (cFE)	
Event Service Executive Service	Software Bus
Table Service Time Service	cFE Users Guide
Flight Event Messages	

What Just Happened?

- OPEN SAT KIT
- The <Start cFS> button invoked a ruby script that created a new terminal window executing the "cFS Framework"
- The cFS Framework is the bottom two layers of the 3-tiered cFS architecture. It is a portable application runtime environment that uses a startup script (cfe_es_startup.scr) to determine which apps to load during initialization. OSK's startup script is configured for SimSat.



Core Flight Executive (cFE)

• The cFE has 5 services

- Executive Services (ES): Manage the embedded software system and create an application runtime environment
- Time Services (TIME): Manage spacecraft time
- Event Services (EVS): Provide a service for sending, filtering, and logging event messages (time stamped text messages).
- Software Bus (SB) Services: Provide an application publish/subscribe messaging service
- Table Services (TBL): Manage application binary file table images

Runtime Envr	Data/File Mgmt	Autonomy	One hutton/screet
ttitude Det/Ctrl	Health Safety	Maintenance	for each convice
			Tor each service
xecutive (CFE)			4
Event Service	Executive Service	Software Bus	
Table Service	Time Service	cFE Users Guide	*
lessages			cFE HTML User's Guide
tt E T	itude Det/Ctrl ecutive (cFE) vent Service Table Service	itude Det/Ctrl Health_Safety ecutive (cFE) vent Service Executive Service Table Service Time Service ssages	Untille Enviro Data/File Mgilt Automotity itude Det/Ctrl Health_Safety Maintenance ecutive (cFE)



cFE Service Screen (1 of 2)

- ĸıt
- Table Service screen shown. All cFE screens have the same layout but may not have every component/button



cFE Service Screen (2 of 2)



Simple Satellite (SimSat)



SimSat provides a reference mission to provide context to

- Illustrate what applications are required and how they are configured and integrated as a system to meet the requirements
- Demonstrate an example integration test script
- Demonstrate an operational script

• This does not include

- Porting SimSat to a new platform
- Integrating hardware devices

SimSat is a

- Low Earth Orbit (LEO) satellite with one nadir-pointing science instrument
- The instrument has
 - A detector that produces 10 bytes of data per second
 - A power the following sequence: Apply power, wait for instrument initialization (~20s), and command to enable science
 - The science team requires
 - A 1Hz auxiliary spacecraft data containing time, attitude, orbit data, and instrument status
 - Start science during a ground contact. Can be automated but ops prefers to monitor instrument health.
 - Ground contact resources/schedule are preplanned
 - Implies autonomous operations can be loaded on board using stored commands

FSW must autonomously monitor instrument health and power off the instrument in the event of a fault

SimSat Applications (1 of 3)



SimSat Applications (2 of 3)

 The previous slide shows a cFS "bubble" chart where each app is a bubble and they communicate via messages on the software bus.

- The blue cFS apps are reusable open source apps that are available on https://github.com/nasa/xx where 'xx' is the abbreviated app name
- The green OSK apps were written specifically for OSK
- The external COSMOS and 42 interfaces use UDP and TCP respectively

 Apps are designed to perform a dedicated function with clear interfaces and they operate in groups to achieve higher level mission objectives

Runtime Environment Apps

- Kit Command Ingest (KIT_CI) receives CCSDS command packets from COSMOS and sends them on the Software Bus
- Kit Telemetry Output (KIT_TO) reads CCSDS telemetry packets from the Software Bus and sends them to COSMOS
- **Kit Scheduler (KIT_SCH)** contains tables that define when to send messages on the Software Bus
 - Apps can use these messages to perform synchronous activities, e.g. sending their housekeeping status packet

SimSat Applications (2 of 3)

Data/File Management

- File Manager (FM) provides a ground interface for performing common directory and file operations
- Data Storage (DS) reads packets from the software bus and writes them to files according to table-defined
- Housekeeping (HK) creates new telemetry packets from pieces of other telemetry packets. The new packets are written to the SB and can be stored and/or telemetered.
- Trivial File Transfer Protocol (TFTP) transfers files between the flight and ground COSMOS. There's an open source CCSDS File Delivery Protocol (CFDP) app that will be added in a future release.

