CCSDS Mission Operations Action Service Core Capabilities

Manned Space Flight

Steve Lucord

Operations Concepts

- Consequences of sending the wrong command (action) are unacceptable
- Command provides a collaborative and distributed work environment for flight controllers and operators.
- Command prescribes a review and approval process where each command is viewed by other individuals before being sent to vehicle.

Introduction

- The Action Service needs additional capabilities to support the operations concepts of manned space flight.
  - Action Service methods
  - Action attributes
  - Action parameter/argument attributes
  - Support for dynamically maintained action data.
  - Publish-subscribe capabilities
- OTF may provide necessary support for any modifications.

Static Action Data

Action Attributes				
Action Identifier	Description	Severity	Default Safe State	Default Safe Group

Action Parameter Attributes						
Parameter Identifier 1	Action Identifier	Description	Data Type	Engineering Units	Lower Limit	Upper Limit
Parameter Identifier N	Action Identifier	Description	Data Type	Engineering Units	Lower Limit	Upper Limit

Dynamic Action Data

Action Instance					
Instance Identifier	Action Identifier	Safe State	Armed State	Safe Group	Time Trigger

Action Instance Parameter		
Parameter Identifier 1	Instance Identifier	Value
Parameter Identifier N	Instance Identifier	Value

Action Service Methods

- preCheckAction
  - No corresponding capability in operations concept based on current definition
  - May need the capability to return the formatted data packet of what would be sent to the vehicle if action was invoked.
- invokeAction
  - Provides a fraction of required capabilities in the operations concepts.
  - Needs capability to specify a time trigger to indicate when action will execute on board the vehicle.

Action Attributes

- Additional Action Attributes
  - Enabled/Inhibited state. No command (action) may be invoked in the inhibited state.
  - Arm/Disarm state. A command (action) must be put in the armed state before being invoked. Only enabled commands may be armed.
  - Owner. A command (action) maintains the position that is allowed to perform the invoke.

Action Parameter Attributes

- Parameter Attributes
  - Parameter type. For example, string, integer or floating point.
  - Engineering Units.
  - Upper and Lower operational limits.

Action Instances

- Pre-built Instance
  - All parameter arguments are constant values
  - Each instance is assigned a unique identifier at initialization of the Action Service.
  - Time trigger application dynamically creates new pre-built instance of the action. A unique identifier for the instance is assigned.
  - Each instance maintains its own safe and armed state.

Action Instances

- User-built Instance
  - User-built instances are dynamically created, replaced or deleted by the user.
  - A unique identifier is assigned to each user-built instance at creation time.
  - User provides parameter arguments when the instance is created or replaced.
  - Time trigger application dynamically creates new user built instance or may be applied at creation time.
  - Each user-built instance maintains its own safe and armed state.

Invoke Action

- Invoke Pre-Built Action Instance
  - The unique identifier is provided as an argument to the invoke action event.
  - The user may provide the time trigger as an additional argument to the invoke action event.
  - The time triggered is applied to the data packet that is sent to the vehicle; however, it is not maintained as part of the pre-built instance data.

Invoke Action Continued

- Invoke User-Built Action Instance
  - The unique identifier is provided as an argument to the invoke action event.
  - The user may provide the parameter values as event arguments. The arguments are applied to the data packet that is sent to the vehicle; however, a user-built instance is not created.
  - The user may provide the time trigger and parameter values as event arguments. The same rules as before apply.
  - The user may provide the unique identifier and time trigger as event arguments. The time trigger is applied to the data packet that is sent to the vehicle but is not maintained as part of the user-built instance attributes.

Action Packages

- Action packages consist of one or more action instances. The data for each command is formatted into one data packet that is sent to the vehicle.
- Action package may be created or deleted. A unique identifier is assigned at creation time.
- Action instances may be added or removed from a packet.
- The order of the instances within a package may be changed.
- The unique package identifier is provided as an event argument to invoke an action package.

Safe Groups

- Safe groups permit the safe state of multiple action instances to be modified with one event.
- A safe group contains the name, description and safe state attributes along with a list of the action instances.
- Action instances may be added or removed from a safe group.
- A user may change the safe state (Enabled/Inhibited) of a group to Enabled. All the action instances in the group are set to the specified value.

Additional Action Service Methods

- Actions
  - buildAction: This method creates a user-built instance of an action. The parameter arguments and time trigger may be provided.
  - replaceAction: This method replaces the parameter arguments and or time trigger for a user-built instance.
  - deleteAction: This method deletes a user-built or pre-built instance with a time trigger.
  - setActionTimeTrigger: This methods sets the execution time for a pre-built or user-built instance of an action.

Additional Action Service Methods

- Action Package
  - createActionPackage: This method creates an action package. The instances to add may be provided as arguments.
  - replaceActionPackage: This method replaces and existing action package. The action instances may be provided as arguments to add, delete or change the order.
  - deleteActionPackage: This method removes an existing action package.
  - invokeActionPackage: This method sends the action package contents to the vehicle

Additional Action Service Methods

- createSafeGroup: This method creates a new safe group. The name and description are provided.
- updateSafeGroup: This method changes the state of the safe group. The state of the action instances in the group are modified. The name and description may be updated also.
- deleteSafeGroup: This method deletes a safe group.
- addSafeGroupAction: This method adds a new action instance to an existing safe group.
- removeSafeGroupAction: This method removes an action instance from an existing safe group.

Monitoring Action State

- Registration for the static Action Definition data is necessary.
  - Filter on Owner
- Pub-sub is necessary for the dynamic data that is maintained for the action instances.
  - Filter on the Enabled/Inhibited state
  - Filter on the Arm/Disarm state
  - Filter on Owner
- Pub-sub is needed for the dynamic action packet data
- Pub-sub is needed for the dynamic safe group data.

