# A Cloud-Based, Open-Source, Command-and-Control Software Paradigm for Space Situational Awareness

## **Ryan Melton and Jason Thomas**

Ball Aerospace, 1600 Commerce St, Boulder, CO 80301

## 1. ABSTRACT

With the rapid growth in the number of space actors, there has been a marked increase in the complexity and diversity of software systems utilized to support SSA target tracking, indication, warning, and collision avoidance. Historically, most SSA software has been constructed with "closed" proprietary code, which limits interoperability, inhibits the code transparency that some SSA customers need to develop domain expertise, and prevents the rapid injection of innovative concepts into these systems. Open-source aerospace software, a rapidly emerging, alternative trend in code development, is based on open collaboration, which has the potential to bring greater transparency, interoperability, flexibility, and reduced development costs. Open-source software is easily adaptable, geared to rapidly changing mission needs, and can generally be delivered at lower costs to meet mission requirements.

This paper outlines Ball's COSMOS C2 system, a fully open-source, web-enabled, command-and-control software architecture which provides several unique capabilities to move the current legacy SSA software paradigm to an open source model that effectively enables pre- and post-launch asset command and control. Among the unique characteristics of COSMOS is the ease with which it can integrate with diverse hardware. This characteristic enables COSMOS to serve as the command-and-control platform for the full life-cycle development of SSA assets, from board test, to box test, to system integration and test, to on-orbit operations. The use of a modern scripting language, Ruby, also permits automated procedures to provide highly complex decision making for the tasking of SSA assets based on both telemetry data and data received from outside sources. Detailed logging enables quick anomaly detection and resolution. Integrated real-time and offline data graphing renders the visualization of the both ground and on-orbit assets simple and straightforward.

### 2. COSMOS BENEFITS

The COSMOS C2 software package provides many key benefits to the SSA ground system architect and operator. The open source nature of COSMOS means it is completely free to start using COSMOS. If additional features or functionality is needed, the source code is freely available for inspection and modification. The open source nature of COSMOS means more and more individuals and companies are being exposed to COSMOS resulting in increased mindshare among operators and developers.

COSMOS can quickly interface with many kinds of targets. Any embedded system that provides a communication interface can be connected to COSMOS. COSMOS ships with interfaces for connecting over TCP/IP, UDP, and serial connections but also supports custom interfaces to connect to anything that a computer can talk to.

All the COSMOS tools are configured with plain text configuration files. This includes the standard COSMOS interfaces and the target command and telemetry definitions. This makes configuration easy by allowing copy and paste as well as more complex templating. It also enables standard configuration management tools which allow text based diffs and searchable history.

COSMOS has a rich API which allows independent tools to interact with the command and telemetry stream. The COSMOS API makes sending commands and checking telemetry easy. However, you are not constrained by your scripting language. COSMOS scripts are written in Ruby, a modern, fully functional scripting language. This allows you to read and write files, and perform live processing that most other systems force you to run offline. An official Python API is supported as well as the native Ruby API. COSMOS is fully cross platform and utilities the QT GUI framework to operate seamlessly on Windows, Linux and Mac OS X.

COSMOS has a rich history in test and can be used at all levels of integration from board level test, box level test, payload integration and test, to spacecraft integration and test. This makes COSMOS an excellent choice to provide

a consistent user interface throughout the full lifecycle of a product. It means you can utilize the same procedures and scripts in operations that were created during development.

Everything in the COSMOS C2 System is logged, and even more importantly, tools are provided to easily interpret and use the logs. Whenever an anomaly occurs there are tools already written that are ready to dig into the logs and help figure out what happened. Data can be visualized with telemetry displays and graphs both in real-time and via log files.

#### 2. COSMOS TOOLS & ARCHITECTURE

Ball Aerospace COSMOS comes with the following set of 17 applications that are directly available for use with minimal to no configuration.

nterfaces Targ	ets Cmd Packets	Tlm Packets	Routers	Logging	) Status					
Interface	Connect/Disconnect	Connected?	Clients	Tx Q Size	Rx Q Size	Bytes Tx	Bytes Rx	Cmd Pkts	TIm Pkts	View Raw
INST_INT	Disconnect	true	0	0	0	46	90974	3	1193	View Raw
INST2_INT	Disconnect	true	0	0	0	0	90852	0	1191	View Raw
EXAMPLE_INT	Connect	false	0	0	0	0	0	0	0	View Raw
TEMPLATED_INT	Connect	false	0	0	0	0	0	0	0	View Raw
SYSTEM_INT	Disconnect	true	0	0	0	0	0	0	61	View Raw
17/08/22 20:23:46	.221 INFO: INST2 HEALT	TH_STATUS GRO	UND1STAT	TUS = CONN	ECTED is GRE	EN				

Fig 1. Command and Telemetry Server

The Command and Telemetry Server acts as the hub of the real-time portion of COSMOS. All commands and telemetry packets pass through this tool ensuring everything that happens is logged. It provides real-time commanding, telemetry reception, logging, limits monitoring, packet routing, and system status.