Autonomy

- Limit Checker (LC) monitors one or more telemetry values and start stored command relative time sequences (RTSs) in response to limit violations
- **Stored Command (SC)** Provides services to execute preloaded, table-defined command sequences at predetermined absolute or relative time intervals

SimSat Applications (3 of 3)



Attitude Determination and Control Apps

- 42 Interface (I42) manages a TCP/IP connection to 42 and transfers actuators/sensor packets to/from 42
- 42 FSW (F42) Implements the "ThreeAxisFsw" attitude control algorithm defined in 42

Maintenance

- Memory Dwell (MD) creates telemetry packets containing contents of memory location specified in dwell tables
- **Memory Manager (MM)** provides read/write access to memory

Health & Safety

- Checksum (CS) monitors checksums across table-defined static code/data regions and reports errors
- Health & Safety (HS) monitors table-defined application check-in and event messages and reporting errors and/or starting a RTS to address the issue

SimSAt Application Screens

Each functional application group screen uses the following layout



SimSat Integration Script



Runs test script using Script Runner

Issues Noop command to every application and verifies telemetry response

SimSat Operational Script

- Integration Scripts
- Operational Scripts



Configuration and Convention Notes

COSMOS Configuration (1 of 2)

COSMOS Target (OpenSatKit/cosmos/config/targets)

- Architectural component, typically on an embedded system, that COSMOS can send commands to and receive telemetry from
- For each target users can define command packets, telemetry packets, screens, and Ruby scripts.
- Each FSW application is defined as a target
- OSK defines a virtual target CFS_KIT to serve as the User's primary interface

OSK scripts in *OpenSatKit/cosmos/lib* extend COSMOS scripting API

- API documentation is under development. See code for details

COSMOS Configuration (2 of 2)

• OSK specific directories defined in *OpenSatKit/cosmos/cfs_kit*

- */docs*: cFE and OSK documentation
- /file_server: Default location for file transferred to/from FSW
 - */table* subdirectory contains table files
 - COSMOS Table Manager file formats defined in /cosmos/config/tools/TableManager
- /tools: cFE and OSK standalone tools
- /tutorials: Tutorial files

OPFN

Minor Inconveniences (1 of 2)

- OSK is a work in progress with a few known issues that you can ignore
- If you cancel an OSK dialogue you may see the follow COSMOS error dialogue.



- The FSW terminal window may display start and stop "FlyWheel" messages
 - OSK is a non-realtime environment so the cFE time service is warning that's it's not operating within its real-time precision limits relative to a 1Hz timer
 - OSK is designed to help users learn functional features and only requires reasonable timing performance in order for the scheduler to execute its schedule correctly

EVS Port1 42/1/CFE_TIME 20: Start FLYWHEEL EVS Port1 42/1/CFE_TIME 21: Stop FLYWHEEL OPFN

Minor Inconveniences (2 of 2)



 Some cFS binary files are variable length. The Table Manager definition files support fixed length files, therefore you may see an error dialog stating the file doesn't contain all of the records. This message is from cFE Executive Service Task Information file.

Table Open Error



Binary size of 1416 not large enough to fully represent table definition of length 3392. The remaining table definition (starting with byte 1416 in CFE_ES TASKINFO) will be filled with 0.



