Customer Avionics Interface Development and Analysis (CAIDA) Lab DEWESoft Display Creation

Connor Coffey[1]
NASA Kennedy Space Center, Kennedy Space Center, FL 32899

The Customer Avionics Interface Development and Analysis (CAIDA) Lab supports the testing of the Launch Control System (LCS), NASA's command and control system for the Space Launch System (SLS), Orion Multi-Purpose Crew Vehicle (MPCV), and ground support equipment. The objectives of the year-long internship were to support day-to-day operations of the CAIDA Lab, create prelaunch and tracking displays for Orion's Exploration Flight Test 1 (EFT-1), and create a program to automate the creation of displays for SLS and MPCV to be used by CAIDA and the Record and Playback Subsystem (RPS).

Nomenclature

CAIDA	=	*Customer Avionics Interface Development and Analysis*
CEV	=	*Crew Exploration Vehicle*
CUI	=	*Compact Unique Identifier*
DEM	=	*Data Exchange Message*
EFT-1	=	*Exploration Flight Test 1*
Eng. Unit	=	*Converted unit (volts, feet, etc.)*
GUI	=	*Graphical User Interface*
ICPS	=	*Interim Cryogenic Propulsion Stage*
IEEE	=	*Institute of Electrical and Electronics Engineers*
KSC	=	*Kennedy Space Center*
LAS	=	*Launch Abort System*
LCC	=	*Launch Control Center*
LCS	=	*Launch Control System*
MPCV	=	*Multi-Purpose Crew Vehicle*
NASA	=	*National Aeronautics and Space Administration*
NIC	=	*Network Interface Card*
NIFS	=	*NASA Internships, Fellowships, and Scholarships*
PCM	=	*Pulse-Code Modulation*
Raw Count	=	*Value read directly from transducer*
RPS	=	*Record and Playback Subsystem*
SLS	=	*Space Launch System*
SOCRRATES	=	*Software Only CEV Risk Reduction Analysis and Test Engineering Simulator*
VALFAC	=	*Validation Facility*
VCTE	=	*VALFAC Common Test Environment*
XTCE	=	*XML Telemetry and Command Exchange*

I. Introduction

The CAIDA Lab is responsible for sending telemetry to LCS to test their Orion MPCV gateway and displays. The gateway processes the raw telemetry into data for the flight controllers. There are three ways CAIDA can send telemetry: MPCV EFT-1 brass board flight computers, the Software Only CEV Risk Reduction Analysis and Test Engineering Simulator (SOCRRATES) Orion simulator, and playbacks. The flight computers are similar to the ones that went up on EFT-1 except they are not radiation hardened; commands can be sent through Honeywell's

[1] Undergraduate Student, Department of Aerospace Engineering, Iowa State University

Validation Facility (VALFAC) Common Test Environment (VCTE) Host Computer to two Time-Triggered (Gigabit) Ethernet Network Interface Cards (NICs), to the flight computers. This is not a closed loop simulation, so for example if a valve is opened or closed there would be no pressure changes. SOCRRATES, an all software MPCV simulator, can be used to send telemetry to LCS; the software simulator in SOCRRATES has much higher fidelity than the VCTE and can run scenarios such as launch, prelaunch, and orbit. SOCRRATES was utilized for testing displays in preparation for EFT-1. The last way CAIDA sends telemetry is through playbacks sent through DEWESoft. DEWESoft is a program used heavily by CAIDA and the Record and Playback Subsystem (RPS) and can be used to easily view telemetry and to play back telemetry files from simulations or when the vehicle was powered up.

I supported the lab in its day-to-day operations including troubleshooting SOCRRATES, patching cables in the Honeywell rack to send telemetry to the correct customer, and troubleshooting with DEWESoft to ensure the customer was receiving the correct data.

I also created two displays used by RPS and CAIDA to view the progress of EFT-1. One would monitor the countdown procedures from when the flight computers were turned on to liftoff. The other display would receive latitude, longitude, altitude, and mission phase from when Launch Abort System (LAS) shroud separation occurred to splashdown and track the spacecraft around the Earth.

The main use for DEWESoft in CAIDA and RPS is to view telemetry. To view telemetry using DEWESoft, a setup must first be created using the information in the XML Telemetry and Command Exchange (XTCE), the user must input into DEWESoft the location of the Compact Unique Identifier (CUI)/measurement in the telemetry stream (start byte, start bit, length in bits, *Data Exchange Message* (DEM)), the data type (IEEE float, signed integer, or unsigned integer), and the raw count to engineering unit conversion polynomial or enumerations if applicable. Then the user can create a display by selecting a widget and selecting the associated CUI in the menu for each desired CUI. This is an easy task for a handful of CUIs but with displays that can have hundreds of CUIs, it is not efficient to search the XTCE and manually create the setup. To solve this problem, Justin Ruscoe and Samuel Harris created two programs. One program would parse through the Orion XTCE and create a SQLite database, and the other would read the data from the database and create a DEWESoft setup for the CUIs and DEMs provided by the user. The major shortcomings of these programs are that they did not support CUIs from the SLS XTCE and did not support raw count to engineering unit conversion polynomials greater than one degree.