💁 Replay										
File Help										
Log File Se										
C:/git/CO	)SMOS/demo/outp	outs/logs/2017_08	_22_20_23_06_	tlm.bin					Bro	owse
Playback C	Iontrol									
	M			•				*	M	
Delay: N	No Delay					,				•
Status: S	Stopped									
File Position										
File Position	חי	20:23:06.164 -06	00 Cur	rrent: 201	17/08/22 20:24:33.	302 -0600	End:	2017/08/22 20	0:24:33.302 -060	D
_	חי	20:23:06.164 -06	00 Cur	rrent: 201	17/08/22 20:24:33.	302 -0600	End:	2017/08/22 20	0:24:33.302 -060	
_	n 2017/08/22	2 20:23:06.164 -06 ,			17/08/22 20:24:33.	302 -0600		2017/08/22 20	0:24:33.302 -060 ,	
Start:	2017/08/22 , , 220:24:16.203 V	v VARN: INST HEALT	H_STATUS GROU	und1status = ur	NAVAILABLE is YELI	1 1 .0W	1	1	0:24:33.302 -060i '	
Start:	2017/08/22 2017/08/22 20124:16.203 V 2 20124:16.203 V 2 20124:17.203 V	VARN: INST HEALT NFO: INST HEALTH NFO: INST HEALTH	H_STATUS GROU _STATUS GROU _STATUS TEMP1	UND1STATUS = UI ND2STATUS = CO L = 23.601250000	NAVAILABLE is YELI NNECTED is GREEN 000007 is GREEN_	, , , , , ,	1	1	0:24:33.302 -060 '	
Start:	2017/08/22 20:24:16.203 V 2 20:24:16.203 I 2 20:24:17.203 II 2 20:24:19.203 II	VARN: INST HEALT NFO: INST HEALTH NFO: INST HEALTH NFO: INST HEALTH	H_STATUS GROU _STATUS GROU _STATUS TEMP1 _STATUS TEMP2	UND1STATUS = UI IND2STATUS = CO I = 23.601250000 2 = -7.525364160	NAVAILABLE is YELI NNECTED is GREEN 1562435 is GREEN	, , , , , , , , , , , , , , , , , , ,	1	1	0:24:33.302 -060 '	
Start:	2017/08/22 20:24:16.203 V 2 20:24:16.203 II 2 20:24:17.203 II 2 20:24:19.203 II 2 20:24:21.204 V	VARN: INST HEALT NFO: INST HEALTH NFO: INST HEALTH NFO: INST HEALTH VARN: INST HEALTH	H_STATUS GROU _STATUS GROU _STATUS TEMP1 _STATUS TEMP2 H_STATUS GROI	UND1STATUS = UI IND2STATUS = CO I = 23.601250000 2 = -7.525364160 UND2STATUS = UI	NAVAILABLE is YELI NNECTED is GREEN_ 1000007 is GREEN_ 1562435 is GREEN NAVAILABLE is YELI	, , , , , , , , , , , , , , , , , , ,	1	1	0:24:33.302 -060 '	
Start:	2017/08/22 20124:16.203 V 20:24:16.203 V 20:24:19.203 I 20:24:19.203 V 20:24:21.204 V 20:24:21.204 V 20:24:25.203 V	VARN: INST HEALT NFO: INST HEALTH NFO: INST HEALTH NFO: INST HEALTH VARN: INST HEALT VARN: INST HEALT	H_STATUS GROU _STATUS GROU _STATUS TEMP1 _STATUS TEMP2 H_STATUS GROU H_STATUS TEMP	UND1STATUS = UI ND2STATUS = CO 1 = 23.601250000 2 = -7.525364160 UND2STATUS = UI P1 = 63.15365 is 1	NAVAILABLE is YELI NNECTED is GREEN_ 1000007 is GREEN_ 1562435 is GREEN NAVAILABLE is YELI	, ow , ow , ow	1	1	0:24:33.302 -060i	
Start:	2017/08/22 20:24:16.203 V 2 20:24:16.203 V 2 20:24:17.203 II 2 20:24:19.203 V 2 20:24:25.203 V 2 20:24:25.203 V 2 20:24:26.203 II 2 20:24:26.203 II	VARN: INST HEALT NFO: INST HEALTH VFO: INST HEALTH VFO: INST HEALTH VARN: INST HEALT VARN: INST HEALTH VFO: INST HEALTH VFO: INST HEALTH	H_STATUS GROU STATUS GROU _STATUS TEMP1 _STATUS TEMP2 H_STATUS GROU _STATUS GROU _STATUS GROU _STATUS GROU	UND1STATUS = UI IND2STATUS = C0 L = 23.601250000 2 = -7.525364160 UND2STATUS = UI 11 = 63.15365 is \ IND1STATUS = C0 ND2STATUS = C0	NAVAILABLE IS YELI INNECTED IS GREEN 1000007 IS GREEN 1562435 IS GREEN NAVAILABLE IS YELI VELLOW_HIGH INNECTED IS GREEN	, ow HIGH .ow	1	1	0:24:33.302 -060i '	
Start:	2017/08/22 20:24:16.203 V 2 20:24:16.203 II 2 20:24:17.203 II 2 20:24:19.203 V 2 20:24:21.204 V 2 20:24:25.203 V 2 20:24:25.203 V 2 20:24:26.203 II 2 20:24:29.204 E	VARN: INST HEALT VFO: INST HEALTH VFO: INST HEALTH VFO: INST HEALT VARN: INST HEALT VFO: INST HEALT VFO: INST HEALTH RFO: INST HEALTH RFO: INST HEALTH	H_STATUS GROU _STATUS GROU STATUS TEMPI3 STATUS TEMPI3 H_STATUS GROU H_STATUS GROU STATUS GROU TH_STATUS TEMPI THUS TEMPI3	UND1STATUS = UI ND2STATUS = CO L = 23.601250000 2 = -7.525364160 UND2STATUS = UI 1 = 63.15365 is \ IND1STATUS = CO IND2STATUS = CO IND2STATUS = CO IP1 = 82.9298500	NAVAILABLE is YELI NINECTED is GREEN 1000007 is GREEN 1562435 is GREEN NAVAILABLE is YELI VINECTED is GREEN NINECTED is GREEN 0000002 is RED_H	.ow HIGH .ow	1	1	0:24:33.302 -060i '	
Start:	2017/08/22 20:24:16.203 V 2 20:24:16.203 II 2 20:24:17.203 II 2 20:24:19.203 V 2 20:24:21.204 V 2 20:24:25.203 V 2 20:24:25.203 V 2 20:24:26.203 II 2 20:24:29.204 E	VARN: INST HEALT VFO: INST HEALTH VFO: INST HEALTH VFO: INST HEALT VARN: INST HEALT VFO: INST HEALT VFO: INST HEALTH RFO: INST HEALTH RFO: INST HEALTH	H_STATUS GROU _STATUS GROU STATUS TEMPI3 STATUS TEMPI3 H_STATUS GROU H_STATUS GROU STATUS GROU TH_STATUS TEMPI THUS TEMPI3	UND1STATUS = UI ND2STATUS = CO L = 23.601250000 2 = -7.525364160 UND2STATUS = UI 1 = 63.15365 is \ IND1STATUS = CO IND2STATUS = CO IND2STATUS = CO IP1 = 82.9298500	NAVAILABLE IS YELI INNECTED IS GREEN 1000007 IS GREEN 1562435 IS GREEN NAVAILABLE IS YELI VELLOW_HIGH INNECTED IS GREEN	.ow HIGH .ow	1	1	0:24:33.302 -060i '	

