SECTION 3 - DATA FORMAT SPECIFICATION

3.1 GENERAL DATA FORMAT SPECIFICATION

3.1.1 ECS MESSAGES

This section describes various messages exchanged between the ECS and the instrumenters as data streams over sockets.

3.1.1.1 General Format of an ECS Message.

A message exchanged over the ECS/IWS interface consists of a 4-byte standard header followed by a data field of variable length as illustrated in Table 3.1.

Table 3.1. General ECS Message Format.

Field	Bytes	Description
Type/Message ID	2	Standard
Length	2	Header
User data dependent on the message type.	variable	Data field

The standard header is comprised of a 2-byte "type field" followed by a 2-byte length field.

(1) The 2-byte type field is defined as:
first byte is
X'01' for messages to control a communication session
X'02' for messages related to telemetry distribution
X'03' for messages related to telecommanding
X'04' for informational messages
second byte identifies the messages within these 4 categories.

(2) The 2-byte length field contains the length in bytes of the message that follows, excluding the 4-byte standard header.

The data field is specific to each type of message and is of variable length.

3.1.1.2 ECS Messages Description.

As much as possible, the ECS/instrumenters functional protocol was kept similar to the protocol implemented between CCS and the Experiment Ground Support Equipment (EGSE) in the AIV environment. Modifications were necessary to apply the AIV protocol to the operational environment, mainly to support the commanding functions. Also, a bi-directional Informational message has been added. Table 3.2. defines the messages used within the EOF.

Table 3.2. ECS Messages.

<table>
<tr><th rowspan="2">Message Name</th><th rowspan="2">Direction</th><th colspan="2">Standard Header</th><th colspan="2">Data Field</th></tr>
<tr><th>Type</th><th>Length</th><th>Bytes</th><th>Description</th></tr>
<tr><td>Session Init</td><td>ECS to IWS</td><td>X'0101'</td><td>X'0004'</td><td>Int 4</td><td>Endian check block data</td></tr>
<tr><td>Session Init Response</td><td>IWS to ECS</td><td>X'0102'</td><td>X'0015'</td><td>ASCII 16
Int 1
Int 4</td><td>ORIG_ID
Endian check result
Endian check block data</td></tr>
<tr><td>Session End</td><td>ECS to IWS</td><td>X'0103'</td><td>X'0001'</td><td>Int 1</td><td>Reason code</td></tr>
<tr><td>NRT Command</td><td>IWS to ECS</td><td>X'0301'</td><td>var</td><td>Int 2
ASCII 6
ASCII var</td><td>Request ID
Instrument name
Command data</td></tr>
<tr><td>Response to NRT Command</td><td>ECS to IWS</td><td>X'0302'</td><td>var</td><td>Int 2
ASCII 6
Int 2
Int 2
ASCII var</td><td>Request ID
Instrument name
Response code
Reason code
Response to command (text)</td></tr>
<tr><td>NRT Command Authority Request</td><td>IWS to ECS</td><td>X'0303'</td><td>X'000A'</td><td>Int 2
ASCII 6
Int 2</td><td>Request ID
Instrument name
Request code</td></tr>
<tr><td>NRT Authority Status</td><td>ECS to IWS</td><td>X'0304'</td><td>var</td><td>Int 2
ASCII 6
Int 2
ASCII var</td><td>Request ID
Instrument name
Status code
Status description (text)</td></tr>
<tr><td>Remote Command Request</td><td>IWS to ECS</td><td>X'0305'</td><td>var</td><td>Int 2
ASCII 6
ASCII 20
ASCII var</td><td>Request ID
Instrument name
PCS name
Instructions/Comments</td></tr>
<tr><td>Remote Procedure Request</td><td>IWS to ECS</td><td>X'0306'</td><td>var</td><td>Int 2
ASCII 6
ASCII 20
ASCII var</td><td>Request ID
Instrument name
STOL Procedure name
Instructions/Comments</td></tr>
<tr><td>TM Packet Distribution Request</td><td>IWS to ECS</td><td>X'0201'</td><td>var:
X'0005'
or
X'0006'</td><td>Int 1
Int 2
Int 2
Int 1</td><td>Spacecraft ID
APID
Request ID
Optional: Q&A capsule Flag</td></tr>
<tr><td>TM Packet Distribution Response</td><td>ECS to IWS</td><td>X'0202'</td><td>X'0004'</td><td>Int 2
Int 1
Int 1</td><td>Request ID corresponding to
Request Response Code
Reason Code</td></tr>
<tr><td>Start of TM Packet Distribution</td><td>ECS to IWS</td><td>X'0203'</td><td>X'0005'</td><td>Int 1
Int 2
Int 2</td><td>Spacecraft ID
APID
Request ID</td></tr>
<tr><td>Telemetry Packet</td><td>ECS to IWS</td><td>X'0204'</td><td>var</td><td>Int 2
Binary
6 bytes</td><td>Request ID
TM source packet
Q&A capsule</td></tr>
<tr><td>Interrupt TM packet transfer</td><td>IWS to ECS</td><td>X'0205'</td><td>X'0002'</td><td>Int 2</td><td>Request ID</td></tr>
<tr><td>End of TM Packet Transfer</td><td>ECS to IWS</td><td>X'0206'</td><td>X'0003'</td><td>Int 2
Int 1</td><td>Request ID
Status code</td></tr>
<tr><td>Informational Message</td><td>ECS to IWS
IWS to ECS</td><td>X'0400'</td><td>var</td><td>ASCII var</td><td>Free form text</td></tr>
</table>

