Subject: ETS MPS/Aura Engineering Release 3.3 Delivery
Date: Thu, 24 Oct 2002 18:11:47 -0400
From: Ernest Quintin <equintin@csc.com>
To: wfuller@pop500.gsfc.nasa.gov

Willie,

We are pleased to deliver Release 3.3 of the ETS Multimode Portable Simulator (MPS) for Aura. This engineering release delivery contains code enhancements that enable the MPS/Aura simulator to connect to the EOC Training Simulator Facility (ETSF) via the 1553 Bus so as to act as an instrument emulator, the addition of a Serial I/O mode to the ETSF Front End capability, and coding corrections to answer Discrepancy Reports (DRs) ETS0465 and ETS0467. Complete descriptions of the changes and enhancements are contained in the attachments. Because of delays, the ETSF 1553 Bus interface capability has not yet been interface tested.

There are seven attachments to this letter.

Attachment A describes the capabilities included in this release.
Attachment B describes installation instructions for this release.
Attachment C describes special operating instructions for this release.
Attachment D contains the resolved DR descriptions
Attachment E contains the system limitations.
Attachment F contains an updated release history summary matrix.
Attachment G contains an updated Mission Systems Configuration Management (MSCM) form.

Attachment C is being delivered as a zip file because its size might overwhelm some mail systems. The updated software executable modules are being delivered on CD-ROM. Two copies of the CD will be given to Guy Cordier, who will forward one copy to Raytheon at Denver and will use the other for installation on the MPS simulator PCs in Building 32. A PC to act as the ETSF Front End is on order.

The updated software is also being installed on the serial card-equipped PCs in the Bldg 25 Simulations Operations Center and on the portable PCs, in the event that any of those units are needed to support upcoming Aura data flows.

The System User's Guide has been updated to include the new capabilities and is undergoing review. It should be available for distribution next week.

If you have any questions about this delivery, please do not hesitate to contact me or Estelle Noone.

Ernest Quintin
301-805-3649

CC: alan.johns@gsfc.nasa.gov, Alexander.Krimchansky.1@gsfc.nasa.gov, gcordier@rattler-e.gsfc.nasa.gov, kcklem@west.raytheon.com, arhabeger@west.raytheon.com, rmesserly@mail.gblt.averstar.com, vhruland@west.raytheon.com, "Pat Burrows" pburrows@csc.com>, "Jean Carlson" <jcarlson@csc.com>, klavery@pop500.gsfc.nasa.gov, eshurie@pop400.gsfc.nasa.gov, "Estelle Noone" <enoone@csc.com>, rwinters@mail.gblt.averstar.com, diana.sexton@averstar.com, jguida@pop500.gsfc.nasa.gov

Attachment A – Summary of Operational Changes

Operational Capabilities of MPS/Aura Release 3.3

New or modified capabilities with this release are noted in **Bold**.

Telemetry:

- Transmit telemetry in IP or Serial (clock/data) mode
- Pack telemetry packets and CLCWs into CADUs when in Serial mode
- Generate one stream of CADUs when in Serial mode
- Generate one stream of telemetry formatted as EDUs when in IP mode
- Start or stop one telemetry stream
- Generate telemetry packets from information contained in the PDB
- Maintain telemetry nodes from information contained in the PDB
- Populate telemetry packets with data values from information contained in the PDB
- Generate correct secondary headers for SC, GIRD, and SUROM-TIE (no secondary header) telemetry packets using information from the PDB
- Generate instrument telemetry packets using secondary key information from the PDB
- Display EDU data when in IP mode
- Display CADU data when in Serial mode
- Set values into telemetry points by mnemonic
- Display telemetry node values by mnemonic
- Convert telemetry values to Engineering Units (EU) for display using information from the PDB
- Accept operator-entered telemetry values in EU and convert to Raw Counts for inclusion in telemetry packets
- Reset packet count for the telemetry stream
- Static packet data can be overwritten (by byte location) and by modification of telemetry mnemonic
- Incrementing packet sequence counters per APID
- Generation of individual APIDs can be inhibited
- Telemetry logs will be created (viewable by offline utility)
- Packet Headers and Packet Data are updated
- Packet data can be shown in hexadecimal or octal format and addressed in hexadecimal or decimal form
- Packet Sequence Counters can be reset
- Packet Sequence Counters can be modified
- Packet Version field can be modified
- Packet APID field can be modified
- Packet Type field can be modified
- Packet Secondary Header Flag field can be modified
- Packet Length field can be modified
- CCSDS Unsegmented TimeCode (CUC) can be modified

- Packet rate may be controlled
- CLCW transmitted via EDUs when in IP mode
- IP packets are transmitted with variable lengths
- CLCW can be overridden by the operator
- Transmission of CLCW can be inhibited when in IP mode
- Scenario file (script) capability to set telemetry nodes and buffers
- Set telemetry data values in response to spacecraft commands received (end-item verification)
- Set initial telemetry data values at initialization
- Allow simultaneous display and set of multiple telemetry container items via GUI screens
- Simulate spacecraft memory dumps
- Use the PDB telemetry state text file to locate end-item verifier values
- Maintain and update telemetry data values in APID 1000
- Telemetry parameters may be set and viewed by Parameter ID
- CLCW Transmit Start and Stop is coupled to H/K Telemetry Start and Stop
- Telemetry values may be set using simple expressions
- Telemetry values may be set using trigonometric expressions
- Telemetry values may be set using Boolean expressions
- Telemetry values may be set to other telemetry mnemonic values
- Telemetry values may be saved in intermediate variables for later use
- TES Segmented Packets are emulated
- CLCW Transmit rate may be set by the operator
- Telemetry data values are validated for fit into packet space
- Current enable status and transmit rate for all APIDs is viewable via status display
- vcProcessor module discards VC63 VCDUs when creating files for playback
- The PDB Red/Yellow Limits file is used to refine initial telemetry values.
- Signed telemetry data values are validated as one's and two's complement integers upon user input, as appropriate.
- Displays of telemetry and command container item names may be saved and restored.
- The VCDU Sequence Counter field occupies 32 bits in APID 1000.
- Direct ingest of telemetry-related PDB flat files
- Accept telemetry and CLCW packets from an external source in IP mode
- Update telemetry parameter values to reflect data received from the external source
- Update CLCW field values to reflect data received from the external source.
- Forward, via IP interface, the telemetry and CLCW packets received from the external source.
- Modify telemetry parameter values and CLCW field values in externally received packets prior to re-transmission, in response to operator directive.
- **Accept CADUs from an external source in serial mode**
- **Extract telemetry packets and CLCWs from externally received CADUs**
- **Pack externally received telemetry packets and CLCWs into CADUs and forward via serial interface**

- **Interface with a 1553 Bus. Transmit telemetry packets over the 1553 Bus.**

Command:

- Identify commands using information from the PDB
- Display event messages with command mnemonics and submnemonics
- Set telemetry points in response to commands received (end-item verification) using information from the PDB
- Recognize spacecraft Command Loads
- Display Command Load data
- Copy Command Load data to a Memory Dump buffer
- Inhibit the Command Load data copy facility via operator directive
- Validate checksums of received Command Loads
- Ingest type AD, BC, and BD commands
- Display Total CLTUs count
- Reset Total CLTUs count
- Display Rejected CLTUs count
- Reset Rejected CLTUs count
- Display Instrument commands count
- Reset Instrument commands count
- Display Spacecraft commands count
- Reset Spacecraft commands count
- Display BC commands count
- Reset BC commands count
- Display BD commands count
- Display current Spacecraft CLCW
- Update Spacecraft and instrument CLCW
- Display current Instrument CLCW
- Validate commands based on individual, all, or none of the following validation criteria: CLTU Start and Tail Sequences, BCH Error Code, Transfer Frame Header Fields, FARM (Valid Frame Sequence), User Command Packet Header
- Generate event messages based on ingest
- Log raw commands (viewable by offline utility)
- Display raw command in hexadecimal or octal format addressed in either hexadecimal or decimal fashion
- Display command packet headers for instrument commands
- Display command packet headers for spacecraft commands
- Update command accepted and rejected counters in telemetry
- Command submnemonics are saved in container items and may be viewed after command receipt
- Expected Spacecraft ID changed to CC Hex
- TES and OMI segmented commands are recognized.
- The Function Code is used to identify HIRDLS commands.
- The two’s complement checksum of instrument commands is validated.

