OpenSatKit (OSK) Introduction for cFS Training Exercises

September 19, 2019

Introduction

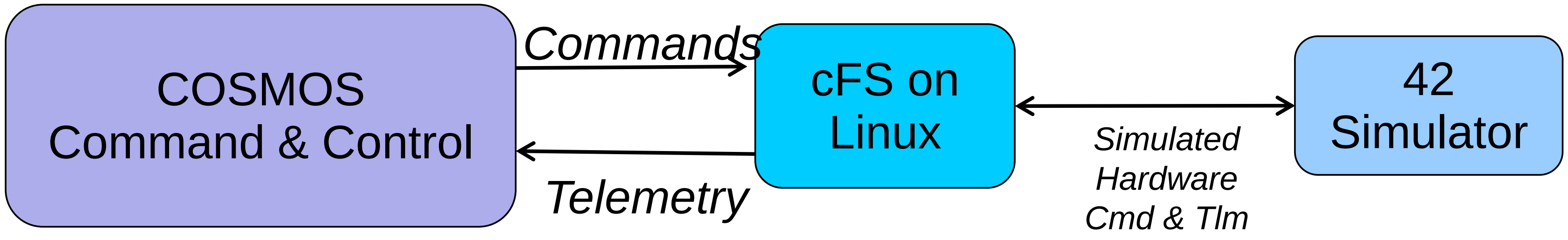
- **Provide an OpenSatKit overview for running the cFS training exercises**
  - Provides a minimal amount of information so you can do the training exercises
  - See OSK ***Quick Start Guide*** and ***User's Guide*** for more comprehensive information
- **Training Exercises**
  - Use a small subset of OSK's functionality so the overview covers enough to do the exercises
  - Use a combination of menus and scripts
  - Scripts are written in Ruby but you do not need to know Ruby to do the exercises

OpenSatKit 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

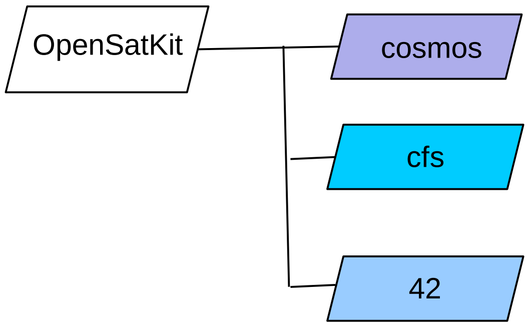
- In addition to the cFS itself, OSK uses two additional open source projects
  - Ball Aerospace's COSMOS command and control platform for embedded systems
  - NASA Goddard's 42 dynamic simulator

- OSK is preconfigured for a reference mission called Simple Satellite (SimSat)
  - See OSK Training – Simple-Sat.pptx