II. Operations

A. Day-To-Day Operations

In order to support LCS, CAIDA uses the flight computers, SOCRRATES, and playbacks. The flight computers can be used for performance testing of the LCS gateway. The gateway will look at each CUI coming off of the spacecraft in raw counts, if the value of the CUI changes the gateway will process the CUIs, if it hasn't the CUI will be ignored. To test the gateway, the values for thousands of CUIs are changed to stress the system and see how many changes the system can handle. I assisted in running these scripts and similar scripts, ensuring that data was successfully flowing from the Honeywell Test Bench to the correct customer, viewing the telemetry through DEWESoft from the test bench to ensure the correct script was running, and troubleshooting connectivity issues on CAIDA's side of the wire. With SOCRRATES, similar troubleshooting was needed. I also created scripts that would send commands to the SOCRRATES simulation to view what CUIs are associated with that command. This was useful in creating and testing the prelaunch EFT-1 display. My mentor and I also troubleshot SOCRRATES' "long prelaunch" scenario because the rocket was not in launch configuration when the rocket was supposed to launch and therefore would not simulate the launch. To troubleshoot we created a script that would send all the necessary commands to the rocket in order for it to be in launch configuration.

B. EFT-1 Displays

I made DEWESoft displays in order to monitor the progress of the prelaunch countdown and tracking of the vehicle in orbit for CAIDA and RPS. The prelaunch countdown display was created using a timeline of the countdown provided by Lockheed Martin. In the timeline it showed what scripts would be run and at what times. These scripts would send a command and then look for a response from an associated CUI. I was able to use an editor within SOCRRATES to send commands to the SOCRRATES simulation and view the changes made to the

CUIs using DEWESoft. The green buttons open that particular sub display so the user can look at that event in greater detail. Figure 1 shows the main prelaunch display.

The second DEWESoft display I created was used to track the spacecraft across the ground using latitude, longitude, and altitude sent from the Orion capsule after LAS shroud separation. We were also able to get video coming off of the spacecraft. It was an extremely satisfying experience to see the capsule was passing over a country on the map and then looking at the video to see the country. Figure 2 shows a picture I took as the spacecraft was passing over Puerto Rico as one of the members on the launch team was from Puerto Rico.

