

Technical Note

A Possible Approach For The CCS – IEGSE Telecommand Interface

Author: Erich Wiezorrek

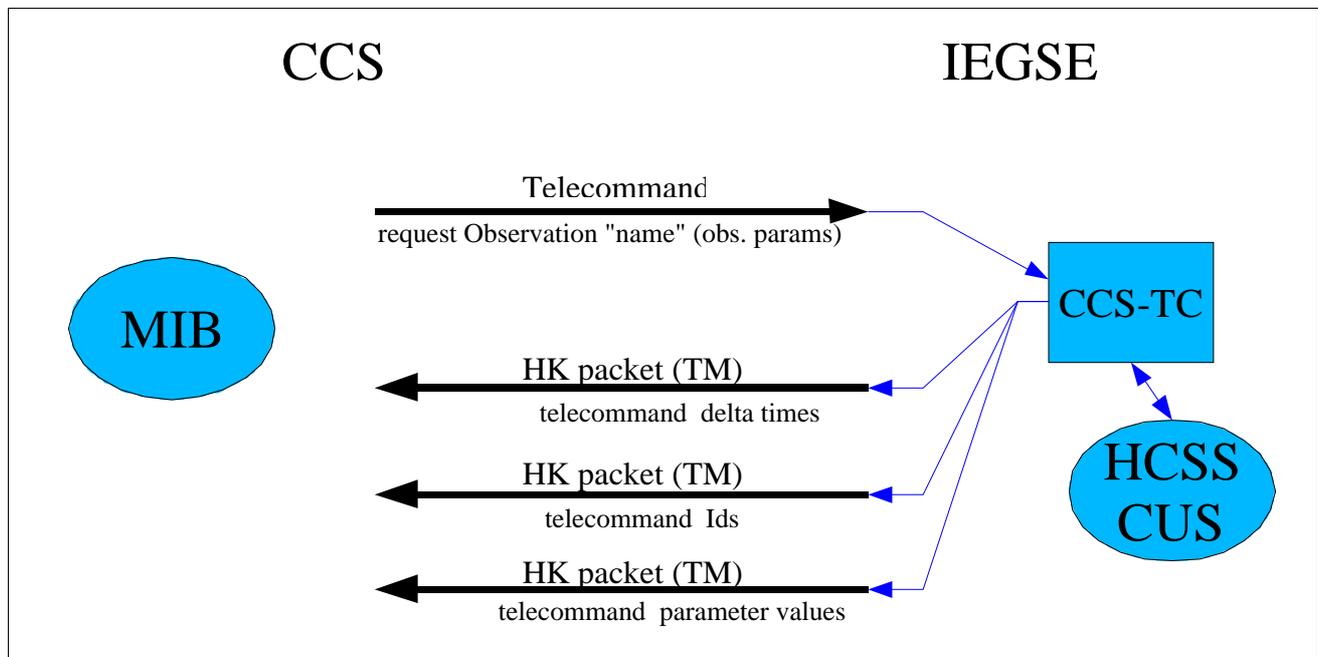
Date: February 28, 2003

Scope

This technical note describes an approach how the CCS can execute a predefined test script holding the complete telecommand sequence but still allowing the IEGSE to provide the values for the telecommand parameters, telecommand Ids and the times to wait between the telecommands.

Concept

The CCS will send a TC packet (via the PIPE protocol) to the IEGSE requesting the TC parameters for a named observation plus its parameters. The IEGSE will use the HCSS system (in particular the CUS) to instantiate the corresponding observation. The IEGSE will retrieve the telecommand parameters, telecommand Ids and the delta times from the HCSS and send them coded in IEGSE HK TM packets back to the CCS (again via the PIPE protocol).