Conclusions

- Action Service needs additional capabilities to support manned space flight operations.
- Modifications may be added to the Core Services Blue Book or a separate blue book may be created
- OTF may provide necessary support

Contacts

- Management
  - Lindolfo Martinez (281) 483-4346 / 2099
  - lindolfo.martinez-1@nasa.gov
- Technical Lead
  - Steve Lucord (281) 483-9711 / 2099
  - steven.a.lucord@nasa.gov
- Project Sponsors
  - Eric Wolfer (281) 483-6709 / 2014AA
  - eric.j.wolfer@nasa.gov

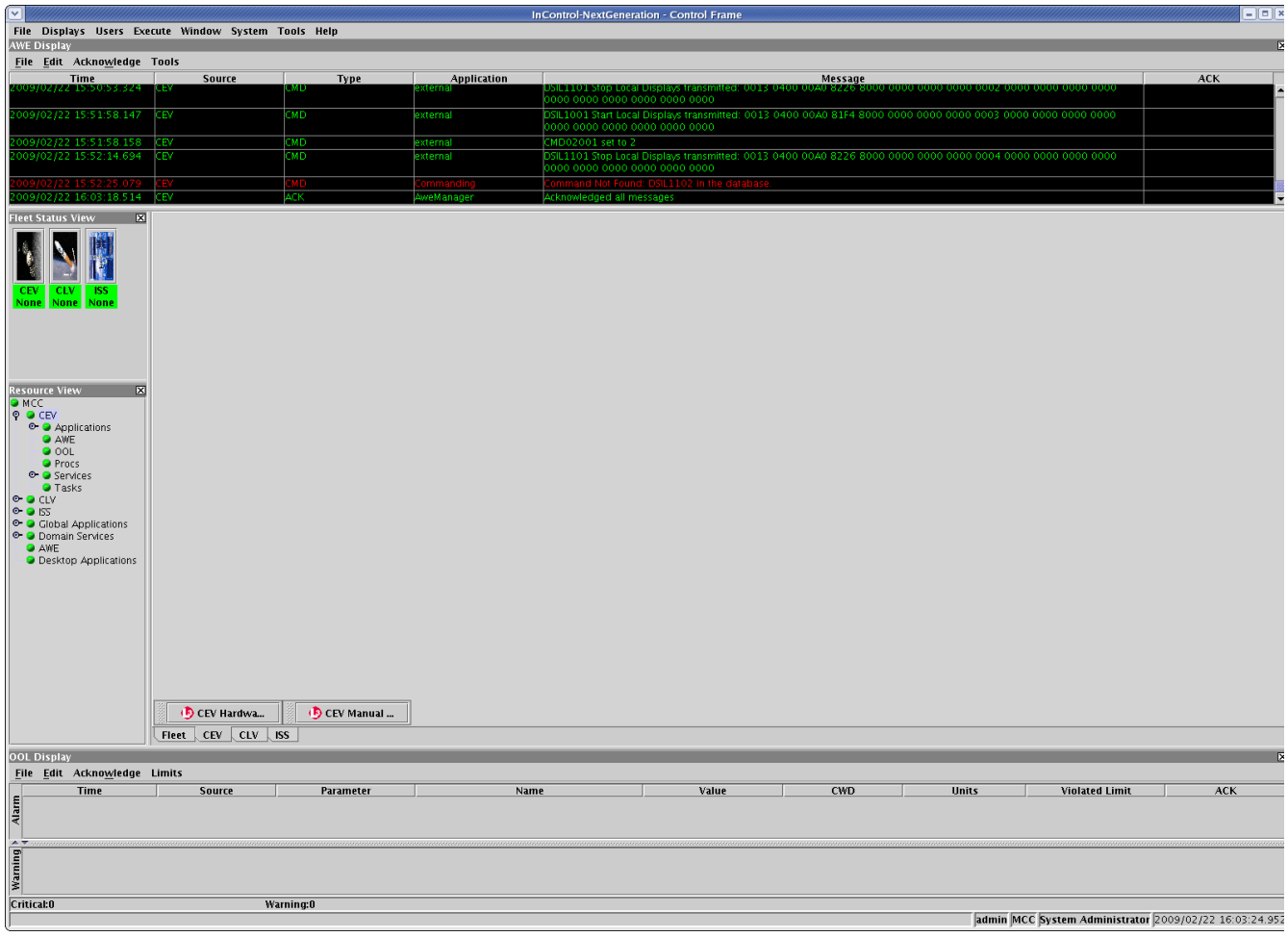
Thank You

- Questions

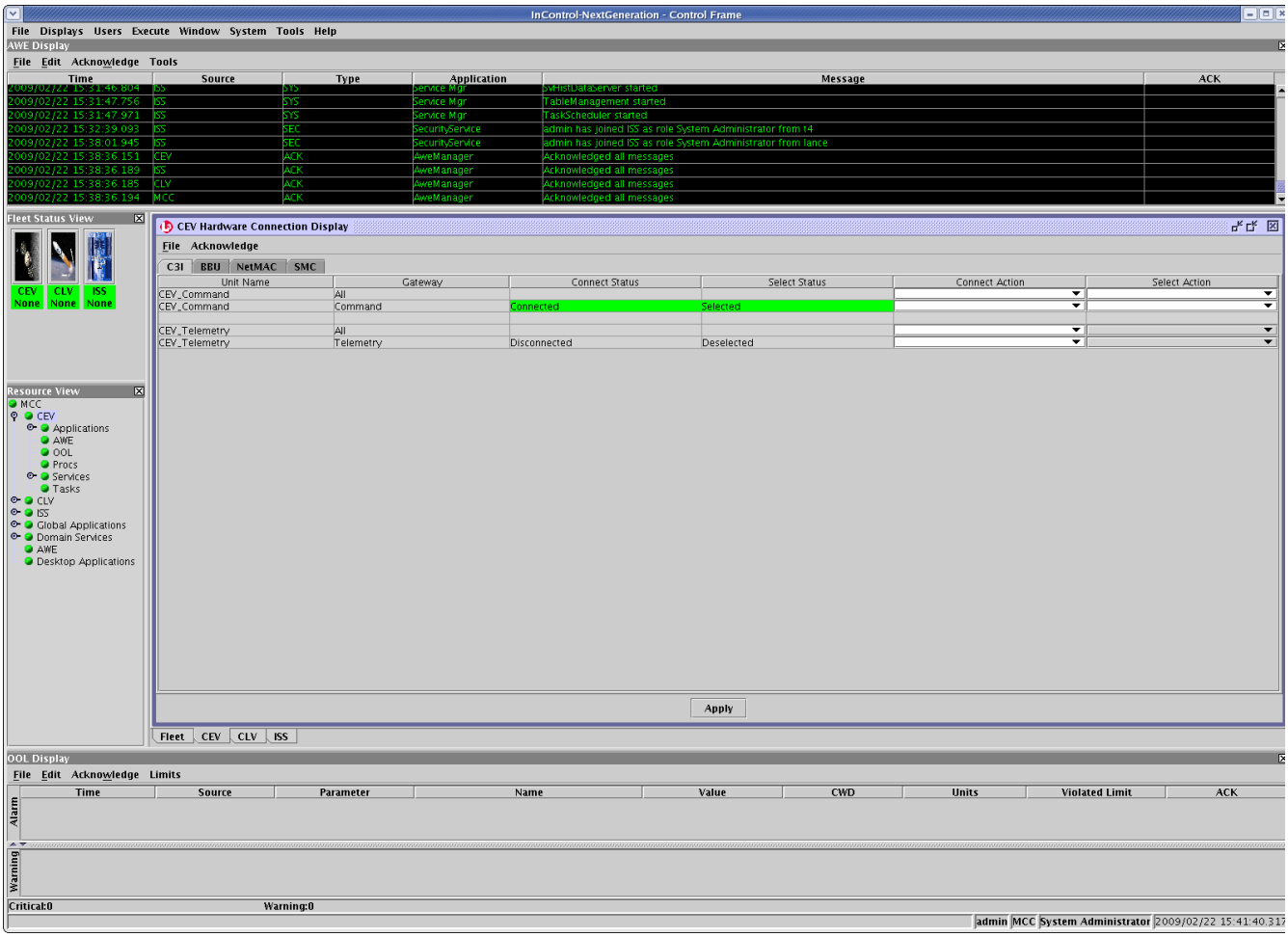
CCSDS Mission Operations
Action Service Demonstration

