CTF Test Instruction Reference

Associated Plug-Ins	CTF Instructions	Data/Argument Name & Description	Instruction Usage/Application	Examples

cfs	***RegisterCfs***	***target*** **:** name of cFS instance; i.e. name of the executable cFS image; this name should match a section in the .ini file in the format [<***target***>] within the section commented as "Base settings for cfs"; default is an empty string; if not specified, it applies to all cFS targets in the .ini file whose name begins with "cfs_".	Register a cFS target with the specified name so CTF can manage its external interfaces. CTF can manage multiple cFS targets.	<pre>{
				"instruction": "RegisterCfs",
				"data": {
				"target": "lx_cfs_1"
				}
				}</pre>

cfs	***BuildCfs***	***target:*** name of a registered cFS target; i.e. executable cFS image	Have CTF build the cFS executable image with the specified name	<pre>{
				"instruction": "BuildCfs",
				"data": {
				"target": "lx_cfs_1"
				}
				}</pre>

cfs	***StartCfs***	***target:*** name of a registered cFS target; i.e. executable cFS image
		run_args: command line argument for cFS executable
			Have CTF start up the cFS target with the specified name and arguments	<pre>{
				"instruction": "StartCfs",
				"data": {
				"target": "lx_cfs_1",
				"run_args": "-R P0"
				}
				}</pre>

cfs	***EnableCfsOutput***	***target:*** name of a registered cFS target; optional; default is empty string; if not specified, it applies to all registered cFS targets	Have CTF establish the connection to cFS target for telemetry downlink	<pre>{
				"instruction": "EnableCfsOutput",
				"data": {
				"target": "lx_cfs_1"
				}
				}</pre>

cfs	***SendCfsCommand***	***target:*** name of a registered cFS target; optional; default is empty string; if not specified, it applies to all registered cFS targets
		mid: CCSDS message ID; can either be a value or its associated name, as defined in the MIDS JSON data file
		cc: CCSDS command/function code; can either be a value or its associated name, as defined in the CMD JSON data file
		args: list of command arguments, as defined in the CMD JSON data file
			Have CTF send a command to the specified target(s)	<pre>{
				"instruction": "SendCfsCommand",
				"data": {
				"target": "lx_cfs_1",
				"mid": "TO_CMD_MID",
				"cc": "TO_ENABLE_OUTPUT",
				"args": {
				"cDestIp": "127.0.0.1",
				"usDestPort": "5011"
				}
				}
				}</pre>

cfs	***SendInvalidLengthCfsCommand***	Same as ***SendCfsCommand*** with the addition of
		payload_length: the size of the invalid payload length
			Have CTF intentionally invalidate the command by having CTF set the message length to payload_length prior to sending it to the specified target(s)	<pre>{
				"instruction": "SendInvalidLengthCfsCommand",
				"data": {
				"target": "lx_cfs_1",
				"mid": "TO_CMD_MID",
				"cc": "TO_ENABLE_OUTPUT",
				"payload_length": "10"
				}
				}</pre>

Associated Plug-Ins	CTF Instructions	Data/Argument Name & Description	Instruction Usage/Application	Examples

cfs	***CheckEvent***	***target:*** name of a registered cFS target; optional; default is empty string; if not specified, it applies to all registered cFS targets
		app_name: the name of the cFS application that sent the event message
		event_id: cFS event ID
		event_str: the event message; optional; default is empty string; if specified, CTF performs a string comparison between the expected and actual strings
		is_regex: boolean flag to denote a regex match instead of string comparison is to be performed on ***event_str*** ; optional; default is false
		event_str_args: arguments that will be inserted into ***event_str*** , similar to printf() function; optional; default is empty string
			Have CTF verify that one or more cFS events matching the specified parameters have been received from the specified cFS target	{
				"instruction": "CheckEvent",
				"data": {
				"target": "lx_cfs_1",
				"args": [
				{
				"app_name": "BEX",
				"event_id": "13",
				"event_str": "Processed MODE(%d) Cmd Rcvd",
				"is_regex": false,
				"event_str_args": "(1)"
				}
				]
				}
				}