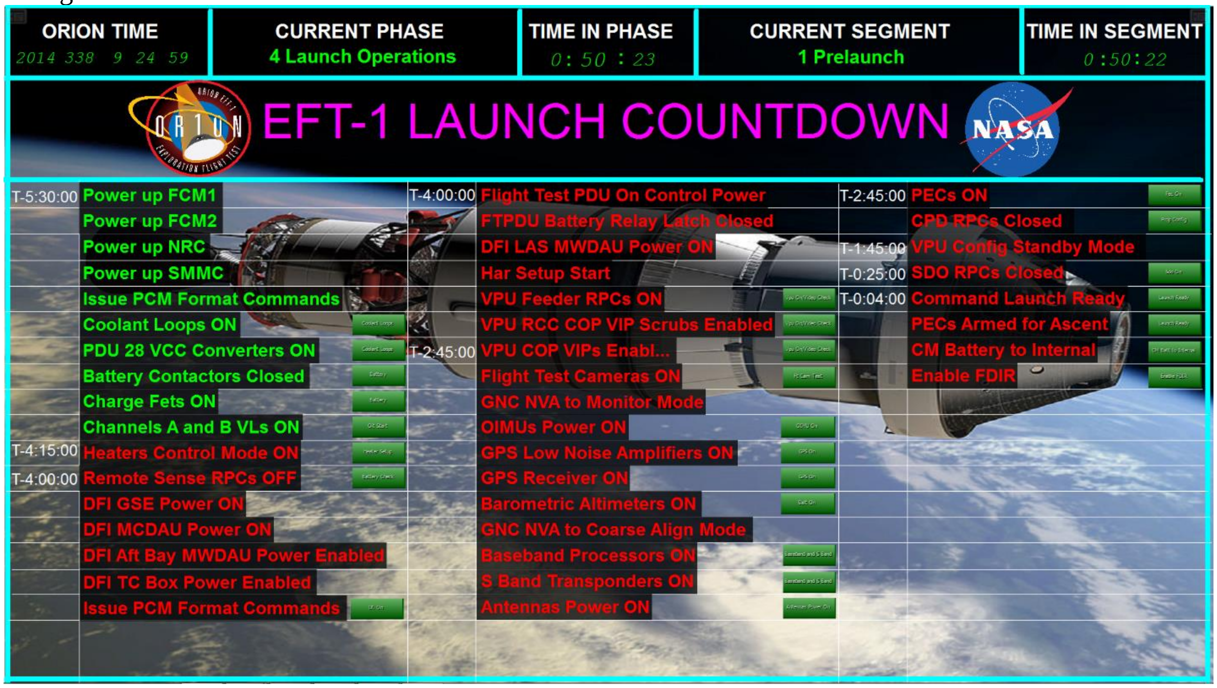


Figure 1. DEWESoft prelaunch display

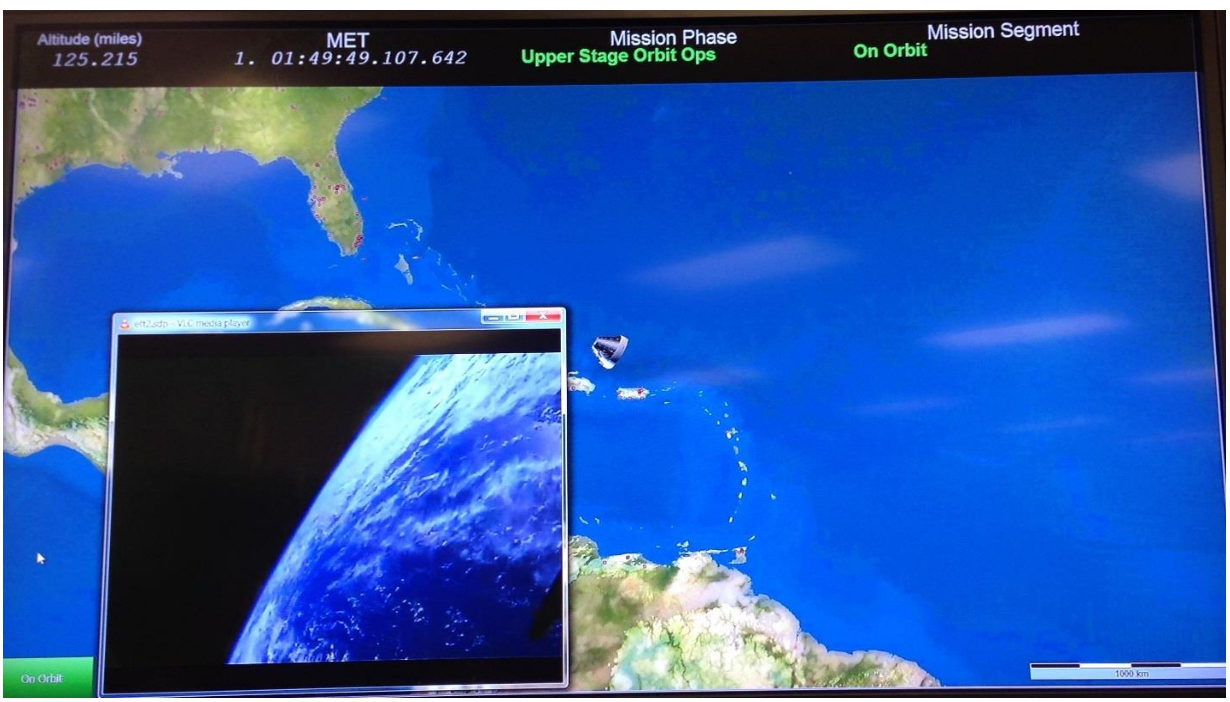


Figure 2. DEWESoft on orbit display with live video from spacecraft (not part of display)

C. Automated Creation of DEWESoft Setup

The first step to automating the creation of DEWESoft setups is parsing out the XTCE and creating a SQLite database for all the information associated with each CUI. The major improvement over the databases created previously is that the new databases support both Orion and SLS. A major difference between the SLS and Orion XTCE is that some SLS CUIs have multiple raw count to engineering unit conversion polynomials depending on the raw count, similarly SLS also uses linear splines where the raw count and engineering unit at certain points are given and the user must interpolate in between. Other minor improvements were made such as the inclusion of the enumeration data type. The enumeration data type has a list of strings associated with integers, a common example would be 0 is "Off" and 1 is "On".