Screen Shots

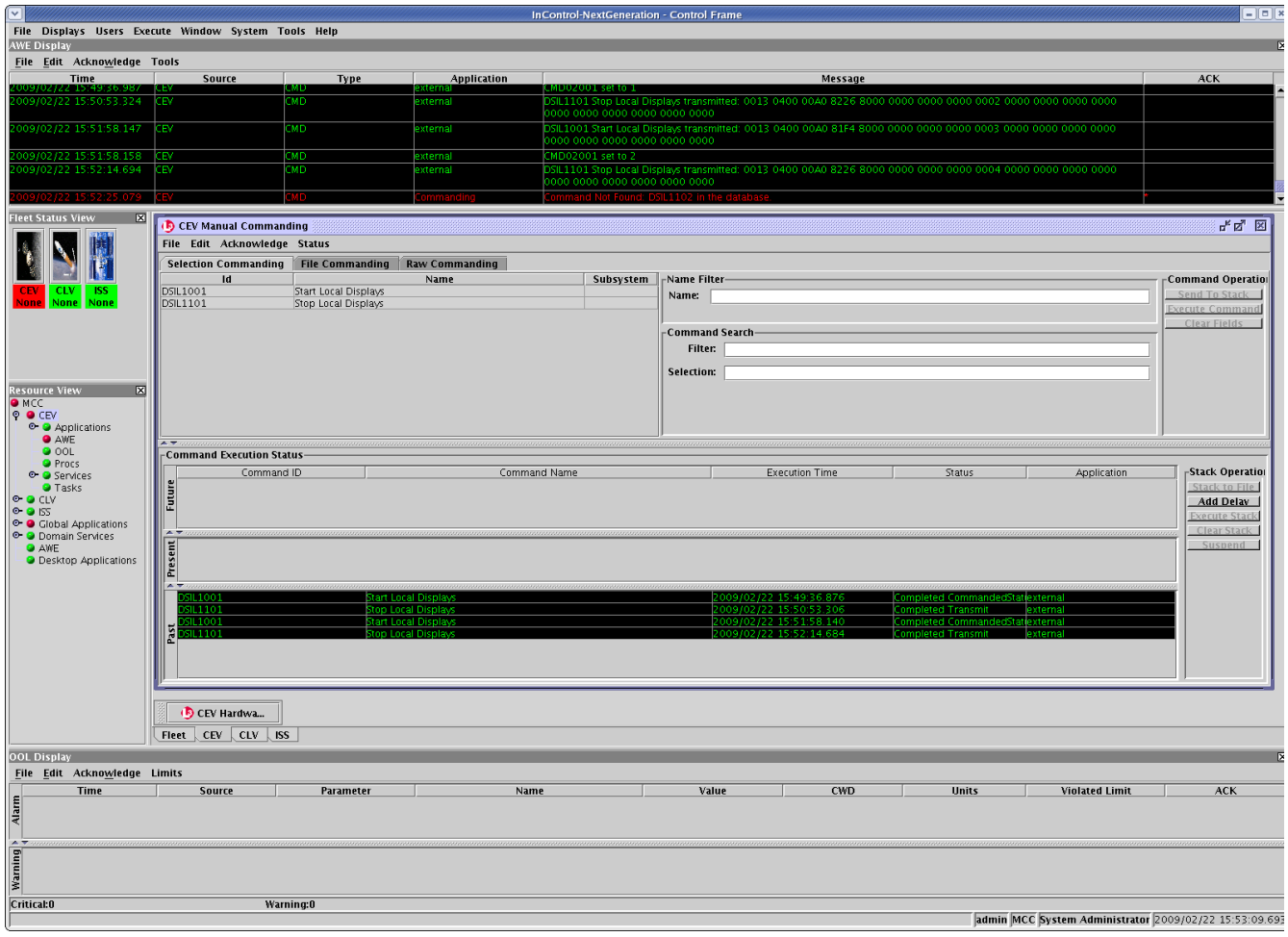
1. L3 In Control Status Frame Display



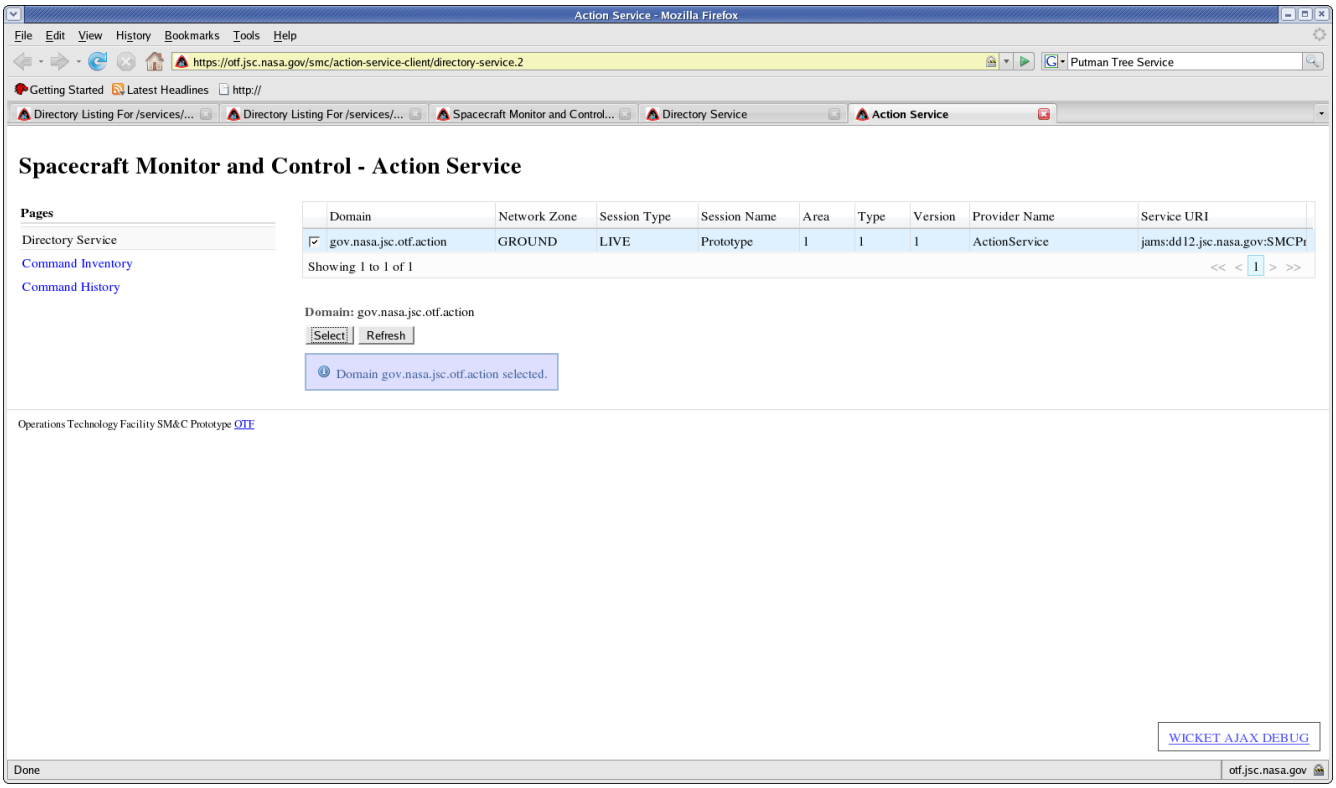
2. L3 In Control Hardware Connection Display



