

CTF Test Instruction Reference

Associated Plug-Ins	CTF Instructions	Data/Argument Name & Description	Instruction Usage/Application	Examples
cfs	<i>RegisterCfs</i>	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.	{ "instruction": "RegisterCfs", "data": { "target": "lx_cfs_1" } }
cfs	<i>BuildCfs</i>	target: name of a registered cFS target; i.e. executable cFS image	Have CTF build the cFS executable image with the specified name	{ "instruction": "BuildCfs", "data": { "target": "lx_cfs_1" } }
cfs	<i>StartCfs</i>	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	{ "instruction": "StartCfs", "data": { "target": "lx_cfs_1", "run_args": "-R P0" } }
cfs	<i>EnableCfsOutput</i>	target: name of a registered cFS target; optional; default is empty string; if not specified, it applies to <u>all</u> registered cFS targets	Have CTF establish the connection to cFS target for telemetry downlink	{ "instruction": "EnableCfsOutput", "data": { "target": "lx_cfs_1" } }
cfs	<i>SendCfsCommand</i>	target: name of a registered cFS target; optional; default is empty string; if not specified, it applies to <u>all</u> 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)	{ "instruction": "SendCfsCommand", "data": { "target": "lx_cfs_1", "mid": "TO_CMD_MID", "cc": "TO_ENABLE_OUTPUT", "args": { "cdDestIp": "127.0.0.1", "usDestPort": "5011" } } }
cfs	<i>SendInvalidLengthCfsCommand</i>	Same as <i>SendCfsCommand</i> 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 0 prior to sending it to the specified target(s)	{ "instruction": "SendInvalidLengthCfsCommand", "data": { "target": "lx_cfs_1", "mid": "TO_CMD_MID", "cc": "TO_ENABLE_OUTPUT", "payload_length": "10" } }

Associated Plug-Ins	CTF Instructions	Data/Argument Name & Description	Instruction Usage/Application	Examples
cfs	<i>CheckEvent</i>	<p>target: name of a registered cFS target; optional; default is empty string; if not specified, it applies to <u>all</u> registered cFS targets</p> <p>app: the name of the cFS application that sent the event message</p> <p>id: cFS event ID</p> <p>msg: the event message; optional; default is empty string; if specified, CTF performs a string comparison between the expected and actual strings</p> <p>is_regex: boolean flag to denote a regex match instead of string comparison is to be performed on msg ; optional; default is false</p> <p>msg_args: arguments that will be inserted into msg , similar to printf() function; optional; default is empty string</p>	Have CTF verify that a cFS event matching the specified parameters <u>has</u> been received from the specified cFS target	<pre>{ "instruction": "CheckEvent", "data": { "target": "1x_cfs_1", "app": "BEX", "id": "13", "msg": "Processed MODE(%d) Cmd Rcvd", "is_regex": false, "msg_args": "(1)" } }</pre>
cfs	<i>CheckNoEvent</i>	Same as <i>CheckEvent</i>	Have CTF verify that a cFS event matching the specified parameters <u>has not</u> been received from the specified cFS target	<pre>{ "instruction": "CheckNoEvent", "data": { "target": "1x_cfs_1", "app": "TO", "id": "3", "msg": "TO - ENABLE_OUTPUT Cmd Rcvd", "is_regex": false, "msg_args": "" } }</pre>
cfs	<i>CheckTlmValue</i>	<p>target: name of a registered cFS target; optional; default is empty string; if not specified, it applies to all registered cFS targets</p> <p>mid: CCSDS message ID</p> <p>args: array of arguments to be verified; see CTF documentation or example test scripts for additional examples</p>	Have CTF verifies that a telemetry matching the specified parameters <u>has</u> been received from the specified cFS target	<pre>{ "instruction": "CheckTlmValue", "data": { "target": "1x_cfs_1", "mid": "TO_HK_TLM_MID", "args": [{ "variable": "usCmdErrCnt", "compare": "<", "value": "1" }] } }</pre>
cfs	<i>CheckTlmPacket</i>	<p>target: name of a registered cFS target; optional; default is empty string; if not specified, it applies to all registered cFS target</p> <p>mid: CCSDS message ID</p>	Have CTF verifies that a telemetry matching the specified MID <u>has</u> been received from the specified cFS target	<pre>{ "instruction": "CheckTlmPacket", "data": { "target": "1x_cfs_1", "mid": "TO_HK_TLM_MID" } }</pre>