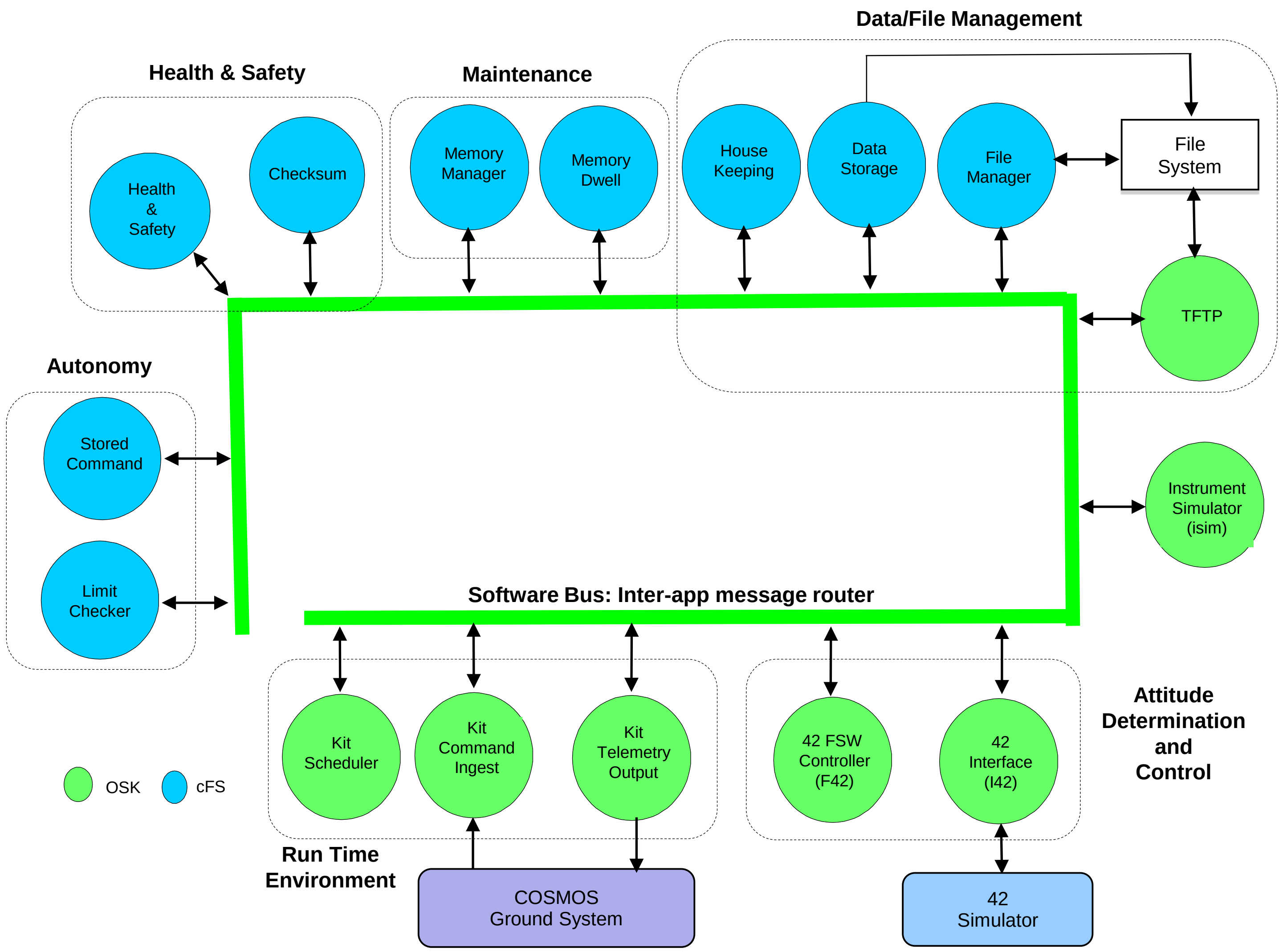


OpenSatKit Directory Structure

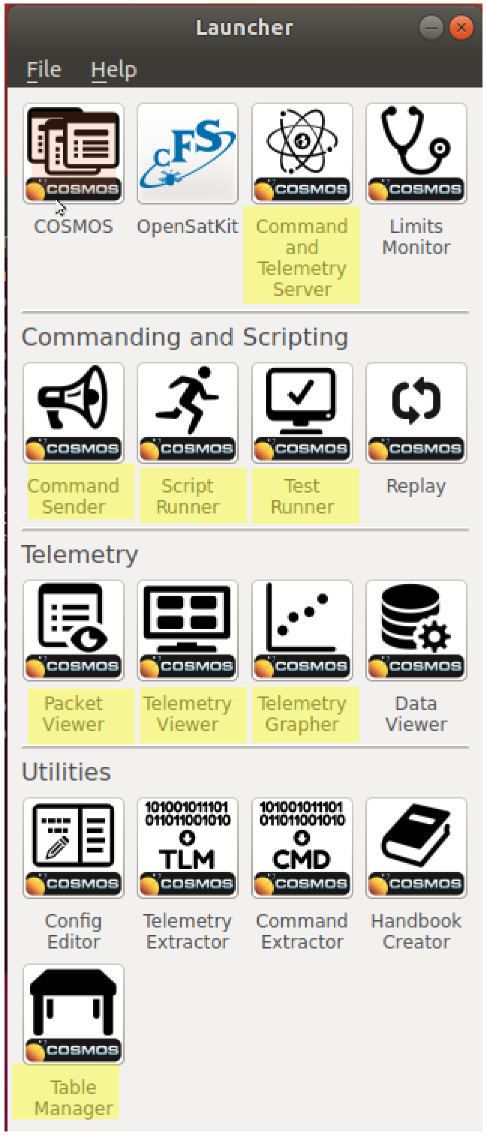
- Each open source project is contained in its own OpenSatKit subdirectory



OSK SimSat FSW Applications



OSK COSMOS Usage



- COSMOS is a collection of tools rather than a console-based ground system like ITOS and ASIST
- Shaded tool titles indicate the COSMOS tools used by OSK
- The “cFS Starter Kit” is a collection of screens and scripts that creates a cFS operational environment

COSMOS Tools (1 of 3)

- **Launcher**
  - 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.

COSMOS Tools (2 of 3)

- **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*

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

COSMOS Tools (3 of 3)

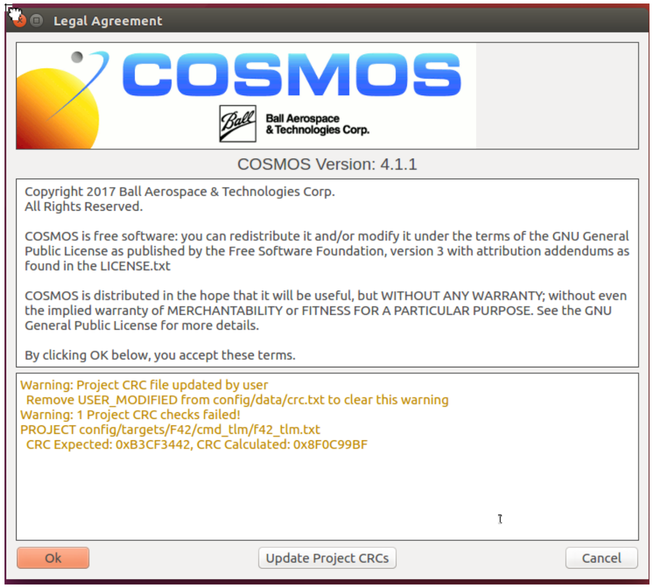
- **Table Manager**
  - Edit and display binary files
  - *OSK provides definitions for most of the cFE binary files and a limited number of cFS application binary files*