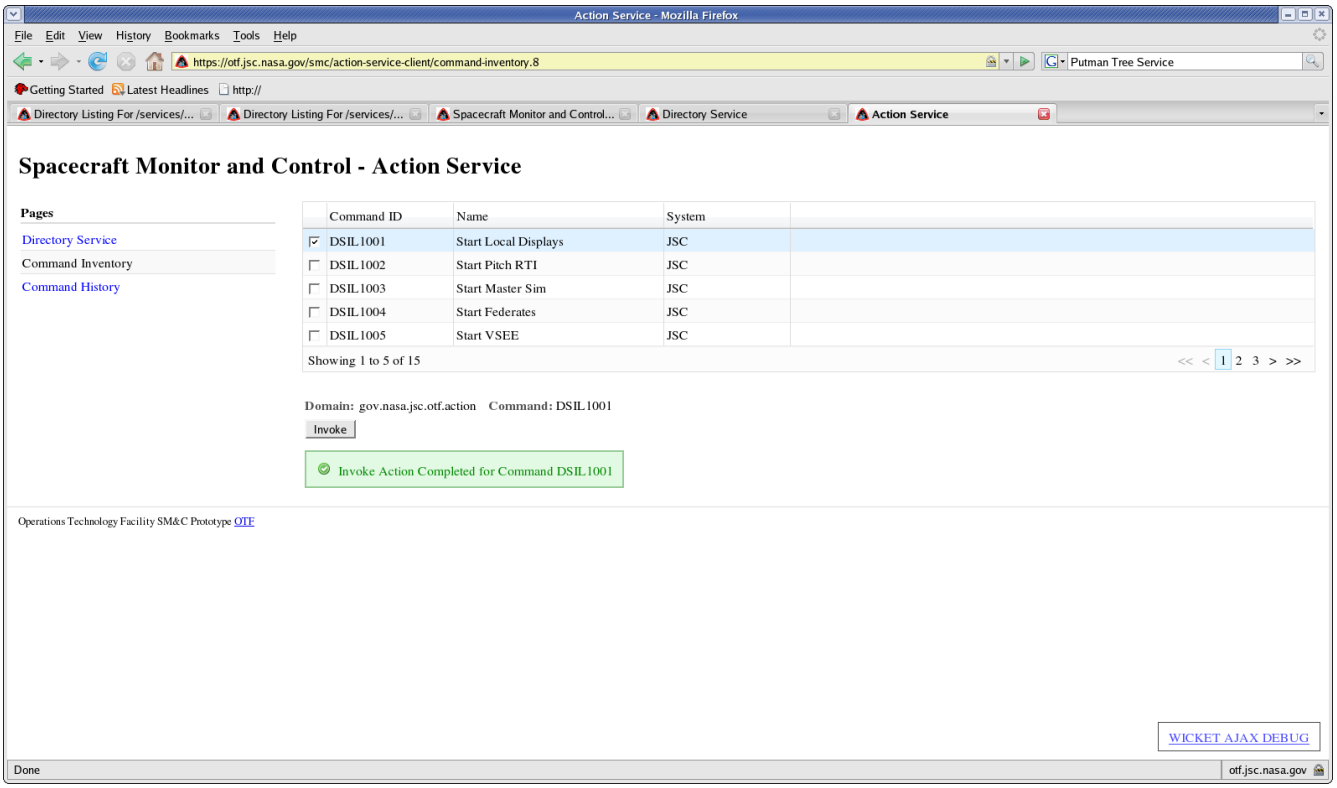
3. L3 In Control Manual Commanding Display