cfs	***CheckNoEvent***	Same as ***CheckEvent***	Have CTF verify that one or more cFS events matching the specified parameters have not been received from the specified cFS target	{
				"instruction": "CheckNoEvent",
				"data": {
				"target": "lx_cfs_1",
				"args": [
				{
				"app_name": "TO",
				"event_id": "3",
				"event_str": "TO - ENABLE_OUTPUT Cmd Rcvd",
				"is_regex": false,
				"event_str_args": ""
				}
				]
				}
				}

cfs	***CheckTlmValue***	***target:*** name of a registered cFS target; optional; default is empty string; if not specified, it applies to all registered cFS targets
		mid: CCSDS message ID
		args: array of arguments to be verified; see CTF documentation or example test scripts for additional examples
			Have CTF verifies that a telemetry matching the specified parameters has been received from the specified cFS target	{
				"instruction": "CheckTlmValue",
				"data": {
				"target": "lx_cfs_1",
				"mid": "TO_HK_TLM_MID",
				"args": [
				{
				"variable": "usCmdErrCnt",
				"compare": "<",
				"value": "1"
				}
				]
				}
				}

cfs	***CheckTlmPacket***	***target:*** name of a registered cFS target; optional; default is empty string; if not specified, it applies to all registered cFS target
		mid: CCSDS message ID
			Have CTF verifies that a telemetry matching the specified MID has been received from the specified cFS target	{
				"instruction": "CheckTlmPacket",
				"data": {
				"target": "lx_cfs_1",
				"mid": "TO_HK_TLM_MID"
				}
				}

Associated Plug-Ins	CTF Instructions	Data/Argument Name & Description	Instruction Usage/Application	Examples

cfs	***CheckNoTlmPacket***	***target:*** name of a registered cFS target; optional; default is empty string; if not specified, it applies to all registered cFS target
		mid: CCSDS message ID
			Have CTF verifies that a telemetry matching the specified MID has not been received from the specified cFS target	<pre>{
				  "instruction": "CheckNoTlmPacket",
				  "data": {
				  "target": "lx_cfs_1",
				  "mid": "TO_HK_TLM_MID"
				  }
				}</pre>

cfs	***CheckTlmContinuous***	Same as ***CheckTlmValue*** , with the addition of
		
		verification_id: a unique ID to identify this verification within a test case
			Similar to ***CheckTlmValue*** , except the verification is done every time the telemetry is received, until the test case ends, or the check is removed by ***RemoveCheckTlmContinuous***	<pre>{
				  "instruction": "CheckTlmContinuous",
				  "data": {
				  "target": "lx_cfs_1",
				  "verification_id": "TO_no_errors",
				  "mid": "TO_HK_TLM_MID",
				  "args": [
				  {
				  "variable": "usCmdErrCnt",
				  "compare": "<",
				  "value": "1"
				  }
				  ]
				  }
				}</pre>

cfs	***RemoveCheckTlmContinuous***	***verification_id:*** the verification_id value of an existing ***CheckTlmContinuous***	Have CTF stop performing continuous telemetry check for ***CheckTlmContinuous*** matching the ***verification_id*** parameter	<pre>{
				  "instruction": "RemoveCheckTlmContinuous",
				  "data": {
				  "verification_id": "TO_no_errors"
				  }
				}</pre>

cfs	***ArchiveCfsFiles***	***target:*** name of a registered cFS target; optional; default is empty string; if not specified; it applies to all registered cFS targets
		source_path: a directory path to copy from, either absolute or relative to the location of CTF
			Have CTF copy files from the specified directory into the test run's log directory	<pre>{
				  "instruction": "ArchiveCfsFiles",
				  "data": {
				  "target": "lx_cfs_1",
				  "source_path": "../../build/exe/lx_cfs_1/cf/"
				  }
				}</pre>

cfs	***ShutdownCfs***	***target:*** name of a registered cFS target; optional; default is empty string; if not specified; it applies to all registered cFS targets	Have CTF shut down cFS target(s) within the test script. CTF will automatically shutdown cFS targets upon test completion.	<pre>{
				  "instruction": "ShutdownCfs",
				  "data": {
				  "target": "lx_cfs_1"
				  }
				}</pre>

ssh	***SSH_RegisterTarget***	***name:*** an arbitrary-but-unique name to identified an SSH target	Register an SSH target with the specified name so CTF can manage its external interfaces. CTF can manage multiple SSH targets.	<pre>{
				  "instruction": "SSH_RegisterTarget",
				  "data": {
				  "name": "ssh_tgt_1"
				  }
				}</pre>