- Direct ingest of command-related PDB flat files
- Enable and disable automatic setting of end-item verifier telemetry points for commands received, in response to operator directive.
- **Interface with a 1553 Bus. Receive command packets from the 1553 Bus.**

Time:

- Maintain and update SC time (GIRD)
- Maintain and update GMT time
- Synchronize SC and GMT times

General:

- Control all simulator module functions via scenario scripts
- Selection of scenario scripts may be via operator type-in or via a file selection browse window
- Start scenario scripts in response to commands received
- Start a scenario script from a scenario script
- Execute multiple scenario scripts simultaneously
- Provide operator control of multiple scenario scripts started by the operator
- Save the last 10 operator directives
- Allow editing of saved operator directives before re-execution
- EDOS Service Header (ESH) fields may be viewed
- ESH field contents may be modified by the operator
- Validation of Command Data Block (CDB) header fields of commands received
- Modification of expected values of CDB header fields
- All viewable buffers may be displayed
- Addition, deletion, and modification of command end-item verifiers via SQL scripts
- Logs of commands received or telemetry transmitted may be retransmitted via IP output or Serial output
- Expected Spacecraft ID may be modified in EOSGS module
- CLCW ESH field contents may be modified by the operator
- Event messages to the screen may be inhibited or enabled by severity (color)
- Scenario scripts may contain IF-then-ELSE-ENDIF and WHILE-ENDWHILE conditional execution directives
- The Scenario module may interface with multiple modules
- Intermediate variables A – Z permit saving values as real numbers – extended to all modules that accept directives
- Intermediate variables Aq – Zq permit saving values as long integers – extended to all modules that accept directives
- The Serial Output module can accept directives from the operator or via a scenario script.

- The Event Message window has been separated from the project window and has been made resizable.
- **Receipt of CADUs from a serial interface and extraction of telemetry packets is via the new EOSXtract module.**
- **Interface with the 1553 Bus is via the new E1553Bus module.**

Attachment B – Installation Instructions for MPS/Aura Release 3.3

This attachment contains the instructions for installing the PDB files and the MPS/Aura Release 3.3 Server and Client. The information presented in this attachment is divided into three major sections. The first section contains abbreviated installation instructions, the second contains a summary of the installation changes, and the third section contains detailed instructions for performing initial and subsequent installations.

The information presented in this attachment has been checked for accuracy by the independent test team.

B-1: Abbreviated Installation Instructions

These instructions are intended for the experienced user.

1. Install the MPS/Aura Release 3.3 Client software by executing the **Setup.exe** program in the Client folder of the CD.

2. Install the MPS/Aura Release 3.3 Server software by executing the **Setup.exe** program in the Server folder of the CD.

3. If not previously done, create a folder under **D:\mps_pdb\AuraPDBs**, or if desired, under **C:\Program Files\CSC\Aura Server 3.3**, to hold the Aura PDB source files. Copy the Aura PDB source files into this new folder. Twelve files are needed. See the list in Paragraph B-3.3 for the files to be copied.

4. When initializing the MPS/Aura simulator for the first time, all Projects needed must be built and saved.

B-2: Summary of changes

The MPS/Aura simulator no longer requires Oracle as a database repository. Instead it ingests the Project Data Base (PDB) flat files directly during initialization. It is suggested that the PDB flat files be stored at a convenient place near the root folder for easy access.

Oracle is still available so that SQL*Plus may be used to make database queries. The method of ingest of the PDB flat files into Oracle is unchanged from Release 3.0. If it is necessary to install/reinstall Oracle, follow the instructions given in Attachment C of the MPS/Aura Release 3.0 Delivery Package.

B-3: Detailed Installation Instructions

This is the complete procedure for performing an initial or subsequent installation of the MPS/Aura simulator Release 3.3, and associated software, data files, and COTS programs on a PC.

Materials Needed:

- One or more versions of the Aura Project Data Base (PDB)
- The CD containing the MPS/Aura Release 3.3 software

B-3.1: Java Runtime Engine Installation

Installation of the Java Runtime Engine product need only be performed if the simulator is being installed on a new PC or one that has had its hard drive replaced. If Java is already installed on the PC then skip to Paragraph B-3.2.

1. Insert the CD containing the MPS/Aura Release 3.3 into the CD drive and navigate to it using either Windows Explorer or My Computer.

2. Double-click on the file named **jdk1_2_2-win.exe.** This will cause the Java Runtime Engine to be installed. Accept all defaults when responding to the installation prompts.

B-3.2: Installation of the Aura Server and Client software

The steps in this paragraph cause the MPS/Aura Client and Server software to be installed on the PC.

1. Insert the delivery media into the appropriate drive.

2. To install the Aura Client:
   a) On the desktop, click on the Start button, and then select Run from the resulting menu.
   b) When the Run window appears select the Browse… button.
   c) From the Browse Window, select the Removable drive that contains the installation CD.
   d) Click on the Client folder.
   e) From within the Client folder, double click on the **Setup.exe** filename.
   f) A window with the title "Run Window" will appear. Click on the Okay button to proceed to the next step.

g) The screen will be filled with an Aura Client background and a smaller window with the title "Welcome to Aura Client 3.3" will appear. Click on the Next button to proceed to the next step.
h) The next window will contain the licensing agreement. Click on Yes to accept the agreement and proceed.
i) After all of the files are copied, a window with the title "Setup Complete" will appear. Click on the Finish button to end.
j) An Aura Client icon will now be installed on the desktop.

3. To install the Aura Server:
   a) On the desktop, click on the Start button, and then select Run from the resulting menu.
   b) When the Run window appears select the Browse… button.
   c) From the Browse Window, select the Removable drive that contains the installation CD.
   d) Click on the Server folder.
   e) From within the Server folder, double click on the **Setup.exe** filename.
   f) A window with the title "Run Window" will appear. Click on the Okay button to proceed to the next step.
   g) The screen will then be filled with an Aura Server background and a window with the title of "Welcome to Aura Server 3.3" will appear. Click the Next button to proceed.
   h) The next window will contain the licensing agreement. Click on Yes to accept the agreement and proceed.
   i) Next a window will show the completion status as the files are copied. When the copying is complete click on the Finish button to finish the installation.
   j) An Aura Server icon will be installed on the desktop.

B-3.3: PDB Download

The next step is to copy the PDB onto the hard drive. You will need at least one version of the Aura PDB. The following PDB flat files are needed, where *xxxxxx* corresponds to the version portion of the filename:

cmd_desc_*xxxxxx*.pdb
cmd_fixdata_*xxxxxx*.pdb
cmd_parm_*xxxxxx*.pdb
cmd_vardata_*xxxxxx*.pdb
cmd_verify_*xxxxxx*.pdb
tlm_calcurve_*xxxxxx*.pdb
tlm_desc_*xxxxxx*.pdb
tlm_dstate_*xxxxxx*.pdb
tlm_packet_*xxxxxx*.pdb
tlm_parm_*xxxxxx*.pdb
tlm_polyconv_*xxxxxx*.pdb
tlm_rylim_*xxxxxx*.pdb

Add a folder to your chosen directory structure to hold the source files of the Aura PDB.

Copy the desired version of the PDB into the folder just created. If desired, more than one version of the PDB may be copied. Be sure to copy each version into its own folder.

