



National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, Texas 77058

October 2003

Interface Design Document
for the
Human Research Facility
Common Software

LS-71062-8D

PROJECT DOCUMENT APPROVAL SHEET

DOCUMENT NUMBER

LS-71062-8D

DATE

10/14/03

**NO. OF
PAGES**

39

TITLE:

Interface Design Document for the Human Research Facility Common Software

APPROVED:

NT3/GFE Assurance Branch

Original Signature on File

APPROVED:EB/L. Bauer
HRF Rack Integration Manager

Original Signature on File 10/14/03

DATE**PREPARED BY****CHANGE APPROVALS****CHANGE
NUMBER**

Report Number	LS-71062-8D	Date	10/07/03
----------------------	-------------	-------------	----------

**Interface Design Document
for the
Human Research Facility
Common Software**

Prepared by:	<u>Original Signature on File</u> Mike Romell Software Developer	<u>10/07/03</u> Date
Approved by:	<u>Original Signature on File</u> Margaret Klee Software Project Manager	<u>10/07/03</u> Date
Approved by:	<u>Original Signature on File</u> Mark G. Scott Software Quality Assurance	<u>10/07/03</u> Date
Approved by:	<u>Original Signature on File</u> Jerry McDonald Technical Work Plan Manager	<u>10/07/03</u> Date

Prepared by:
Lockheed Martin Space Operations
Houston, Texas
for
National Aeronautics and Space Administration
Johnson Space Center

REVISION/CHANGE APPROVALS					
Date	Change Number	Prepared by	Approved by:		
			Unit Manager	SR&QA Manager	Projects Manager

DOCUMENT NUMBER LS-71062-8		DOCUMENT CHANGE/ REVISION LOG		PAGE <u> 1 </u> OF <u> 1 </u>
CHANGE/ REVISION	DATE	DESCRIPTION OF CHANGE	PAGES AFFECTED	
BASIC	07/12/99	BASELINE	All	
A	05/09/00	Complete Revision	All	
B	9/26/01	Complete Revision	All	
C	3/15/02	Complete Revision	All	
D	10/14/03	Complete Revision	All	
Altered pages must be typed and distributed for insertion.				

CONTENTS

Section		Page
1.0	<u>INTRODUCTION</u>	1-1
1.1	COMMON SOFTWARE OVERVIEW	1-1
1.2	COMMON SOFTWARE DEVELOPER ENVIRONMENTS	1-3
1.3	COMMON SOFTWARE NAMED PIPE INTERFACE	1-3
1.4	COMMON SOFTWARE FILE INTERFACE	1-3
2.0	<u>REFERENCED DOCUMENTS</u>	2-1
3.0	<u>INTERFACE DESIGN</u>	3-1
3.1	FUNCTION CALLS FOR COMMUNICATING WITH THE COMMON SOFTWARE RACK INTERFACE	3-1
3.2	CLIENT COMMANDS ISSUED TO COMMON SOFTWARE RACK INTERFACE	3-2
3.2.1	<u>Connect Message</u>	3-2
3.2.2	<u>Update File Location Message</u>	3-2
3.2.3	<u>Batch Downlink On Message</u>	3-3
3.2.4	<u>Batch Downlink Off Message</u>	3-3
3.2.5	<u>Ancillary Data Control Message</u>	3-4
3.2.6	<u>Rack Time Request Message</u>	3-4
3.2.7	<u>Health and Status Message</u>	3-5
3.2.8	<u>Payload Executive Processor (PEP) Bundle Request Message</u>	3-5
3.2.9	<u>PEP Execution Request Message</u>	3-6
3.2.10	<u>Local Client Message</u>	3-6
3.2.11	<u>Remote Client Message</u>	3-7
3.2.12	<u>Sending Data Packets to the Common Software Rack Interface Server</u>	3-7
3.3	MESSAGES AND RESPONSES ISSUED TO CLIENT BY COMMON SOFTWARE RACK INTERFACE	3-8
3.3.1	<u>Ancillary Data Response</u>	3-8
3.3.2	<u>Broadcast Ancillary Data Response</u>	3-8
3.3.3	<u>Rack Time Response</u>	3-10
3.3.4	<u>Generic Client Message</u>	3-11
3.4	COMMON SOFTWARE GROUND COMMANDS	3-11
3.4.1	<u>CONTROL COMMANDS</u>	3-12
3.4.2	<u>FILE COMMANDS</u>	3-17
3.5	ACCESSING THE COMMON SOFTWARE DYNAMIC LINK LIBRARY	3-26
3.5.1	<u>Common Software Dynamic Link Library Function Prototypes and Descriptions</u>	3-26
3.6	COMMON SOFTWARE HEALTH AND STATUS FORMAT	3-29

CONTENTS (Cont'd)

Section	Page
3.7 BATCH DOWNLINK	3-30
3.7.2 <u>Using Same File Name</u>	3-31
3.7.3 <u>Downlink File ID File</u>	3-32
3.7.4 <u>Downlink List</u>	3-32
APPENDIX A COMMON SOFTWARE TELEMETRY HEADER DEFINITIONS	A-1
APPENDIX B CSW SPECIFIC FILE IDS USED IN TELEMETRY HEADERS	B-1

LIST OF TABLES

Table		Page
3.2.1-1	CONNECT MESSAGE STRUCTURE	3-2
3.2.2-1	UPDATE FILE LOCATION FILE MESSAGE STRUCTURE	3-2
3.2.3-1	BATCH DOWNLINK ON MESSAGE STRUCTURE	3-3
3.2.4-1	BATCH DOWNLINK OFF MESSAGE STRUCTURE	3-3
3.2.5-1	ANCILLARY DATA CONTROL MESSAGE STRUCTURE	3-4
3.2.6-1	RACK TIME REQUEST MESSAGE STRUCTURE	3-4
3.2.7-1	HEALTH AND STATUS MESSAGE STRUCTURE	3-5
3.2.8-1	PEP BUNDLE REQUEST MESSAGE STRUCTURE	3-5
3.2.9-1	PEP EXECUTION REQUEST MESSAGE STRUCTURE	3-6
3.2.10-1	LOCAL CLIENT MESSAGE STRUCTURE	3-6
3.2.11-1	REMOTE CLIENT MESSAGE STRUCTURE	3-7
3.2.12-1	DATA PACKET STRUCTURE	3-7
3.3.1-1	ANCILLARY DATA RESPONSE STRUCTURE	3-8
3.3.2-1	BROADCAST ANCILLARY DATA RESPONSE STRUCTURE	3-8
3.3.2-2	CCSDS PRIMARY HEADER STRUCTURE	3-9
3.3.2-3	CCSDS SECONDARY HEADER STRUCTURE	3-9
3.3.3-1	RACK TIME RESPONSE STRUCTURE	3-10
3.3.4-1	GENERIC CLIENT MESSAGE STRUCTURE	3-11

ACRONYMS AND ABBREVIATIONS

APID	Application Programming Interface Definition
CCSDS	Consultative Committee for Space Data Systems
CRC	Cyclic Redundancy Checksum
CSCI	Computer Software Configuration Item
CSW	Common Software
DLL	Dynamic Link Library
EMU	EXPRESS Memory Unit
EXPRESS	EXpedite the PROcessing of Experiments to the Space Station
FM	File Manager
GUI	Graphical User Interface
HRF	Human Research Facility
I/O	Input/Output
ICD	Interface Control Document
ID	Identification
LSDS	Life Science Data Systems
LSB	Least Significant Byte
MDM	Multiplexer Demultiplexer
MSB	Most Significant Byte
N/A	Not Applicable
PC	Portable Computer
PEP	Payload Executive Processor
PLD	Payload
RI	Rack Interface
RIC	Rack Interface Controller
SDD	Software Design Document
SRS	Software Requirement Specifications
UI	User Interface
UIL	User Interface Language
UTC	Universal Time Code
VDD	Version Description Document

1.0 INTRODUCTION

This document describes the interfaces of the Human Research Facility (HRF) Common Software (CSW) required for payload client software connectivity. The CSW provides user and client software access to launching payload software and controlling telemetry of payload data to the ground by passing the corresponding data to the Rack Interface Controller (RIC). This document provides all information needed to generate an interface to the CSW.