3.1.2 ECS FILES

Files exchanged between the ECS and the instrumenters have a standard transfer format consisting of a file header followed by a file body. The file header uses keywords to provide information about the file and is in the general format "KEYWORD = value". Each Keyword is followed by '=' and each record is ended by a New Line (X'0A'). The file body contains ASCII character data that is specific to each type of data contained in the file. See Appendix B for examples of file formats.

3.1.2.1 System Directory Organization for Files

The ECS files are organized among various system directories, one directory being provided for each type of file, and sub-directories being provided as needed in each case. The ECS main system directories for the data exchanged with the instrumenters are listed and described in Appendix B.

3.1.2.2 File Naming Conventions

Each file is referenced by a unique name representative of the type of data it contains. Several file naming schemes are necessary in the EOF to better describe the data contained within each file or to satisfy already existing naming conventions with other ECS external interfaces. The specific conventions are described in Appendix A.

3.1.2.3 File Header Format

All file headers described in this document have the same general format: a series of records of ASCII characters, each record being of the form "KEYWORD = value", the last character being a New Line (NL).

3.1.3 TIME FIELD FORMAT

All time fields, unless specified otherwise in individual cases, will contain both the date and time in a single 19 character format as follows:

YYYY/MM/DD HH:MM:SS
where:
YYYY/MM/DD is the date:

YYYY	represents the year (for example 1995)
MM	represents the month (01 for January to 12 for December)
DD	represents the day of the month (01 to 31)

HH:MM:SS is the time:

HH	represents the hours (00 to 23)
MM	represents the minutes (00 to 59)
SS	represents the seconds (00 to 59)

The date and time fields are separated by an ASCII blank.

Except where specifically mentioned otherwise, all times mentioned in this document are in reference to GMT.

3.1.4 INSTRUMENT NAME FIELD FORMAT

Unless specified otherwise, all fields specifying the Instrument name are 6 ASCII characters in length, and must be one of the following, left-justified and padded with ASCII blanks if necessary:

CDS
CELIAS
CEPAC
EIT
GOLF
LASCO
MDI
SUMER
SWAN
UVCS
VIRGO

3.2 COMMANDING DATA SPECIFICATION

3.2.1 OBDH BLOCK COMMAND

The routing and formatting of the command data from the instrumenters to the spacecraft is illustrated in figure 3.1. The basic unit of command input provided by an instrumenter is an OBDH block command as illustrated in figure 3.2. Each OBDH block command consists of a series of 16-bit words of data as follows:

- one word representing the block header
- up to 30 words of data
- one word containing the checksum of the preceding words (header and data).

The block header is 16 bits long and of the form:

XXYYYYZZZZZLLLLL

where:

XX	bits 0 and 1	Reserved				
YYYY	bits 2 through 5	Destination address				
		0100	for CDS	0101	for CELIAS	
		0110	for CEPAC	0111	for EIT	
		1000	for GOLF	1001	for LASCO	
		1010	for MDI	1011	for SUMER	
		1100	for SWAN	1101	for UVCS	
		1110	for VIRGO			
ZZZZZ	bits 6 through 10	Command identifier				
LLLLL	bits 11 through 15	Block length (1 to 31): number of 16-bit words in the block, excluding the checksum.				

3.2.2 INSTRUMENT COMMAND INPUT

The method and format for submitting instrument commands described in this section are based on the following understanding:

Figure 3.1. SOHO Command Formatting.

REPRESENTATION OF AN OBDH BLOCK COMMAND

BINARY FORMAT FORMAT		OBDH COMMAND BLOCK		MNEMONIC
4 Hex Characters	------->	Header		
4 Hex Characters	------->	Data		Mnemonic
4 Hex Characters	------->	Data		+
.		.	<-------	Representations of words with user-specified values
.		.		(Variable number as
.		.		defined in PDB)
.		.		
4 Hex Characters	------->	Data		
4 Hex Characters	------->	Data		
4 Hex Characters	------->	Checksum		

Mnemonic representation:

```
MNEMO1,0x77AF;      /* one parameter in hexadecimal */
MNEMO1,123;         /* one parameter in decimal */
MNEMO1,0777;        /* one parameter in octal */
```

Binary representation:

```
BINARY 0x1203,0x2401,0x77AF,0xADB3;   /* Only hexadecimal allowed */
```

Figure 3.2. Instrumenters Command Specification.

- The instrumenters can only use OBDH block commands when commanding their instruments through the ECS. However this does not apply to VIRGO.

- A mnemonic for an instrument command uniquely defines a single OBDH block command.

- For each such mnemonic, the PDB provides the binary equivalent of the OBDH block command header (including destination address, command identifier and number of data words associated with this command).