Associated Plug-Ins	CTF Instructions	Data/Argument Name & Description	Instruction Usage/Application	Examples

ssh	***SSH_InitSSH***	***host:*** network hostname or IP to connect to
		name: a registered SSH target; optional
		user: user name for the connection; do not specified if already included in ***host*** ; optional
		port: port number for the connection; optional
		gateway: SSH gateway command to proxy the connection to ***host*** ; optional
		ssh_config_path: path to an ssh config file that contains host definitions or additional parameters; optional; default is "~/.ssh/config"
		args: additional SSH connection options, as needed
			Have CTF establish the SSH connection with a target host	{
				"instruction": "SSH_InitSSH",
				"data": {
				"name": "ssh_tgt_1",
				"host": "123.123.123.1",
				"user": "lander_demo",
				"port": 22,
				"gateway": "ssh -W %h:%p myproxy",
				"ssh_config_path": "./ssh/config"
				}
				}

ssh	***SSH_RunRemoteCommand***	***name:*** a registered SSH target
		command: the shell command to be executed; can contain multiple commands separated with semicolon
			Have CTF execute a shell command on the SSH target	{
				"instruction": "SSH_RunRemoteCommand",
				"data": {
				"name": "ssh_tgt_1",
				"command": "cd lander_fsw_ctf/;rm -rf build; make; make install;"
				}
				}

ssh	***SSH_RunLocalCommand***	***name:*** a registered SSH target; optional
		command: the shell command to be executed; can contain multiple commands separated by semicolons
			Have CTF execute a shell command on the local host	{
				"instruction": "SSH_RunLocalCommand",
				"data": {
				"name": "ssh_tgt_1",
				"command": "cd lander_fsw_ctf/;rm -rf build; make; make install;"
				}
				}

ssh	***SSH_CheckOutput***	***name:*** a registered SSH target
		output_contains: a substring that must be contained in the output
		output_does_not_contain: a substring that should not be contained in the output; optional
		exit_code: the expected exit code after the command is executed
			Have CTF verify the output of the most recently executed SSH command	{
				"instruction": "SSH_CheckOutput",
				"data": {
				"name": "ssh_tgt_1",
				"output_contains": "Built target mission-install",
				"output_does_not_contain": "Error",
				"exit_code": 0
				}
				}

Associated Plug-Ins	CTF Instructions	Data/Argument Name & Description	Instruction Usage/Application	Examples

ssh	***SSH_PutFile***	***name:*** a registered SSH target
		local_path: path to the local file/directory as source path
		remote_path: path to the remote file/directory as destination path
		args : parameters for the transfer
		args.delete: a boolean option for rsync command with --delete argument; if set to true; rsync will remove remote files that no longer exists locally; optional; default is false
		args.exclude: a string of option for rsync command with --exclude argument; optional; default is an empty string
			Have CTF copy files from the specified local directory into the specified remote directory	{
				"instruction": "SSH_PutFile",
				"data": {
				"name": "ssh_tgt_1",
				"local_path": "./cfs",
				"remote_path": "/tmp/workspace/cfs",
				"args": {
				"delete": true,
				"exclude": "*.git"
				}
				}
				}

ssh	***SSH_GetFile***	***name:*** a registered SSH target
		remote_path: path to the remote file/directory as source path
		local_path: path to the local file/directory as destination path
		args: parameters for the transfer
		args.delete: a boolean option for rsync command with --delete argument; if set to true; rsync will remove remote files that no longer exists locally; optional; default is false
		args.exclude: a string of option for rsync command with --exclude argument; optional; default is an empty string
			Have CTF copy files from the specified remote directory into the specified local directory	{
				"instruction": "SSH_GetFile",
				"data": {
				"name": "ssh_tgt_1",
				"local_path": "./results.txt",
				"remote_path": "./data/output.dat"
				}
				}