- **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.*

Starting OSK

1. **Open a terminal window (Ctrl-Alt-t)**
2. **Change directory to cosmos**
   - [~] cd OpenSatKit/cosmos
3. **Start COSMOS**
   - [~/OpenSatKit/cosmos]ruby Launcher
   - You'll see a screen similar to below. Select <OK>



Legal Agreement

COSMOS Version: 4.1.1

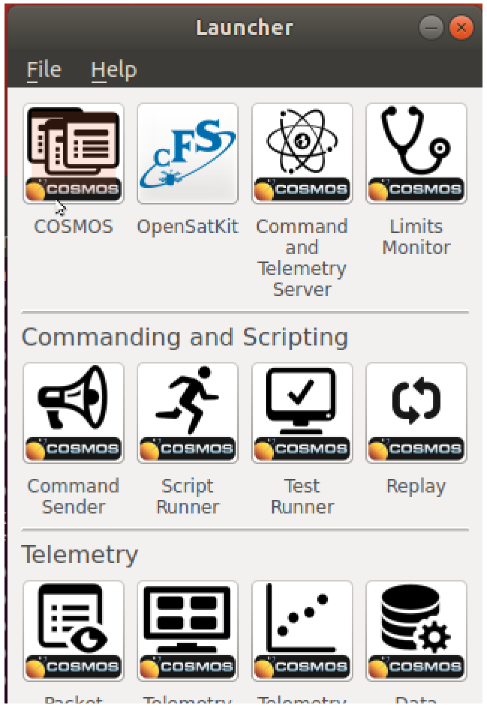
Copyright 2017 Ball Aerospace & Technologies Corp.
All Rights Reserved.

COSMOS is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 3 with attribution addendums as found in the LICENSE.txt

COSMOS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

By clicking OK below, you accept these terms.

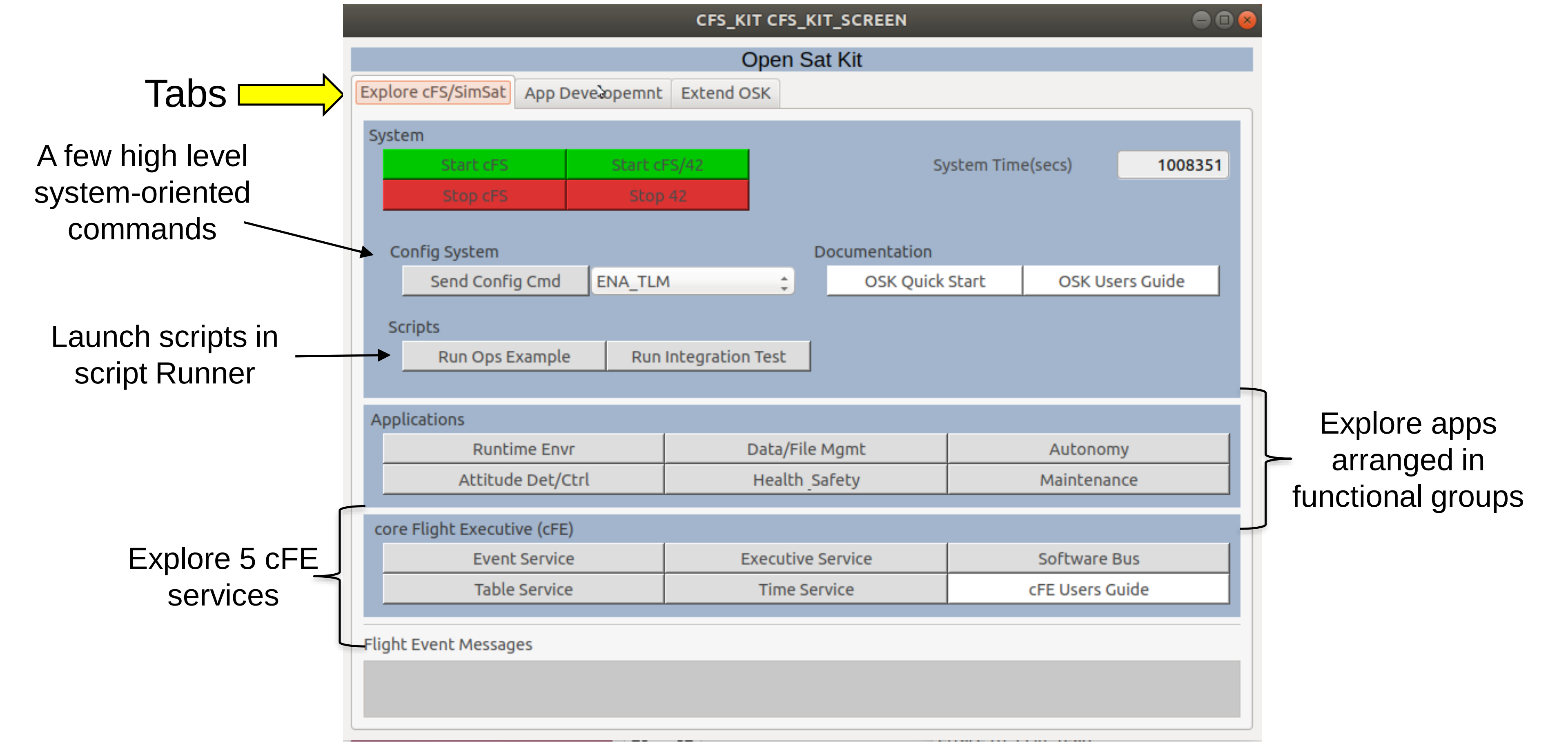
```
Warning: Project CRC file updated by user
 Remove USER_MODIFIED from config/data/crc.txt to clear this warning
Warning: 1 Project CRC checks failed!
PROJECT config/targets/F42/cmd_tlm/F42_tlm.txt
 CRC Expected: 0xB3CF3442, CRC Calculated: 0x8F0C99BF
```