- For each mnemonic, the PDB also provides the binary equivalent of each 16-bit data word in the OBDH block command. Some data words may contain variable bits that the user must specify. These parameters specifications can be done using ASCII characters representing their decimal, octal or hexadecimal value.

The content of an OBDH block command may be represented in one of two general formats: binary format and mnemonic format as described in the following sections. Note that a procedural language such as STOL or ELISA is not used in the EOF.

3.2.2.1 Binary Format

In the binary format, the commanding data consist of a series of ASCII hexadecimal representations of all the 16-bit binary words contained in one OBDH block command including header and checksum:

1) The keyword "BINARY" indicates the start of an OBDH block command. It must be in upper case.

2) The content of an OBDH block command is represented as a series of up to 32 4-character hexadecimal words:
   - The first hexadecimal word represents the 16-bit OBDH block header.
   - The following words represent up to 30 16-bit data words.
   - The last word represents the 16-bit checksum of the header and data words.

3) Hexadecimal words are separated by a comma.

4) The end of the OBDH block command is indicated by a semicolon.

5) Comments in the form "/* text */" are allowed after the semicolon.

Example: BINARY 0x1203,0x2401,0x77AF,0xADB3; /* optional comment */

3.2.2.2 Mnemonic Format

Each mnemonic specification defines a single OBDH block command. It consists of a mnemonic optionally followed by ASCII representations of the data words which have been defined as user-specified in the PDB.

1) The mnemonic is followed by up to 30 user-specified parameters. The mnemonic must be defined in the telecommand description files of the PDB and the number of user-specified

parameters must correspond to the number of user-defined values defined in the PDB. All command mnemonics must be specified in upper-case characters to conform to the PDB definition.

The user-specified 16-bit words are defined using ASCII character representations:

hexadecimal:	0xhhhh	'0x' followed by hexadecimal digits
octal:	0oooooo	'0' followed by octal digits without leading zeros
decimal:	ddddd	decimal digits without leading zeros

2) The representation of data words are separated by a comma.

3) The end of the command specification (mnemonic and optional data words) is indicated by a semicolon.

4) Comments in the form "/* text */" are allowed after the semicolon.

Examples:

```
CBEFILI,0x77AF;     /* in this command, only one word has user-defined values */
CBEFILI,073657;     /* same command using octal representation */
CBEFILI,30639; /* same command using decimal representation */
```

Note: The checksum is not provided by the instrumenters in the case of the mnemonic format. It will be calculated by the CMS.

3.2.3. NEAR-REAL-TIME COMMANDING DATA SPECIFICATION

The general format of the messages exchanged over the interface is defined in section 3.1.1. These messages all contain a 4-byte standard header followed by a data field. This section defines the specific functions and the content of the data field for the messages exchanged for near-real-time commanding.

3.2.3.1 Session-Init (ECS to IWS). This message is used to initiate a communication session between the ECS and a given IWS.

Data field content:

Integer*4	Endian check block data X'01020304' used to insure that the data transmitted by the ECS will correctly be interpreted by the IWS. The ECS uses the big endian format.

3.2.3.2 Session-Init-Response (IWS to ECS). This message is used to acknowledge a Session-Init message and to identify the entity with which the session is established.

Data field content:

Character*16	IWS Identification padded with ASCII blanks.
Integer*1	Endian check result indicating whether the endian check block included in the session-init message was correctly interpreted by the IWS: X'00': successful X'01': unsuccessful If the check is not successful, ECS will consequently terminate the session.

Integer*4 Endian check block data X’01020304’ used to insure that the data transmitted by the IWS is compatible with the endian characteristics of ECS.

3.2.3.3 <u>Session-End</u> (ECS and IWS). This message is used to terminate a session, either immediately after the reception of a Session-Init-Response if a problem was found or to end an on-going session.

Data field content:

Integer*1 Reason Code

X'00': normal end
X'01': invalid ORIG_ID (IWS identification in Session-Init-Response)
X'02': endian check error
X'03': unsuccessful session-init-response or no Session-Init-Response received
X'04': error reading/writing to socket
X'05': invalid IWS message received
X'06': lost connection to CMS

3.2.3.4 <u>NRT-Command</u> (IWS to ECS). This message is used to transfer the near-real-time command data. The message data field for a near-real-time command message is illustrated in Table 3.3.

Data field content:

Integer*2 Request ID generated by the IWS to uniquely identify this command message. ECS and SMOCC use the combination "instrument/request ID" to uniquely identify all NRT-Command messages and NRT-Command-Authority-Request messages. ECS uses the value X'FFFF' for special purposes (see section 3.2.3.7)

Character*6 Instrument name

Character var The command data as defined in section 3.2.2. Each message will contain one and only one OBDH block command definition, that is one mnemonic definition or one "BINARY" keyword.