OSK Conventions



- Most cFE services have commands that can generate a telemetry as part of the response or write information to a file
 - The verbs *list* and *send* indicate information is sent in a telemetry packet.
 - Write is used when information is written to a file
- The FSW directory /cf (compact flash) is used as the default location for onboard file creation and flight-ground file transfers
 - This is mapped to OpenSatKit/cfs/build/exe/cpu1/cf
- OpenSatKit/cosmos/cfs_kit/file_server is used as the default ground file location
 - Table are located in the *tables* subdirectory
- OSK often uses osk_tmp_bin.dat as a standard temporary binary file name to avoid clutter
 - **OSK does not** "cheat" when working with ground and flight tables
 - Files are transferred between flight and ground locations and not accessed via shared locations within the VM

Running SimSat with 42

Needs 2.0 Updates

OpenSatKit – 2.0

Tools: Preparing 42 Simulation



- From the kit main page on the previous slide select <42 Simulator> and the screen to the left will appear.
- The 2nd row of buttons allow you to change the behavior of the control algorithms running in the FSW and are described on the next slides
- Before running the sim you will open some additional windows that will be used for your class exercise
 - Manage Control Table
 - Plot Attitude Errors

42 Sim: Manage Control Table



	F42 (Control	Table			
Get Current Val	ues Load	d Screen V	/alues	Rest	ore Defa	aults
D Cain Baramot						
	CI .	7				
VV						
/heel Target Mo	mentum Lin	nits				
Vheel Target Mo Lower	mentum Lin	nits Upp	er			
Vheel Target Mo	mentum Lin	nits Upp	er)
Vheel Target Mo	mentum Lin	nits Upp	er)
Vheel Target Mo Lower	mentum Lin	nits Upp	er	7		
Vheel Target Mo Lower Moment of Inerti	mentum Lin a	nits Upp	er	Z		
Vheel Target Mo Lower Moment of Inerti	mentum Lin a Y	nits Upp	er	Z		
Vheel Target Mo Lower Moment of Inerti X	a Y	nits Upp	er	Z)
Vheel Target Mo Lower Moment of Inerti X	a X-Axis	nits Upp	er	Z	-Axis	
Vheel Target Mo Lower Moment of Inerti X Control Gains Kr	a X-Axis	nits Upp	er Axis 0.087	Z 2 920	-Axis 0.09	98000

OpenSatKit – 2.0

- Selecting *<Manage Control Table>* on the 42 Sim screen produces the screen to the left.
- Select <*Get Current Values*> and it will populate the screen with the current control table values. This takes a little time because it is transferring a file from flight to ground
- Edit the screen as desired and click <Load Screen Values> to replace the current control table values
- The defaults can be restored by clicking <*Restore Defaults*>

🖲 🗊 F42 TBL_	SCR				
	F42 C	ontro	l Table		
Get Current Va	lues Load S	Screen	Values	R	estore Defaults
D Gain Parame	ter	_			
W		Z			
0.628		0.7			
Lower -0.9		Up 0.9	рег		
Noment of Inert	ia				
x	Y			Ζ	
0.119835	0.1477	78		0.04	4908
Control Gains					
	X-Axis	Y	-Axis		Z-Axis
Кг	0.1053	59	0.129	9928	0.03948
	0.0470				

42 Sim: Plot Attitude Errors

• Selecting <Plot> button next to the attitude errors produces the screen below



42 Sim: Starting the Simulation



Run 42 Sim	Perco			
Run 42 Sim	Peco			
	Kecol	nnect 42	Disco	onnect 42
Set Whl Tgt Morr	n Manag	e Ctrl Tbl	Config	g SunValid
42		F42		
Cmd Valid	0	Cmd Val	id	C
Cmd Error	0	Cmd Err	ог	C
42 Connected	FALSE	Control	Exec Cnt	C
42 Cycles	0	Sun Vali	d	
Sensor Pkts	0	OVR Sur	n Valid	SE 42 SIN
Actuator Pkts	0			
ttitude Control				
Att Err X	Att Err Y	Att Er	r Z	
0.000000	0.0000	00 0	0.000000	Plot
Wheel 1 Cmd	Wheel 2 Cm	d Wheel	3 Cmd	-1.4
0.000000	0.0000	00 0	.000000	Plot
Aomentum Contro	ι			
Mom Err X	Mom Err Y	Mom	Err Z	
0.000000	0.0000	00 0	0.000000	Plot
MTB 1 Cmd	MTB 2 Cmd	MTB 3	Cmd	
0.000000	0.0000	00 0	0.000000	Plot
Whl 1 Tgt Mom	Whl 2 Tgt Mor	m Whl 3 To	gt Mom	
	0 0000	00 0	0.000000	Plot