To view the old databases Samuel Harris and Justin Ruscoe created a SQLite GUI so the user could view what was in the database without having to know the command prompt commands. I made it so it could read both the new database and the old database.

Samuel Harris and Justin Ruscoe created an XML generator for DEWESoft that would create a setup if the user gave a list of CUIs and Content IDs. DEWESoft setups and displays are saved as XML files, so what their program did was create the DEWESoft XML line by line. I redesigned most of the program and added features, other ways to create the setup as requested by the customer, and a way to update and add to existing displays. The new XML generator I created supports multiple streams, meaning it can now simultaneously support both SLS and Orion displays. The user can input the SQLite database, a text file containing all of the Content IDs to be used (figure 3), and text files containing lists of CUIs to be used (figure 4). There are four options to creating a setup: create a setup for all the Content IDs in the database for each CUI in the CUI file provided by the user, create a setup for all the Content IDs in the Content ID file provided by the user associated with each CUI in the CUI file, create a setup where the user selects individually which Content IDs to include for each CUI, and create a setup for all CUIs in each Content ID in the list of Content IDs (figure 5). The enumeration data type is now included in the program so that the user no longer has to look through the XTCE to find the strings associated with each integer value for that data type. One of the major problems with the previous automated setups was that only one degree raw count to engineering unit conversion polynomials were included because DEWESoft can only apply a factor and offset directly to the CUI (figure 9). To get past this, with CUIs that have greater than one degree polynomials a math channel is created so that all factors can be applied to the CUI (figures 8 and 10). Another feature of the program is the inclusion of Orion and DEM time, the automation of this process saves the users the time it takes to convert the raw time from seconds to years, days, hours, minutes, and seconds. A major improvement is the ability to update old setups and add to setups while preserving the display. When a new XTCE comes out, where each CUI is located in the telemetry stream could change which would mean the display would no longer show that CUI correctly. To solve this, each CUI is read from the existing setup and the index of that CUI is added to a dictionary where the keys are the CUI names and the values are that CUI's old and new indexes. Once all the CUIs are added, the new indexes will be added for each CUI in the same dictionary. After that the previous display is read through and all the old indexes are updated with the new indexes. Some minor features include each Content ID is sorted in numerical order, each CUI within the Content ID is sorted in alphanumerical order, the colors of each CUI match their respective Content ID, Content IDs are created with the minimum amount of extra features to save on memory, CUIs include their respective Content ID in their name, each CUI has the short description and units included in its setup, the textboxes are automatically populated to save user time where the defaults are read from text files in the main "User Files"

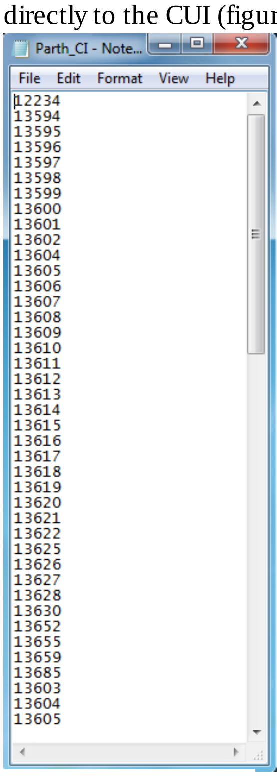


Figure 3 Content IDs provided by the user

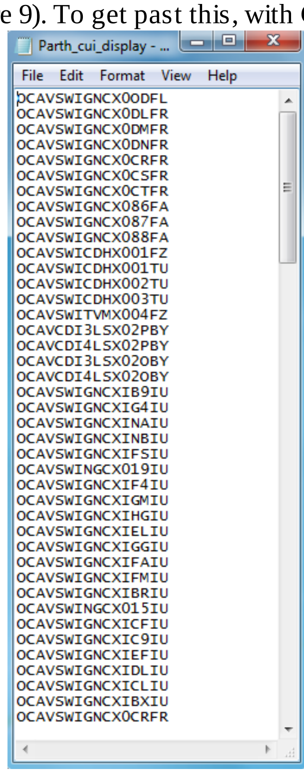


Figure 4 CUIs provided by the user

folder (figure 6), the output file name and location can be specified by the user (figure 7), and documentation and shared files such as the database files are in the same folder so there is no need for duplicate copies that take up space. For the ease of the user I created a way to install the program onto the computer so when there is an update, the previous installed program will be overwritten.