Table 3.3. Near-Real-Time Command Message Format

Field	Bytes	Description
Standard Header	4	Type (X'0301') and length
Request ID	2	2-Byte integer uniquely identifying this message
Instrument Name	6	Instrument commanded
Command data	var	Command data in mnemonic or binary format. See Section 3.2.2

3.2.3.5 <u>Response-to-NRT-Command</u> (ECS to IWS). This message is used to answer a NRT-Command message. There is one response for each NRT-Command message to indicate its processing status. If an error is detected either by the ECS, the CMS or the POCC, the Response-to-NRT-Command is sent immediately back to the IWS currently commanding the instrument, with a description of the error. Otherwise, an OK status will be sent once the successful uplink status is received from the POCC. This may represent a few seconds delay between the reception of the NRT-command and the response.

Data field content:

Integer*2	Request ID identical to the request ID in the corresponding NRT message.
Character*6	Instrument name
Integer*2	Response Code 00 NRT command message OK 01 NRT command message was rejected
Integer*2	Reason Code (see table 3.4)
Character var	Text explaining reason code (see table 3.4). Also indicates if the instrument error flag has been set, resulting in the need for an instrument reset (described in section 3.2.3.6).

Table 3.4. Response-to-NRT-Command Format Definition

Response Code	Reason Code	Reason Text
01	1	Rejected- throughput mode is disabled (ERROR FLAG SET)
01	2	Rejected - syntax error found in this command group (ERROR FLAG SET)
01	3	Rejected - mnemonic not found in PDB (ERROR FLAG SET)
01	4	Rejected - format error found in message received (ERROR FLAG SET)
01	5	Rejected - duplicate request ID for this instrument (ERROR FLAG SET)
01	6	Rejected - binary format disallowed for this instrument (ERROR FLAG SET)
01	7	Rejected- reserved time commanding in progress
01	8	Rejected - message received had invalid message type
01	9	Rejected - invalid instrument for this socket (ERROR FLAG SET)
01	10	Rejected- start command request not received for instrument
01	11	Rejected- instrument disabled by FOT or previous error
01	12	Uplink failed BARM verification (ERROR FLAG SET)
01	13	Uplink failed - NASCOM link is down (ERROR FLAG SET)
01	14	RCR rejected - RCR processing disabled (ERROR FLAG SET)
01	15	RCR rejected - was not on the approved list (ERROR FLAG SET)
01	16	RPR rejected - STOL procedure not found (ERROR FLAG SET)
01	17	Rejected - invalid first character in name (ERROR FLAG SET)
01	18	RCR terminated - contained invalid command (ERROR FLAG SET)
01	19	RCR terminated - contained critical command (ERROR FLAG SET)
01	20	RCR rejected - not found (ERROR FLAG SET)
00	0	OK - command group successfully uplinked
00	-1	OK - command group uplinked without BARM verification
00	-2	RPR notify - FOT has been notified to start the requested RPR

The Response-to-NRT-Command is also used to notify an instrumenter of the processing status of RCRs and RPRs. This message may indicate one of the following:

- the request was rejected
- the FOT has been notified to start the requested procedure
- the uplink status of the requested PCS

For RPR's, the final status is not systematically made available by the POCC and will not be sent to the instrumenters via this message. An informational message defined between the instrumenters and the FOT may be incorporated into the procedure itself.

Data field content for RPR's and RCR's:

Integer*2	Request ID as provided by instrumenter in the RPR or RCR message.
Character*6	Instrument name
Integer*2	Response Code 00 RPR/RCR message OK 01 RPR/RCR message was rejected
Integer*2	Reason Code (see table 3.4)
Character var	Text explaining reason code (see table 3.4). Also indicates if the instrument error flag has been set, resulting in the need for an instrument reset (described in section 3.2.3.6).

3.2.3.6 **<u>NRT-Command-Authority-Request</u> (IWS to ECS)**. This message identifies the commanding functions to be performed by a given IWS. It allows the ECS to verify that only one IWS is commanding a given instrument at a given time. This message can be used to start commanding, to stop commanding, or to reset commanding after an error for an instrument.

Data field content:

Integer*2	Request ID generated by the IWS to uniquely identify this message.
Character*6	Instrument name. ECS and SMOCC use the combination "instrument/request ID" to uniquely identify all NRT-Command messages and NRT-Command-Authority-Request messages. ECS uses the value X'FFFF' for special purposes (see NRT-Authority-Status messages, section 3.2.3.7).
Integer*2	Request code (see table 3.5)

Table 3.5. NRT-Command Authority Request Format Definition

Request Code	Description Text
00	Start commanding the instrument specified
01	Stop commanding the instrument specified
02	Reset after error: commanding for this instrument to restart with the near-real-time command message immediately following.

3.2.3.7 **<u>NRT-Authority-Status</u> (ECS to IWS)**. This message is used to respond to a NRT-Command-Authority-Request, or to notify an IWS of changes affecting the commanding session. The possible values for the status code transmitted via this message are defined in table 3.6.

Data field content:

Integer*2	Request ID corresponding to ID in NRT-Command-Authority-Request, when applicable. ECS will use a fixed value of X'FFFF' for status codes 3, 5, 6, 7, 8, 9, 10, and 11.
Character*6	Instrument name
Integer*2	Status code (see table 3.6)
Character var	Text explaining status code (see table 3.6)

Table 3.6. NRT-Authority-Status Format Definition

Status Code	Description Text
-1	Command Authority Request denied - duplicate request ID (ERROR FLAG SET)
-2	Start commanding request denied - instrument already commanded
-3	Command Authority Request denied - instrument specified incorrect (ERROR FLAG SET)
-4	Start commanding request denied - session not properly established (incorrect Session-Init response)

Table 3.6. NRT-Authority-Status Format Definition (Cont.)