Associated Plug-Ins	CTF Instructions	Data/Argument Name & Description	Instruction Usage/Application	Examples
cfs	<i>CheckNoTlmPacket</i>	<i>target:</i> name of a registered cFS target; optional; default is empty string; if not specified, it applies to all registered cFS target <i>mid:</i> CCSDS message ID	Have CTF verifies that a telemetry matching the specified MID <u>has not</u> been received from the specified cFS target	{ "instruction": "CheckNoTlmPacket", "data": { "target": "lx_cfs_1", "mid": "TO_HK_TLM_MID" } }
cfs	<i>CheckTlmContinuous</i>	Same as <i>CheckTlmValue</i> , with the addition of <i>verification_id:</i> a unique ID to identify this verification <u>within</u> a test case	Similar to <i>CheckTlmValue</i> , except the verification is done every time the telemetry is received, until the test case ends, or the check is removed by <i>RemoveCheckTlmContinuous</i>	{ "instruction": "CheckTlmContinuous", "data": { "target": "lx_cfs_1", "verification_id": "TO_no_errors", "mid": "TO_HK_TLM_MID", "args": { { "variable": "usCmdErrCnt", "compare": "<", "value": "1" } } } }
cfs	<i>RemoveCheckTlmContinuous</i>	<i>verification_id:</i> the verification_id value of an existing <i>CheckTlmContinuous</i>	Have CTF stop performing continuous telemetry check for <i>CheckTlmContinuous</i> matching the <i>verification_id</i> parameter	{ "instruction": "RemoveCheckTlmContinuous", "data": { "verification_id": "TO_no_errors" } }
cfs	<i>ArchiveCfsFiles</i>	<i>target:</i> name of a registered cFS target; optional; default is empty string; if not specified; it applies to all registered cFS targets <i>source_path:</i> 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	{ "instruction": "ArchiveCfsFiles", "data": { "target": "lx_cfs_1", "source_path": "../build/exe/lx_cfs_1/cf/" } }
cfs	<i>ShutdownCfs</i>	<i>target:</i> 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) <u>within</u> the test script. CTF will automatically shutdown cFS targets upon test completion.	{ "instruction": "ShutdownCfs", "data": { "target": "lx_cfs_1" } }
ccsds	<i>ValidateCfsCcsdsData</i>			
ssh	<i>SSH_RegisterTarget</i>	<i>name:</i> 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.	{ "instruction": "SSH_RegisterTarget", "data": { "name": "ssh_tgt_1" } }

Associated Plug-Ins	CTF Instructions	Data/Argument Name & Description	Instruction Usage/Application	Examples
ssh	<i>SSH_InitSSH</i>	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	<pre>{ "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" } }</pre>
ssh	<i>SSH_RunRemoteCommand</i>	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	<pre>{ "instruction": "SSH_RunRemoteCommand", "data": { "name": "ssh_tgt_1", "command": "cd lander_fsw_ctf;/rm -rf build; make; make install;" } }</pre>
ssh	<i>SSH_RunLocalCommand</i>	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	<pre>{ "instruction": "SSH_RunLocalCommand", "data": { "name": "ssh_tgt_1", "command": "cd lander_fsw_ctf;/rm -rf build; make; make install;" } }</pre>
ssh	<i>SSH_CheckOutput</i>	name: a registered SSH target output_contains: a substring that must be contained in the output output_does_not_contain: a substring that should <u>not</u> 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	<pre>{ "instruction": "SSH_CheckOutput", "data": { "name": "ssh_tgt_1", "output_contains": "Built target mission-install", "output_does_not_contain": "Error", "exit_code": 0 } }</pre>
ssh	<i>SSH_PutFile</i>	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	<pre>{ "instruction": "SSH_PutFile", "data": { "name": "ssh_tgt_1", "local_path": "./cfs", "remote_path": "/tmp/workspace/cfs", "args": { "delete": true, "exclude": "*.git" } } }</pre>

Associated Plug-Ins	CTF Instructions	Data/Argument Name & Description	Instruction Usage/Application	Examples
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	<pre>{ "instruction": "SSH_GetFile", "data": { "name": "ssh_tgt_1", "local_path": "./results.txt", "remote_path": "./data/output.dat" } }</pre>
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 CTF download file/directory from a remote directory via FTP	<pre>{ "instruction": "SSH_GetFTP", "data": { "name": "ssh_tgt_1", "host": "ftphost", "local_path": "./results.txt", "remote_path": "./data/output.dat" } }</pre>
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	<pre>{ "instruction": "SSH_PutFTP", "data": { "name": "ssh_tgt_1", "host": "ftphost", "local_path": "./results.txt", "remote_path": "./data/output.dat" } }</pre>
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>

Associated Plug-Ins	CTF Instructions	Data/Argument Name & Description	Instruction Usage/Application	Examples
variable	<i>CheckUserVariable</i>	<i>variable_name</i> : user-defined variable name <i>operator</i> : comparison operator to apply to variable <i>value</i> : 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	<i>BeginLoop</i>	<i>label</i> : user-define label; must be unique within the scope of the test case <i>conditions</i> : a list of comparison conditions <i>condition.name</i> : name of an existing user-defined variable <i>condition.operator</i> : comparison operator <i>condition.value</i> : 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	<i>EndLoop</i>	<i>label</i> : user-defined label; must match an existing label from <i>BeginLoop</i> instruction	Have CTF create a loop exit point and associate it with the specified looping label	<pre>{ "instruction": "EndLoop", "data": { "label": "loop_1" } }</pre>