4. **Select <OpenSatKit> with a single click**
   - This launches COSMOS's Command and Telemetry Server, Telemetry Viewer, the OSK main screen
   - You can minimize the COSMOS tools, but don't close them

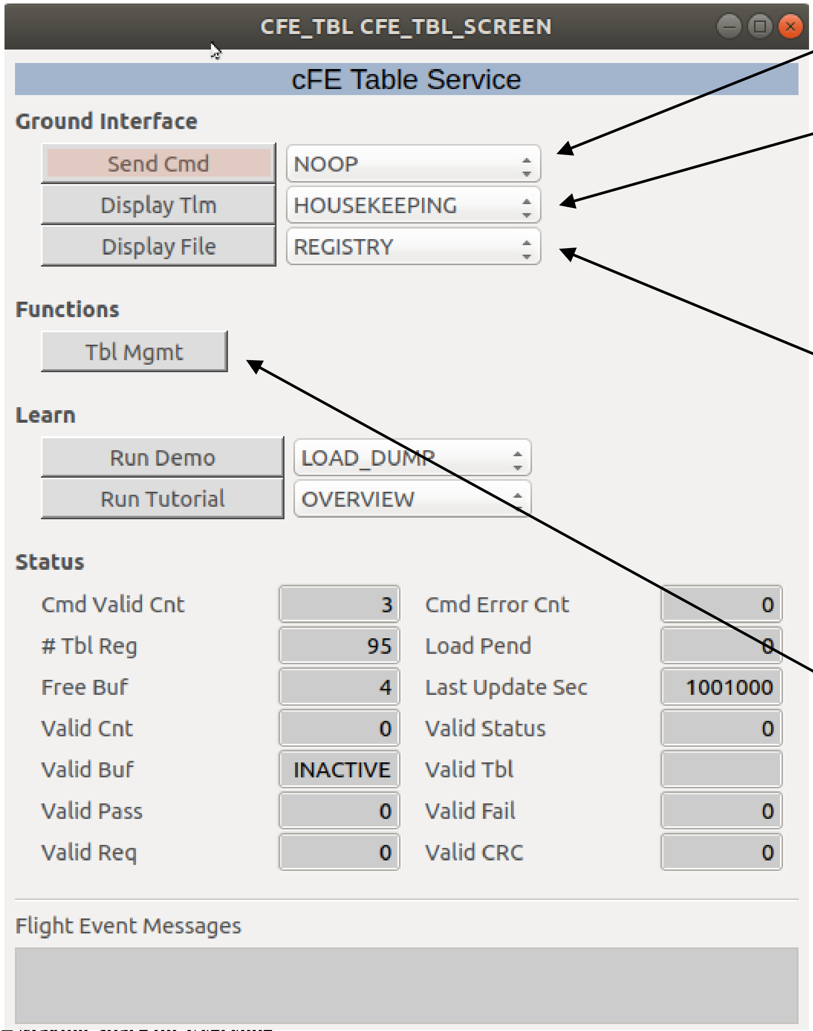
OSK Home Page

- 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



cFE Service Screen

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



CFE_TBL CFE_TBL_SCREEN

cFE Table Service

Ground Interface

Send Cmd	NOOP
Display Tlm	HOUSEKEEPING
Display File	REGISTRY

Functions

Tbl Mgmt

Learn

Run Demo	LOAD_DUMP
Run Tutorial	OVERVIEW

Status

Cmd Valid Cnt	3	Cmd Error Cnt	0
# Tbl Reg	95	Load Pend	0
Free Buf	4	Last Update Sec	1001000
Valid Cnt	0	Valid Status	0
Valid Buf	INACTIVE	Valid Tbl	
Valid Pass	0	Valid Fail	0
Valid Req	0	Valid CRC	0

Flight Event Messages

Select and send commands

Display a telemetry packet using COSMOS's Packet Viewer.