1.1 COMMON SOFTWARE OVERVIEW

The CSW consists of three Computer Software Configuration Items (CSCIs): the Server or Rack Interface (RI), the File Manager (FM), and the User Interface (UI). These CSCIs communicate through named pipes. The UI and RI communicate through a single synchronized named pipe. Commands, requests, and status messages flow between the UI and RI. The FM communicates with the Server through two named pipe instances. The FM writer pipe is used for writing downlink packets and messages to the Server, and the FM reader pipe is used for reading uplink messages and commands from the Server. (Ref. Figure 1-1)

The UI and FM communicate indirectly by sending messages and commands to each other through the Server. The UI and FM also share access to the downlink list file. This file contains the list of all files, experiment and data files, to be sent to the ground by batch downlink telemetry. The downlink list file is created by the FM and can be changed by either the FM or the UI. Additionally, the UI and FM share access to the downlink file ID file which holds the next file identifier to be assigned to a file added to the downlink list file.

Client software may also communicate with the server using a named pipe. Two named pipe connections are required for communication. One pipe is solely used for writing to the Server and the other for reading from the Server. Once a connection has been established, client software may transmit real time telemetry data to the server as well as receive periodic health and status or ancillary data.

The FM primarily handles batch downlink of files found in the downlink list. The FM takes each file on the downlink list, breaks it up into packets, and sends each packet to the Server.

The UI provides a Graphical User Interface (GUI) for the crewmember to launch HRF payload software, view CSW log files, and control batch downlink telemetry. In the UI, HRF payload software is listed under an experiments listing and a payload or instruments listing. Additionally, the UI displays the downlink list that the FM uses. From this display, the crewmember can impede files from being sent, add files to the downlink list, and start or stop the batch downlink telemetry process.

Example scenario: The server, FM, and UI are all running on the HRF PC connected to the RIC by Ethernet. The UI and FM have connected to the Server. Batch file downlink is initiated from a command sent from the ground through the RIC. The “start telemetry” command is received by the Server which sends it to the FM.

Upon receiving the command, the FM goes through the downlink list. The FM takes each enabled file on the downlink list, breaks it up into packets, and sends each packet to the Server. The Server accepts the packets, puts on the appropriate Expedite

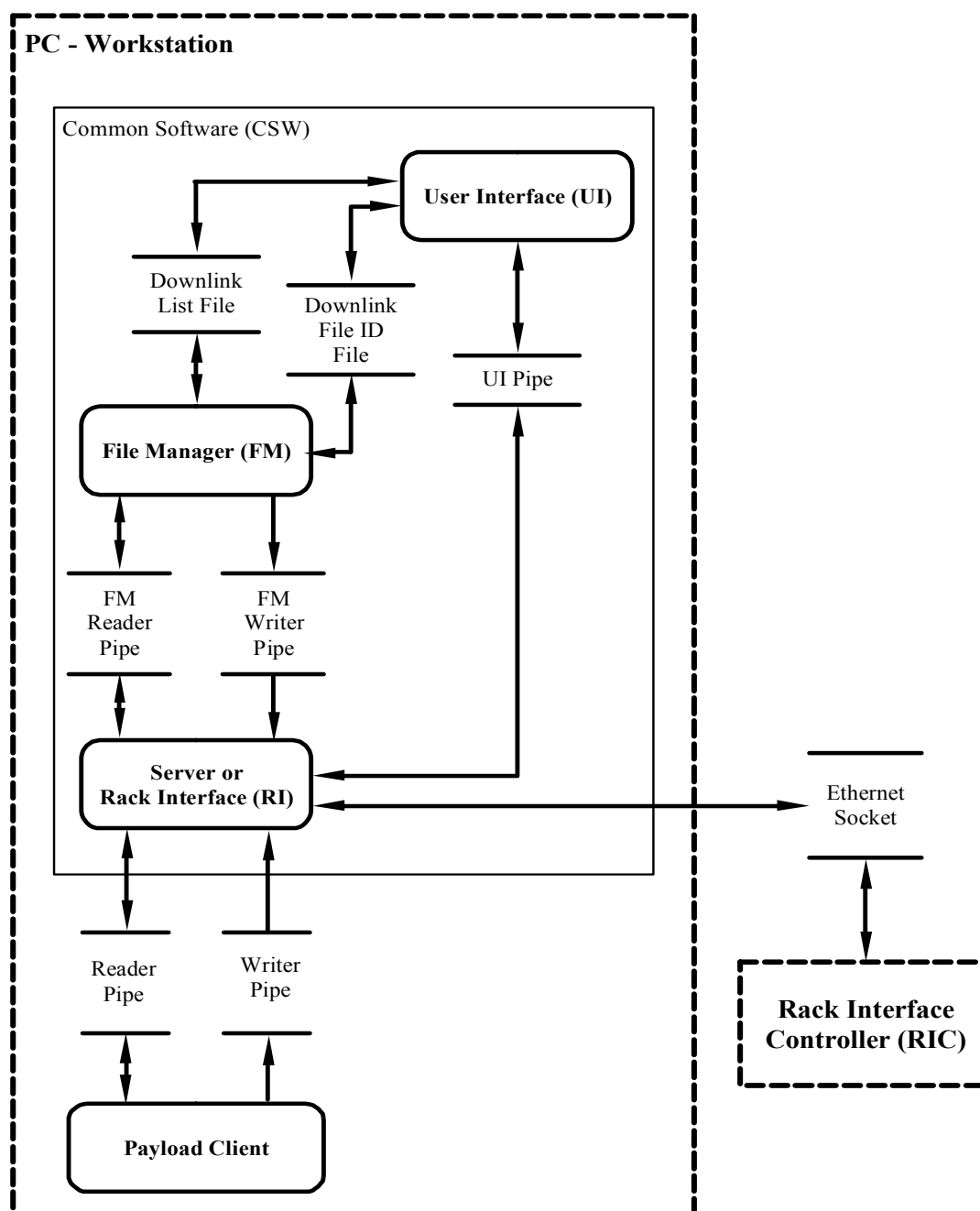


Figure 1-1. CSW Interface Diagram

the Processing of Experiments to the Space Station (EXPRESS) header and sends it to the RIC, which sends the data packets to the ground.

1.2 COMMON SOFTWARE DEVELOPER ENVIRONMENTS

All CSW CSCIs were developed for MS Windows NT 4.0 and Windows 2000.

Developer Environments

- Server - MS Visual C++ 6.0
- FM - MS Visual C++ 6.0
- UI - MS Visual BASIC 6.0
- UI help file - MS Help Workshop 6.0

NOTE: The MS Help Workshop is an application bundled with the MS Visual C++ developer.

1.3 COMMON SOFTWARE NAMED PIPE INTERFACE

The named pipes used by the CSW team in the RI software are overlapped pipes which allow for asynchronous I/O. An overlapped structure must be associated with each read and write operation. These structures are used to signal that a read or write action has occurred on the named pipe. The CreateFile function is used in Visual C++ and BASIC to connect to the server. The ReadFile and WriteFile functions are used in Visual C++ and BASIC to communicate with the Server.

Reference Section 3.5 for specific functions to communicate with the server.

1.4 COMMON SOFTWARE FILE INTERFACE

Each payload client should create a file location file that defines the location of their data files. The file location files are used by the CSW FM, to add data files to the downlink list. This downlink list can be updated by a user through the UI, by payload client software through a command to the CSW Server, or by a ground issued command.

The user can inhibit files on the downlink list from being sent. Additionally, the user can add files to the downlink list individually or can add files by using the 'Add Data' button on the Experiment window.

The Add Data button sends a command to the FM that includes a path to a file location file. The FM then reads the file location file and updates the downlink list accordingly by adding files that meet the file parameters contained in the file location file.

Each line in the file location file must specify a descriptive experiment name, an experiment identifier, a priority level from 1 to 3, most important to least important, a value of 0 or 1 for the recursion flag, and a file prototype. The priority level determines which files get sent first when telemetry is active. See Section 3.7 for information on file naming convention and file location file format.

NOTE: The location of the data files and their priority levels will be determined on a case by case basis.