Fig 2. Replay

Replay simulates the Command and Telemetry Server for telemetry packet log file playback. This enables use of any of the real-time tools with logged data. Replay is great for playing back scenarios and viewing them on telemetry screens.

Limits Monitor			
File Help			
Limits Log			
Monitored Limits State:		Green - Some Items Ignored	34 31
INST HEALTH_STATUS TEMP2:	-16.031 C (G)		Ignore Item Ignore Packet
INST HEALTH_STATUS TEMP1:	-31.479 C		Ignore Item Ignore Packet
INST HEALTH_STATUS GROUND2STATUS:	UNAVAILABLE		Ignore Item Ignore Packet
INST HEALTH_STATUS GROUND1STATUS:	CONNECTED	_	Ignore Item Ignore Packet
INST HEALTH_STATUS TEMP3:	-8.451 ⊂ (G)		Ignore Item Ignore Packet
Warning: Some Telemetry Items are Igno		T * */ 3.4 */	

Fig 3. Limits Monitor

Limits Monitor monitors telemetry with defined limits and shows items that are currently out of limits or have violated limits since the tool was started. Expected violations can be easily ignored.

🔂 Command Se File Mode I	ender Help				x
Target: INST	•	•	Com	mand: COLLECT   Send	^
Description: Star Parameters:	rts a collect on the instrume	int			
Name	Value or State		Units	Description	Ξ
TYPE:	SPECIAL	1		Collect type	-
DURATION:		1.0		Collect duration	
OPCODE:		0xAB		Collect opcode	
TEMP:		0.0	С	Collect temperature	
ommand History:	(Pressing Enter on the line	re-exe	cutes th	e command)	
	CT with TYPE NORMAL, DUI CT with TYPE SPECIAL, DUI				
cmd("INST COL	LECT with TYPE SPECIAL,	, DURA	TION 1	.0, OPCODE 171, TEMP 0.0") sent. (2)	
		F	Fig 4. (	Command Sender	

e

Command Sender provides a graphical interface for manually sending individual commands. Drop down selection of every command and command parameter in the system makes sending individual commands easy. A history pane makes resending previous commands easy.

📕 Command Sequ	uence : C	:/git/COSMOS/	demo/	output/	s/sequences/seque	ence.txt		
File Actions H	lelp							
	Stopped	ł					Start Pause	Stop
Target: INST					▼ Command: CC	OLLECT		▼ Add
Time (Delay or Abs	olute)	Command						
2017/08/22 20:34	:45.647	INST ABORT						×
2		INST CLEAR (Ha	izarous	)				×
5		INST COLLECT W	ith TYP	E NORM	AL, DURATION 1.0,	OPCODE 171, TEMP	0.0	×
Description: Star Parameters: Name		e or State	SULC.	Units	Description			
TYPE:	NC	DRMAL	0		Collect type			
DURATION:			1.0		Collect duration			
OPCODE:			0×AB		Collect opcode			
TEMP:			0.0	С	Collect temperatur	ire		
2017/08/22 20:35: 2017/08/22 20:35:	25.701: 25.702: 27.776: 1 32.887:	WARNING: Start I INST ABORT INST CLEAR INST COLLECT wi	time 20 th TYPE	17-08-2 E NORM	2 20:34:45 -0600 ha AL, DURATION 1.0, C	as already passed! OPCODE 171, TEMP 0	.0	
				Fi	g 5. Command	Sequence		

Command Sequence allows for creating a sequence of commands that can be executed as agroup. Commands can be absolute time tagged or have relative delays between commands. The full GUI capability of Command Sendor is available to customize indivual commands.

Script Runner : C:/git/COSMOS/demo/proce File Edit Search Script Help	dures/test.rb	
Running		Go Pause Stop
2 puts "2" 3 puts "3" 4 puts "4" 5 puts "5" 6 puts "6" 7 puts "7" 8 puts "8" 9 puts "9" 10 puts "10" 11 12 play_wav_file(File.join( 13 value = ask('Enter the v 14 15 cmd("INST CLEAR") 16 17 5.times do [index] 18 puts index 19 end 20 21 if false 22 puts false 23 else 24 puts true 25 end Script Output: 2017/08/22 20:54:19.415 (test.rb:7): 7 2017/08/22 20:54:19.519 (test.rb:8): 8 2017/08/22 20:54:19.519 (test.rb:10): 10	Ask ? X	data','tada.wav'))
		ـــــــــــــــــــــــــــــــــــــ

Fig 6. Script Runner

Script Runner executes test scripts and provides highlighting of the currently executing line. Scripts pause if any error occurs, breakpoints can be added, and lines can be re-executed after a problem has been corrected.