-5	Command Authority Request denied - IWS not currently commanding instrument specified
-6	Reset denied - instrument not in error
1	Start commanding request granted
2	Stop commanding request granted
3	Received SOC request to cancel this session (1)
4	Reset accepted
5	Throughput mode status = enabled / RCRs allowed (1) (2)
6	Throughput mode status = enabled / RCRs disallowed (1) (2)
7	Throughput mode status = disabled (1) (2)
8	Throughput mode status = paused (1) (2)
9	Warning: Throughput mode shutdown soon (1) (2)
10	Reserved time commanding has ended (1) (2)
11	Warning: reserved time commanding will begin soon (1) (2)
12	Reserved time commanding now in effect for this session (1)
13	Reserved time commanding no longer in effect for this session (1)

(1) Request ID is not applicable and will have a fixed value of X'FFFF'
(2) Message sent to all IWSs with open commanding sessions

3.2.3.8 <u>Remote Command Request and Remote Procedure Request</u> (IWS to ECS).
For RCRs, the message data field identifies the name of a Predefined Command Sequence (PCS); For RPRs, the message data field identifies the name of a STOL procedure.

Data field content:

Integer*2	Request ID generated by the IWS to uniquely identify this message.
Character*6	Instrument name
Character*20	The name of the predefined command sequence or the STOL procedure, in lower case characters, left-justified and padded with ASCII blanks if necessary. The first character of the name must be as follows: c for CDS f for CELIAS h for CEPAC e for EIT g for GOLF l for LASCO m for MDI s for SUMER n for SWAN u for UVCS v for VIRGO
Character var	Comments and execution instructions in the form /* text */, maximum of 256 characters, are allowed for RPRs.

3.2.3.9 <u>Informational Message</u> (Bi-directional).

This message is supplied for the exchange of free text between the ECS and the IWSs. For instance, it will be used by ECS to provide execution status for RPRs. The SMOCC passes an informational message to ECS that contains a text defined between the instrumenters and the FOT as part of the definition of the STOL procedures. ECS will forward that message to the IWS currently commanding the instrument.

Data field content:
Character var ASCII Free text (maximum of 256 characters)

3.2.4 DELAYED COMMANDING DATA SPECIFICATION

The delayed commanding data is received by the ECS as files. Each file is comprised of a file header followed by a file body that contains the command data as specified in section 3.2.2. The maximum number of OBDH block commands that can be included in a single delayed command file is 1000. If this number is exceeded, ECS will reject the file and indicate so in the command validation report. The file header is described in Table 3.7. It specifies the earliest and latest uplink times.

Table 3.7. File Header Format for Delayed Commanding.

KEYWORD	DESCRIPTION
DATATYPE	"DELAYED"
FILENAME	Name of this file: iiicccccccc.DEL (see Appendix A)
INSTRUME	Instrument (full name) being commanded
ORIG_ID	ID of originating entity (IWS ID or remote host). No embedded blanks allowed.
OBSERVER	Person who generated this file
DATE_CRE	Date file was created YYYY/MM/DD HH:MM:SS (GMT)
NUM_CMDS	Number of commands (OBDH Block commands) in this file.
EARLIEST	Earliest uplink time YYYY/MM/DD HH:MM:SS (GMT)
LATEST	Latest uplink time YYYY/MM/DD HH:MM:SS (GMT)
COMMENT	Free text. May contain special instructions (i.e., contingency, end-item verification, etc...). This keyword may be repeated to allow several comment lines
END	

3.2.5 BACKGROUND-QUEUE COMMANDING DATA SPECIFICATION

The file header is defined in Table 3.8. It specifies the total number of commands contained in the file and an optional uplink window. The uplink window may be specified when it is critical to uplink the data by a given time. However, SMOCC does not guarantee uplink within that window and would only reject the data that has not yet been uplinked by the specified latest uplink time. In most cases the window will not be specified, and the data will be uplinked whenever possible.

The file body contains the commanding data which consists of a series of command specifications either in binary format or mnemonic format as specified in section 3.2.2. The command data, once expanded into the binary form of OBDH block commands, should be less than 0.5 Kbytes in length.

Table 3.8. File Header Format for Background-Queue Commanding.