2.0 REFERENCED DOCUMENTS

<u>Document Number</u>	<u>Rev.</u>	<u>Document Title</u>
LS-71083-A	Basic	Software Design Document for the Human Research Facility
LS-71062-9-2		Common Software Version Description Document (VDD) for the Human Research Facility
LS-71062	C	Software Requirements Specification (SRS) for the Human Research Facility (HRF) Common Software
Boeing D683-43525-1	Issue A Draft 8	EXPRESS Software Interface Control Document - HRF
MIL-STD-1553	B	Digital Time Division Command/Response Multiplex Data Bus

3.0 INTERFACE DESIGN

The interface between the CSW RI and payload client software must handle the following actions:

- Sending Messages and Data Packets to the RI
- Receiving Messages from the RI
- Accessing the CSW Dynamic Link Library (DLL)

3.1 FUNCTION CALLS FOR COMMUNICATING WITH THE COMMON SOFTWARE RACK INTERFACE

The following function calls are examples that may be used in payload software to connect to the CSW as well as read and write to the CSW. Use of the CSW DLL is recommended. (Ref. Section 3.5)

```
HANDLE m_write_handle = CreateFile("\\\\.\\PIPE\\CSW\\",GENERIC_WRITE || GENERIC_READ,
    FILE_SHARE_WRITE || FILE_SHARE_READ,NULL,OPEN_EXISTING,FILE_ATTRIBUTE_NORMAL
    & FILE_FLAG_OVERLAPPED,NULL);
```

```
int STDCALL Read_CSW (void* data_buffer, int data_size)
{
    DWORD bytes_read = 0;
    OVERLAPPED olread;
    olread.hEvent = CreateEvent (NULL,TRUE, FALSE, NULL);
    read_ret = ReadFile(m_read_handle,data_buffer,data_size,&abytes_read,&olread);
}
```

```
int STDCALL Write_CSW (HANDLE send_handle,void* data)string,int send_size)
{
    int send_ret = 0;
    DWORD bytes_written = 0;
    OVERLAPPED olsend;
    olsend.hEvent = CreateEvent(NULL,TRUE,FALSE,NULL);
    send_ret = WriteFile(send_handle,data_string,send_size,&bytes_written,&olsend);
}
```

3.2 CLIENT COMMANDS ISSUED TO COMMON SOFTWARE RACK INTERFACE

3.2.1 Connect Message

The connect message is used to tell the server the name of the client connecting. This message must be sent to the server immediately after connection. Use the “Client Type” to inform the server whether reading or writing will occur over this pipe connection. If reading from the server with this connection, set the “Client Type” to 0x13. If writing to the server with this connection, set the “Client Type” to 0x14.

TABLE 3.2.1-1. CONNECT MESSAGE STRUCTURE

Client Message Field	Description	Size
Client ID High Byte	ID Assigned to Client	byte
Client ID Low Byte	ID Assigned to Client	byte
Client Type	Reader - 0x13 Writer - 0x14	byte

3.2.2 Update File Location Message

The update file location file message is used to tell the server that new data files have been recorded and should be added to the downlink list. Give a file path to the location of the file location file in the “Additional Data” field. The Size Byte fields should be set to the size of the entire message from the Sync High field to the end of the message.

TABLE 3.2.2-1. UPDATE FILE LOCATION FILE MESSAGE STRUCTURE

Client Message Field	Description	Size
Sync High Byte	0x55	byte
Sync Low Byte	0xAA	byte
Destination	File Manager (FM) - 0x03	byte
Client ID High Byte	ID Assigned to Client	byte
Client ID Low Byte	ID Assigned to Client	byte
Size High Byte	Message Packet Size in Bytes (High)	byte
Size Low Byte	Message Packet Size in Bytes (Low)	byte
Message High Byte	Update File Location File - 0x00	byte
Message Low Byte	Update File Location File - 0x81	byte
Additional Data	File Path of the File Location File - Null Terminated (Supports all Win32 file names)	up to 128 bytes

3.2.3 Batch Downlink On Message

The batch downlink on message is used to enable the batch downlink capabilities of the server after time critical downlink has finished. It is important that this message be sent immediately after completing the real time work.

TABLE 3.2.3-1. BATCH DOWNLINK ON MESSAGE STRUCTURE

Client Message Field	Description	Size
Sync High Byte	0x55	byte
Sync Low Byte	0xAA	byte
Destination	Server - 0x01	byte
Client ID High Byte	ID Assigned to Client	byte
Client ID Low Byte	ID Assigned to Client	byte
Size High Byte	Message Packet Size in Bytes (High) - 0x00	byte
Size Low Byte	Message Packet Size in Bytes (Low) - 0x09	byte
Message High Byte	Batch Downlink On - 0x00	byte
Message Low Byte	Batch Downlink On - 0x18	byte

3.2.4 Batch Downlink Off Message

The batch downlink off message is used to stop the server from beginning batch downlink during time critical real time data downlink.

TABLE 3.2.4-1. BATCH DOWNLINK OFF MESSAGE STRUCTURE

Client Message Field	Description	Size
Sync High Byte	0x55	byte
Sync Low Byte	0xAA	byte
Destination	Server - 0x01	byte
Client ID High Byte	ID Assigned to Client	byte
Client ID Low Byte	ID Assigned to Client	byte
Size High Byte	Message Packet Size in Bytes (High) - 0x00	byte
Size Low Byte	Message Packet Size in Bytes (Low) - 0x09	byte
Message High Byte	Batch Downlink Off - 0x00	byte
Message Low Byte	Batch Downlink Off - 0x19	byte

3.2.5 Ancillary Data Control Message

The ancillary data control message is used to request ancillary data sets.

TABLE 3.2.5-1. ANCILLARY DATA CONTROL MESSAGE STRUCTURE

Client Message Field	Description	Size
Sync High Byte	0x55	byte
Sync Low Byte	0xAA	byte
Destination	Server - 0x01	byte
Client ID High Byte	ID Assigned to Client	byte
Client ID Low Byte	ID Assigned to Client	byte
Size High Byte	Message Packet Size in Bytes (High) - 0x00	byte
Size Low Byte	Message Packet Size in Bytes (Low) - 0x0B	byte
Message High Byte	Ancillary Data Control - 0x00	byte
Message Low Byte	Ancillary Data Control - 0x20	byte
Request ID	One Time Data Request - 0x00 Add Entry - 0x01 Delete Entry - 0x03	byte
Data Set ID	Data Sets - 0x01 to 0x64 All Broadcast Ancillary Data - 0xE4 Request Broadcast - 0x80 to 0xE3	byte

3.2.6 Rack Time Request Message

The rack time request message is used to request rack time from the server. One shot of time will be sent to the client per request.

TABLE 3.2.6-1. RACK TIME REQUEST MESSAGE STRUCTURE

Client Message Field	Description	Size
Sync High Byte	0x55	byte
Sync Low Byte	0xAA	byte
Destination	Server - 0x01	byte
Client ID High Byte	ID Assigned to Client	byte
Client ID Low Byte	ID Assigned to Client	byte
Size High Byte	Message Packet Size in Bytes (High) - 0x00	byte
Size Low Byte	Message Packet Size in Bytes (Low) - 0x09	byte
Message High Byte	Rack Time Request - 0x00	byte
Message Low Byte	Rack Time Request - 0x4D	byte

3.2.7 Health and Status Message

The health and status message is used by clients sending health and status data to the ground in order to update their data being sent. The Size Byte fields should be set to the size of the entire message from the Sync High field to the end of the message. The cyclic counter should begin with zero and increment by one for consecutive packets. The size of the health and status data is assigned to the client. The size assigned is based on both need and availability.

TABLE 3.2.7-1. HEALTH AND STATUS MESSAGE STRUCTURE

Client Message Field	Description	Size
Sync High Byte	0x55	byte
Sync Low Byte	0xAA	byte
Destination	Server - 0x01	byte
Client ID High Byte	ID Assigned to Client	byte
Client ID Low Byte	ID Assigned to Client	byte
Size High Byte	Message Packet Size in Bytes (High)	byte
Size Low Byte	Message Packet Size in Bytes (Low)	byte
Message High Byte	Health and Status - 0x00	byte
Message Low Byte	Health and Status - 0x9B	byte
Cyclic Counter High Byte	Starts at 0 and increment by one for every packet.	byte
Cyclic Counter Low Byte	Starts at 0 and increment by one for every packet.	byte
Health and Status Data	Assigned to Client and Configurable	bytes

3.2.8 Payload Executive Processor (PEP) Bundle Request Message

The PEP bundle request message is used to send a series of commands to the RIC. This message must be sent before the PEP execution request message.

TABLE 3.2.8-1. PEP BUNDLE REQUEST MESSAGE STRUCTURE

Client Message Field	Description	Size
Sync High Byte	0x55	byte
Sync Low Byte	0xAA	byte
Destination	Server - 0x01	byte
Client ID High Byte	ID Assigned to Client	byte
Client ID Low Byte	ID Assigned to Client	byte
Size High Byte	Message Packet Size in Bytes (High) - 0x00	byte
Size Low Byte	Message Packet Size in Bytes (Low) - 0x0D	byte
Message High Byte	PEP Bundle Request - 0x00	byte
Message Low Byte	PEP Bundle Request - 0x49	byte
Request ID High Byte	Any request message - 0x00	byte
Request ID Low Byte	Install - 0x15 Halt - 0x16 Remove - 0x17	byte
Bundle ID High Byte	Unique Value Relating to the UIL Bundle Assigned by the PEP	byte
Bundle ID Low Byte	Unique Value Relating to the UIL Bundle Assigned by the PEP	byte

3.2.9 PEP Execution Request Message

The PEP execution request message is used to tell the RIC to perform the PEP bundle request previously sent.