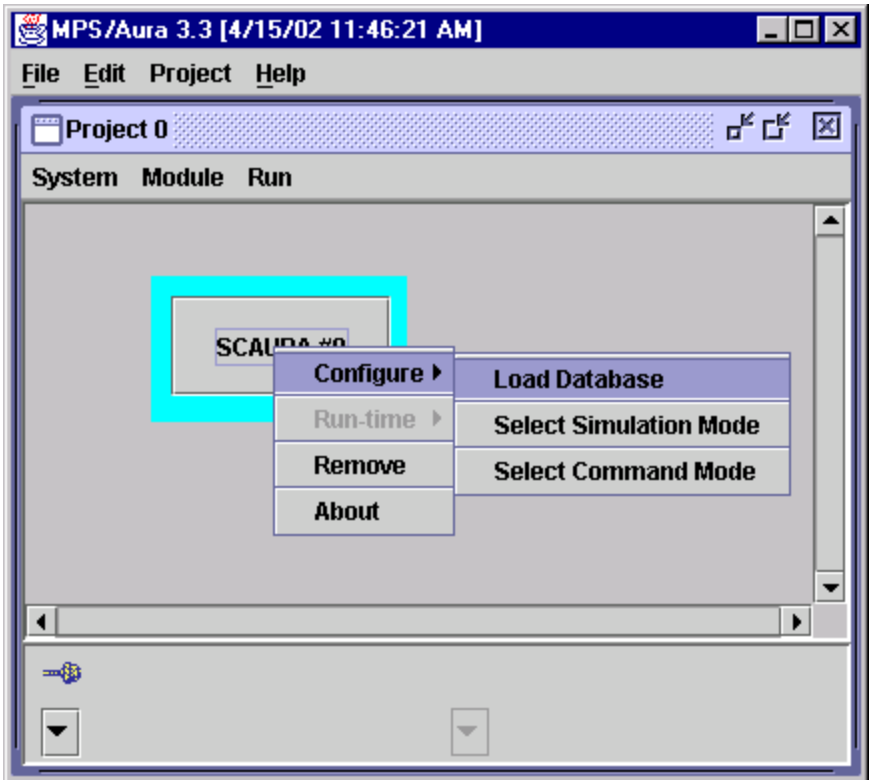
Attachment C - Special Operating Instructions

This attachment contains new special operating instructions for MPS/Aura Release 3.3. The information presented in this attachment has been checked for accuracy by the independent test team.

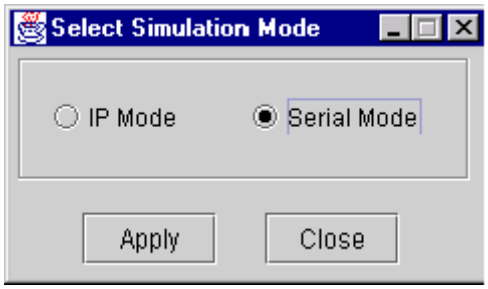
A User's Guide is being updated to include the information presented in this section. The User's Guide will be available from the ETS home page at **http://esdis-it.gsfc.nasa.gov/ETS.**

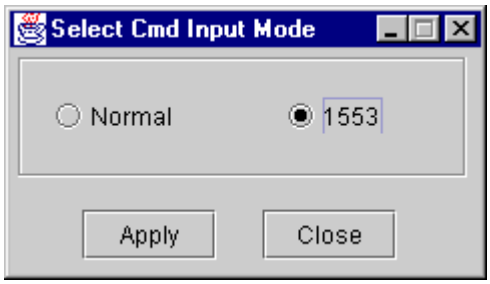
Save and Restore of operating mode has been added to SCAura

The ability to save the telemetry transmission and command receipt modes has been added to the SCAura module. When a Project is being built, if other than the default mode is desired, click on the SCAura module and select the Configure option from the drop-down menu. The Simulation Mode defaults to IP telemetry transmission. The Command Mode defaults to CLTU receipt.



Select the Simulation Mode or the Command Mode menu item as desired. In the resulting window, select the desired mode as shown below.



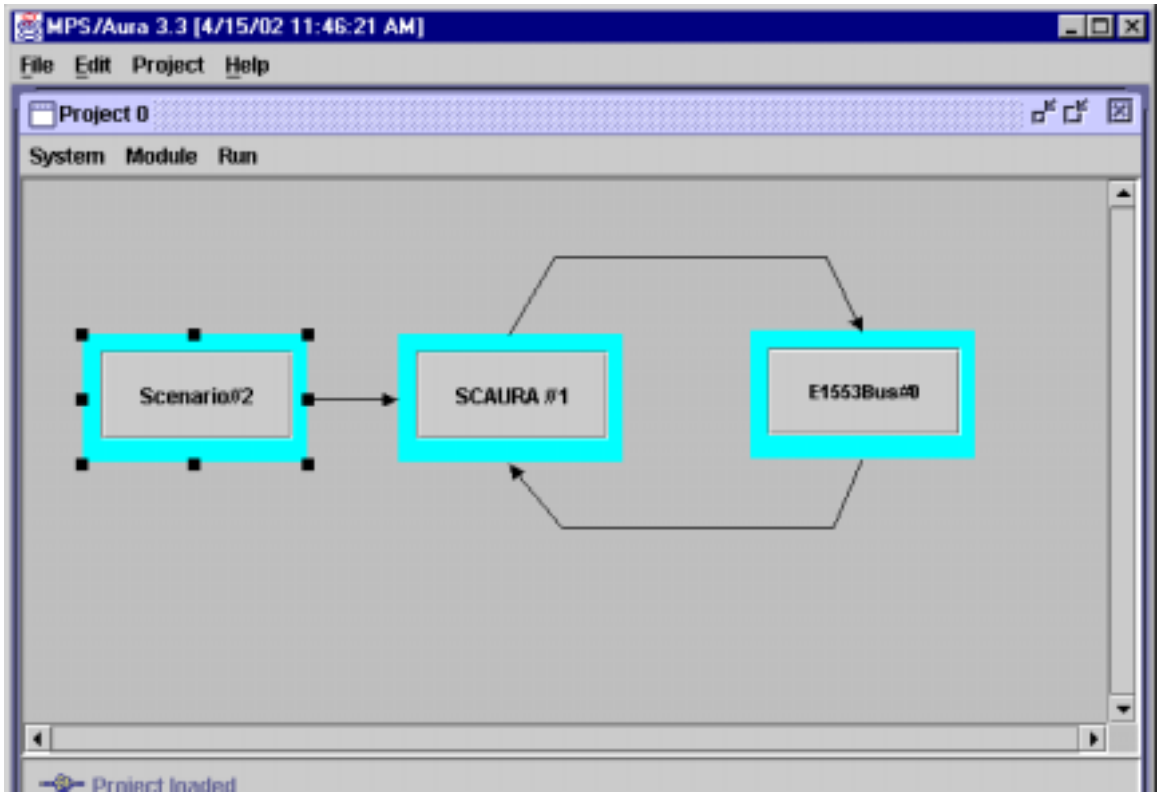


Emulation of Aura instruments via ETSF 1553 Bus Interface

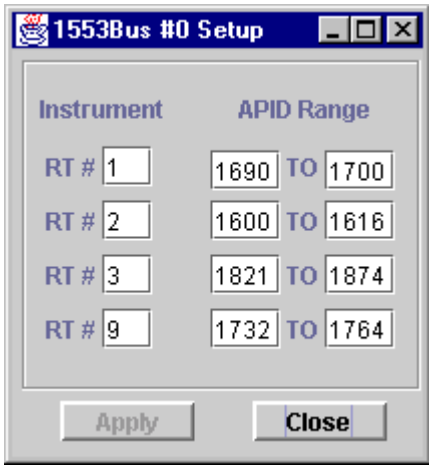
The capability to connect to the 1553 Bus of the EOC Training Simulator Facility (ETSF) is being added with this release. When connected to the ETSF, the MPS can act as an instrument emulator, transmitting instrument telemetry APID packets and receiving command packets via the 1553 Bus.

When operating in this mode, the SCAura IP telemetry transmit channel is connected to the input channel of the E1553Bus module and the E1553Bus output channel is connected to the SCAura command receipt channel. The picture shown below contains a typical 1553 Bus interface Project. The E1553Bus module handles all interfaces with the ETSF 1553 bus. Only the A-side bus is supported.