- Select <*Run 42 Sim*> which will start the 42 simulator in a new terminal window.
- The 42 configuration files used in the simulation are located in directory *OpenSatKit/42/OSK*
- The simulation takes a while to initialize

42 Sim: Additional Configuration Options KIT

- The kit includes two additional configuration options that can be manipulated
 - **1.** Wheel target Momentum
 - 2. Sun Valid Configuration

34

OPEN

42 Sim: Set Wheel Target Momentum

	Set Wheel Target Momentum Command						
Current Tar	get Momentum						
	Wheel 1	Wheel 2	Wheel 3				
	0.000000	0.000000	0.000000				
Command T	arget Momentu	n					
Wheel 1	N	/heel 2	Wheel 3				
-							

- The controller allows a non-zero (default) momentum to be stored in the wheels
- Enter new values and click <*Send*> to change the values
- The plot below shows a jump in momentum errors when new targets were selected

Sellu	Cancer		
	🏹 🖨 🕼 File Tab Plot Data Object F	Help	
	Running Start Pause Stop Seconds Plotted: 100.00 Points Saved: 1000000 Points Plotted: 1000 Refresh Rate Hz: 10.0 Data Objects: F42 CONTROL_PKT MOM_ERR_X F42 CONTROL_PKT MOM_ERR_Y	Add Housekeeping Data Object:	04:10:10.914
	F42 CONTROL_PKT MOM_ERR_Z	F42 CONTROL_PKT MOM_ERR_Y F42 CONTROL_PKT MOM_ERR_Z	

Cancol

OpenSatKit – 2.0

Cond

42 Sim: Configure SunValid

- Selecting <*Config SunValid*> to override the current sun valid flag
- The plot below shows gimbal command
 - The linear portion had a valid sun and the bend occurred when the SunValid was overridden to false.



OpenSatKit – 2.0

OPFN

42 Sim: Termination



- 1. Click *<Disconnect* 42*>* to end a 42 simulation that is running with the FSW
- 2. To terminate the flight software click on the terminal window with the FSW messages and then enter ctrl-c
- 3. Each of the cosmos windows will need to be closed individually. If you close the COSMOS TlmViewer window first it prompt you to close all of the telemetry screens at once.



Manage Applications

Needs 2.0 Updates

OpenSatKit – 2.0

Tools: Create Application



• Seven quick steps and a new app is created and integrated into the kit

Tools: Create Application



• Follow the instructions in the center of the dialogue. Create app generates the fsw source/make files, the cosmos target, and edits the COSMOS cmd-tlm-server config file.

 <Install App> has not been implemented. Follow the instructions on the previous slide

OpenSatKit - 2.0



Kit App Integration



Goal is to provide easy access to COSMOS, KIT_TO, and KIT_SCH to integrate a new app



Extending OSK

OpenSatKit – 2.0

Tools: Benchmarks

Coming Soon...



Tools: Performance Monitor



- Capture FSW performance data using screen
- Download file and <Launch Analysis Tool>



Tools: PiSat Control



This requires a PiSat which is currently not in the public domain



Demos

OpenSatKit – 2.0





• Each demo follows a common user screen configuration



Demo Structure – FM Example (2 of 2)

<More Info> provides

information

detailed context-specific



۲

🕽 🗇 🗊 CFS_KIT FILE_MGMT_DEMO_SCREEN

File Management Demo

Create a new directory. After the directory is created, FM's SEND_DIR_PKT is sent to display the latest /cf directory contents. The new directory appears as the first file in the directory listing.