KEYWORD	DESCRIPTION
DATATYPE	"BACKGROUND"
FILENAME	Name of this file: iiicccccccccc.BCK (see Appendix A)
INSTRUME	Instrument being commanded
ORIG_ID	ID of originating entity (IWS ID or remote host). No embedded blanks allowed.
OBSERVER	Person who generated this file
DATE_CRE	Date file was created YYYY/MM/DD HH:MM:SS (GMT)
NUM_CMDS	Number of commands (OBDH Block commands) in this file.
EARLIEST	Optional. Earliest uplink time YYYY/MM/DD HH:MM:SS (GMT)
LATEST	Optional. Latest uplink time YYYY/MM/DD HH:MM:SS (GMT)
COMMENT	Free text. May contain special instructions (i.e., contingency, end-item verification , etc...). This keyword may be repeated to allow several comment lines
END	

3.2.6 COMMAND VALIDATION REPORTS

These reports are generated by the CMS as soon as CMS receives and processes a delayed command group or a background-queue command group. They contain an echo of the original commanding data: list of mnemonics or binary specification, and error messages when applicable. There will be one validation report for each group (one uniquely named file) of delayed or background-queue commanding data generated by the instrumenters.

Table 3.9. File Header Format for Command Validation Reports.

KEYWORD	DESCRIPTION
DATATYPE	"COMMAND VALIDATION REPORT"
FILENAME	Name of this file: (see Appendix A) iiicccccccccc.DRP for Delayed command validation reports iiicccccccccc.BRP for Background queue validation reports
INSTRUME	Instrument commanded in original delayed or background command group
ORIGFILE	Filename of delayed/background command group this report applies to. iiicccccccccc.DEL or iiicccccccccc.BCK
DATE_CRE	Date this file was created YYYY/MM/DD HH:MM:SS (GMT)
NUM_CMDS	Number of commands (OBDH Block commands) covered in this report
COMMENT	Free text. This keyword may be repeated to allow several comment lines
END	

3.2.7 SPECIAL COMMANDING FOR THE VIRGO INSTRUMENT

The VIRGO instrument requires special commanding data specifications. At the present time, VIRGO sends time-tagged commands as delayed command files. These files are rejected by CMS as invalid and are then processed manually by the CMS operator. A final process to be used during operations still needs to be defined and accepted by CMS.

3.3 TELEMETRY DATA SPECIFICATION

3.3.1 REAL-TIME TELEMETRY

The general format of the messages exchanged over the interface was defined in section 3.1.1. They all contain a 4-byte standard header followed by a data field. This section defines the specific functions and the data field content of the messages exchanged for real-time telemetry distribution.

The Session-Init message (ECS to IWS), the Session-Init-Response message (IWS to ECS) and the Session-End messages (ECS and IWS) are identical to the messages used for NRT commanding, defined in section 3.2.3.

3.3.1.1 Telemetry-Packet-Distribution-Request (IWS to ECS). This message is used by an IWS to indicate what type of telemetry packets (i.e., what APID) it wants to receive. There must be one such message for every APID to be transmitted. This message contains an optional field that allows an instrumenter to choose to receive the Quality and Accounting capsule at the end of each telemetry packet for a given APID. If the flag is not specified, the Quality and Accounting capsule will not be provided at the end of the telemetry packets.

Data Field content:

Integer*1	Spacecraft ID (X'00').
Integer*2	APID to be received.
Integer*2	Request ID: non-negative number generated by IWS to uniquely identify this telemetry exchange. It will be found in all the messages related to this TM distribution request.
Integer*1	Optional: flag indicating if the Q&A capsule should be included (01) or omitted (00) after each telemetry packet for this APID for this telemetry exchange.

3.3.1.2 Telemetry-Packet-Distribution-Response (ECS to IWS). This message is used to indicate the result of the telemetry-packet-distribution-request:

Data Field content:

Integer*2	Request ID corresponding to request ID in TM Packet Distribution Request message.
Integer*1	Request Response Code X'00' successful X'01' unsuccessful
Integer*1	Reason Code giving an explanation for success or failure X'00' Success X'01' Bad APID X'02' APID already requested X'03' Duplicate Request ID X'04' Request ID in TM-packet-distribution-request is missing X'05' TM data not available X'06' ECS system capacity exceeded X'07' to X'0F' other reasons

3.3.1.3 Start-of-Telemetry-Packet-Distribution (ECS to IWS). This message is used to indicate the start of telemetry transmission.

Data Field content:

Integer*1 Spacecraft ID (X'00').
Integer*2 APID of telemetry packets included in this distribution
Integer*2 Request ID corresponding to request ID in TM Packet Distribution Request message

3.3.1.4 Interrupt-Telemetry-Packet-Transfer (IWS to ECS). This message is used by an IWS to ask for immediate termination of the current telemetry transfer.

Data Field content:

Integer*2 Request ID corresponding to request ID in TM Packet Distribution Request message

3.3.1.5 End-of-Telemetry-Packet-Transfer (ECS to IWS). This message is sent by ECS to confirm the fact that the telemetry transfer is terminated. This may happen either upon receipt by ECS of an interrupt-telemetry-packet-transfer message, in the case of a system problem, or at the end of the real-time contact period.

Data Field content:

Integer*2 Request ID corresponding to request ID in TM Packet Distribution Request message
Integer*1 Status code
X'00' Canceled by IWS
X'01' Canceled by ECS
X'02' Telemetry transfer interrupted by PACOR
X'03' to X'0F' other reasons.