TABLE 3.2.9-1. PEP EXECUTION REQUEST MESSAGE STRUCTURE

Client Message Field	Description	Size
Sync High Byte	0x55	byte
Sync Low Byte	0xAA	byte
Destination	Server - 0x01	byte
Client ID High Byte	ID Assigned to Client	byte
Client ID Low Byte	ID Assigned to Client	byte
Size High Byte	Message Packet Size in Bytes (High) - 0x00	byte
Size Low Byte	Message Packet Size in Bytes (Low) - 0x0D	byte
Message High Byte	PEP Execution Request - 0x00	byte
Message Low Byte	PEP Execution Request - 0x4C	byte
Request ID High Byte	Any request message - 0x00	byte
Request ID Low Byte	Start - 0x12 Stop - 0x13 Resume - 0x14	byte
Payload Sequence ID High Byte	Unique Value Relating to the Sequence Assigned by the PEP	byte
Payload Sequence ID Low Byte	Unique Value Relating to the Sequence Assigned by the PEP	byte

3.2.10 Local Client Message

The local client message is defined by the client receiving the message. The local client message is from a client to another client running on the same system. The Size Byte fields should be set to the size of the entire message from the Sync High field to the end of the message.

TABLE 3.2.10-1. LOCAL CLIENT MESSAGE STRUCTURE

Client Message Field	Description	Size
Sync High Byte	0x55	byte
Sync Low Byte	0xAA	byte
Destination	Local Client ID to Send to	byte
Client ID High Byte	ID Assigned to Client	byte
Client ID Low Byte	ID Assigned to Client	byte
Size High Byte	Message packet size in bytes (High)	byte
Size Low Byte	Message packet size in bytes (Low)	byte
Message High Byte	Local Client Message - 0x00	byte
Message Low Byte	Local Client Message - 0x03	byte
Message Word	Message Data	2 bytes
Message Size High	Size of following message in bytes (High)	byte
Message Size Low	Size of following message in bytes (Low)	byte
Message		up to 1248 bytes

3.2.11 Remote Client Message

The remote client message is understood by the RIC. The remote client message is sent from a client to another client on a remote system. The Size Byte fields should be set to the size of the entire message from the Sync High field to the end of the message.

TABLE 3.2.11-1. REMOTE CLIENT MESSAGE STRUCTURE

Client Message Field	Description	Size
Sync High Byte	0x55	byte
Sync Low Byte	0xAA	byte
Destination	RIC 0x01	byte
Client ID High Byte	ID Assigned to Client	byte
Client ID Low Byte	ID Assigned to Client	byte
Size High Byte	Message Packet Size in Bytes (High)	byte
Size Low Byte	Message Packet Size in Bytes (Low)	byte
Message High Byte	Local Client Message - 0x00	byte
Message Low Byte	Local Client Message - 0x03	byte
Function Code	0x0040 to 0xFFFFE	2 bytes
Message Word	Message Word in EXPRESS Header	2 bytes
Message		Up to 1248 bytes

3.2.12 Sending Data Packets to the Common Software Rack Interface Server

The data packets are used for the payload clients to send data to the server. The Size Byte fields should be set to the size of the entire message from the Sync High field to the end of the message.

TABLE 3.2.12-1. DATA PACKET STRUCTURE

Client Message Field	Description	Size
Sync High Byte	0x55	byte
Sync Low Byte	0xAA	byte
Destination	Server - 0x01	byte
Client ID High Byte	ID Assigned to Client	byte
Client ID Low Byte	ID Assigned to Client	byte
Size High Byte	Message Packet Size in Bytes (High)	byte
Size Low Byte	Message Packet Size in Bytes (Low)	byte
Message High Byte	Data Packet - 0xDD	byte
Message Low Byte	Data Packet - 0xDD	byte
Life Science Data Systems (LSDS) Client ID Byte High	Highest bit is set to 1 when data is in LSDS format	byte
LSDS Client ID Byte Low	Rest of the two bytes is the Client ID	byte
Client Data		Up to 316200 bytes (255*1240 bytes)

3.3 MESSAGES AND RESPONSES ISSUED TO CLIENT BY COMMON SOFTWARE RACK INTERFACE

For messages received by payload software from the CSW RI Server, the first two bytes are read to determine the message.

3.3.1 Ancillary Data Response

Ancillary data is sent to the client after it has been requested.

TABLE 3.3.1-1. ANCILLARY DATA RESPONSE STRUCTURE

Client Message Field	Description	Size
Message High Byte	Ancillary Data Response - 0x00	byte
Message Low Byte	Ancillary Data Response - 0x98	byte
Data Set High Byte	Data Set ID (High)	byte
Data Set Low Byte	Data Set ID (Low)	byte
Ancillary Data	Data	46 bytes

3.3.2 Broadcast Ancillary Data Response

Broadcast ancillary data is received at a 10 Hz rate from the RIC and sent to the payload clients that have requested broadcast ancillary data.

TABLE 3.3.2-1. BROADCAST ANCILLARY DATA RESPONSE STRUCTURE

Client Message Field	Description	Size
Message High Byte	Broadcast Ancillary Data Response - 0x00	byte
Message Low Byte	Broadcast Ancillary Data Response - 0x99	byte
Consultative Committee for Space Data Systems (CCSDS) Primary Header	Reference Boeing Document D683-43525-1	6 bytes
CCSDS Secondary Header	Reference Boeing Document D683-43525-1	10 bytes
Broadcast Ancillary Data	Data	112 bytes

The CCSDS primary and secondary headers are used in commands sent from the PEP to the RIC across the PLD MDM Local MIL-STD-1553B bus interface. The headers consist of the following parameters: description, type transmit/receive, sub-address, bit assignment, and parameter name. The secondary header contains command-specific data used to identify the type of command packet being sent and when it was issued.

TABLE 3.3.2-2. CCSDS PRIMARY HEADER STRUCTURE

Field	Description	Size
Version Number		3 bits
Application Programming Interface Definition (APID) Type	System - 0 Payload - 1	1 bit
Secondary Header Flag	0 - No Secondary Header 1 - Secondary Header	1 bit
APID	Variable	11 bits
Sequence Flag		2 bits
Package Sequence Count	Increments by one with each transmission. Auto roll over at maximum.	14 bits
Package Length	Number of octets.	2 bytes

NOTE: Verify this structure with Boeing Document Number D683-4352501.

TABLE 3.3.2-3. CCSDS SECONDARY HEADER STRUCTURE

Field	Description	Size
Time - Most Significant Bytes (MSBs) of Course Time		2 bytes
Time - Least Significant Bytes (LSBs) of Course Time	LSB - 1 second	2 bytes
Fine Time		byte
Time ID		2 bits
Checkword Indicator		1 bit
Spare - Empty	0	1 bit
Packet Type	Variable	4 bits
Packet ID Word 1	Variable	2 bytes
Packet ID Word 2	Broadcast Ancillary frame number.	2 bytes

NOTE: Verify this structure with Boeing Document Number D683-4352501.

3.3.3 Rack Time Response

The rack time response is the message high and low bytes followed by the rack time described in the EXPRESS Software Interface Control Document - HRF Issue A, Draft 8.

TABLE 3.3.3-1. RACK TIME RESPONSE STRUCTURE

Client Message Field	Description	Size
Message High Byte	Rack Time Response - 0x00	byte
Message Low Byte	Rack Time Response - 0x80	byte
CCSDS Preamble	01010000 (binary)	byte
MSB of Year	19-20	byte
LSB of Year	0-99	byte
Month	1-12	byte
Day	1-31	byte
Hour	0-23	byte
Minutes	0-59	byte
Seconds	0-59	byte
Spare Sub Seconds High - High Byte	Spare sub seconds high and low bytes together range 0-15.	byte
Spare Sub Seconds High - Low Byte	Spare sub seconds high and low bytes together range 0-15.	byte
Binary Sub Seconds Low - High Byte*	Binary sub seconds high and low bytes together range 0-65525.	byte
Binary Sub Seconds Low - Low Byte*	Binary sub seconds high and low bytes together range 0-65525.	byte
Universal Time Code (UTC) - High Byte	0	byte
UTC - Low Byte	0	byte
Non-CCSDS Seconds - High Byte**	Non-CCSDS seconds high and low bytes together range 0-65535.	byte
Non-CCSDS Seconds - Low Byte**	Non-CCSDS seconds high and low bytes together range 0-65535.	byte

* **NOTE:** Combining binary sub seconds high and low, the total range is 0-1048575 at one microsecond (1 μ sec) per count.