<Demo> Send FM's CREATE_DIR to create a new directory /cf/aatmp

	More Info	Demo	Next ->
File Management Demo	ight vent Messages		
The SEND_DIR_PKT command takes an offset argument that specifies the starting index into the directory listing. An offset of 0 is used through out this demo. FM's WRITE_DIR_TO_FILE cmd can be used to write an entire directory listign to a file. Application command execution counters typically mean a command has been successfully processed. However there are often situations when a command may take a while to process and the activity canoccur in the background. In these situations a child task performs the function and its commandexecution counters (pass/fail) indicate whether the command was completed sucessfully. The parentapplication's execution counter simply means the command was successfully/unsucessfully parsed and passed to the shild task.			
Press for More Information			V

Application Functional Screens

OPEN SAT

File Management



🐻 🖲 🗉 СЕЅ_КІТ ЕІІ	_E_MGMT_SCREE	N					
File Management							
Directory Manageme	ent	File Manager Directory Listing					
Create	Delete	DIRNAME:					
List to Packet	Write to File	TOTALFILES:	0				
File Management		PACKETFILES:	0				
Сору	Move	FIRSTFILE:	0				
Rename	Decompress	FILE01_NAME:					
Delete	Delete All	FILE02_NAME:					
Concat	Get Info	FILE03_NAME:					
List Open		FILE04_NAME:					
File Manager House	eeping	FILE05_NAME:					
Cmd Valid Cnt	0	FILE06_NAME:					
Cmd Error Cnt	0	FILE07_NAME:					
Child Cmd Valid Cn	t 0	FILE08 NAME:					
Child Cmd Error Cn	t 0	- FILE10 NAME:					
		- FILE11 NAME:					
		FILE12 NAME:					
File Transfer		_					
Put I	File	Get F	ile				
PUT_FILE_COUNT:	0	GET_FILE_COUNT:	0				
Ground Working Di	rectory						
Flight Working Dire	ectory						
Event Messages							

- <List to Packet> commands File Manage (FM)
 - To send a directory listing
 - The command uses a directory listing alphabetical "offset" to determine which file to start with in the listing
- OSK uses the verbs *list* and *send* to indicate information is sent in a telemetry packet.
- Write is used when information is written to a file

- <List to Packet> commands File Manage (FM)
 - To send a directory listing
 - The command uses a directory listing alphabetical "offset" to determine which file to start with in the listing

Table Management



CFS_KIT TABLE_MGMT_SCREE	N		
Table	e Management		Load a new FSW table
Table Management			< <i>Put File</i> > transfers file from ground to flight
Load Table	Validate	Activate	<load table=""> into table buffer</load>
Abort Load	Dump Table	Display Table	<i>Validate</i> > table via app validation function
Table Registry	Table Registry L	sting 🗸	<i><activate></activate></i> new table
Display Registry Write Registry to File	e NAME:		
Table Manager Housekeeping Cmd Valid Cnt Cmd Error Cnt Last Updated Table Last File Loaded	SIZE: CRITICAL: TABLE_LOADE LOAD_PENDIN DUMP_ONLY:	D_ONCE: 0 G: 0	• < <i>Display Registry</i> > sends a table's registry information in a telemetry packet
Last Table Loaded	DBL_BUFFERE LAST_UPD_TIN FILE_CREATE_T LAST_FILE_LO/ OWNER_APP_	D: 0 IE_SECONDS: 0 TIME_SECS: 0 ADED: 0 NAME: 0	• Dump and display FSW table <dump table=""> to onboard file <get file=""> transfers file from flight to ground <display table=""> launches COSMOS Table</display></get></dump>
File Transfer			Manager to view file. Requires binary file
Put File		Get File	definition.