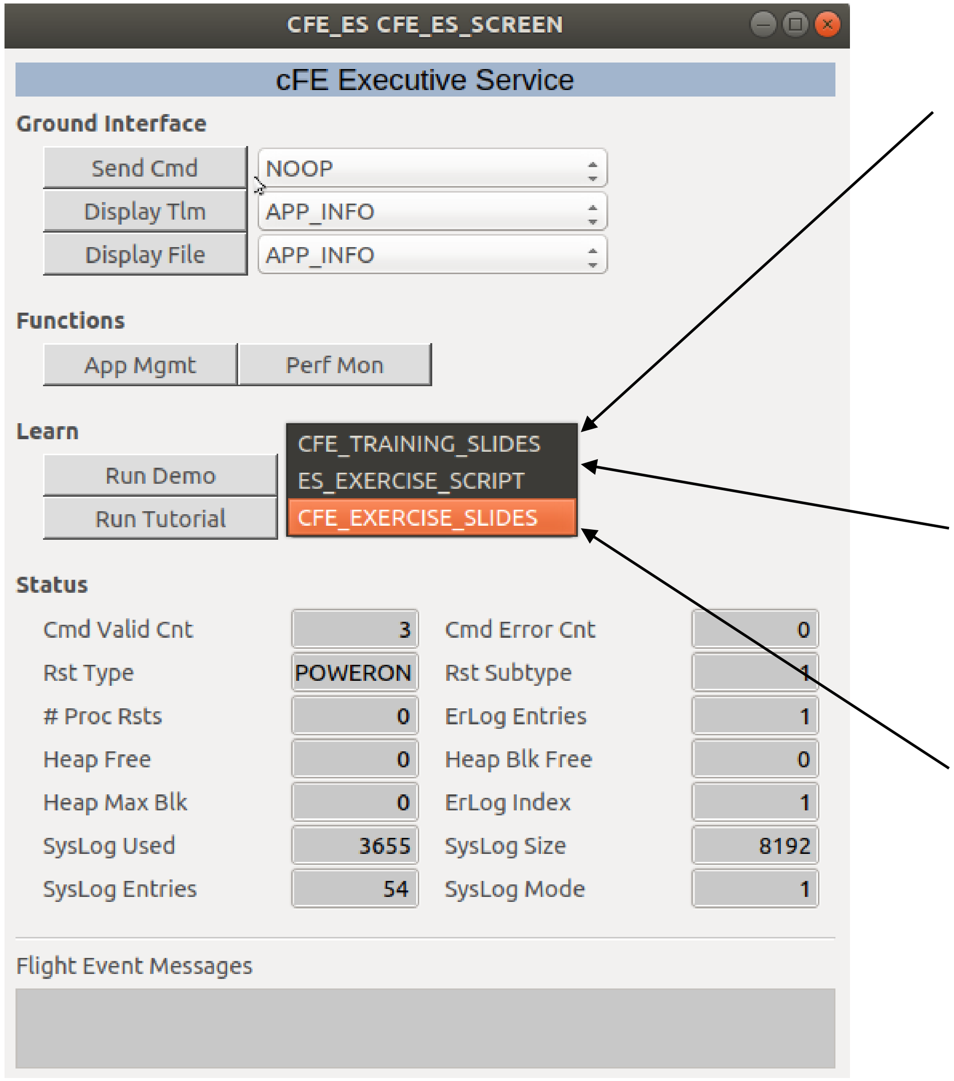
- Telemetry packets can be generated in response to a command
  - E.g. Telemeter the registration information for a single table

Display a binary file using COSMOS's Table Manager

- Binary files can be generated in response to a command.
  - E.g. Dump the entire table registry to a file