** **NOTE:** Non-CCSDS seconds high and low bytes represent the one's portion of the seconds plus the sub seconds information of the CCSDS time converted to a straight binary count rounded to the nearest 256 microseconds (256 μ sec).

3.3.4 Generic Client Message

The generic client message is defined by the client receiving the message. The Size Byte fields will be set to the entire message size from the Message High Byte to the end of the message.

TABLE 3.3.4-1. GENERIC CLIENT MESSAGE STRUCTURE

Client Message Field	Description	Size
Message High Byte	Defined by Client	byte
Message Low Byte	Defined by Client	byte
Size High Byte	Message Packet Size in Bytes (High)	byte
Size Low Byte	Message Packet Size in Bytes (Low)	byte
Message	Defined by Client	Up to 90 bytes

3.4 COMMON SOFTWARE GROUND COMMANDS

The following tables show the formats for ground issued Common Software commands. Each command begins with the EXPRESS header and does not show the CCSDS and other required headers.

Generic Ground Command Format

The table below shows the generic format for a CSW command issued from the ground.

MSB																LSB
EXPRESS Version (0000)				EXPRESS Size												
EXPRESS Message Type (0xFFFF)																
Function Code Source (0x000F)																
Function Code Destination High								Function Code Destination Low								
CSW Destination High								CSW Destination Low								
CSW Message Size High								CSW Message Size Low								
CSW Message Type High								CSW Message Type Low								
Message Data Word 1																
...																
Message Data Word N																
The size of the message data may be anything from 0 to 45 words long. The CSW Message size includes the CSW Message Type bytes and the Message Data. EXPRESS Size = CSW Message Size + 10																

3.4.1 CONTROL COMMANDS

3.4.1.1 Batch Mode On

This command places the CSW in batch downlink mode. Batch downlink of data files may only be initiated if the CSW is in batch downlink mode.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x01)							
CSW Message Size High (0x00)								CSW Message Size Low (0x02)							
CSW Message Type High (0x00)								CSW Message Type Low (0x18)							
EXPRESS Size = CSW Message Size + 10															

3.4.1.2 Batch Mode Off

This command takes the CSW out of batch downlink mode. The CSW must be commanded back into batch downlink mode in order to downlink data files.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x01)							
CSW Message Size High (0x00)								CSW Message Size Low (0x02)							
CSW Message Type High (0x00)								CSW Message Type Low (0x19)							
EXPRESS Size = CSW Message Size + 10															

3.4.1.3 Debug Mode On

This command places the CSW in debug mode. The CSW server logs the contents of all output and input when in this mode. This extra data logged will increase the size of log files dramatically and should only be used if absolutely necessary.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x01)							
CSW Message Size High (0x00)								CSW Message Size Low (0x02)							
CSW Message Type High (0x00)								CSW Message Type Low (0x36)							
EXPRESS Size = CSW Message Size + 10															

3.4.1.4 Debug Mode Off

This command takes the CSW out of debug mode.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x01)							
CSW Message Size High (0x00)								CSW Message Size Low (0x02)							
CSW Message Type High (0x00)								CSW Message Type Low (0x37)							
EXPRESS Size = CSW Message Size + 10															

3.4.1.5 Health and Status Configuration Change

This command updates the CSW Health and Status configuration on the fly. If the Permanent Change Flag value is set (0x01) then the contents of HSConfig.txt will be rewritten with this commands configuration information. If the value is set to zero, the configuration change is temporary and will revert back to the old configuration when the CSW Server restarts.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low							
CSW Message Type High (0x00)								CSW Message Type Low (0x39)							
Num Clients								Permanent Change Flag (0x00 or 0x01)							
Total Data Words High								Total Data Words Low							
Client 1 ID High								Client 1 ID Low							
Client 1 Subset ID High								Client 1 Subset ID 1 Low							
Client 1 Data Words High								Client 1 Data Words Low							
...															
Client N ID High								Client N ID Low							
Client N Subset ID High								Client N Subset ID 1 Low							
Client N Data Words High								Client N Data Words Low							
Maximum file path size is 90 bytes. CSW Message Size is 6*(N + 1) where N is the number of clients in the Health and Status configuration. EXPRESS Size = CSW Message Size + 10															

3.4.1.6 Request File CRC-32

This command requests that the CSW File Manager calculate the CRC-32 for a file. The File Manager will immediately send down a telemetry response that contains the CRC-32. The format of that telemetry packet can be found in Appendix A.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low							
CSW Message Type High (0x00)								CSW Message Type Low (0x3F)							
File Path Byte 1								File Path Byte 2							
File Path Byte 3								File Path Byte 4							
...															
File Path Byte N – 1								File Path Byte N							
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10															

3.4.1.7 Shutdown Windows

This command will shut down the Windows operating system and place the system in a state where the power may be switched off. If the CSW File Manager and/or User Interface are running, they will be terminated cleanly before the system goes down.

MSB														LSB
EXPRESS Version (0000)				EXPRESS Size										
EXPRESS Message Type (0xFFFF)														
Function Code Source (0x000F)														
Function Code Destination High							Function Code Destination Low							
CSW Destination High (0x00)							CSW Destination Low (0x03)							
CSW Message Size High (0x00)							CSW Message Size Low (0x02)							
CSW Message Type High (0x00)							CSW Message Type Low (0x95)							
EXPRESS Size = CSW Message Size + 10														

3.4.1.8 Start Batch Telemetry

This command begins telemetry of batch downlink files.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low (0x02)							
CSW Message Type High (0x00)								CSW Message Type Low (0x35)							
EXPRESS Size = CSW Message Size + 10															

3.4.1.9 Stop Batch Telemetry

This command stops telemetry of batch downlink files.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low (0x02)							
CSW Message Type High (0x00)								CSW Message Type Low (0x25)							
EXPRESS Size = CSW Message Size + 10															

3.4.1.10 Start Downlink List Telemetry

This command starts telemetry of the CSW batch downlink list file.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low (0x02)							
CSW Message Type High (0x00)								CSW Message Type Low (0x32)							
EXPRESS Size = CSW Message Size + 10															

3.4.1.11 Start Current Logs Telemetry

This command starts telemetry of all current CSW log files.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low (0x02)							
CSW Message Type High (0x00)								CSW Message Type Low (0x33)							
EXPRESS Size = CSW Message Size + 10															

3.4.1.12 Terminate File Manager

This command terminates execution of the CSW File Manager.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low (0x02)							
CSW Message Type High (0x00)								CSW Message Type Low (0x98)							
EXPRESS Size = CSW Message Size + 10															

3.4.1.13 Terminate Rack Interface

This command terminates execution of the CSW Server. The Server will ignore this command if in the middle of batch downlink. If the CSW File Manager is running, the Server will command the File Manager to terminate before halting its own execution.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x01)							
CSW Message Size High (0x00)								CSW Message Size Low (0x02)							
CSW Message Type High (0x00)								CSW Message Type Low (0x99)							
EXPRESS Size = CSW Message Size + 10															

3.4.1.14 Terminate User Interface

This command terminates execution of the CSW User Interface.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x01)							
CSW Message Size High (0x00)								CSW Message Size Low (0x02)							
CSW Message Type High (0x00)								CSW Message Type Low (0x97)							
EXPRESS Size = CSW Message Size + 10															

3.4.2 FILE COMMANDS

A wildcard may be used to force some file commands to act on multiple files. The wildcard can appear in the following ways in a file path:

C:\hrf_data*.txt
C:\hrf_data\!*.txt
C:\hrf_data\!*
C:\hrf_data*
C:\hrf_data*.*

Commands that accept wildcards have “**wildcard ready**” next to their names.

3.4.2.1 Add Files with File Location File

This command adds files to the batch downlink list that are successfully compared with the paths with wildcards contained in a file location file. The file path in the command is the path to the file location file.

MSB																LSB
EXPRESS Version (0000)				EXPRESS Size												
EXPRESS Message Type (0xFFFF)																
Function Code Source (0x000F)																
Function Code Destination High								Function Code Destination Low								
CSW Destination High (0x00)								CSW Destination Low (0x03)								
CSW Message Size High (0x00)								CSW Message Size Low								
CSW Message Type High (0x00)								CSW Message Type Low (0x81)								
File Path Byte 1								File Path Byte 2								
File Path Byte 3								File Path Byte 4								
...																
File Path Byte N – 1								File Path Byte N								
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10																

3.4.2.2 Add File (**wildcard ready**)

This command will add a file or files (using a wildcard) to the downlink list.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low							
CSW Message Type High (0x00)								CSW Message Type Low (0x2B)							
File Path Byte 1								File Path Byte 2							
File Path Byte 3								File Path Byte 4							
...															
File Path Byte N – 1								File Path Byte N							
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10															

3.4.2.3 Add File List

This command adds files to the downlink list that are listed on individual lines in a text file. The file path in the command is the path to this text file.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low							
CSW Message Type High (0x00)								CSW Message Type Low (0x6B)							
File Path Byte 1								File Path Byte 2							
File Path Byte 3								File Path Byte 4							
...															
File Path Byte N – 1								File Path Byte N							
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10															

3.4.2.4 Confirm File (**wildcard ready**)

This command removes a file or files (using a wildcard) with “sent” status from the downlink list. The file is then logged as confirmed in the downlink log file.