OpenSatKit – 2.0

	Put File		Get File	
	PUT_FILE_COUNT:	0	GET_FILE_COUNT:	0
	Ground Working Directory			
]
	Flight Working Directory			
Fli	ght Event Messages			

Memory Management



🖢 💿 CFS_KIT MEMORY_MGMT_SCREEN

_							
	Memory N	lanagement					
Memory Manager							
Lookup Symbol	Peek Address	Poke Address	Dump to Event				
Fill Block	Load from File	Dump to File	Manage Checksums				
Memory Dwell		Memory Manager S	tatus				
Start Dwell	Stop Dwell	Cmd Valid Cnt	0				
Jam Dwell Tbl	Dwell Tbl 1 Pkt	Cmd Error Cnt	0				
Memory Dwell House	keeping	Last Action	NONE				
Cmd Valid Cpt		Mem Type	0				
Cmd Error Cot		Address	0000000				
Enable Mask	0000	Fill Pattern	0000000				
	0000	Bytes Processed	0				
Last Memory Manager	File						
File Transfer							
Pul	t File	Get File					
PUT_FILE_COUNT:	C	GET_FILE_COUNT:	0				
Ground Working Dir	ectory						
Flight Working Directory							
Elight Event Messages							
Fught Event Messages							

- Memory Manager (MM) and Memory Dwell (MD) apps are typically used for inflight maintenance.
- MM commands allow direct access to any memory location
- MD generates telemetry packets that contain the contents of table-specified memory locations
 - Only 1 dwell table telemetry packet is defined
 - <*Jam Dwell Table*> allows the dwell table to be loaded without using the table load service
- The FSW can easily be corrupted using memory manager
- The memory management demo is a good place to start since it demonstrates MM and MD using safe memory locations

Recorder Management

•		RDER	_MGMT_SCRE	EN							
Recorder Management											
Da	ita Storage App Statu	5									
	Enable/Disable		Dest File	Dest File 58 Info							
	Cmd Valid Cnt		0 Cmd Error Cnt		0 State			0			
50											
Enable/Disable Sequence Count Eilonamo Tuno											
	Enable/Disable		Sequen		Filename Type						
	File Path Name		File Bas	se Name	File Extension			_			
	Max File Size		Max F	lle Age		LIOSE 1/F	All Files				
	Tbl Load Count		0	Tbl Access Err	Cnt			0			
	File Write Valid Cnt		0	File Write Inva			0				
	Hdr Update Valid Cnt		0	Hdr Update Invalid Cnt				0			
				, .				Ռո			
Set Packet Filter Configuration											
	Dest File	Ad	d Message	Algorithm		Fil	ter Type				
	Tbl Load Cnt		0	Tbl Access Err			0				
	Pkt Discard Cnt		0	Pkt Ignored Cr			0				
	Pkt Filtered Cnt	0		Pkt Stored Cnt			0				
	Packet Filter File										
Fi	File Transfer										
	Put File			Get File							
	PUT_FILE_COUNT:		0	GET_FILE_COUNT:				0			
Ground Working Directory											
Flight Working Directory											
el:	abt Event Messages										
FU	ght Event Messages										

OPEN SAT

KI.