🖳 Test Runner									
File Script Help									
Pause on Error	Test Suite:	ExampleTestSuite	🔹 💽 Star	: Setup	Teardown				
☑ Continue Test Case after Error 🗌 Loop Testing	Test Group:	ExampleTest	🔹 💽 Star	: Setup	Teardown				
Abort Testing after Error     Break Loop after	Error Test Case:	test_2	- Star	:					
Executing Test Case: ExampleTest : setup		Pass: 0	Skip: O Fail:	0	0%				
Paused			Go	Pause	Stop				
ExampleTestSuite example_test.rb									
<pre>6 # with 'test_' to be picked up by TestRunner. 7 class ExampleTest &lt; Cosmos::Test 8 def initialize 9 super() 10 end 11 12 # Setup the test case by doing stuff 13 def setup 14 status_bar("setup") 15 puts "Running #{Cosmos::Test.current_test}:#</pre>									
Script Output: 2017/08/22 20:54:55.542 (SCRIPTRUNNER): Starting script: Example 2017/08/22 20:54:55.794 (ExampleTestSuite:1): Marshal load success 2017/08/22 20:54:55.266 (example_test.rb:15): Running ExampleTes 2017/08/22 20:55:00.270 (SCRIPTRUNNER): User pressed Pause	s: C:/git/cosmos/demo/outp	uts/tmp/marshal_f363	175994a55b127caa65b190	389682.bin					

Fig 7. Test Runner

Test Runner provides a high-level framework for system level testing including automatic test report generation. Test Runner brings the best features of software unit level testing to system level integration and test by breaking tests down into easy understandable test cases. Users can execute entire test procedures or just the specific test cases they need to run for integration or regression tests.

	iewer : Formatted Teleme Help	etry with Units 📃 🔲	x	Instrument Health and Status General Telemetry	
			Q	General relemetry Start Collect	
arget: INS	5T 🔻	Packet: HEALTH_STATUS	•	0×00000002	
			_	COLLECT_TYPE:	NORMA
escription:	Health and status from the	e instrument		DURATION:	5.
	Item	Value	*	ASCIICMD:	
8	*TEMP1MAX:	94,9438000000001			
9	*TEMP1MIN:	-95.05595		Temperatures	-
10	*TEMP1MEAN:	-4.821699999999999		TEMP1: 64.602 C -25.28	
11	*TEMP1STDDEV:	52.39506403915517		TEMP2: 3.5297242187500046 (G)	
19	TIMESEC:	1503456994		темр2: 33944.171875 (G)	
20	TIMEUS:	484347	Ξ	TEMP2: 3.530 (G)	
21	PKTID:	1		TEMP2; 3.530 C (G)	
22	COLLECTS:	2		TEMP3: 59.387 C (R)	
23	TEMP1:	-76.991 C			
24	TEMP2:	45.400 C		1.123 ⊂ (G)	
25	TEMP3:	4.914 C		Ground Station	
26	TEMP4:	8.745 C		GROUND1STATUS: CON	INECTE
27	ARY:	["0 V", "1 V", "2 V", "3 V", "4 V", "		GROUND2STATUS: UNAV	AILABL
			Ŧ		

Fig 9. Telemetry Viewer

Packet Viewer provides real-time visualization of every telemetry packet that has been defined. Values within packets are displayed in a simple key-value format that requires no configuration. An autocomplete search bar makes finding values easy.

Telemetry Viewer provides custom telemetry screen functionality with advanced layout and visualization widgets. Tabs, graphs, limits bars, and other animated displays can be quickly created. Also, Telemetry Viewer can autogenerate a base set of screens for every telemetry packet that can be customized as needed.

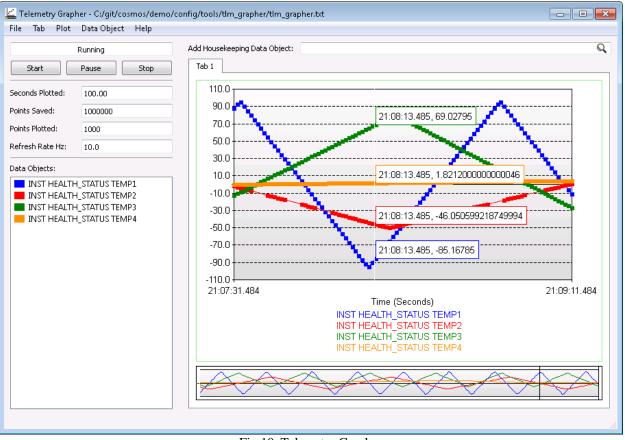


Fig 10. Telemetry Grapher

Telemetry Grapher provides real-time and offline graphing of telemetry data. Supports both line and x-y style plotting, with multiple tabs, plots, and items per plot. Includes built-in analysis functionality to graph min, max, difference, and standard deviation.

	Run	ning																Sta	ť			Pau	lse		Stop
lealth Status	4	ADC:	5	Oth	ner F	acke	ets	I	NST	Heal	th St	atus		INS	T2 H	lealth	Statu	IS							
*******	****	***	* * * *	****	***	* * * *	****	***	***	***	***	***	* * * *	****	***	****	****	**	***	***	***	***	***	**	-
* INST HEA	LTF	I ST	FATI	JS																					
* Received		_			/08,	/22	20:	: 58	36	. 48	5														
* Received	i Co	ount	t: 1	1860	)																				
********	****	***	* * * *	****	* * * *	* * * *	****	***	* * * *	* * * *	* * * *	* * * *	* * * *	***	* * * *	****	****	**	***	***	**1	***	***	**	
00000000:	00	01	С7	44	01	81	59	9C	EF	5C	00	07	63	$\mathbf{FB}$	00	01		D	Y	1	С				
00000010:	00	02	ЕС	B8	7F	CO	00	00	Al	BD	83	8E	00	01	02	03									
00000020:																			0						
00000030:																									
00000040:																									
00000050:																									
00000060:																									
00000070:																									
00000080:																									
00000090:																									
000000A0:																									
000000B0:																									
000000CO:																									
000000D0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00									 -
												Tex													

Fig 11. Data Viewer

Data Viewer provides text based telemetry visualization for items that don't fit into other data visualization paradigms. It is great for scrolling log displays and memory dumps.