Telemetry packets are transmitted by SCAura exactly as when operating in IP mode. Command packets received from the ETSF have already been reassembled from codeblocks and have had their CLTU headers, CTLU tail sequences, and transfer frame headers removed. A new SCAura module container item, **CommandInputType**, must be set to non-zero to inform SCAura that command packets are to be expected vs. command CLTUs. This container item may be set by selecting “1553” as Command Input Mode during initialization, as shown above, or entered at any time the simulator is running.

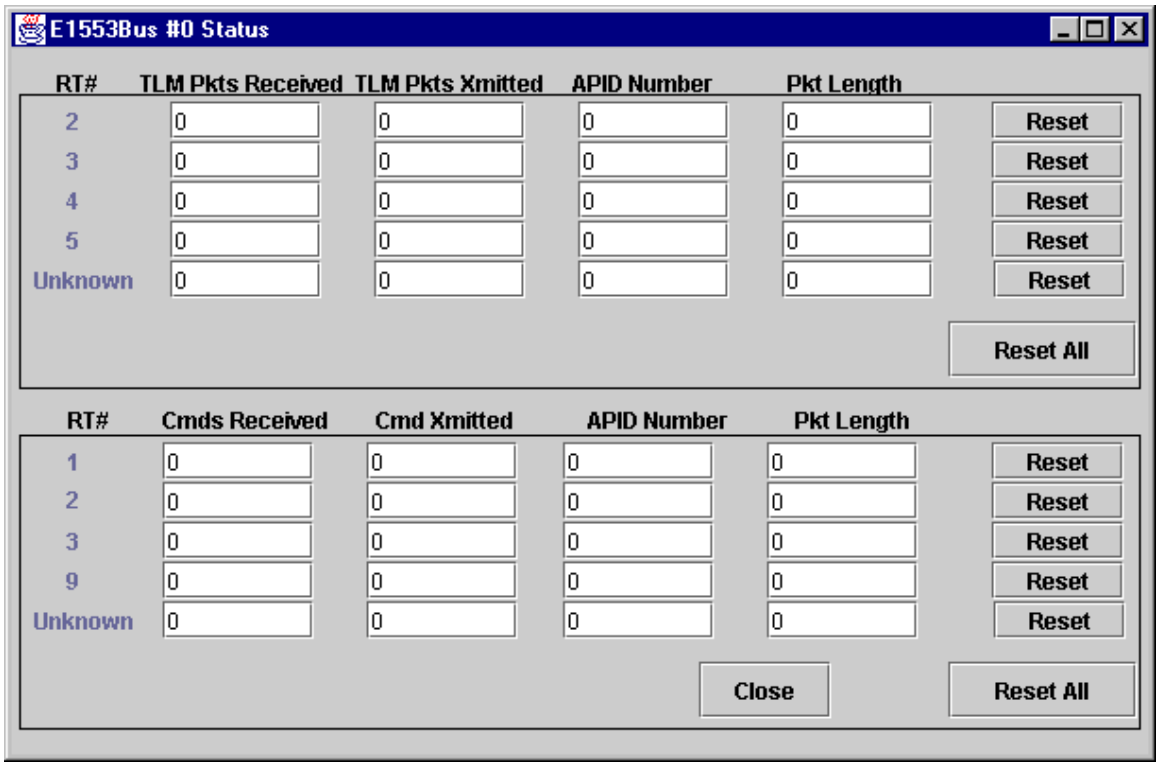


The E1553Bus module has two displays. The configuration display may be accessed during simulator initialization by clicking on the E1553Bus module and selecting **Configure** from the drop-down menu. In the resulting window, shown below, enter the Remote Terminal numbers and associated range of APIDs for each instrument to be emulated. Note that the APID range must include all telemetry and command APIDs expected.



While the simulator is running the E1553Bus module status display, shown below, will show counts of all packets transmitted and received over each Remote Terminal. If a command packet is received or SCAura transmits a telemetry APID that does not fall

within any of the selected ranges, it will be displayed in the “Unknown” row and will not be forwarded by the E1553Bus module.



Processing of externally received telemetry

The capability to accept, modify, and retransmit telemetry from an external source, such as a disk file, another PC, or the EOC Training Simulator Facility (ETSF) is enhanced in this release with the addition of a Serial mode receive capability. In Serial mode, the MPS is capable of receiving CADUs and extracting the telemetry packets and CLCW information from them. In IP mode the MPS is capable of receiving telemetry and CLCW packets via Input IP modules. Screen snaps with explanatory information are included below.

As each telemetry packet is received the MPS will apply any changes directed by the operator. If the simulator is operating in Serial mode, the packets are immediately placed into CADUs for transmission. If the simulator is operating in IP mode, the telemetry packets are immediately retransmitted. The following paragraphs describe the processing in detail.

External Packet Input Processing:

1. When operating in Serial mode a new module, EOSXtract, must be added to the Project between the Serial Input module that will receive the telemetry, and the

SCAura module. EOSXtract will extract all telemetry APIDs from the CADUs received, create EOS Enhanced CLCWs from information within the CADUs, and forward all data to the downstream module, usually SCAura. See Figure C-1 for a sample Project.

2. When operating in IP mode, two Input IP modules, one for telemetry APIDs and one for Enhanced CLCWs, are connected to the SCAura. See Figure C-2 for a sample Project.

3. External data must be connected to input channel 3 of the SCAura module. Packets may be received from one or more external sources via Input IP modules, from one or more disk files via the TxFile module, a Serial line via the EOSXtract module, or a combination of these methods.

4. The data packets input to the SCAura module must be either EOS Enhanced CLCW packets or EOS Telemetry packets. The Aura PDB is used to identify all received telemetry packets. CLCW packets are recognized by the three fill bytes at the front of the data area. When operating in Serial mode the EOSXtract module creates EOS Enhanced CLCW packets from information contained within the CADU.

5. The external packet may optionally have an EDOS Service Header preceding the data. The simulator presumes that the packets do not have an EDOS Service Header. **If a header is present, ALL packets must have the same header. The SCAura module must be told to remove and discard that many bytes from the packets received.** There is a single container item that defines the size of this header in bytes. (All new container items are listed in tables later in this document.)

6. If the received packet cannot be identified, a warning event message is produced and the input is counted as an error. For telemetry APID packets, this means that all packets expected from the external source must be properly defined in the database. The event message also provides the current length in bytes of the optional service header in case this value needs adjustment.

CLCW Processing:

1. Since the CLCW buffers within SCAura are provided to telemetry threads and may now be overwritten with external data, critical section processing was added so that a consistent buffer is available for transmission.

2. Filtering of CLCWs: There is a flag, per CLCW buffer, to enable/disable external input. See the container item list below for the names of these flags. It is possible to have the external source provide the spacecraft CLCW but not the instrument CLCW, and vise versa. If a CLCW is enabled for external input, the externally received data overwrites MPS-generated values but not operator-entered values.

The following table should help to make the preceding CLCW processing description clear. The states described apply separately to Spacecraft and Instrument CLCW processing. CLCW transmission may be disabled in IP mode but not when in Serial mode.

CLCW Transmission	Externally Enabled	Resulting Affect
Disabled	Don't Care	Only applies to IP mode. No data is transmitted
ENABLED	Disabled	CLCWs are generated and transmitted solely by MPS
ENABLED	ENABLED	Data received from an external source overwrites the MPS generated value, but NOT operator-entered values.

3. When an external CLCW packet is processed, it is copied to the appropriate CLCW buffer. If the operator has modified any fields of the CLCW within MPS, these modifications will be made to the buffer before it is re-transmitted.

4. Each field of each CLCW has an associated update flag. Whenever the operator modifies a field, this flag will be set. This causes the change to persist in all following CLCW packets. If a temporary change was intended, the operator MUST clear the associated update flag for the specific field immediately after the CLCW is transmitted. Then the value from the next external CLCW will be used for this field.