3.3.1.6 Telemetry-Packet (ECS to IWS). This message is used to transmit the real-time telemetry data, that is one complete telemetry source packet corresponding to the APID requested. Additional data may be included, such as the Quality and Accounting (Q&A) capsule. If requested by the receiving instrumenter in the Telemetry-Packet-Distribution-Request message, a 6-byte Q&A capsule will be appended to the end of each telemetry packet associated with this Request ID.

Data Field content:

Integer*2 Request ID corresponding to request ID in TM-Packet-Distribution-Request.
Binary data Telemetry data packet, including packet header as defined in table 3.10. Note that for some APIDs, the packets do not contain a time field.
Integer*6 (Optional, supplied only if requested in Telemetry-Packet-Distribution-Request) Real-time Q&A capsule, as provided by PACOR (see Reference 9) and defined in table 3.11.

Table 3.10. Telemetry Data Packet

PACKET HEADER			PACKET DATA FIELD	
Packet ID	Sequence Control			

Version No.	Packet type	Data field header flag	APID	Segment flags	Source sequence count	Packet length	Time field (OBT or LOBT)	Source data
3 bits	1 bit	1 bit	11 bits	2 bits	14 bits	16 bits	48 bits	variable
(2 bytes)				(2 bytes)		(2 bytes)	(6 bytes)	variable

Table 3.11. Real-Time Quality and Accounting Capsule

Field Name	**Length**	**Description**
Virtual Channel ID	1 byte	Virtual channel the packet was transmitted on
Data Type Flag	1 byte	PACOR uses each bit is a flag indicating the data type: 00010000 for real-time and 00001000 for test telemetry
Sequence Continuity Flag	1 byte	00 (hexadecimal) no sequence discontinuity 01 (hexadecimal) sequence discontinuity
Reed-Solomon Error Flag	1 byte	00 (hexadecimal) no Reed-Solomon correction 01 (hexadecimal) Reed-Solomon error corrected.
Data Fill Location	2 bytes	Location of the start of fill in the source data unit. A value of 0000 hex indicates there is no fill.

3.3.1.7 <u>Informational Message</u> (Bi-directional).

This message is supplied for the exchange of free text between the ECS and the IWSs. For instance, ECS sends an informational message to an IWS when a telemetry session is canceled by the ECS operator. ECS displays the informational messages it receives from the IWSs on the ECS event page.

Data field content:
Character var ASCII Free text (maximum of 256 characters)

3.3.2 ARCHIVED TELEMETRY DATA

The telemetry data is stored in the ECS for retrieval by the instrumenters. The archived telemetry is sorted by APID and by time. Each file contains approximately 2 hours of telemetry and contains a header followed by the file body. The telemetry data is organized among several system directories, one directory per APID. Under each directory, each file contains packets consecutively received by ECS for the given APID. The instrumenters can obtain the archived telemetry data via file transfer. In order to select the files they want to retrieve, the instrumenters will use the file naming conventions described in Appendix A or will formulate a standing request with the SOC to receive files corresponding to a given APID as soon as these files are available within ECS.

3.3.2.1 <u>Archived Telemetry File Header</u>.

Table 3.12 defines the format of the file header.

3.3.2.2 <u>Archived Telemetry File Body</u>.

The file body contains the telemetry packets followed by quality and accounting information as illustrated in table 3.13. The format of the file body is illustrated in Appendix C. It was kept as much as possible identical to the format of the production data as defined in the ICD between the SDPF and the SOHO Consumers (Reference 9).

Table 3.12. Archived Telemetry File Header

DATATYPE	"ARCHIVED REAL-TIME TELEMETRY" or "ARCHIVED RETRANSMITTED REAL-TIME TELEMETRY" or "ARCHIVED TAPE RECORDER DUMP TELEMETRY"
FILENAME	As defined in Appendix A
APID	APID of the telemetry packet (see Appendix A)
DATE_CRE	Date file was created by ECS YYYY/MM/DD HH:MM:SS (GMT)
NUM_PACK	Number of telemetry packets stored in this file
STARTIME	Start of period covered (Time stamp of first packet) YYYY/MM/DD HH:MM:SS when applicable. Othewise, time of reception by ECS of the first data packet.
ENDTIME	End of period covered (Time stamp of last packet) YYYY/MM/DD HH:MM:SS when applicable. Othewise, time of reception by ECS of the last data packet.
COMMENT	Free text. When applicable, will indicate if the file name is based on the ECS system time (time of first packet not available)
END	

Table 3.13. Archived Telemetry File Body

Source Data Units	Series of telemetry packets: Telemetry packet 1 Telemetry packet 2 . . . Telemetry packet n
Quality and Accounting List Length	Length in bytes of the Quality and Accounting List
Quality and Accounting List	Series of Quality and Accounting Capsules: Quality and Accounting capsule for first packet in error Quality and Accounting capsule for second packet in error . . . Quality and Accounting capsule for m^{th} packet in error
Missing Data Units List Length	Length in bytes of the Missing Data Units List
Missing Data Units List	Series of Missing Data Units Entries: Offset, "From" packet and "To" packet . . .