Display a screen that simplifies user interaction with a service

cFE Service Tutorials



CFE_ES CFE_ES_SCREEN

cFE Executive Service

Ground Interface

Send Cmd	NOOP
Display Tlm	APP_INFO
Display File	APP_INFO

Functions

Status

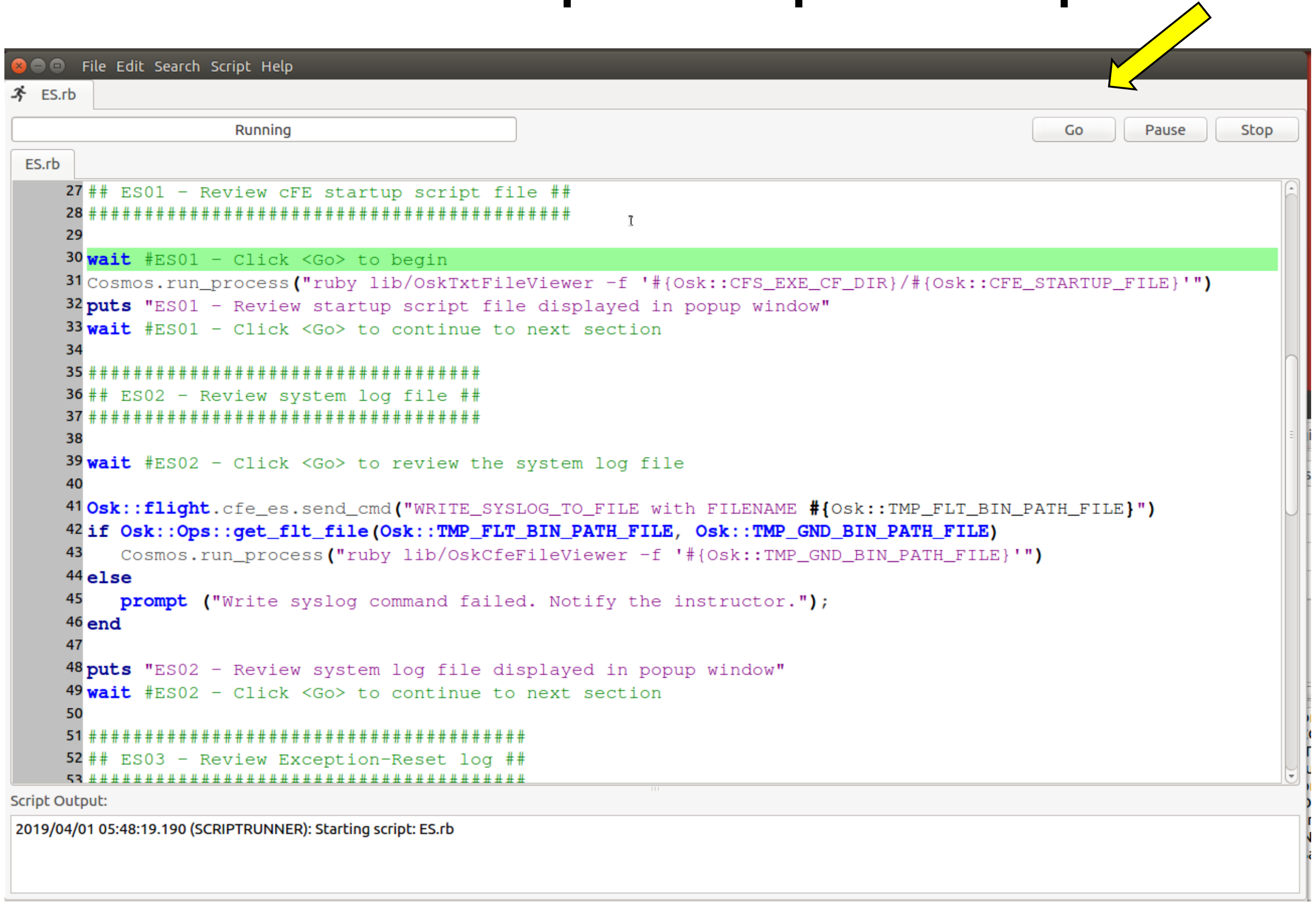
Cmd Valid Cnt	3	Cmd Error Cnt	0
Rst Type	POWERON	Rst Subtype	1
# Proc Rsts	0	ErLog Entries	1
Heap Free	0	Heap Blk Free	0
Heap Max Blk	0	ErLog Index	1
SysLog Used	3655	SysLog Size	8192
SysLog Entries	54	SysLog Mode	1

Flight Event Messages

- **cFE service tutorials use publically available cFE training slides**
- **The slides bundle all of the services so the same slides are launched from each service page**
- **The exercise scripts are unique to each service and compliment the cFE Exercise Slides**
- **The cFE Exercise bundle all of the services so the same slides are launched from each service page**