Orion CUIs will be created in Stream 0, SLS CUIs will be created in Stream 1, and Stream 2 is not used

The Orion database used is called “Orion.db”, the SLS database used is called ”SLS.db”, ICPS is not used. The databases are located in “Databases and XTCEs” folder

Name of file to be created. Path can be changed with “File” menu (see images below)

When checked, enumerations will be added to that CUI’s “Discrete Display” design widget

When checked, enumerations from old DEWESoft setup will be updated, if not checked old enumerations will not be updated

Button creates DEWESoft setup for all Content IDs in the database associated with each CUI provided in CUI list by the user

File containing list of Content IDs. Press button then navigate to desired Content ID file or type full path in text box

Files containing list of CUIs. Type name of file if file is located in “Input Files” folder or click “Find” to find path of file

Click “Find Setup” to find DEWESoft file to update and add to or type full path in textbox. If no CUIs are to be added, any of the four buttons will update the setup

When checked, a math channel will be created if raw count to engineering unit polynomial is greater than one factor and offset

When checked, an Orion Time math channel will be added for active Content IDs that have “OCAVSWICDHX001FZ” in them and DEM time will be added for each DEM

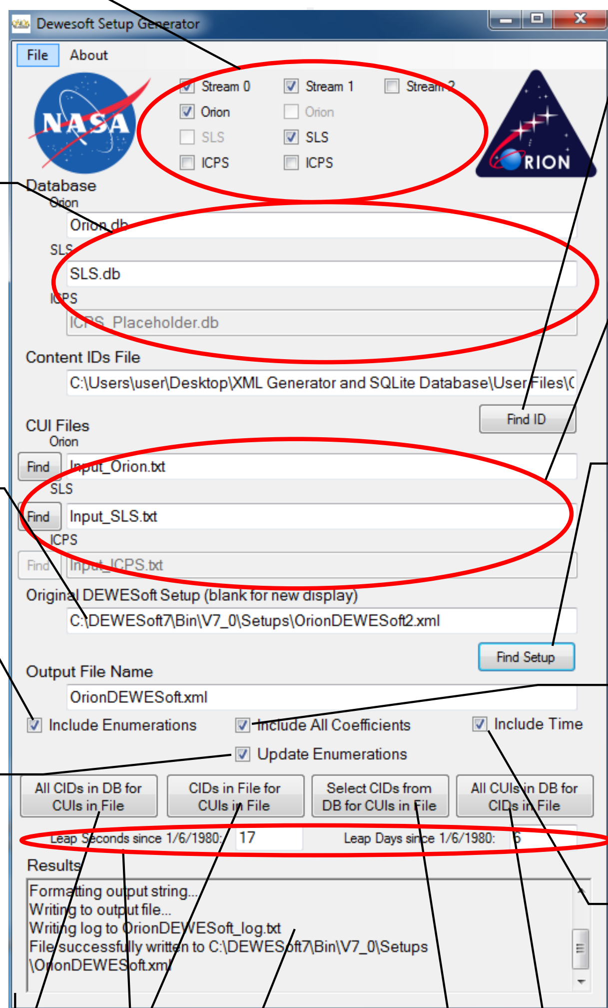


Figure 5 DEWESoft Setup Generator

Button creates DEWESoft setup for all Content IDs in the Content ID list provided by user associated with each CUI provided in CUI list by the user

Number of leap seconds and days since 1/6/1980, used for DEM Time and Orion Time

The results text shows what the program is doing and will display errors. The text from the results is written to a file in the “Setup Logs” folder

Button creates DEWESoft setup by taking all Content IDs in Content ID list provided by the user and creating a setup for each CUI in that Content ID from the database

Button creates DEWESoft setup by allowing the user to select the desired Content IDs for each CUI one at a time. The Content IDs are provided by the database and CUIs are from list of CUIs provided by the user