5. After any modifications have been applied, values from the CLCW buffer are copied into the individual field variables. This keeps the CLCW displays properly updated.

6. Both ignored and processed external CLCW packets are counted.

Telemetry APID Packet Processing:

1. Since the APID packet buffers within SCAura may now be overwritten with external data, critical section processing was added so that a consistent buffer is available for transmission.

2. Filtering of external APID packets:

    a. If a received packet is transmit “enabled" within MPS, it is assumed to be controlled solely by the SCAura module. The external packet input thread will count but otherwise ignore the packet. Thus the generation and transmission of "enabled" packets works in the same way as in previous releases.

    b. If a packet is received which is transmit "disabled" within MPS, its external load enabled flag will be checked. If the external load flag is enabled, the packet will be accepted and retransmitted. It is assumed that the transmission timing for external packets is controlled externally. The packet is copied into the appropriate packet buffer, formatted, then immediately transmitted. This eliminates the need for packet timing coordination between MPS and the external source. It also solves the problem of receiving dump packets since these packets are in the database and "disabled" by default. If the external load flag is set to disabled, the received external packet is counted but otherwise ignored.

    The following table should help to make the preceding APID processing description clear:

Enabled	**Externally Enabled**	**Resulting Affect**
Disabled	Disabled	No data is transmitted
ENABLED	Disabled	APIDs are generated and transmitted solely by MPS
Disabled	ENABLED	Data received from an external source is forwarded.
ENABLED	ENABLED	APIDs are generated and transmitted solely by MPS Data received from an external source is discarded

3. When an external packet is accepted for retransmission the following happens:

    a. It is copied into the appropriate telemetry buffer. The EDOS Service Header, if present, is stripped off and discarded. For packets with secondary keys, the logic makes the key value of the received packet

buffer the current one, thus ensuring transmission of secondary keys in the same order and with the same timing as the external source.

b. The packet header is formatted, and any operator-directed changes to telemetry values are applied. When the external load flag is enabled, new updates (from operator or scenario files) to this packet's telemetry are saved in an update vector and applied to every packet received.

c. All telemetry points associated with the APID are set to the values from the formatted buffer. This means that parent and child mnemonics will be in agreement for externally received packets and scenario scripts may access telemetry point values set by the external source or by the MPS operator. The telemetry points may also be monitored using the container item display.

d. The packet is immediately transmitted. If it is sent through the EOSGS module, as is standard when operating in IP mode, a new EDOS Service Header is applied. While there will be a small delay from processing each packet, the interval between packets will reflect the external timing.

Operations Concepts:

When the MPS/Aura simulator is being used without externally supplied data, nothing has changed. The simulator will operate exactly as it has in previous releases.

MPS defaults to disable of external input for both CLCW packets and all defined telemetry packets. When the simulator is being used with externally supplied data, directives must be issued to enable external input of specific APIDs and CLCW packets. A boilerplate scenario script to enable external packet receipt is being supplied with the delivery package. The user may issue directives or run another scenario script to specify timing and transmit "enable" flags for those packets that are to be generated by the SCAura module.

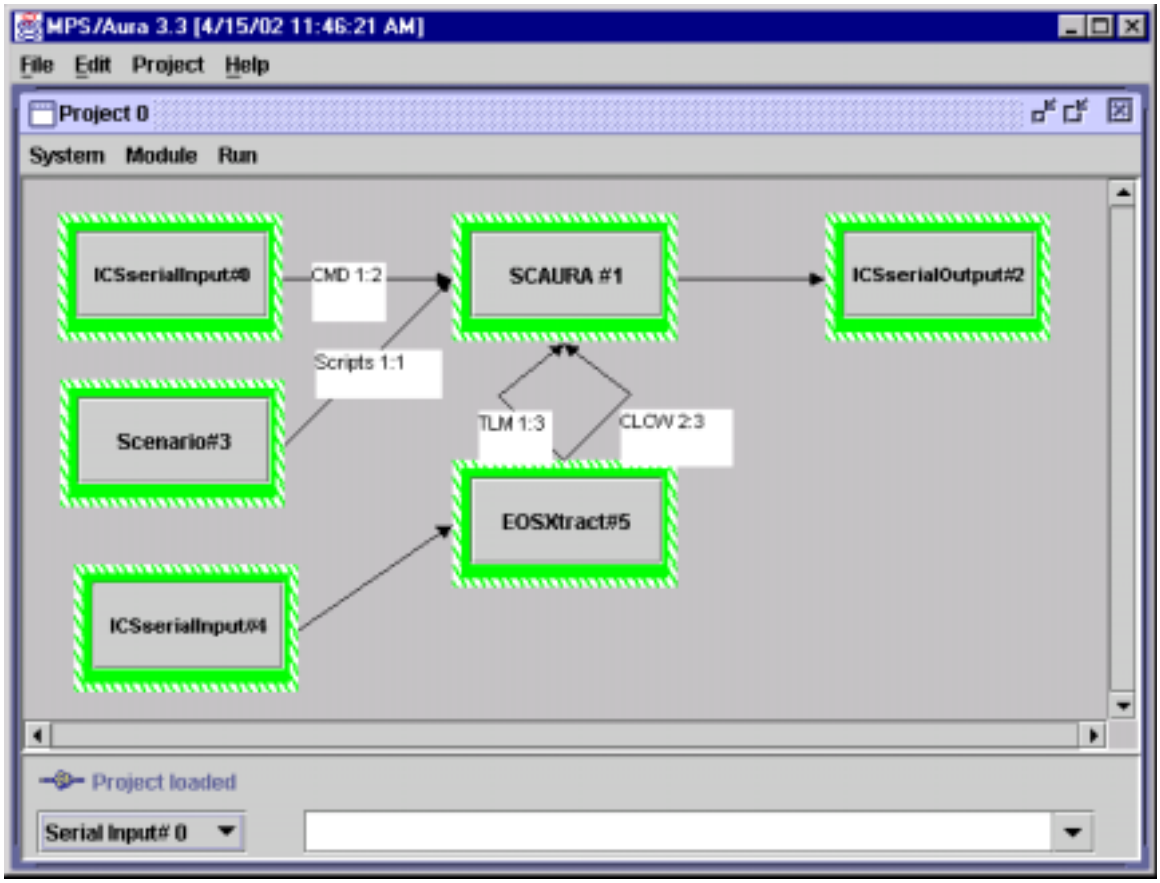
If APID 1000 is enabled for external receipt, the simulator will not apply data to the time fields or set the VCDU count. APID 1000 will be handled in the same way as any other externally received packet. However, should any of the telemetry points within APID 1000 be modified by the MPS operator, these changes will be processed and applied to the next and all subsequent packets.

Setting of end-item verifier telemetry points in response to commands received is enabled by default. When packets are being received externally and commands are being received, it may be desirable to disable setting of end-item verifiers. To do so, set the container item, **commandverification**, to zero.

Whenever a user makes a change to a CLCW field, the field is immediately copied into the associated CLCW buffer and the update flag for that field is then set. If the operator modifies a CLCW field when the CLCW is being received externally, these changes will

persist. This may not always be desired. For example, the operator may change the externally loaded CLCW to set the lockout flag to true. Because the change is "sticky", the lockout flag will stay true even when new external data packets would reset the field to false. If the operator wants the field to match the external data again, the field's update flag must be cleared. Scenario scripts are suggested as an efficient means of setting a field and clearing the update flag. The names of all CLCW flags are given in the container item table below.

The following picture shows how MPS could be configured for receipt of data from an external source when operating in Serial mode.