ssh	***SSH_GetFTP***	***name:*** a registered SSH target
		host: hostname or address of the FTP server
		remote_path: remote path to the file/directory is to be downloaded from
		local_path: local path to where the file/directory is to be downloaded to
			Have CTFdownload file/directory from a remote directory via FTP	{
				"instruction": "SSH_GetFTP",
				"data": {
				"name": "ssh_tgt_1",
				"host": "ftphost",
				"local_path": "./results.txt",
				"remote_path": "./data/output.dat"
				}
				}

ssh	***SSH_PutFTP***	***name:*** a registered SSH target
		host: hostname or address of the FTP server
		local_path: local path to where the file/directory is to be uploaded from
		remote_path: remote path to the file/directory on the FTP server where local file/directory is to be uploaded to
			Have CTF upload file/directory to a remote directory via FTP	{
				"instruction": "SSH_PutFTP",
				"data": {
				"name": "ssh_tgt_1",
				"host": "ftphost",
				"local_path": "./results.txt",
				"remote_path": "./data/output.dat"
				}
				}

Associated Plug-Ins	CTF Instructions	Data/Argument Name & Description	Instruction Usage/Application	Examples

userio	***WaitForUserInput***	***prompt:*** a string to be displayed to prompt for user input; optional; default is an empty string	Have CTF pause the test execution until it receives user input to either resume or stop	<pre>{
				  "instruction": "WaitForUserInput",
				  "data": {
				  "prompt": "Enter Y or N:",
				  }
				}</pre>

variable	***SetUserVariable***	***variable_name:*** user-defined variable name
		operator : assignment or math operator to apply to variable (e.g.: =, +, -, *, /)
		value: value of primitive data type (e.g., integer, float or string)
			Have CTF create a variable and assign a value to it; commonly use for loop counter	<pre>{
				  "instruction": "SetUserVariable",
				  "data": {
				  "variable_name": "my_var_1",
				  "operator": "=",
				  "value": 10
				  }
				}</pre>

variable	***SetUserVariableFromTlm***	***variable_name:*** user-defined variable name
		mid: CCSDS message ID
		tlm_variable: name of the telemetry item as defined in the json definition for that telemetry message
			Have CTF create a variable and assign a value of a telemetry item to it	<pre>{
				  "instruction": "SetUserVariableFromTlm",
				  "data": {
				  "variable_name": "my_var_2",
				  "mid": "TO_HL_TLM_MID",
				  "tlm_variable": "usCmdCnt"
				  }
				}</pre>

variable	***CheckUserVariable***	***variable_name:*** user-defined variable name
		operator: comparison operator to apply to variable
		value: value of primitive data type (e.g., integer, float or string)
			Have CTF compare the value of an existing user-defined variable to the specified value	<pre>{
				  "instruction": "CheckUserVariable",
				  "data": {
				  "variable_name": "my_var_1",
				  "operator": "==",
				  "value": 4
				  }
				}</pre>

control_flow	***BeginLoop***	***label:*** user-define label; must be unique within the scope of the test case
		conditions: a list of comparison conditions
		condition.name: name of an existing user-defined variable
		condition.operator: comparison operator
		condition.value: value of primitive data type
			Have CTF create a loop entry point and associate it with the specified looping label; commonly use to loop thru a set of test instructions	<pre>{
				  "instruction": "BeginLoop",
				  "data": {
				  "label": "loop_1",
				  "conditions": [
				  { "variable": "my_var_1",
				  "compare": "<",
				  "value": 4
				  },
				  { "variable": "my_var_2",
				  "compare": ">",
				  "value": 1
				  }
				  ]
				  }
				}</pre>

control_flow	***EndLoop***	***label:*** user-defined label; must match an existing label from ***BeginLoop*** instruction	Have CTF create a loop exit point and associate it with the specified looping label	<pre>{
				  "instruction": "EndLoop",
				  "data": {
				  "label": "loop_1"
				  }
				}</pre>