le Edit Search File Type Help		
Name	inst_tlm.txt	
<ul> <li>config</li> <li>data</li> <li>system</li> <li>targets</li> <li>EXAMPLE</li> <li>INST</li> <li>INST</li> <li>cond_tim</li> <li>_ccsds_cmd.txt</li> <li>_ccsds_tim.txt</li> <li>inst_cmds.txt</li> <li>inst_tim.txt</li> </ul>	<pre>1 TELEMETRY INST HEALTH_STATUS BIG_ENDIAN "Health and sta 2 &lt;\= render "_ccsds_tlm.txt", locals: {apid: 1} \&gt; 3 APPEND_ITEM COLLECTS 16 UINT "Number of a 4 APPEND_ITEM TEMP1 16 UINT "Temperature 5 POLY_READ_CONVERSION -100.0 0.00305 6 POLY_WRITE_CONVERSION 32768.885246 327.86885 7 UNITS CELCIUS C 8 FORMAT_STRING "\0.3f" 9 LIMITS DEFAULT 1 ENABLED -80.0 -70.0 60.0 80.0 -20 10 LIMITS TVAC 1 ENABLED -80.0 -30.0 30.0 80.0 11 LIMITS_RESPONSE example_limits_response.rb 12 APPEND_ITEM TEMP2 32 FLOAT "Temperature 15 DEVENDENT 100 00 10 00 00 00 00 00 00 00 00 00 00</pre>	Defines a new telemetry     packet     Parameters:     Target (Required)     Name of the target this telemetry     packet is associated with
<ul> <li>o doc</li> <li>ib</li> <li>o procedures ≡</li> <li>o screens</li> <li>o sequences</li> <li>tables</li> <li>o tables</li> <li>o tools</li> </ul>	13         FOLY_READ_CONVERSION -100.0 0.00305           14         FOLY_WRITE_CONVERSION 32768.885246 327.86885           15         UNITS_CELCIUS C           16         FORMAT_STRING "%0.3f"           17         LIMITS_DEFAULT 1 ENABLED -60.0 -55.0 30.0 35.0           18         LIMITS_TVAC 1 ENABLED -60.0 20.0 30.0 35.0           19         APPEND_ITEM TEMP3           16         UINT           20         POLY_READ_CONVERSION -100.0 0.00305	Command (Required) Name of this telemetry packet. Also referred to as its mnemonic. Must be unique to telemetry packets in this target. Ideally will be as short and clear as possible.
	<pre>21 POLY_WRITE_CONVERSION 32768.885246 327.86885 22 UNITS CELCIUS C 23 FORMAT_STRING "%0.3f" 24 LIMITS DEFAULT 1 ENABLED -25.0 -10.0 50.0 55.0 25 LIMITS TVAC 1 ENABLED -15.0 -10.0 20.0 30.0 26 APPEND_ITEM TEMP4 16 UINT "Temperature 27 POLY_READ_CONVERSION -100.0 0.00305 28 POLY_WRITE_CONVERSION -100.0 0.00305 29 UNITS CELCIUS C 30 FORMAT_STRING "%0.3f" 31 LIMITS DEFAULT 1 ENABLED -80.0 -70.0 60.0 80.0 32 APPEND_ARRAY_ITEM ARY 8 UINT 80 "Array data" 33 UNITS VU </pre>	HEALTH_STATUS Endianness (Required) Indicates if the data in this packet is in Big Endian or Little Endian format BIG_ENDIAN Description (Optional) Description of this telemetry packet which must be enclosed with quotes alth and status from the instrument"

Fig 12. Config Editor

Config Editor provides GUI contextual help when editing COSMOS configuration files. The left pane allows you to easily navigate the COSMOS project in a directory tree. The right pane provides descriptions and drop downs for the current line being edited.