All the necessary communication steps between the CCS and IEGSE are part of the delivered test script:

- the IEGSE telecommand request
- the wait statement to wait for the arrival of all IEGSE HK packets

- the retrieval of the TC parameters, Ids and delta times from the IEGSE HK packets

Functionality requested by the CCS TOPE implementation

- Capability to send telecommands to the EGSE with variable string parameters
- Capability to wait for the reception of TM packets
- Capability to read parameters from TM packets
- Capability to set the (external) telecommand request ID
- Capability to wait a specified delta time

Prototyping

To prove the concept the MIB files describing the IEGSE TC and TM packets were prototyped and ingested into the PACS ILT EGSE SCOS 2000 system. The attached test script example was executed using the TOPE subsystem. The arrival of the IEGSE TC packet was confirmed using a special TC packet dumping utility. The corresponding IEGSE HK packets were generated using a general packet generation utility. The resulting TC sequence from the test script was checked using the SCOS 2000 telecommand history application.

Deliveries by the instrument teams

The instrument teams will deliver the MIB describing the IEGSE TC and TM packets. This MIB will be the same for all three instrument teams.

For each test the instrument teams will deliver a test script containing

- a) the TC sequence itself
- b) all TOPE commands needed for the communication between CCS and IEGSE

Open points

- The test script example attached to this note is written in the TOPE syntax known by the IEGSEs. It must be converted to the CCS format.
- It is assumed that the TOPE installation at the CCS knows the TCrequestID argument of the TCsend (or a similar) command. This TCrequestID argument is used to transfer the external TC ID to SCOS 2000 (known since SCOS 2000 version 2.3e).
- The MIB used for prototyping is not following yet the naming conventions.
- Due the length limitation for Herschel telecommands the observation requesting IEGSE TC is limited to 242 characters.
- Due the length limitation for Herschel telemetry packets (1006 bytes in the source data field) the transfer of the resulting telecommand parameters has to be broken into several TM packets.
- The way how parameters are passed to the test script is not known yet. One way could be that the test script is defined as TCL procedure which later can be called with arguments

Test script example

```
#
# make sure the variable set obsid2 doesn't exist
# catch errors because it may not exist
#
catch {Unsubscribe obsid2}

#
# trigger observation generation by sending a TC to IEGSE
# TC parameters have to be terminated with a semicolon because
# TOPE/S2K does not prefix the parameters with the number of characters.
# So the semicolon is used to parse the telecommand.
#
TCsend ZCREQOBS {ZPOBSMOD "IntegrationTest;"} \
  {ZPOBSNOP 2} \
  {ZPOBSPAR "bbid_offset=2;"} \
  {ZPOBSPAR "dummy=null"}

#
# subscribe to the OBSID value of the TC parameter packet (which is the last one)
# wait for the variable obsid2_raw to be created
#
Subscribe ZMCPOLID2 ReferBy obsid2 Whenever UPDATE_raw
vwait obsid2_raw

#
# all packets are arrived, fetch all the other
# TC parameters
# delta times to wait between sending the TC
# TC Ids
#
Fetch ZMCPL000 into bbID1
Fetch ZMCPL001 into bbID2
Fetch ZMCPL002 into bbID3
Fetch ZMCDT000 into dtime0
Fetch ZMCDT001 into dtime1
Fetch ZMCDT002 into dtime2
Fetch ZMCDT003 into dtime3
Fetch ZMCID000 into tcId0
Fetch ZMCID001 into tcId1
Fetch ZMCID002 into tcId2
Fetch ZMCID003 into tcId3

Fetch ZMCPOLID0 into obsid0
Fetch ZMCPOLID1 into obsid1

puts "ObsIds: $obsid0_raw $obsid1_raw $obsid2_raw"
puts "delta times: $dtime0_raw $dtime1_raw $dtime2_raw $dtime3_raw"
puts "TcIds: $tcId0_raw $tcId1_raw $tcId2_raw $tcId3_raw"
puts "TC parameters: $bbID1_raw $bbID2_raw $bbID3_raw"

#
# finally execute the list of TCs with the actual parameters
#
TCsend ZCOBSID TCrequestID $tcId0_raw [list ZPOBSID $obsid2_raw]
WaitTime $dtime0_raw

TCsend ZCBPID TCrequestID $tcId1_raw [list ZPBPID $bbID1_raw]
WaitTime $dtime1_raw

TCsend ZCBPID TCrequestID $tcId2_raw [list ZPBPID $bbID2_raw]
WaitTime $dtime2_raw

TCsend ZCBPID TCrequestID $tcId3_raw [list ZPBPID $bbID3_raw]
WaitTime $dtime3_raw
```