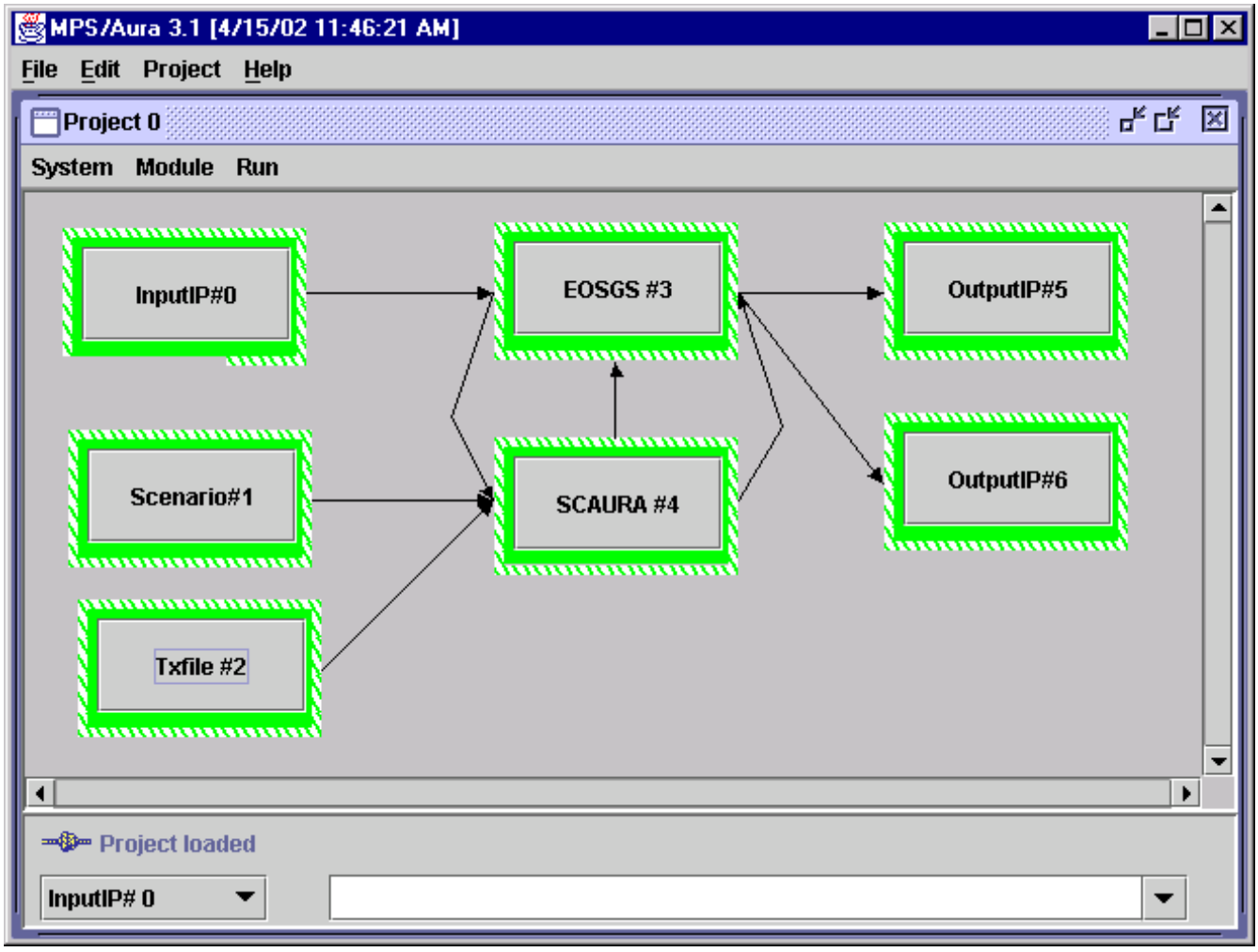
- ICSserialInput #0 is set up to receive command CLTUs. It is connected to input channel #2 of the SCAura module.
- ICSserialInput #4 is set up to receive telemetry CADUs from the external source. Every CADU received is passed to EOSXtract.
- EOSXtract extracts telemetry packets and builds Enhanced CLCWs from the CADUs it receives. Telemetry APID packets are output on channel #1 and CLCW packets are output on channel #2.
- SCAura receives telemetry APID packets and CLCW packets on its input channel #3.
- SCAura transmits the telemetry CADUs it generates on its output channel #3.

All of the channel connections are summarized in the following table.

Purpose	Source	Source Channel Output	Destination	Destination Input Channel
Command Receipt	ICSserialInput #0	1*	SCAura	2
Scenario Script Execution	Scenario	1	SCAura	1
CADU Receipt	ICSserialInput #4	1*	EOSXtract	1*
Telemetry Packet Forward	EOSXtract	1	SCAura	3
CLCW Packet Forward	EOSXtract	2	SCAura	3
Telemetry CADU Transmit	SCAura	3	ICSserialOutput #2	1*

*The module has only one input or output channel.

The following picture shows how MPS might be configured for receipt of data from an external source such as a log file with retransmit in IP mode.



- InputIP #0 is set up to receive command data transmitted by EMOS. It is connected to input channel #2 of the SCAura module via the EOSGS module.
- TxFile reads telemetry packets from a disk file. It is connected to input channel #3 of the SCAura module.
- Output IP #6 and Output IP #7 transmit the telemetry and CLCW packets, respectively.

All of the channel connections are summarized in the following table.

Purpose	Source	Source Channel	Destination	Destination Channel
Command Receipt	Input IP #0	1*	EOSGS	1
Command Receipt	EOSGS	1	SCAura	2
Scenario Script Execution	Scenario	1	SCAura	1
Data File Read	TxFile	1*	SCAura	3
Telemetry Packet Transmit	SCAura	1	EOSGS	2
CLCW Packet Transmit	SCAura	2	EOSGS	3
Telemetry EDU Transmit	EOSGS	2	Output IP #6	1*
CLCW EDU Transmit	EOSGS	3	Output IP #7	1*

*The module has only one input or output channel.

The following container items have been added to the SCAura module to support external packet receipt and retransmission. The packet counts may be viewed by invoking the **External Packet Summary** display from the **Telemetry** menu. The **APID Status** display has been augmented with a column showing the external enabled status.

Container Item Name for Global Flags and Counts	Description
ExternalPacketHeaderBytes	Size of optional header on all external packets. **Defaults to 0.** Set this item to 20 bytes to accommodate an EDOS Service Header, if needed.
ExternalCLCWCount	Count of all external CLCW packets processed
ExternalAPIDCount	Count of all external APID packets processed
ExternalIgnoredCLCWCount	Count of all CLCW packets ignored because external loading was disabled
ExternalIgnoredAPIDCount	Count of all APID packets ignored because external loading was disabled
ExternalPacketErrorCount	Count of all external packets in error

Container Item Name for APID items	Description
TlmPacketxxxxExternalEnabled where xxxx = APID in decimal	External load enable flag for Telemetry packet xxxx (0=disabled, 1=enabled) **Default is disabled**.

Container Item Name for CLCW items	Description
SpaceClcwExternalEnabled	External load enable flag for Spacecraft CLCW buffer (0=disabled, 1=enabled) **Default is disabled.**
InstrClcwExternalEnabled	External load enable flag for Instrument CLCW buffer (0=disabled, 1=enabled) **Default is disabled**.
UpdatedSpaceClcwCWT UpdatedInstrClcwCWT	Sticky update flag for spacecraft/instrument CLCW Control Word Type field (0=false, 1=true)
UpdatedSpaceClcwVersion UpdatedInstrClcwVersion	Sticky update flag for spacecraft/instrument CLCW Version field (0=false, 1=true)
UpdatedSpaceClcwStatus UpdatedInstrClcwStatus	Sticky update flag for spacecraft/instrument CLCW Status field (0=false, 1=true)
UpdatedSpaceClcwCOP UpdatedInstrClcwCOP	Sticky update flag for spacecraft/instrument CLCW COP In Effect field (0=false, 1=true)
UpdatedSpaceClcwVCID UpdatedInstrClcwVCID	Sticky update flag for spacecraft/instrument CLCW VCID field (0=false, 1=true)
UpdatedSpaceClcwSpare1 UpdatedInstrClcwSpare1	Sticky update flag for spacecraft/instrument CLCW spare 1 field (0=false, 1=true)
UpdatedSpaceClcwNoRfAvail UpdatedInstrClcwNoRfAvail	Sticky update flag for spacecraft/instrument CLCW No RF Avail field (0=false, 1=true)
UpdatedSpaceClcwNoBitLock UpdatedInstrClcwNoBitLock	Sticky update flag for spacecraft/instrument CLCW No Bit Lock field (0=false, 1=true)
UpdatedSpaceClcwLockout UpdatedInstrClcwLockout	Sticky update flag for spacecraft/instrument CLCW Lockout field (0=false, 1=true)
UpdatedSpaceClcwWait UpdatedInstrClcwWait	Sticky update flag for spacecraft/instrument CLCW Wait field (0=false, 1=true)
UpdatedSpaceClcwRetransmit UpdatedInstrClcwRetransmit	Sticky update flag for spacecraft/instrument CLCW Retransmit field (0=false, 1=true)
UpdatedSpaceClcwFarmCount UpdatedSpaceClcwFarmCount	Sticky update flag for spacecraft/instrument CLCW FARM-B Count field (0=false, 1=true)
UpdatedSpaceClcwSpare2 UpdatedInstrClcwSpare2	Sticky update flag for spacecraft/instrument CLCW spare 2 field (0=false, 1=true)
UpdatedSpaceClcwReport UpdatedInstrClcwReport	Sticky update flag for spacecraft/instrument CLCW Report field (0=false, 1=true)