MSB																LSB
EXPRESS Version (0000)				EXPRESS Size												
EXPRESS Message Type (0xFFFF)																
Function Code Source (0x000F)																
Function Code Destination High								Function Code Destination Low								
CSW Destination High (0x00)								CSW Destination Low (0x03)								
CSW Message Size High (0x00)								CSW Message Size Low								
CSW Message Type High (0x00)								CSW Message Type Low (0x31)								
File Path Byte 1								File Path Byte 2								
File Path Byte 3								File Path Byte 4								
...																
File Path Byte N – 1								File Path Byte N								
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10																

3.4.2.5 Confirm File List

This command removes files from the downlink list with “sent” status that are listed on individual lines in a text file. The file path in the command is the path to this text file.

MSB																LSB
EXPRESS Version (0000)				EXPRESS Size												
EXPRESS Message Type (0xFFFF)																
Function Code Source (0x000F)																
Function Code Destination High								Function Code Destination Low								
CSW Destination High (0x00)								CSW Destination Low (0x03)								
CSW Message Size High (0x00)								CSW Message Size Low								
CSW Message Type High (0x00)								CSW Message Type Low (0x71)								
File Path Byte 1								File Path Byte 2								
File Path Byte 3								File Path Byte 4								
...																
File Path Byte N – 1								File Path Byte N								
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10																

3.4.2.6 Delete File (**wildcard ready**)

This command deletes a file or files (using a wildcard) from the hard drive of the destination computer. This command does not apply to the batch downlink file list.

MSB																LSB
EXPRESS Version (0000)				EXPRESS Size												
EXPRESS Message Type (0xFFFF)																
Function Code Source (0x000F)																
Function Code Destination High								Function Code Destination Low								
CSW Destination High (0x00)								CSW Destination Low (0x03)								
CSW Message Size High (0x00)								CSW Message Size Low								
CSW Message Type High (0x00)								CSW Message Type Low (0x11)								
File Path Byte 1								File Path Byte 2								
File Path Byte 3								File Path Byte 4								
...																
File Path Byte N – 1								File Path Byte N								
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10																

3.4.2.7 Delete File List

This command deletes files that are listed on individual lines in a text file. The file path in the command is the path to this text file.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low							
CSW Message Type High (0x00)								CSW Message Type Low (0x51)							
File Path Byte 1								File Path Byte 2							
File Path Byte 3								File Path Byte 4							
...															
File Path Byte N – 1								File Path Byte N							
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10															

3.4.2.8 Enable File (**wildcard ready**)

This command will enable a file or files (using a wildcard) that have “inhibited” status on the downlink list.

MSB																LSB
EXPRESS Version (0000)				EXPRESS Size												
EXPRESS Message Type (0xFFFF)																
Function Code Source (0x000F)																
Function Code Destination High								Function Code Destination Low								
CSW Destination High (0x00)								CSW Destination Low (0x03)								
CSW Message Size High (0x00)								CSW Message Size Low								
CSW Message Type High (0x00)								CSW Message Type Low (0x23)								
File Path Byte 1								File Path Byte 2								
File Path Byte 3								File Path Byte 4								
...																
File Path Byte N – 1								File Path Byte N								
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10																

3.4.2.9 Enable File List

This command enables files on the downlink list with “inhibited” status that are listed on individual lines in a text file. The file path in the command is the path to this text file.

MSB																LSB
EXPRESS Version (0000)				EXPRESS Size												
EXPRESS Message Type (0xFFFF)																
Function Code Source (0x000F)																
Function Code Destination High								Function Code Destination Low								
CSW Destination High (0x00)								CSW Destination Low (0x03)								
CSW Message Size High (0x00)								CSW Message Size Low								
CSW Message Type High (0x00)								CSW Message Type Low (0x63)								
File Path Byte 1								File Path Byte 2								
File Path Byte 3								File Path Byte 4								
...																
File Path Byte N – 1								File Path Byte N								
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10																

3.4.2.10 Inhibit File (**wildcard ready**)

This command will inhibit a file or files (using a wildcard) that have “enabled” status on the downlink list.

MSB																LSB
EXPRESS Version (0000)				EXPRESS Size												
EXPRESS Message Type (0xFFFF)																
Function Code Source (0x000F)																
Function Code Destination High								Function Code Destination Low								
CSW Destination High (0x00)								CSW Destination Low (0x03)								
CSW Message Size High (0x00)								CSW Message Size Low								
CSW Message Type High (0x00)								CSW Message Type Low (0x29)								
File Path Byte 1								File Path Byte 2								
File Path Byte 3								File Path Byte 4								
...																
File Path Byte N – 1								File Path Byte N								
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10																

3.4.2.11 Inhibit File List

This command inhibits files on the downlink list with “enabled” status that are listed on individual lines in a text file. The file path in the command is the path to this text file.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low							
CSW Message Type High (0x00)								CSW Message Type Low (0x69)							
File Path Byte 1								File Path Byte 2							
File Path Byte 3								File Path Byte 4							
...															
File Path Byte N – 1								File Path Byte N							
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10															

3.4.2.12 Remove File (**wildcard ready**)

This command will remove a file or files (using a wildcard) from the downlink list regardless of the file status.

MSB																LSB
EXPRESS Version (0000)				EXPRESS Size												
EXPRESS Message Type (0xFFFF)																
Function Code Source (0x000F)																
Function Code Destination High								Function Code Destination Low								
CSW Destination High (0x00)								CSW Destination Low (0x03)								
CSW Message Size High (0x00)								CSW Message Size Low								
CSW Message Type High (0x00)								CSW Message Type Low (0x2F)								
File Path Byte 1								File Path Byte 2								
File Path Byte 3								File Path Byte 4								
...																
File Path Byte N – 1								File Path Byte N								
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10																

3.4.2.13 Remove File List

This command deletes files from the downlink list regardless of status that are listed on individual lines in a text file. The file path in the command is the path to this text file.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low							
CSW Message Type High (0x00)								CSW Message Type Low (0x6F)							
File Path Byte 1								File Path Byte 2							
File Path Byte 3								File Path Byte 4							
...															
File Path Byte N – 1								File Path Byte N							
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10															

3.4.2.14 Resend File/Resend File Block (**wildcard ready**)

This command enables a file or files (using a wildcard) with “sent” status on the batch downlink list. For resending an individual file segment, append the segment number followed by a dash at the beginning of the file path. For resending a contiguous set of file segments, append the start segment followed by ‘>’, the last segment desired, and finally a dash. Below are examples of resending file segments:

2-c:\hrf_data\file.dat (resends segment 2 of file1.dat)
 2>5-c:\hrf_data\file.dat (resends segments 2 to 5 of file1.dat)

Also, file segment commands can be queued up. For instance, use the following paths in resend commands prior to starting batch downlink of files:

1-c:\hrf_data\file.dat
 3-c:\hrf_data\file.dat
 7-c:\hrf_data\file.dat

After these three resend commands are received, the next telemetry session will transmit segments 1, 3, and 7 to the ground.

NOTE: Wildcards cannot be used for resending file segments.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low							
CSW Message Type High (0x00)								CSW Message Type Low (0x21)							
File Path Byte 1								File Path Byte 2							
File Path Byte 3								File Path Byte 4							
...															
File Path Byte N – 1								File Path Byte N							
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10															

3.4.2.15 Resend File List

This command enables files on the downlink list with “sent” status that are listed on individual lines in a text file. The file path in the command is the path to this text file.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low							
CSW Message Type High (0x00)								CSW Message Type Low (0x61)							
File Path Byte 1								File Path Byte 2							
File Path Byte 3								File Path Byte 4							
...															
File Path Byte N – 1								File Path Byte N							
Maximum file path size is 90 bytes. CSW Message Size is N + 2. EXPRESS Size = CSW Message Size + 10															

3.4.2.16 Common Software File Transfer Command

This command tells the CSW file manager to initiate a file transfer. The local file path must be 60 characters. Any unused characters should be set to null.