Autonomy Management

			,	lanag	enne	ant			
tored Command	(SC) App	- Relati	ive Tin	ne Seq	ueno	:es(R	TS)		
Start RTS		Stop R	TS	E	nabl	e RTS	5	Dis	able RTS
Start Group		Stop Gr	oup	En	able	Grou	ıp	Disa	ble Group
Cmd Valid Cnt			0	Cmd	Егго	r Cnt			0
TS Status									
	RTS	64	49 48	833	32	. 17	16	. 1	
EX	ECUTING	00	00	0000	0	000	0	000	
DI	SABLED	00	00	0000	0	000	0	000	
Start Cnt	0000	Start B	Err Cnt		0000	N C	ext Ti	ime	000000
Active Cnt	0000	Next F	RTS Nu	m	0000) RT	'S CM	D Cnt	000000
CMD Err Cnt	0000	Err RT	S#		0000) Er	r RTS	Offsel	0000
mit Checker(LC)	Арр								
Reset WP S	Stats	Reset	AP Sta	its !	Set A	P Sta	te	Set Al	P Prem Off
Reset WP S	stats ate	Reset App	AP Sta o State		Set A	P Sta	ote 0	Set Al	P Prem Off
Reset WP S Set App St Cmd Valid Cnt	ate	Reset App	AP State	ots :	Set A	P Sta	nt	Set Al	P Prem Off
Reset WP S Set App St Cmd Valid Cnt Watch Points(V	stats sate VP) Actio	Reset App on Poin	AP State	o Cm Statu	Set A nd Err	P Sta	nt	Set Al	P Prem Off 0
Reset WP S Set App St Cmd Valid Cnt Watch Points(V Watch Point	stats sate VP) Actions (2-bits p	Reset App on Point per WP)	AP State	o Cm Statu 0 0	Set A nd Err s	P Sta	nt 0 0	Set Al	P Prem Off
Reset WP S Set App St Cmd Valid Cnt Watch Points(V Watch Point Action Point	stats ate VP) Actio s (2-bits p	Reset App on Point oer WP) er AP)	AP State State ts(AP) 0 0 0 0 0 0	o Cm Statu 0 0 0 0 0 0	Set A nd Err s 0 0	P Sta	ate 0 0 0	Set Al	P Prem Off
Reset WP S Set App St Cmd Valid Cnt Watch Points(V Watch Point Action Point	stats sate VP) Actio s (2-bits p	Reset App on Poin ber WP) er AP)	AP State State ts(AP) 0 0 0 0 0 0 0 0 0 0	Its ! 0 Cm Statu 0 0 0 0 0 0 0 0 0	Set A nd Err s 0 0 0	P Star	ate 0 nt 0 0 0	Set Al	P Prem Off
Reset WP S Set App St Cmd Valid Cnt Watch Points(V Watch Point Action Point PASS RTS EXE	Stats State VP) Actions s (2-bits point c (4-bits point E Cnt	Reset App on Poin ber WP) er AP)	AP State State ts(AP) 0 0 0 0 0 0 0 0 0 0 0 0	its ! 0 Cm Statu 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Set A nd Err s 0 0 0 0 XE C	P Sta ror C 0 0 0 0 0 0 0 0 0 0	ate 0 0 0 0 0 0	Set Al	P Prem Off
Reset WP S Set App St Cmd Valid Cnt Watch Points(V Watch Point Action Point PASS RTS EXE WPs in Use	Stats Stats State	Reset App on Poin ber WP) er AP)	AP State State ts(AP) 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ets : 0 Cm 5 Cm 5 Cm 0 Cm 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Set A nd Err 0 0 0 XE C	P Sta ror C 0 0 0 0 0 0 0 0 0 0 0 0 0 0	o 0 0 0 0 0 0 0 0 0 0 0 0 0	Set Al	P Prem Off

🕽 🖨 🗉 CFS_KIT AUTONOMY_MGMT_SCREEN



Application Management

	App Mai	nagement			
	Executive Service S	itatus			
App Summary	Cmd Ctr	0	Cmd Err Ctr	0	
App/Task Registry	Registered Apps	0 F	egistered Tasks	0	
Enable App Events					
Disable App Events	App Info				
Add KIT_TO Msg	Name		Entry Point		
Start App	Main Task Name	0	Main Task ID	0	
Stop App	Туре	0	# Child Tasks	0	
Reload App	File Name			0	
Get App Info	Code Size	0	Data Size	0	
Create App Tool	BSS Size	0	Stack Size	0	
e Transfer	rile	1	Cabrila		
PUT_FILE_COUNT:		GET_FILE_COU			
	ory				
Flight Working Director	У			ſ	
ght Event Messages					

• *<Get App Info>* commands cFE executive services to send a telemetry packet with the command-specified app

 <*App/Task Registry*> commands cFE executive services to write app or task information to a file that can be transferred to ground via a <*Get File*>



COSMOS Extras

OpenSatKit – 2.0



Utilities OpenSatKit – 2.0 **Offline Analysis Tools**