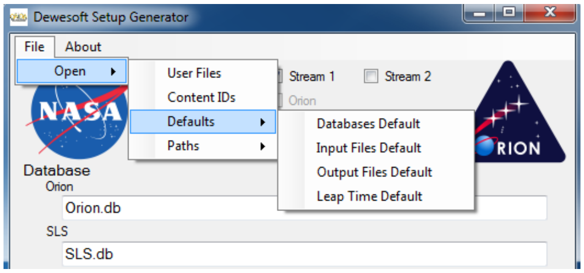


Figure 6 Easy access to file specifying defaults

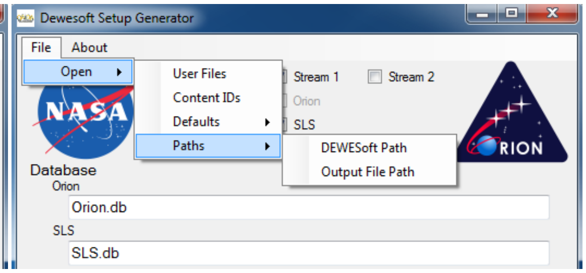
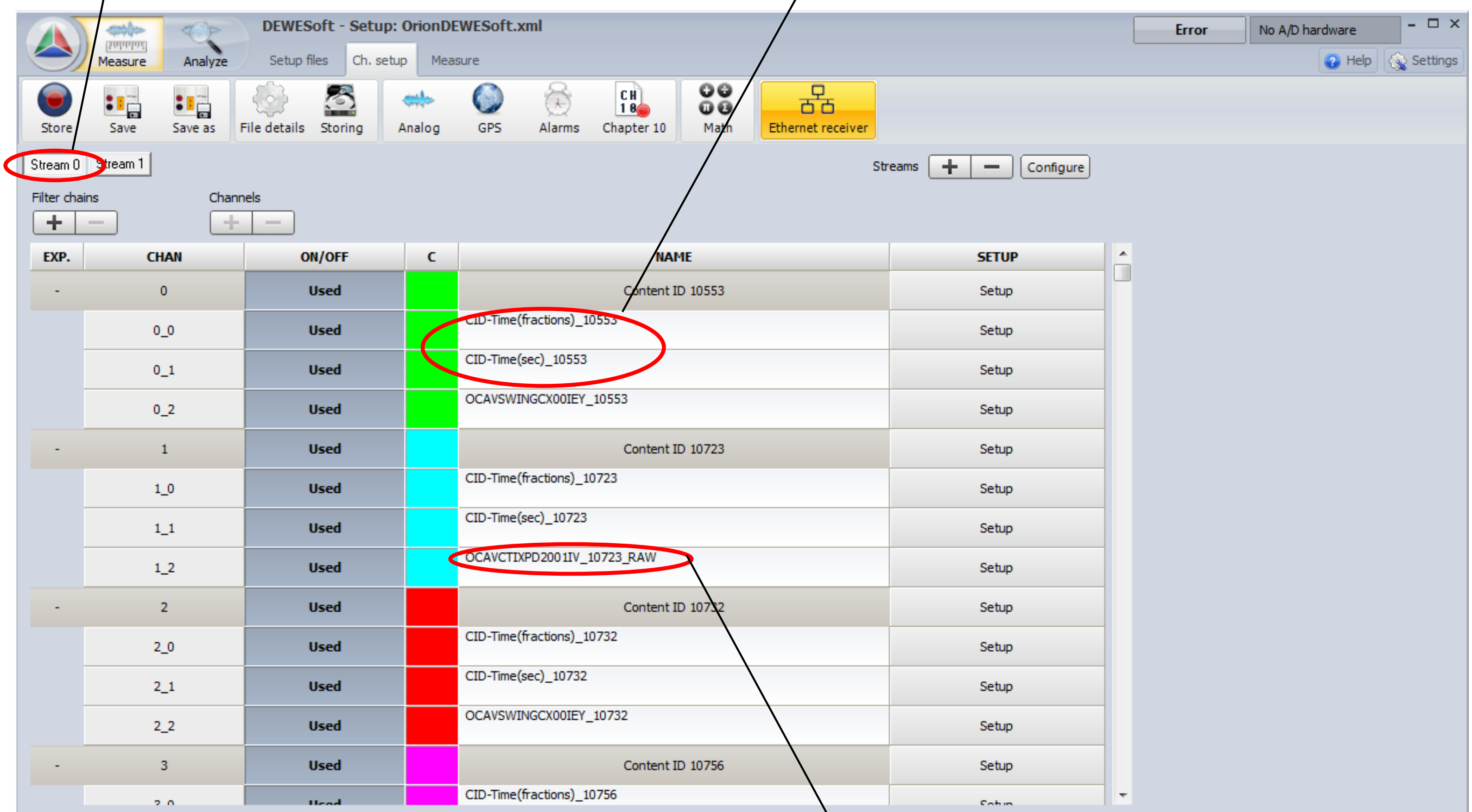


Figure 7 Easy access to files specifying paths to DEWESoft and the output file

Stream 0 setup was created with Orion CUIs using the "All Content IDs for Database for Each CUI in file" button