Container Item Name for end-item verifier setting	Description
commandverification	Controls setting of end-item verifier telemetry points in response to commands received. (0=disabled, 1=enabled) **Default is enabled.**

PDB Ingest

The following information is being repeated from the MPS/Aura Release 3.1 delivery package for the convenience of the user.

With Release 3.1 the Project Data Base (PDB) is no longer taken from the Oracle repository. Instead the PDB flat files are ingested directly into the simulator during initialization, or whenever the user wishes to change the PDB version in use.

The PDB flat files must be present on the PC hard disk, or in a reachable point of the network neighborhood. The following PDB flat files are needed, where *xxxxxx* corresponds to the version portion of the filename:

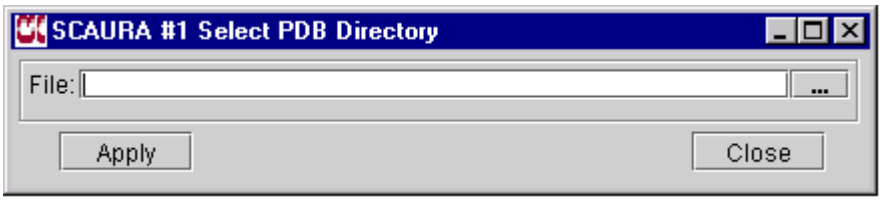
cmd_desc_*xxxxxx*.pdb
cmd_fixdata_*xxxxxx*.pdb
cmd_parm_*xxxxxx*.pdb
cmd_vardata_*xxxxxx*.pdb
cmd_verify_*xxxxxx*.pdb
tlm_calcurve_*xxxxxx*.pdb
tlm_desc_*xxxxxx*.pdb
tlm_dstate_*xxxxxx*.pdb
tlm_packet_*xxxxxx*.pdb
tlm_parm_*xxxxxx*.pdb
tlm_polyconv_*xxxxxx*.pdb
tlm_rylim_*xxxxxx*.pdb

It is suggested that these files be stored in a common area near the root folder or in a subfolder to the Server folder to make navigation easy.

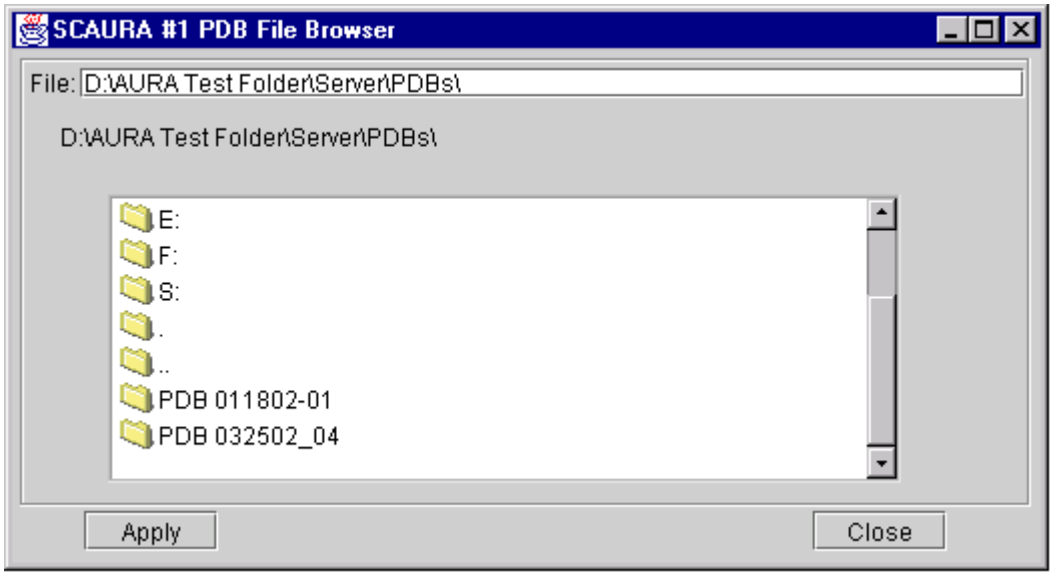
Follow these steps to ingest the PDB at simulator initialization:

1. Create, or restore, a desired Project in the same manner as all previous releases.

2. When the **Configure/Load Database** option of the SCAura module menu is accessed, the **Select PDB Directory** window will appear:



3. Single-click on the Browser button, which is the button with three dots at the right. The **PDB File Browser** window, which appears as follows, will appear:



4. To locate the desired version of the PDB, move the Slider as necessary and single-click folder names until the folder containing the PDB files is located. To move up the directory structure, single-click the folder with two dots as its name. Single-click the folder containing the PDB files, then click **Apply.** The **PDB File Browser** window will disappear.

5. Ensure that the complete path to the PDB folder appears in the **Select PDB Directory** filename field, then click **Apply** followed by **Close.**

The PDB files will be read in and internal tables created. A series of messages indicating progress at reading the files will be displayed in the Event Log window. If any messages indicate Warnings or Errors, more informational messages will be written to the Event Log disk file.

The simulator may now be reloaded with a different version of the PDB without completely stopping it. To load a new PDB version, select the **STOP** option followed by the **UNLOCK** option from the **RUN** menu. Then follow steps 2 through 5 above. Then restart the simulator.

Miscellaneous

- The file, SegmentedApids.txt, has been renamed AuraSegmentedApids.txt to avoid conflict and confusion with a segmented APID file used by another mission.

Attachment D – Resolved Discrepancy Reports

In addition to the new capabilities, the following Discrepancy Reports (DRs) and Change Requests (CRs) have been closed by and are being delivered with MPS/Aura Release 3.3. The DRs/CRs are listed in the table below, which provides the DR/CR Number, Status, Severity, and a short description. A full description of each DR/CR follows the summary table. Complete information on all DRs/CRs may be accessed via the Internet at address **http://edosultra30.gsfc.nasa.gov/ddts/**

Summary of Closed Discrepancy Reports

Critical (Severity 1)	Urgent (Severity 2)	Routine (Severity 3)	Change Requests	Total
0	2	3	3	8

Status Definitions

N – New	A – Assigned Analysis	R – Analysis Entered
V – Assigned Verification	T – Tested	C – Closed
W – Withdrawn	P – Postponed	X – Duplicate

ETS No.	SMO No.	Type	Severity	Description
ETS0465	SMOdr17551	DR	3	Stopping SCAura to load PDB causes GUI crash.
ETS0467	SMOdr17816	DR	3	Cmd Sub-mnemonics not shown in Event window

DR: SMOdr17551 (ETS0465) Related NCR: Submitted: 020809
Status: NEW Class: ETS

Title: Stopping SCAura to load PDB causes GUI crash

SUBMITTAL INFORMATION

Project:	ETS
DR Type:	Problem
Rel/Ver:	3.1
Subsystem:	Aura
Module:	Simulator
Affected-Requirement:	
Test Phase:	system I&T
Severity:	3
Date found:	020805
Location:	GSFC
Submitter:	Ernest Quintin
Organization:	ETS Dev Group
Phone number:	301-805-3649
Email:	equintin@csc.com

********** Problem (Added 020809 by equintin) **********
Please describe the problem you are experiencing below, including what you did, what you expected to happen, and what actually happened:

If the user forgets to load the Project Data Base (PDB) when initializing the simulator, stopping the SCAura module to load a PDB results in a Client crash. The entire Project must be stopped via the Run/Stop and Run/Unlock menu items.