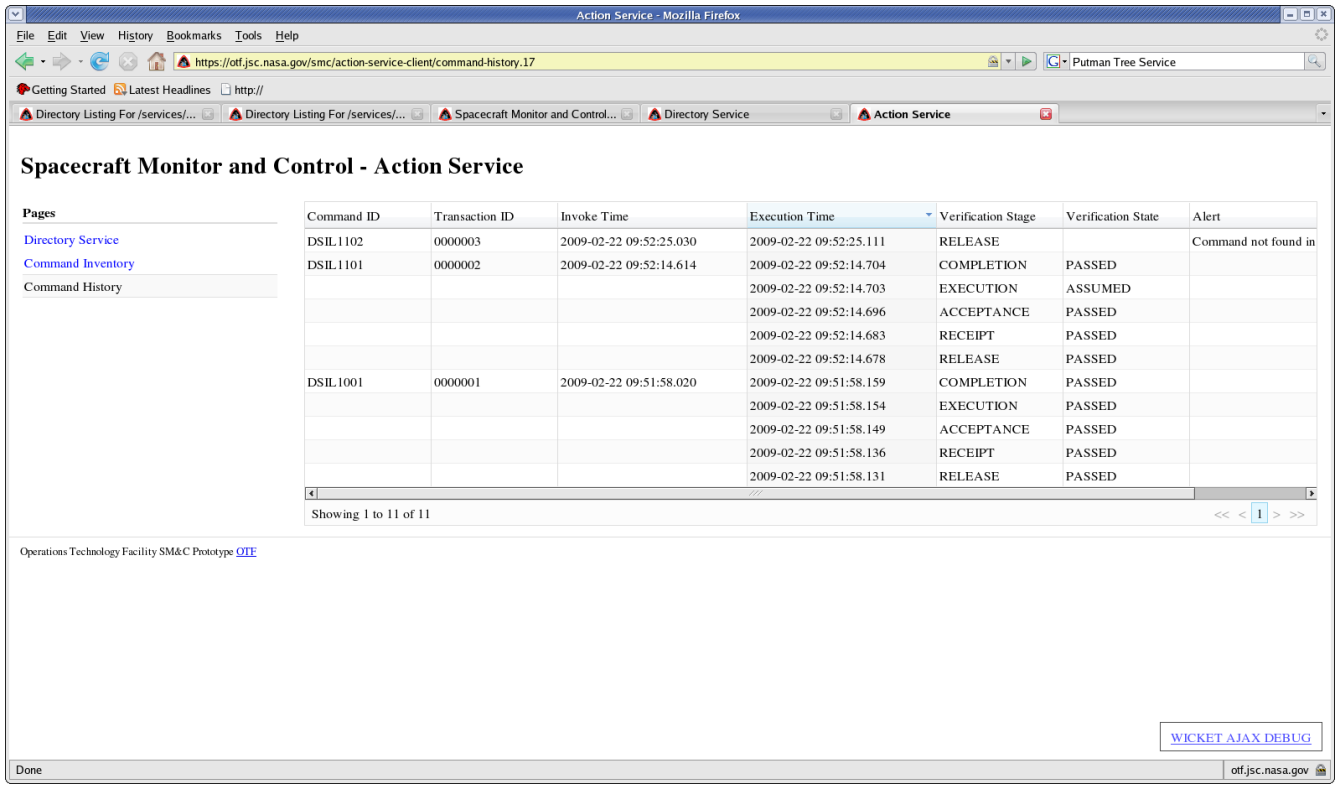
4. Action Service Directory Page



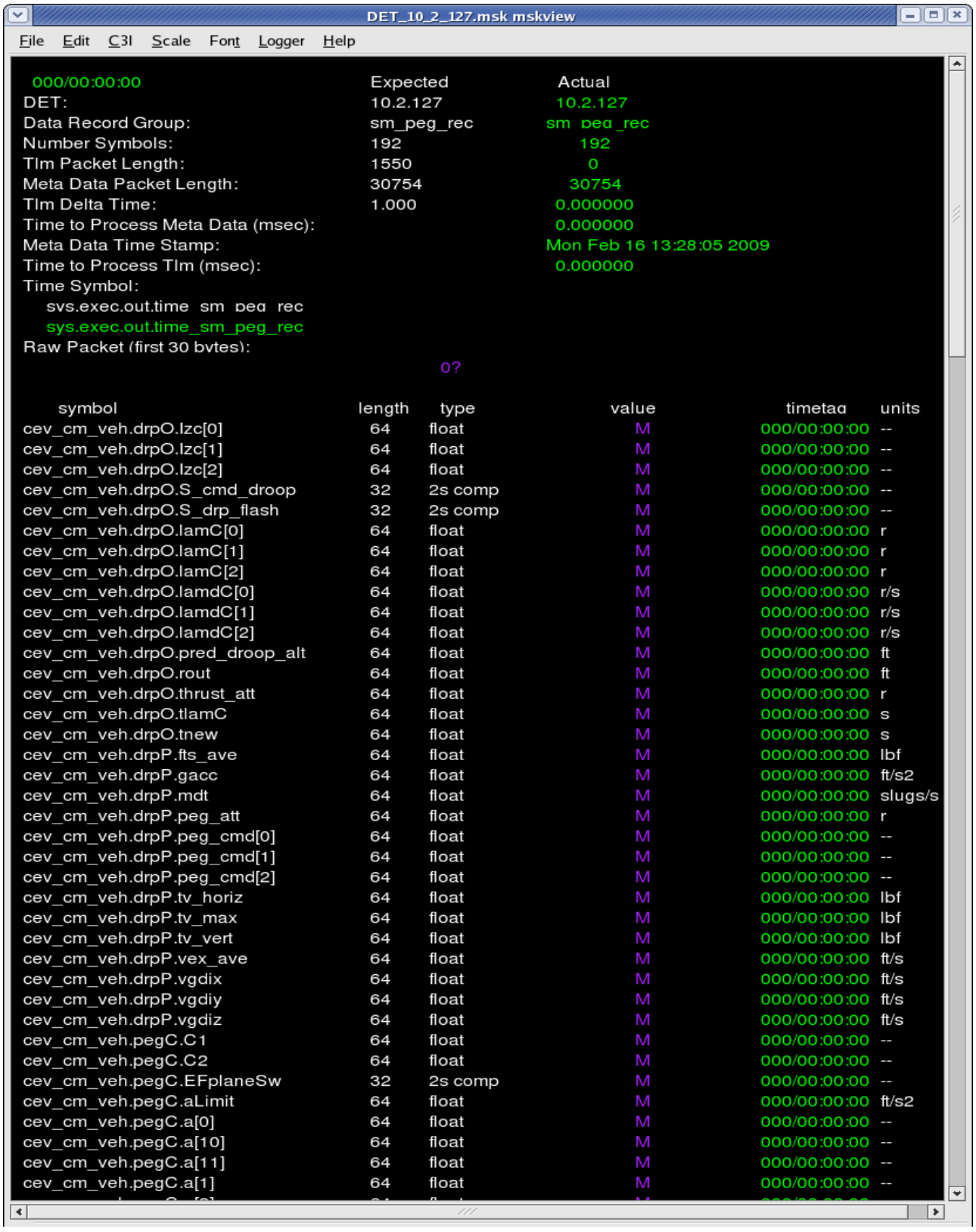
5. Action Service Inventory

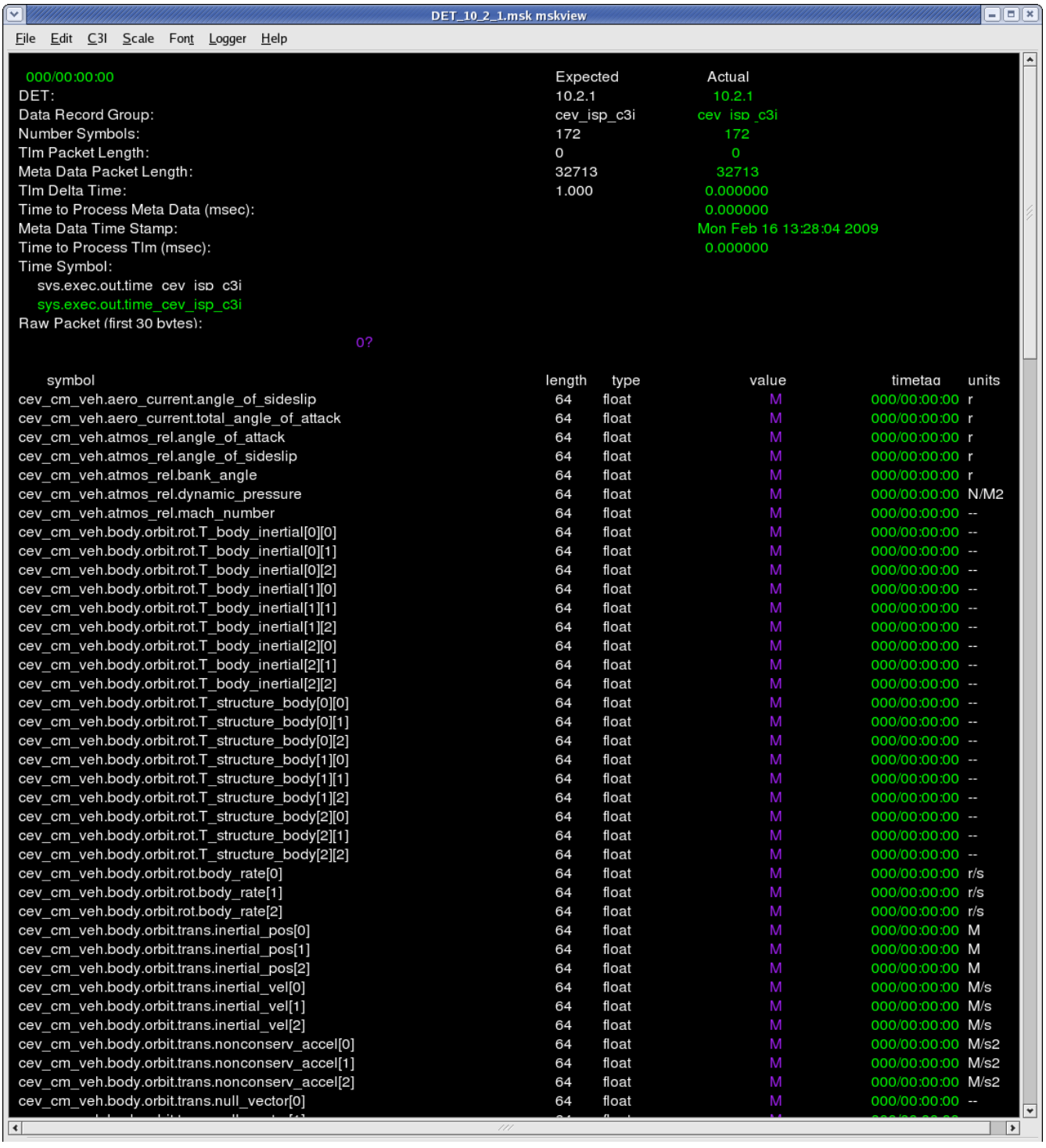


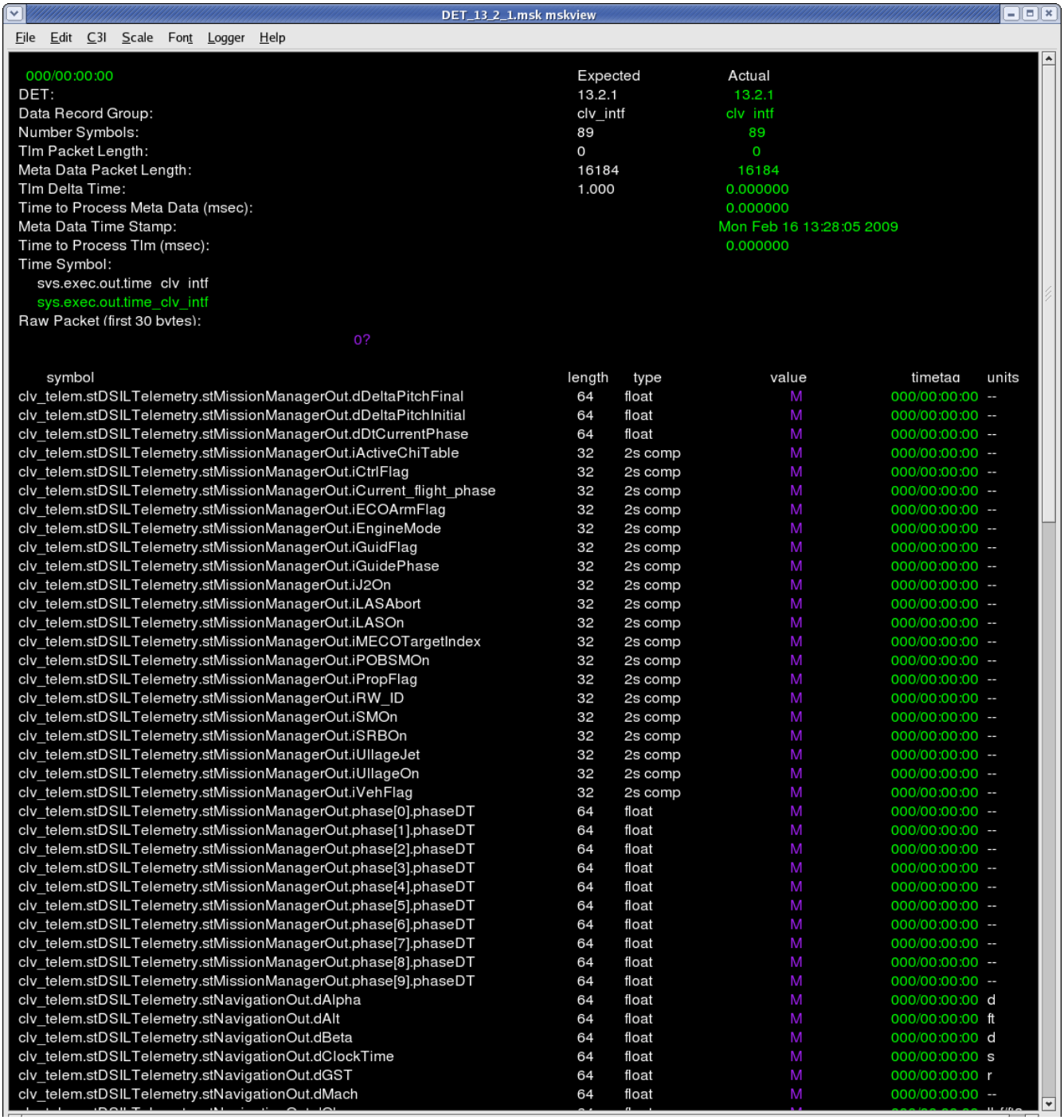
6. Action Service History



7. Test Telemetry Displays




DET_10_2_1.msk mskview
File Edit C3I Scale Font Logger Help
000/00:00:00
Expected
Actual
DET:
10.2.1
10.2.1
Data Record Group:
cev_isp_c3i
cev_isp_c3i
Number Symbols:
172
172
Tlm Packet Length:
0
0
Meta Data Packet Length:
32713
32713
Tlm Delta Time:
1.000
0.000000
Time to Process Meta Data (msec):
0.000000
Meta Data Time Stamp:
Mon Feb 16 13:28:04 2009
Time to Process Tlm (msec):
0.000000
Time Symbol:
sys.exec.out.time_cev_isp_c3i
sys.exec.out.time_cev_isp_c3i
Raw Packet (first 30 bytes):
0?
symbol
length
type
value
timetag
units
cev_cm_veh.aero_current.angle_of_sideslip 64 float M 000/00:00:00 r
cev_cm_veh.aero_current.total_angle_of_attack 64 float M 000/00:00:00 r
cev_cm_veh.atmos_rel.angle_of_attack 64 float M 000/00:00:00 r
cev_cm_veh.atmos_rel.angle_of_sideslip 64 float M 000/00:00:00 r
cev_cm_veh.atmos_rel.bank_angle 64 float M 000/00:00:00 r
cev_cm_veh.atmos_rel.dynamic_pressure 64 float M 000/00:00:00 N/M2
cev_cm_veh.atmos_rel.mach_number 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_body_inertial[0][0] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_body_inertial[0][1] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_body_inertial[0][2] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_body_inertial[1][0] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_body_inertial[1][1] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_body