Added together in math channel to get DEM Time

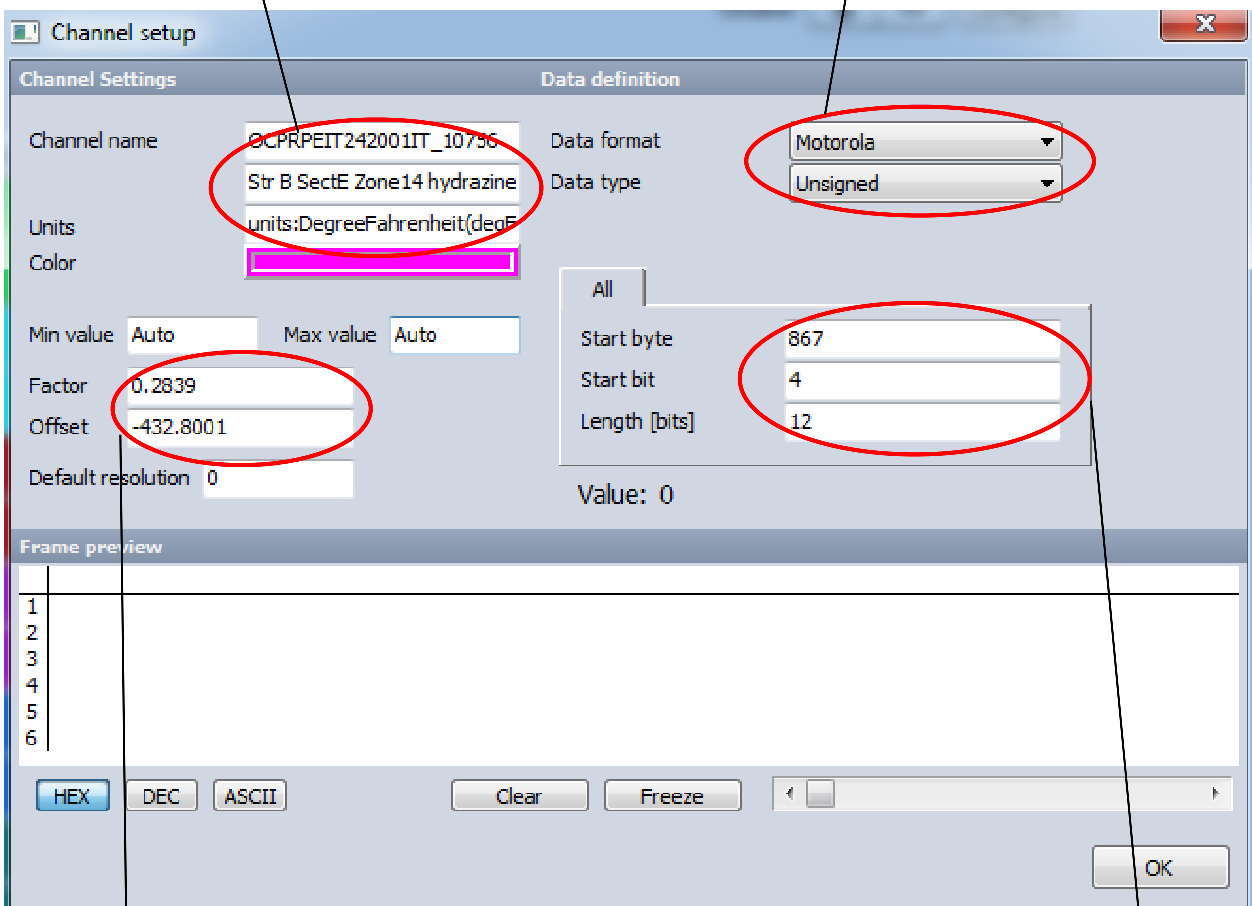


CUI with "RAW" at end indicates that polynomial is greater than one degree which is not currently supported by DEWESoft. This is solved with the creation of a math channel.

Figure 8 Stream 0: Orion

The XML Generator populates the first textbox with the CUI name and Content ID, the second textbox with the short description, and the third textbox with the engineering unit type

Data format is always "Motorola", Data type can be "Unsigned", "Signed", or "IEEE Float" and are set by program



For the raw count to engineering unit polynomial if there is only a factor and offset, it will be applied to the CUI itself, if there are more coefficients, it will be applied in a math channel