DR: SMOdr17816 (ETS0467) Related NCR: Submitted: 020912
Status: NEW Class: ETS

Title: Cmd Sub-mnemonics not shown in Event window

SUBMITTAL INFORMATION

Project:	ETS
DR Type	Problem
Rel/Ver:	3.1
Subsystem:	Aura
Module:	Simulator
Affected-Requirement:	
Test Phase:	system I&T
Severity:	3
Date found:	020826
Location:	GSFC
Submitter:	Ernest Quintin
Organization:	ETS Dev Group
Phone number:	301-805-3649
Email:	equintin@csc.com

********** Problem (Added 020912 by equintin) **********
Please describe the problem you are experiencing below, including
what you did, what you expected to happen, and what actually happened:

Command identification messages in the Event window and Event
Log give incorrect information regarding command sub-mnemonics.

Workaround:
Issue directives or display sub-mnemonic names in the
Display/Set Container Items window using the name format

<sub-mnem name>#<cmd parm-id>

Attachment E – System Limitations

E.1 MPS/Aura Release 3.3 Limitations

The following limitations apply to MPS/Aura Release 3.3. Some of these are Discrepancy Reports (DRs) against SIMSS baseline products and have been recorded in their DR repository.

Problem Description	Workaround
The Scenario module File Selection window does not always show all of the files in the selected folder.	Click the Accept button of the File Selection window without selecting any file. Then type the scenario file name into the Filename field of the Scenario Control window, or copy and paste it from Windows Explorer.
The Save Project (Extended) and Restore From (Extended) options are intended for another application where a remote server runs simultaneously with the local application. The options are included with MPS/Aura so that only one version of the NeTTCore code needs to be maintained.	Avoid use of the Save Project (Extended) and Restore From (Extended) options.
The Generic Container Buffer display is limited to 1400 bytes of data (= 700 words, or 350 double words). A request for more data than that will result in a display of 1400 bytes of information. *This is SIMSS Defect # 102.*	To view data that is beyond byte 1400 of the buffer, set the offset to 1400, or as required to view the data.
When using SQL*Plus to select entries from the Oracle calcurve table via the conversion type field, *conv_type*, it is necessary to put a space after the type entry. e.g. "U_5D ", not "U_5D".	Given at left.
If the user forgets to load a PDB when initializing the simulator, Stopping the SCAura module to load the database will result in a Client GUI crash. This has been written up as an MPS DR ETS0465.	Use Run/Stop and Run/Unlock to stop the entire Project.

Problem Description	Workaround
Certain APIDs are included in the PDB tlm_packet file without any Interval or Slot number information. MPS supplies a default Interval of one second and a default Slot number of zero. However, the packet timing appears to be less accurate than for those APIDs that have at least one Interval field filled in the tlm_packet file. More investigation is necessary before writing this as an MPS DR.	Explicitly supply an Interval when enabling an APID that has no non-zero Interval in the tlm_packet file.
Not all of the Reset buttons on the E1553Bus module GUI work.	None.
The E1553Bus module will crash MPS if it is invoked on a PC that does not have a 1553 Bus interface board installed.	Avoid adding the E1553Bus module to any Project if the PC does not have a 1553 Bus interface board.

Attachment F - Release History Summary Matrix

Attached is the MPS/Aura simulator release history summary matrix, updated to reflect the MPS/Aura Release 3.3 delivery. Modules inherited from the SIMSS baseline have the SIMSS Release Number, while the MPS/Aura modules EOSGS, EOSXtract, and SCAura have the current MPS/Aura Release Number.

Release History Summary Matrix

System: **MPS/Aura**

Release Number		1.0	2.0	3.0 Beta	3.0	3.1	3.2	3.3						
Delivery Date		3/16/01	6/15/01	9/28/01	1/11/02	8/9/02	8/23/02	10/25/02						
Configuration Item	CI No.													
Core (Client)	1.1	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
Core (Server)	1.2	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
SCAURA (Client)	1.3	1.0	2.0	3.0	3.0	3.1	3.2	3.3						
SCAURA (Server)	1.4	1.0	2.0	3.0	3.0	3.1	3.2	3.3						
EOSGS (Client)	1.5	1.0	2.0	3.0	3.0	3.0	3.0	3.0						
EOSGS (Server)	1.6	1.0	2.0	3.0	3.0	3.0	3.0	3.0						
IP Input (Client)	1.7	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
IP Input (Server)	1.8	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
IP Output (Client)	1.9	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
IP Output (Server)	2.0	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
Logging (Client)	2.1	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
Logging (Server)	2.2	4.0	4.1	5.0	6.0	6.0	6.0	7.0						

Delivery Date		3/16/01	6/15/01	9/28/01	1/11/02	8/9/02	8/23/02	10/25/02						
Configuration Item	CI No.													
Scenario (Client)	2.3	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
Scenario (Server)	2.4	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
Serial Input (Client)	2.5	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
Serial Input (Server)	2.6	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
Serial Output (Client)	2.7	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
Serial Output (Server)	2.8	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
TxFile (Client)	2.9	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
TxFile (Server)	3.0	4.0	4.1	5.0	6.0	6.0	6.0	7.0						
vcProcessor (Client)[1]	3.1		4.1	5.0	6.0	6.0	6.0	7.0						
vcProcessor (Server)[1]	3.2		4.1	5.0	6.0	6.0	6.0	7.0						
EOSXtract (Client)[2]								1.0						
EOSXtract (Server)[2]								1.0						

[1] Added with Release 2.0
[2] Added with Release 3.3

Attachment G — Mission Systems Configuration Management Form

This attachment contains the completed Mission Systems Configuration Management (MSCM) form for the delivery of MPS/Aura Release 3.3.

Mission Systems Configuration Management Form

1. ORIGINATOR	2. ORGANIZATION	3. PHONE	4. E-MAIL ADDRESS
Estelle Noone	CSC	301-805-3653	enoone@csc.com

5. ELEMENT	6. INSTALLATION PRIORITY	7. TRACKING NUMBER (Assigned by CM Office)
ETS (MPS/Aura)	Routine	

8. SOURCE CHANGE REQUEST(S):

ETS delivery of MPS for EOS Aura (MPS/Aura)

9. APPROVALS

Element Manager _____ / /

Flight Ops Director _____ / /

Operations Manager _____ / /

10. DELIVERED SYSTEM (Check all that apply)

	Name	Version	Media Identification	Identification Date
☐ Hardware				
☒ Software	MPS/Aura	R3.3	CD-ROM	10/25/02
☐ Database				
☒ Documentation:				
MPS/Aura delivery package		N/A	via email	10/25/02
MPS/Aura Release 3.3 User's Guide		R3.3	http://esdis-it.gsfc.nasa.gov/ETS/etsdoc.html	TBS
☐ Other				

11. CHANGE DESCRIPTION

Release 3.3 of MPS/Aura

12. ATTACHMENT(S): Check if YES ☒

Description: MPS/Aura Release 3.3 delivery package (cover letter with attachments) dated 10/25/02

13. CM OFFICE USE

	Location (Bldg/Room)	Slot location(s)
Hardware	/	
Media	/	
Documentation	/	

Installation date ____/____/____ CM Office Signature _____

Form MSCM (970327)