MSB															LSB
EXPRESS Version (0000)				EXPRESS Size											
EXPRESS Message Type (0xFFFF)															
Function Code Source (0x000F)															
Function Code Destination High								Function Code Destination Low							
CSW Destination High (0x00)								CSW Destination Low (0x03)							
CSW Message Size High (0x00)								CSW Message Size Low (0x54)							
CSW Message Type High (0x00)								CSW Message Type Low (0xF1)							
EMU Filename Character 1								EMU Filename Character 2							
EMU Filename Character 3								EMU Filename Character 4							
EMU Filename Character 5								EMU Filename Character 6							
EMU Filename Character 7								EMU Filename Character 8							
EMU Filename Character 9								EMU Filename Character 10							
EMU Filename Character 11								EMU Filename Character 12							
Request Type Transfer ID								Block Restart High							
Block Restart Low								Transfer Type (41h – PLD EMU, 42h PEP EMU)							
PEP File ID High								PEP File ID Low							
Local Filename Character 1								Local Filename Character 2							
Local Filename Character 3								Local Filename Character 4							
. . .															
Local Filename Character 59								Local Filename Character 60							
EMU Directory Character 1								EMU Directory Character 2							
EMU Directory Character 3								EMU Directory Character 4							

3.4.2.17 Request Type Transfer IDs

Description	Value
Start PEP to EMU Transfer	00h
Stop PEP to EMU Transfer	40h
Restart PEP to EMU Transfer	80h
Start EMU to PEP Transfer	10h
Stop EMU to PEP Transfer	50h
Restart EMU to PEP Transfer	90h
Start EMU to Ground Transfer	20h
Stop EMU to Ground Transfer	60h
Restart EMU to Ground Transfer	A0h
Start PLD to EMU Transfer	00h
Stop PLD to EMU Transfer	40h
Restart PLD to EMU Transfer	80h
Start EMU to PLD Transfer	10h
Stop EMU to PLD Transfer	50h
Restart EMU to PLD Transfer	90h

3.5 ACCESSING THE COMMON SOFTWARE DYNAMIC LINK LIBRARY

The CSW DLL has been designed for client software to use for communicating with the CSW. The following section describes each function of the DLL, including parameters used in function calls and function return values.

3.5.1 Common Software Dynamic Link Library Function Prototypes and Descriptions

Files Needed: csw_dll.dll
csw_dll.lib
csw_dll.h

3.5.1.1 Connect Command

Call one of the following functions to connect to the CSW RI Server:

```
int STDCALL Connect_To_Server(BYTE client_id);
int STDCALL Connect_To_Server_2(WORD client_id);
```

NOTE: The function “Connect_To_Server_2” must be used.
“Connect_To_Server” exists for legacy software applications only.

Each of these function calls will connect a client to the CSW RI Server. Two named pipe connections are established. One connection is for reading data and the other is for writing data. The client_id parameter tells the CSW RI Server which client is connecting. The client_id will be assigned by the HRF Software Integration Team. Only one connect call may be made by a single process.

A return value of 1 means that a connection was successfully established, and a value of 0 means connection was unsuccessful.

3.5.1.2 Request/Send Commands

Call the following functions to write to the CSW RI Server:

int STDCALL Request_Rack_Time();

This function requests rack time from the CSW RI Server. The RIC will return the rack time to the CSW RI Server, and the time will be passed on to the client. The client will continue to receive time every time a request goes out to the RIC. The function returns 0 if successful otherwise a negative number. Rack time will be returned in the form found in Section 3.3.3.

int STDCALL Request_Ancillary_Data(BYTE data_set_id, BYTE request_id);

This function requests configuration of an ancillary data set. The first parameter is the data set id and the second argument is the request id. The value for the data_set_id may be 0x01 to 0x64 for ancillary, 0x80 to 0xE3 for broadcast frames, and 0xE4 for all broadcast frames. The request id may be as follows:

0x00, One shot request	ONE_SHOT to return once
0x01, Continuous request	ADD_ENTRY to start sending
0x03, Delete request	DELETE_ENTRY to stop sending

The function returns 2 if successful or 0 if the function was unsuccessful. Ancillary data packet format is in Sections 3.3.1 and 3.3.2.

int STDCALL Send_File_Location_File(char file_path);*

This function sends a data file location message. This should be called when new data has been recorded and needs to be added to the batch downlink list. Sending this message will put the new data on the batch downlink list. The parameter is the path to a file location file which lists all the paths where data may be found.

The function returns the path length including null characters if successful or 0 if the function was unsuccessful.

int STDCALL Batch_Downlink_Control(int Control_id);

This function turns on and off the batch downlink capability of the HRF CSW. The parameter value must either be 101 for enabling batch downlink or 102 for disabling batch downlink. Disabling batch downlink will allow a client to send real time telemetry data to the RIC without interference from batch downlink. The function returns 0 if successful otherwise a negative number.

int STDCALL Update_Health_and_Status(BYTE health_buffer, int buffer_size);*

This function sends health and status data to the CSW RI Server. The first parameter is a pointer to the health and status data, and the second parameter is the data size in bytes. Health and status data must be configured for a client. If a client is not configured for health and status, the CSW RI Server will discard Health and Status

data packets. The function returns the health buffer size if successful or 0 if the function was unsuccessful. The buffer size must match the health and status size assigned to the client. Excess buffer data will be discarded. The first WORD (2 bytes) of health_buffer must be a data cycle counter. The initial value of this counter must be zero. The counter must increment by one for every new call to this function. The counter should return to zero after reaching 65,535.

int STDCALL Send_PEP_Bundle_Request(int request_type ,int bundle_parameter);

This function sends a Payload Executive Processor (PEP) Bundle Request to the CSW RI Server which is passed on to the PEP through the RIC. The first parameter is the request type, and the second parameter is the bundle parameter. The value for the bundle parameter will vary depending on the request. The request_type can be INSTALL_BUNDLE_REQ (21 decimal), HALT_BUNDLE_REQ (22 decimal), or REMOVE_BUNDLE_REQ (23 decimal). The function returns 4 if successful or 0 if the function was unsuccessful.

int STDCALL Send_PEP_Execution_Request(int request_type,int payload_seq_id);

This function sends a PEP Execution Request to the CSW RI Server which is passed on to the PEP through the RIC. The first parameter is the request type, and the second parameter is the payload sequence id.

The payload sequence id will vary for different requests. The request_type can be START_EXEC (18 decimal), STOP_EXEC (19 decimal), or RESUME_EXEC (20 decimal). The function returns 4 if successful or 0 if the function was unsuccessful.

int STDCALL Send_Data(void data_buffer,int data_size);*

This function writes data packets to the CSW RI Server and returns the number of bytes written to the server. The first parameter is a pointer to the data buffer, and the second parameter is the size of the data buffer. The size may be no greater than 4090 bytes. The function returns immediately. The client software is responsible for making sure that all data is written. The first word in the data buffer must be the two LSDS bytes described in Section 3.2.12. Set the highest bit of the first LSDS byte when the data is in LSDS format. The remaining 15 bits are used for the client ID. This is the same ID passed in the Connect_To_Server function call.

int STDCALL Read_Server(void data_buffer, int data_size);*

int STDCALL Read_Server_Wait(void data_buffer, int data_size);*

These functions will return the number of bytes read from the server and data from the named pipe buffer. The following items may be read from the CSW RI Server:

- Rack Time
- Ancillary Data Sets
- Broadcast Ancillary Data
- Generic Client Messages

The Read_Server_Wait function will wait for data on the pipe, but the Read_Server function will return immediately.

Void Disconnect_Server();

This function disconnects the client software from the CSW RI Server. The function should be called by the client software before program termination.

int STDCALL Send_Client_Message(void message, int message_size, BYTE destination)*

This function sends a client specific message to the server for transfer to another client. The destination should be the client id of the client receiving the message. The return value is the message size.

3.6 COMMON SOFTWARE HEALTH AND STATUS FORMAT

55AA	Sync Word
XXXX	Byte Count
XXXX	Message Type
0100	Source Function Code
0001	Destination Function Code
0000	Caution and Warning Data Word
0000	Cycle Counter (increases with each packet and cycles back to zero after FFFF)
0100	Common Software Source Word (tells whether the data is from Laptop or Workstation)
000X	Common Software Health Word - 0009 Server Running (batch mode on)
	000B Server/User Interface Running (batch mode on)
	000F Server/User Interface/File Manager Running (batch mode on)
	000D Server/File Manager Running (batch mode on)
0000	CSW Commands Received Word
0000	# Enabled Downlink Files Word (will be FFFF if the File Manager is not connected to the Server)
0046	10 Words that show which experiment software clients are connected to the CSW Server
0000	Each word is 0000 if a client is not connected. In this case only client id 0046 is connected
0000	The number of words used for connected clients may be changed in the Rack Interface (Server)
0000	configuration file, "CFRIconfig.txt". Ten words were used in this case as an example.
0000	
0000	
0000	
0000	
0000	
XXXX	Client 1 subset id
XXXX	Client caution and warning word
0000	Client cycle counter (increases with each packet received from the client and cycles back to zero)
XXXX	Client Health Data
XXXX	
XXXX	
XXXX	
XXXX	Client 2 subset id
XXXX	Client caution and warning word
0000	Client cycle counter (increases with each packet received from the client and cycles back to zero)
XXXX	Client Health Data
XXXX	
XXXX	
XXXX	

3.7 BATCH DOWNLINK

3.7.1 File Naming Convention Restrictions - Using Unique File Names

One of two naming conventions must be used for files created for batch downlink. One convention involves using a unique file name for each file created. The other convention involves using the same file name every time information such as experiment data is written to file. Use of this secondary convention is described in the next section. The remainder of this section discusses using unique file names.