Tutorial Scripting

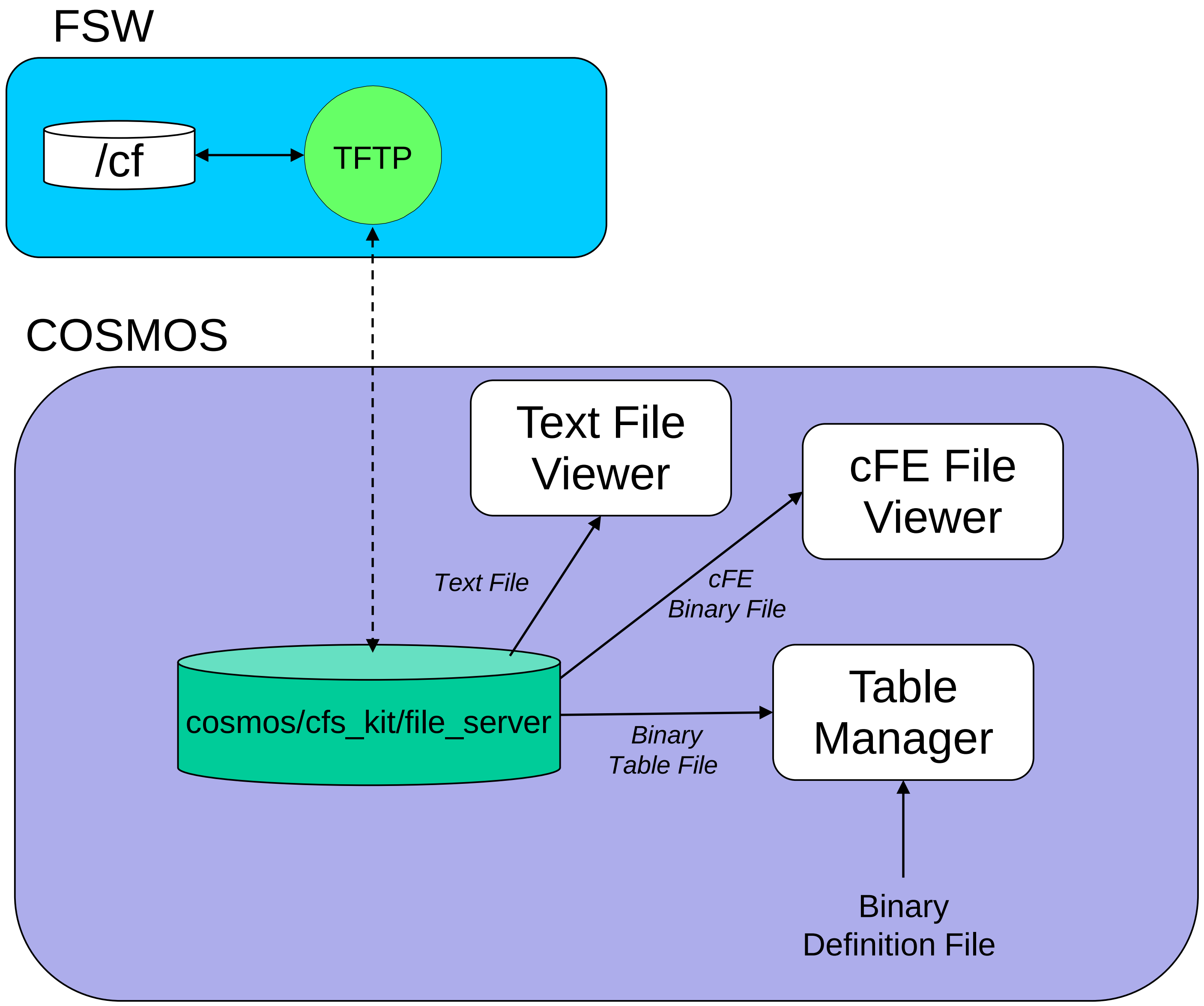
- **Each tutorial launches COSMOS Script Runner to run the exercise script**
- **Exercises are numbered**
- ***Wait* state statements suspend scripts and require *Go* to resume**



```
## ES01 - Review cFE startup script file ##
###########################################

wait #ES01 - Click <Go> to begin
Cosmos.run_process("ruby lib/OskTxtFileViewer -f '#{Osk::CFS_EXE_CF_DIR}/#{Osk::CFE_STARTUP_FILE}'")
puts "ES01 - Review startup script file displayed in popup window"
wait #ES01 - Click <Go> to continue to next section

###################################
## ES02 - Review system log file ##
###################################

wait #ES02 - Click <Go> to review the system log file

Osk::flight.cfe_es.send_cmd("WRITE_SYSLOG_TO_FILE with FILENAME #{Osk::TMP_FLT_BIN_PATH_FILE}")
if Osk::Ops::get_flt_file(Osk::TMP_FLT_BIN_PATH_FILE, Osk::TMP_GND_BIN_PATH_FILE)
   Cosmos.run_process("ruby lib/OskCfeFileViewer -f '#{Osk::TMP_GND_BIN_PATH_FILE}'")
else
   prompt ("Write syslog command failed. Notify the instructor.");
end

puts "ES02 - Review system log file displayed in popup window"
wait #ES02 - Click <Go> to continue to next section

#######################################
## ES03 - Review Exception-Reset log ##
#######################################
```

```
2019/04/01 05:48:19.190 (SCRIPTRUNNER): Starting script: ES.rb
```

File Management



Exercises

Executive Service Exercises

- **ES01 – Review the cfe_es-startup.scr file**
  - Note Memory Dwell (MD) priority and stack size
- **ES02 – Review the system log file**
  - File size limited by CFE_ES_SYTEM_LOG_SIZE defined in cfs/osk_defs/cpu1_platform_cfg.h
- **ES03 – Review the Exception-Reset log file**
  - Uses Table Manager to view the log
- **ES04 – Review the Critical Data Store Registry**
  - Uses Table Manager to view the registry
- **ES05 – Run the App Management Demo**
  - Runs built in demo
- **ES06 – Run the Performance Analyzer Demo**