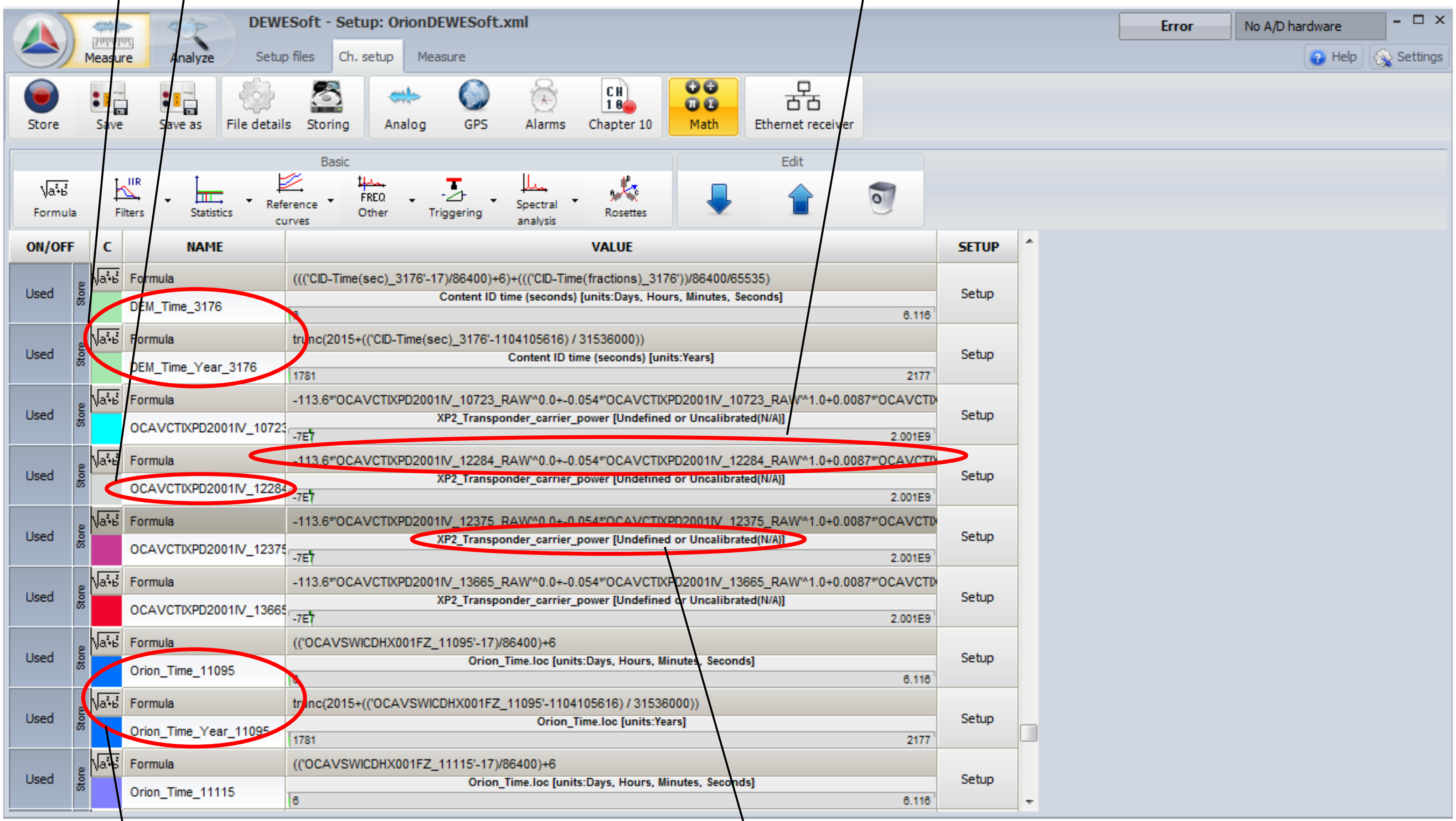
Start byte, start bit, and length in bits are found with generator

Figure 9 Channel Setup

DEM_Time (seconds, minutes, hours, days) and DEM_Time_Year (year) are created

Math channel created with name of CUI and Content ID

Formula of math channel. The XML Generator supports multiple coefficients, multiple raw count to engineering unit polynomials (SLS), and splines (SLS)



Short description and engineering unit type

Orion_Time (seconds, minutes, hours, days) and Orion_Time_Year (year) are created

Figure 10 Math

III. Conclusion

The work of previous interns and of myself regarding DEWESoft setups should significantly reduce the amount of time it takes to create setups for CAIDA and RPS. These programs now support both SLS and Orion and are far more comprehensive than they were previously. However, the programs do not support the use of ICPS telemetry but placeholders have been put in place in the event CAIDA or RPS will use ICPS. ICPS was not included in this version because the telemetry will come down as PCM whereas SLS and MPCV telemetry is Ethernet. The work with regard to EFT-1 helped RPS track the progress of the countdown and mission. RPS was one of the few mission critical systems from the LCC involved with EFT-1. The assisting in the day-to-day operations in the lab helped reduce workload for my mentor and assured CAIDA could still run routine work if he was gone.

Acknowledgments

I would like to thank my mentor Curtis Williams for selecting me for this amazing opportunity as well as William Brim, and William Johnston all of whom have taught me much regarding telemetry, NASA, and life in general. I would also like to thank the education office especially Rose Austin and Grace Johnson for all the work they have done to run the intern program. Lastly, I would like to thank USRA for providing the funds to enable this internship.