File       Mode       Item       Help         Configuration       Configuration       Save       Browse         ITEM INST ADCS TIMESECONDS       ITEM INST ADCS TIMESECONDS       Item INST ADCS TIMEFORMATTED         ITEM INST ADCS QI FORMATTED       ITEM INST ADCS QI FORMATTED       Item INST ADCS QI FORMATTED         ITEM INST ADCS QI FORMATTED       ITEM INST ADCS VELX       Item         ITEM INST ADCS VELX       FORMATTED       Item         ITEM INST ADCS VELX       FORMATTED       Item         ITEM INST ADCS VELX       FORMATTED       Item         Target:       EXAMPLE         Packet:       STATUS          Column Name:       Text:       Add Item       Add Packet       Add Target         Column Name:       Text:       Add Item       Add Text       Downsample Seconds: 0.0         File Selection       Item:       Packet_gas/2017_08_22_20_27_37_ttm.bin       Items       C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_ttm.bin       Select         Output File:       C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_ttm.bin       Celear       Select         Time Period End:       N/A       Celear       Select       Select       Select         Time Period End:       N/A       Celear       Select       <	🚡 Telemetry Extracto	r					- • •
Configuration File: Save Browse ITEM INST ADCS TIMESECONDS ITEM INST ADCS TIMEFORMATTED ITEM INST ADCS Q1 KAW ITEM INST ADCS Q1 FORMATTED ITEM INST ADCS Q1 FORMATTED ITEM INST ADCS VELX RAW ITEM INST ADCS VELX ITEM INST ADCS VELX RAW ITEM INST ADCS VELX RAW ITEM INST ADCS VELX RAW ITEM INST ADCS VELX ITEM INST	File Mode Item	Help					
Configuration File: Save Browse ITEM INST ADCS TIMESECONDS ITEM INST ADCS TIMEFORMATTED ITEM INST ADCS Q1 RAW ITEM INST ADCS Q1 FORMATTED ITEM INST ADCS Q1 FORMATTED ITEM INST ADCS VELX RAW ITEM INST ADCS VELX ITEM INST	Configuration						
ITEM INST ADCS TIMESECONDS         ITEM INST ADCS TIMESECONDS         ITEM INST ADCS Q1 RAW         ITEM INST ADCS Q1 FORMATTED         ITEM INST ADCS VELX RAW         ITEM INST ADCS VELX RAW         ITEM INST ADCS VELX         Output Seconds: 0.0         File Selection         Log Files:         Downsample Seconds: 0.0         File Selection         Log Files:         Output File:       C:\gt\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin         Output File:       C:\gt\COSMOS\demo\outputs\logs\2017_08_22_2.20_27_37_tlm.bin         Item Period							
ITEM INST ADCS TIMEFORMATTED ITEM INST ADCS QI RAW ITEM INST ADCS QI PORMATTED ITEM INST ADCS QI WITH_UNITS ITEM INST ADCS VELX RAW ITEM INST ADCS VELX RAW ITEM INST ADCS VELX RAW ITEM INST ADCS VELX EORMATTED Target: EXAMPLE Packet: STATUS Item: PACKET_ID Add Item Add Packet Add Target Column Name: Text: Add Item Add Packet Add Target Column Name: Text: Add Item Add Packet Downsample Seconds: 0.0 File Selection Log Files: Browse Remove C:\git\COSMOS\demo\outputs\\logs\2017_08_22_20_27_37_tlm.bin Output File: C:\git\COSMOS\demo\outputs\\logs\2017_08_22_20_27_37_tlm.bin Itme Period Start: N/A Clear Select Packet Log Reader: Cosmos::PacketLogReader Select	Configuration File:					Save	Browse
ITEM INST ADCS TIMEFORMATTED ITEM INST ADCS QI RAW ITEM INST ADCS QI PORMATTED ITEM INST ADCS QI WITH_UNITS ITEM INST ADCS VELX RAW ITEM INST ADCS VELX RAW ITEM INST ADCS VELX RAW ITEM INST ADCS VELX EORMATTED Target: EXAMPLE Packet: STATUS Item: PACKET_ID Add Item Add Packet Add Target Column Name: Text: Add Item Add Packet Add Target Column Name: Text: Add Item Add Packet Downsample Seconds: 0.0 File Selection Log Files: Browse Remove C:\git\COSMOS\demo\outputs\\logs\2017_08_22_20_27_37_tlm.bin Output File: C:\git\COSMOS\demo\outputs\\logs\2017_08_22_20_27_37_tlm.bin Itme Period Start: N/A Clear Select Packet Log Reader: Cosmos::PacketLogReader Select							
ITEM INST ADCS Q1 RAW   ITEM INST ADCS Q1   ITEM INST ADCS Q1 FORMATTED   ITEM INST ADCS Q1 WITH_UNITS   ITEM INST ADCS VELX RAW   ITEM INST ADCS VELX RAW   ITEM INST ADCS VELX RAW   ITEM INST ADCS VELX   Target:   EXAMPLE • Packet:   STATUS • Item:   PACKET_ID • Add Item   Add Target   Column Name:   Text:   Downsample Seconds:   0.0   File Selection  Log Files:   C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin  Output File:  C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin  City File:   Output File:   C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin     Charles  Clear Select  Packet Log Reader:  Cosmos::PacketLogReader  Select							Â
ITEM INST ADCS Q1 ITEM INST ADCS Q1 FORMATTED ITEM INST ADCS Q1 WITH_UNITS ITEM INST ADCS VELX ITEM INST ADCS							
ITEM INST ADCS Q1 FORMATTED ITEM INST ADCS Q1 WITH_UNITS ITEM INST ADCS Q1 WITH_UNITS ITEM INST ADCS VELX RAW ITEM INST ADCS VELX FORMATTED Add Item Add Packet Add Target Target: EXAMPLE Packet: STATUS Tem: PACKET_ID Add Item Add Packet Add Target Column Name: Text: Add Item Add Packet Add Target Column Name: Text: Add Item Add Packet Add Target Downsample Seconds: 0.0 File Selection Log Files: Browse Remove C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin Output File: C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin Output File: C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin Ime Period Start: N/A Clear Select Time Period End: N/A Clear Select Packet Log Reader: Cosmos::PacketLogReader Select		•					=
ITEM INST ADCS Q1 WITH_UNITS ITEM INST ADCS VELX RAW ITEM INST ADCS VELX RAW ITEM INST ADCS VELX COMMATTED Q Add Item Add Packet Add Target Add Item Add Packet Add Target Add Item Add Packet Add Target Column Name: Text: Add Item Add Packet Add Target Column Name: Text: Add Item Add Packet Add Target Downsample Seconds: 0.0 File Selection Log Files: Browse Remove C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin Output File: C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin Output File: C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin Dutput File: C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin Difference File: N/A Clear Select Packet Log Reader: Cosmos::PacketLogReader Select							
ITEM INST ADCS VELX RAW   ITEM INST ADCS VELX   Target:   EXAMPLE • Packet:   STATUS • Item:   PACKET_ID • Add Item   Add Target   Column Name:   Text:   Add Text   Downsample Seconds:   0.0   File Selection  C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin  Output File: C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin  Output File: C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bit   Select   Time Period End:   N/A   Clear   Select   Packet Log Reader:   Cosmos::PacketLogReader							
TTELM INIST ADOC VELX EORMAATTED     Add Item     Add Item   Add Item   Add Text   Add Text   Downsample Seconds:   0.0   File Selection   Cigit/COSMOS/demo/outputs/logs/2017_08_22_20_27_37_tlm.bin							
Add Item Add Packet Add Target   Target: EXAMPLE • Packet: STATUS • Item: PACKET_ID • Add Item Add Packet Add Target   Column Name: Text: Add Text Add Text   Downsample Seconds: 0.0 Remove   File Selection Log Files: Browse Remove C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin Output File: C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin Citine Period Start: N/A Clear Select Packet Log Reader: Cosmos::PacketLogReader Select	ITEM INST ADOS	/ELX					
Target:       EXAMPLE ▼ Packet:       STATUS ▼ Item:       PACKET_ID       Add Item       Add Packet       Add Target         Column Name:       Text:       Add Text       Add Text         Downsample Seconds:       0.0       Item:       File Selection         Cog Files:       Browse       Remove         C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin       Item:       Select         Output File:       C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.txt       Select         Time Period Start:       N/A       Clear       Select         Packet Log Reader:       Cosmos::PacketLogReader       Select	TTEM INST ADOS	/ELX EORMATTED					<b>T</b>
Column Name: Text: Add Text Downsample Seconds: 0.0 File Selection Log Files: Browse Remove C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin Output File: C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin Output File: C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.txt Select Time Period Start: N/A Clear Select Time Period End: N/A Clear Select Packet Log Reader: Cosmos::PacketLogReader Select				Q	Add Item	Add Packet	Add Target
Downsample Seconds:       0.0         File Selection       Browse         Log Files:       Browse         C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin         Output File:       C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.txt         Output File:       C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.txt         Select       Time Period Start:         N/A       Clear         Packet Log Reader:       Cosmos::PacketLogReader	Target: EXAMPLE	Packet: STATUS	r Item: PACKET_ID	•	Add Item	Add Packet	Add Target
Downsample Seconds:       0.0         File Selection       Browse         Log Files:       Browse         C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin         Output File:       C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.txt         Output File:       C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.txt         Select       Time Period Start:         N/A       Clear         Packet Log Reader:       Cosmos::PacketLogReader	Column Name:		Text:				Add Text
File Selection         Log Files:       Browse         C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin         Output File:       C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.txt         Select         Time Period End:       N/A         Packet Log Reader:       Cosmos::PacketLogReader	Coldmit Hamor						Had love
Log Files:       Browse       Remove         C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin	Downsample Second	s: 0.0					
Log Files:       Browse       Remove         C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin	City ColorMan						
C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin         Output File:       C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.txt         Select         Time Period Start:       N/A         Clear       Select         Time Period End:       N/A         Packet Log Reader:       Cosmos::PacketLogReader	File Selection						
C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.bin         Output File:       C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.txt         Select         Time Period Start:       N/A         Clear       Select         Time Period End:       N/A         Packet Log Reader:       Cosmos::PacketLogReader	Log Files:					Browse	Remove
Output File:       C:\git\COSMOS\demo\outputs\logs\2017_08_22_20_27_37_tlm.txt       Select         Time Period Start:       N/A       Clear       Select         Time Period End:       N/A       Clear       Select         Packet Log Reader:       Cosmos::PacketLogReader       Select			7 00 22 20 27 27 11 11				
Time Period Start:     N/A     Clear     Select       Time Period End:     N/A     Clear     Select       Packet Log Reader:     Cosmos::PacketLogReader     Select	C:\git\CUSMUS\d	iemo\outputs\logs\20J	./_08_22_20_27_37_tim.bin				
Time Period Start:     N/A     Clear     Select       Time Period End:     N/A     Clear     Select       Packet Log Reader:     Cosmos::PacketLogReader     Select							
Time Period Start:     N/A     Clear     Select       Time Period End:     N/A     Clear     Select       Packet Log Reader:     Cosmos::PacketLogReader     Select							
Time Period Start:     N/A     Clear     Select       Time Period End:     N/A     Clear     Select       Packet Log Reader:     Cosmos::PacketLogReader     Select							
Time Period Start:     N/A     Clear     Select       Time Period End:     N/A     Clear     Select       Packet Log Reader:     Cosmos::PacketLogReader     Select							
Time Period End:     N/A     Clear     Select       Packet Log Reader:     Cosmos::PacketLogReader     Select	Output File:	C:\git\COSMOS\demo\o	utputs\logs\2017_08_22_20_27	_37_tlm.t×	t		Select
Time Period End:     N/A     Clear     Select       Packet Log Reader:     Cosmos::PacketLogReader     Select	Time Period Start:	N/A				Clear	Select
Packet Log Reader: Cosmos::PacketLogReader Select							
	Time Period End:	N/A				Clear	Select
Process Files Open in Text Editor Open in Excel	Packet Log Reader:	Cosmos::PacketLogRead	ler				Select
Process Files Open in Text Editor Open in Excel							
	Proce	ss Files	Open in Text Edito	or		Open in Exc	el