Performance Monitor Demo Scenario

- **Log three markers**
  - 26: Memory Dwell execution
  - 44: File Manager Application
    - Pends for ground command and responds to housekeeping telemetry requests
  - 26: File Manager Child Task
    - Implements FM directory commands
- **Configure Memory Dwell as the trigger**
  - Memory Dwell configured to execute at 1Hz
- **Data collection scenario**
  - Start data collection
  - Wait 4 seconds
  - Issue FM command to send a directory in a telemetry packet
  - Wait 4 seconds
  - Issue FM command to write a directory to a file
  - Wait 4 seconds
  - Issue FM command to send a directory in a telemetry packet
  - Wait 4 seconds
  - Stop data collection

Time Service Exercises

- **TIME01 – Review default time configuration using "cFE Service" screen**
  - Information comes from housekeeping packet
  - Send commands to reconfigure time
- **TIME02 – Review diagnostic telemetry packet**
- **TIME03 – Demonstrate 1Hz adjustment**
  - Hokey demo that plots an incrementing 1Hz STCF adjustment

Event Service Exercises

- **EVS01 - Browse Event Log**
  - Captured startup messages, note SB no subscriber message

- **EVS02 - Browse Event Application Registry/Status**
  - `“EVS Port1 42/1/CFE_SB 14: No subscribers for MsgId 0x808, sender xxx”`
  - `APP1_NAME: CFE_SB`
  - `APP1_ENA_BITMASK: 0x0E   # (3..0) => (Critical, Error, Info, Debug)`
  - `APP1_FLTR1_EVENT_ID: 14  # CFE_SB_SEND_NO_SUBS_EID defined in cfe_sb_events.h`
  - `APP1_FLTR1_BITMASK: 0xFFFC  # Send 4 then stop`
  - `APP1_FLTR1_COUNT: xx    # What does this value tell you?`

- **Configure type**
  - Enable/disable EVS debug messages and notice response

Software Bus Exercises

- **SB01 - Review Pipe Definition File**
  - Should be no overflow errors
  - Pipe IDs helpful if you ever need to issue route enable/disable commands
- **SB02 – Request SB Statistics telemetry packet**
  - Peak in use can help tune pipe depths

Table Service Exercises

- **TBL01 - Browse Existing Table Registry**
  - Memory Dwell (MD) app is used for the load/dump exercises
  - ENTRY7: Memory Dwell Table #1
- **TBL02 – Jam MD table #1**
  - MD table defines locations to be telemetered
- **TBL03 – Load/Dump Tables**
  - Try various load/dump scenarios

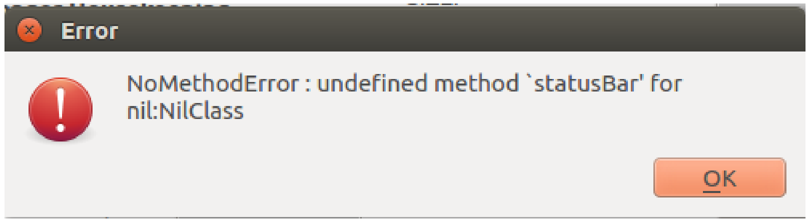
Additional OSK Material

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

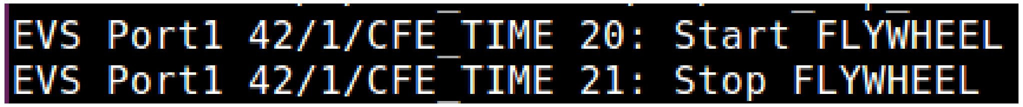
Minor Inconveniences (1 of 2)

- **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.**



NoMethodError : undefined method `statusBar' for nil:NilClass

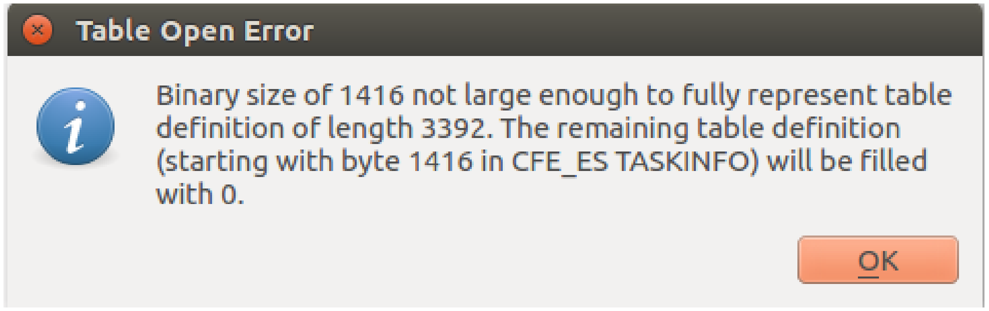
- **The FSW terminal window may display start and stop "FlyWheel" messages**
  - OSK is a non-realtime environment so the cFE time service is warning that's it's not operating within its real-time precision limits relative to a 1Hz timer
  - OSK is designed to help users learn functional features and 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
```

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.**



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.