_inertial[1][2] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_body_inertial[2][0] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_body_inertial[2][1] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_body_inertial[2][2] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_structure_body[0][0] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_structure_body[0][1] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_structure_body[0][2] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_structure_body[1][0] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_structure_body[1][1] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_structure_body[1][2] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_structure_body[2][0] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_structure_body[2][1] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.T_structure_body[2][2] 64 float M 000/00:00:00 --
cev_cm_veh.body.orbit.rot.body_rate[0] 64 float M 000/00:00:00 r/s
cev_cm_veh.body.orbit.rot.body_rate[1] 64 float M 000/00:00:00 r/s
cev_cm_veh.body.orbit.rot.body_rate[2] 64 float M 000/00:00:00 r/s
cev_cm_veh.body.orbit.trans.inertial_pos[0] 64 float M 000/00:00:00 M
cev_cm_veh.body.orbit.trans.inertial_pos[1] 64 float M 000/00:00:00 M
cev_cm_veh.body.orbit.trans.inertial_pos[2] 64 float M 000/00:00:00 M
cev_cm_veh.body.orbit.trans.inertial_vel[0] 64 float M 000/00:00:00 M/s
cev_cm_veh.body.orbit.trans.inertial_vel[1] 64 float M 000/00:00:00 M/s
cev_cm_veh.body.orbit.trans.inertial_vel[2] 64 float M 000/00:00:00 M/s
cev_cm_veh.body.orbit.trans.nonconserv_accel[0] 64 float M 000/00:00:00 M/s2