There are two basic ways to create a unique file name. The preferred method is to include a date and time stamp in the file name. The other method is to include a sequence number in the file name. If the latter approach is used, the sequence number shall not be dependent on the existence of any previously created file. A separate file located outside of the downlink directory should be used to maintain the sequence count.

The file names below are examples of both these methods.

Exp_data10142003065830.dat (with date and time stamp)

Exp_data0001.dat (with sequence number)

The following tables show the format and example contents of the file location text file for this naming convention.

Line	Section (separated by tabs)	Represents	Description
1	N/A	File Location File Method	The type of file location file. For example, "SAME FILE OFF."
2+	1	Data Type Name	The name of the type of data files. For example, "HRFdata".
2+	2	Experiment ID	Identification for the data.
2+	3	Priority Level	The priority of the data files: "1", "2", or "3".
2+	4	Recursion Flag	Set the recursion flag to "0" to gather the files from the folder specified only. Set the recursion flag to "1" to gather files from the folder and all of its subfolders.
2+	5	Data Path	The complete path of the data files, including wildcards. For example, "d:\HRF_data*.dat"

FileLocation.txt Example with "SAME FILE OFF"

SAME FILE OFF				
hrfdata	45	2	1	d:\hrfdata*.dat
ws diagnostics	55	1	0	c:\workstation\test*.log

3.7.2 Using Same File Name

In some cases, an experiment or instrument may save their data to one file and each data session just appends to that data file. In this event, the file location file should use the “SAME FILE ON” method.

When using the “SAME FILE ON” method, a number file is used to track how many data sessions has occurred with the experiment/instrument. This setup is needed so the File Manager can rename and move the data session file to appear as a nominal, separate data file. Then the file can be added to the downlink list and telemetry can occur as normal.

Also note then when using the “SAME FILE ON” method, the recursion flag must be set to “0”, the data folder path must have a subfolder called “same_file_name_data”. In the example below, “d:\hrf_data\evarm\” is the data folder path and so, the path “d:\hrf_data\evarm\same_file_name_data\” must exist.

The following tables show the format and example contents of the file location text file for the naming convention.

Line	Section (separated by tabs)	Represents	Description
1	N/A	File Location File Method	The type of file location file. For example, “SAME FILE ON”
2	N/A	Number File	Path of file designated to track the number of data sessions.
3+	1	Data Type Name	The name of the type of data files. For example, “HRFdata”.
3+	2	Experiment ID	Identification for the data.
3+	3	Priority Level	The priority of the data files: “1”, “2”, or “3”.
3+	4	Recursion Flag	The recursion flag must be set to “0”.
3+	5	Data Path	The complete path of the data files, including wildcards. For example, “d:\HRF_data*.dat”.

FileLocation.txt Example with “SAME FILE ON”

SAME FILE ON			
d:\hrf_prog\evarm\evarm_number.txt			
evarm files	33	1	0
			d:\hrf_data\evarm*.log

NOTE: In this example, the “d:\hrf_prog\evarm\evarm_number.txt” file must exist and be set to a starting number, e.g., “1”.

3.7.3 Downlink File ID File

The downlink file ID file (downlink_file_id.txt) is monitored by the CSW FM and UI only. No payload client software should modify this file.

The file contains the next unique file ID number for the FM or UI to use when adding a file to the downlink list.

3.7.4 Downlink List

The downlink list files (downlinklist.txt, csw_archive_list.txt, csw_largefilesegment_list.txt) are monitored by the CSW FM and UI only. No payload client software should modify this file.

The format of a line in a downlink list file is as follows with the fields separated by tabs.

File ID	Experiment ID	Folder	File	Size	Priority	Status	Modified Time	Sent/Inhibited Time
---------	---------------	--------	------	------	----------	--------	---------------	---------------------

*Note: Priority - 1, 2, or 3; highest to lowest priority of files

*Note: Status - sent, inhibited, or enabled

*Note: Modified Time - time stamp when the file was last modified, ddMmmyy hh:mm:ss

*Note: Sent/Inhibited Time - time stamp when the file is sent or inhibited, in format ddMmmyy hh:mm:ss

The following is an example of a downlink list file.

1	8	c:\data\e308data\	re2379.dat	79085	2	enabled	02Apr99 12:39:03	xxxxxxxxxxxxxxxxxxxx
2	10	c:\exper\yphb\	o0752.sw1	5241	1	sent	01Apr99 03:59:24	03Apr99 11:34:28

APPENDIX A

COMMON SOFTWARE TELEMETRY HEADER DEFINITIONS

EXPRESS packet format (header 1)

Version is the high 3 bits of the Version-Byte count word

Byte count is the size of the entire message not counting the version-byte count word.

MSB	LSB
Version – Byte Count	Version Byte Count
Message Type	Message Type
Function Code Source	Function Code Source
Function Code Destination	Function Code Destination

EXPRESS packet format (header 2)

Telemetry Data Type

0x100 Batch Telemetry

0x101 Real Time Telemetry

0x102 CRC packet

MSB	LSB
Telemetry Data Type	Telemetry Data Type
Sequence Count	Sequence Count
Coarse Time (High Word)	Coarse Time
Coarse Time (Low Word)	Coarse Time

Real Time Packet Format (TSC header)

The highest bit in the Format-Experiment ID word is set if the data is in LSDS format.

The other 15 bits represent the experiment ID.

MSB	LSB
Current Packet	Total Packets
Current Packet Byte Count	Current Packet Byte Count
Total Byte Count	Total Byte Count
Format-Experiment ID	Format-Experiment ID

Batch Downlink Header 1

Current segment and total segment words are zero if the file is not segmented.

MSB	LSB
Block ID	Header Version
Experiment-Instrument ID	Experiment-Instrument ID
Downlink File ID	Downlink File ID
Packet Data Size	Packet Data Size
Current Packet Number	Current Packet Number
Total File Packets	Total File Packets
Current Segment Number	Current Segment Number
Total File Segments	Total File Segments
Cyclic Redundancy Checksum	Cyclic Redundancy Checksum
Cyclic Redundancy Checksum	Cyclic Redundancy Checksum
Filename Byte 1	Filename Byte 2
Filename Byte 3	Filename Byte 4
....
Filename Byte 127	Filename Byte 128

Batch Downlink Header 2

MSB	LSB
Block ID	Header Version
Experiment-Instrument ID	Experiment-Instrument ID
Downlink File ID	Downlink File ID
Packet Data Size	Packet Data Size
Current Packet Number	Current Packet Number
Total File Packets	Total File Packets
Current Segment Number	Current Segment Number
Total File Segments	Total File Segments

Cyclic Redundancy Checksum (CRC) packet

This packet is sent to the ground after a ground command is sent to the CSW requesting a calculation of a CRC for a file.

MSB	LSB
Cyclic Redundancy Checksum	Cyclic Redundancy Checksum
Cyclic Redundancy Checksum	Cyclic Redundancy Checksum
Filename Byte 1	Filename Byte 2
Filename Byte 3	Filename Byte 4
....
Filename Byte 93	Filename Byte 94

APPENDIX B

CSW SPECIFIC FILE IDS USED IN TELEMETRY HEADERS

File Type	File ID
Archived Logs	65000
Current Logs	65000
Archive Downlink List	65533
File Segment Downlink List	65534
Downlink List	65535

Max File ID for user data files is 64,999.

DISTRIBUTION LIST
FOR
LS-71062-8D

NASA/JSC

EB/E. Bauer

NT3/GFE Assurance Branch

NT52/Frances Simmons

SM/C. Haven

LOCKHEED MARTIN

B07/M. Scott

B08/W. Tinch

C42/Science Payloads Library

C42/M. Klee

S03/D. Babic

S03/W. Cohen

S03/A. Nordheim

S03/M. Romell

S3362/TDI Center/Bldg. 36 (5)