Fig 13. Telemetry Extractor

Telemetry Extractor extracts telemetry packet log files into CSV data. Highly configurable and supports batch processing to output multiple files at once.

🧱 Command Extra	ctor		- • •
File Mode Hel	p		
Log Files:		Browse	Remove
C:\git\COSMOS\a	emo\outputs\logs\2017_08_22_20_27_40_cmd.bin		
Output File:	C:\git\CO5MO5\demo\outputs\logs\2017_08_22_20_27_40_cmd.txt		Select
Time Period Start:	N/A	Clear	Select
Time Period End:	N/A	Clear	Select
Packet Log Reader:	Cosmos::PacketLogReader		Select
	Process Files Open in	Text Editor	

Fig 14. Command Extractor

Command Extractor extracts command packet logs into human readable text.

🛃 Handbook Creator	
File Help	
Create HTML Handbooks	
Create PDF Handbooks	
Create HTML and PDF Handbooks	
🕞 Open in Web Browser	

Fig 15. Handbook Creator

Handbook Creator creates html and pdf documentation of available commands and telemetry packets.

	nos/demo/config/t MOS/demo/outpul		ager/ConfigTables_def.txt ¡Tables.dat	
MC CONFIGURATION	TLM MONITORI	NG PPS SEL	ECTION	
		Value		-
SCRUB REGION 1 START ADDR		0×0		
SCRUB REGION 1 END ADDR		0x3FFFFFF		
SCRUB REGION 1 ERROR CHECK SIZE		0×10000		
SCRUB REGION 1 THROTTLE COUNT		0×1770		
SCRUB REGION 1 THROTTLE TICKS		0×2		
SCRUB REGION 2 START ADDR		0×0		=
SCRUB REGION 2 END ADDR		0x3FFFFF		
SCRUB REGION 2 ERROR CHECK SIZE		0×10000		
SCRUB REGION 2 THROTTLE COUNT		6000		
SCRUB REGION 2 THROTTLE TICKS		2		
DUMP PACKET THROT	TLE (SEC)	2		
MEMORY SCRUBBING		ENABLE		
SIOC MEMORY CONFI	G	3		-

Table Manager is a binary file editor that can be used to create or edit configuration tables or other binary data.