cev_cm_veh.body.orbit.trans.nonconserv_accel[1] 64 float M 000/00:00:00 M/s2
cev_cm_veh.body.orbit.trans.nonconserv_accel[2] 64 float M 000/00:00:00 M/s2
cev_cm_veh.body.orbit.trans.null_vector[0] 64 float M 000/00:00:00 --


DET_13_2_1.msk mskview
File Edit C3I Scale Font Logger Help
000/00:00:00 Expected Actual
DET: 13.2.1 13.2.1
Data Record Group: clv_intf clv_intf
Number Symbols: 89 89
Tlm Packet Length: 0 0
Meta Data Packet Length: 16184 16184
Tlm Delta Time: 1.000 0.000000
Time to Process Meta Data (msec): 0.000000
Meta Data Time Stamp: Mon Feb 16 13:28:05 2009
Time to Process Tlm (msec): 0.000000
Time Symbol:
svs.exec.out.time clv intf
sys.exec.out.time_clv_intf
Raw Packet (first 30 bytes):
0?
symbol length type value timetag units
clv_telem.stDSILTelemetry.stMissionManagerOut.dDeltaPitchFinal 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.dDeltaPitchInitial 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.dDtCurrentPhase 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iActiveChiTable 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iCtrlFlag 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iCurrent_flight_phase 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iECOArmFlag 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iEngineMode 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iGuidFlag 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iGuidePhase 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iJ2On 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iLASAbort 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iLASOn 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iMECOTargetIndex 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iPOBSMOn 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iPropFlag 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iRW_ID 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iSMOn 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iSRBOn 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iUllageJet 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iUllageOn 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.iVehFlag 32 2s comp M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.phase[0].phaseDT 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.phase[1].phaseDT 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.phase[2].phaseDT 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.phase[3].phaseDT 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.phase[4].phaseDT 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.phase[5].phaseDT 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.phase[6].phaseDT 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.phase[7].phaseDT 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.phase[8].phaseDT 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stMissionManagerOut.phase[9].phaseDT 64 float M 000/00:00:00 --
clv_telem.stDSILTelemetry.stNavigationOut.dAlpha 64 float M 000/00:00:00 d
clv_telem.stDSILTelemetry.stNavigationOut.dAlt 64 float M 000/00:00:00 ft
clv_telem.stDSILTelemetry.stNavigationOut.dBeta 64 float M 000/00:00:00 d
clv_telem.stDSILTelemetry.stNavigationOut.dClockTime 64 float M 000/00:00:00 s
clv_telem.stDSILTelemetry.stNavigationOut.dGST 64 float M 000/00:00:00 r
clv_telem.stDSILTelemetry.stNavigationOut.dMach 64 float M 000/00:00:00 --