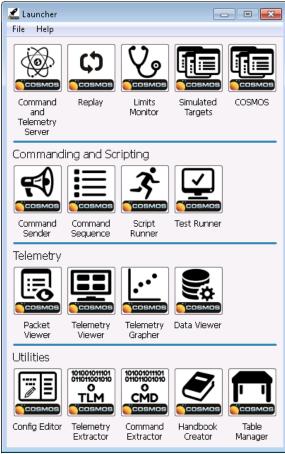


Fig 17. Launcher

Launcher provides a graphical user interface for launching each of the tools that make up the COSMOS system. Supports launching any application that can be started from the command line.

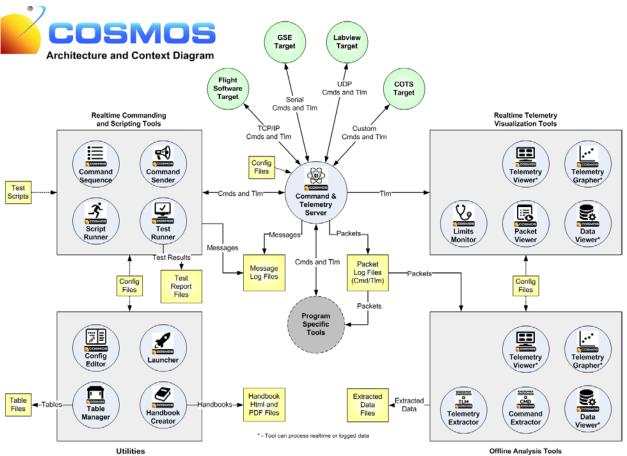


Fig 18. COSMOS Architecture

Fig 18 shows how the 17 applications that make up the COSMOS system relate to each other and to the targets that COSMOS is controlling.

## 5. CLOUD DEPLOYMENT

Deploying to the cloud provides several benefits for the SSA C2 ground system operator. Once COSMOS has been deployed to the cloud any user can connect to the deployed instance to operate COSMOS. The cloud also provides scalable infrastructure to allow COSMOS to handle increased processing loads by simply scaling the deployed instance. Once on the cloud, COSMOS can interface with existing cloud applications and services much easier than if it were deployed on a standard workstation.

Due to the cross-platform nature of COSMOS, it can be deployed to the cloud in many different instantiations. Amazon Web Services (AWS) provides images for Windows Server, Red Hat Linux and Ubuntu which have all successfully hosted COSMOS. Installation on the Windows Server instance is as simple as running the COSMOS installer and starting the Launcher.

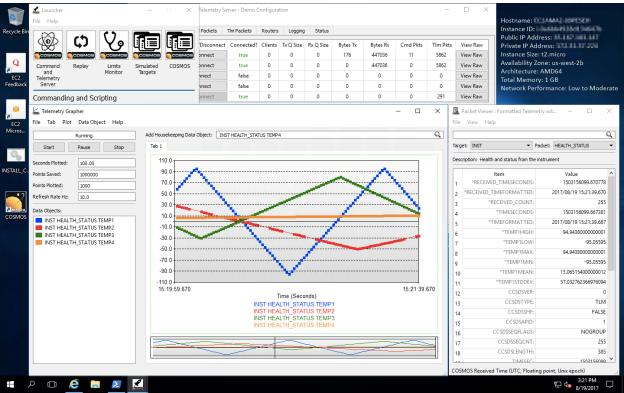


Fig 19. Windows Server AWS

Deploying COSMOS to Red Hat or Ubuntu on AWS requires installing a GUI desktop and then simply running the standard COSMOS Linux installer script. COSMOS has successfully been run through XWindows or VNC on both architectures. Fig 20 depicts COSMOS running on Red Hat Linux with SSH X forwarding to a Mac OS machine running XQuartz.

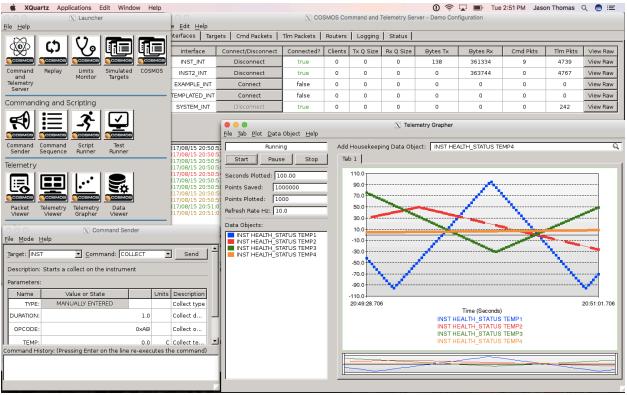


Fig 20. Red Hat AWS X Server

Fig 21 depicts COSMOS running on Ubuntu Linux through a VNC Client. This solution was found to have much better performance than SSH X forwarding.



Fig 21. Ubuntu AWS VNC Client

## 6. CONCLUSION

Ball Aerospace COSMOS is a free and open source command and control system that is immediately available for use. It provides a wealth of functionality, much of which is not even available in expensive proprietary tools. It has been deployed to AWS and is ready for use in operational programs. COSMOS has extensive heritage as it was first developed in 2006 and has since been used to develop and test more than 30 flight programs at Ball Aerospace including GMI, OLI, Kepler, WISE, OMPS, Ares, Orion, and numerous defense programs. Since being open sourced in January 2015 it is now being used with at least 10 major corporations and numerous Universities. For more information and to get started with Ball Aerospace COSMOS please see <a href="http://cosmosrb.com">http://cosmosrb.com</a>.

## REFERENCES

1. Free Software Foundation, Inc., GNU GENERAL PUBLIC LICENSE, Version 3, 29 June 2007

7.