3.3.3 TELEMETRY GAP REPORT

The ECS generates a Telemetry Gap Report for each operational day. This report is generated from the archived telemetry Selective reports for shorter time ranges can be generated by SOC. Appendix B contains an example of the Telemetry Gap Report.

3.4 MISSION SUPPORT DATA SPECIFICATION

3.4.1 SUMMARY DATA

The summary data received by ECS from the instrumenters is in FITS format. There may be one or more input files for each instrument and each day (some instruments may provide more than one file for the same day, while other instruments may not provide a file for every day). The SOC is responsible for gathering the instrumenters' input before sending it to CDHF which requires detached SFDU headers.

The instrumenters submit their input to ECS in FITS format optionally with detached SFDU headers. The file naming convention used for the summary data will comply with the CDHF conventions (see Appendix A).

3.4.2 ORBIT AND ATTITUDE DATA

Orbit and attitude data are received by the ECS from CDHF at a frequency that will be defined within the EOF and operationally agreed upon with CDHF. These data are received in files in CDF format with detached SFDU headers. Each file contains data for one operational day.

That data is provided to the instrumenters in CDF format with detached SFDU headers, and the CDHF file naming conventions described in Appendix A are used.

The files will be organized among system directories in the ECS. The data files and the corresponding SFDU header files are contained in the same directory.

3.4.3 COMMAND HISTORY

The command history file covers one operational day. ECS receives a file from the SMOCC typically at the end of every DSN contact. The ECS merges the individual files into a single file per day. It then generates an SFDU header before transmitting the final report to CDHF.

The command history is available to the instrumenters as an ASCII text file with a detached SFDU header. See Appendix B for an example of the command history file format as proposed by the SMOCC (ASCII text with fixed fields).

3.4.4 TIME CORRELATION LOG

The time correlation log is an ASCII text file containing a cumulative log of all SOHO spacecraft clock time offsets since the start of the mission. Once each day that the spacecraft clock is adjusted, ECS updates the time correlation file by appending the new information at the end. ECS extracts the time correlation information from a command history file by recognizing commands that were used to reset the spacecraft clock.

The time correlation log is made available to the instrumenters as an ASCII text file with detached SFDU header.

3.4.5 SYNOPTIC DATA

The format of that data is not defined in this ICD. The synoptic data will be gathered by the SOC and will reside in a dedicated ECS file directory from where the instrumenters will be able to retrieve it. The management of the synoptic data is not an ECS function.

3.4.6 PROJECT DATA BASE

The ECS will make the PDB available to the instrumenters as ASCII files in the Data Format Control Document (DFCD) format as supplied by the POCC.

3.4.7 PROJECT DATA BASE UPDATE REQUEST

The format of this free-form text E-mail message exchanged between the instrumenters and the FOT is not described in this ICD.

3.4.8 SOHO DAILY REPORT

The SOHO Daily Report is an ASCII text file with detached SFDU header. Each file covers one operational day. The files are named according to the CDHF file naming conventions described in Appendix A. The format of the SOHO Daily Report will be defined in an operational agreement between the FOT and the Instrumenters.

3.5 PLANNING AND SCHEDULING DATA SPECIFICATION

There are two types of data related to the planning functions: the ECS activity (EAP) plan which ECS sends to the instrumenters and the instrumenters' input to the activity plan (IAP). These data are in a file format as described in the following sections.

3.5.1 INSTRUMENTERS INPUT TO THE ACTIVITY PLAN

The IAP is a file used by each instrument team to specify their activity requests. One input file relates to a single instrument and covers an entire operational day. When updates and modifications are necessary, the IAP for the entire operational day must be re-submitted to the ECS. In order to allow for greater flexibility in the input, the "keyword=value" format is used for the IAP files.

3.5.1.1 Input to the Activity Plan File Header

The IAP file header is defined in table 3.14.

Table 3.14. File Header for the Instrumenter Input to the Activity Plan

KEYWORD	DESCRIPTION
DATATYPE	"INSTRUMENTER INPUT TO THE ACTIVITY PLAN"
FILENAME	Name of this file: iiiccccccccc.IAP (see Appendix A)
INSTRUME	Instrument for which this IAP is submitted (see section 3.1.4)
ORIG_ID	E-Mail address where a validation report on the IAP will be sent
OBSERVER	Person who generated this file
DATE_CRE	Date file was created YYYY/MM/DD HH:MM:SS (GMT)
STARTIME	Start time of period covered YYYY/MM/DD HH:MM:SS (GMT) ECS will assume that HH:MM:SS is 00:00:00

ENDTIME	End time of period covered YYYY/MM/DD HH:MM:SS (GMT) ECS will assume that HH:MM:SS is 00:00:00 for the day following the start time
COMMENT	Free text. This keyword may be repeated to allow several comment lines
END	

3.5.1.2 Input to the Activity Plan File Body Format

The file body is in the keyword format. It contains a list of statements, each statement consisting of a series of fields of the form KEYWORD = value. A list of keywords that may be used in the IAP is provided in Appendix B.

Notes:
- Comment lines may be inserted anywhere in the IAP file. They consist of the keyword COMMENT= followed by free text.

- Typically, the duration of an activity is specified using the STARTIME= and ENDTIME= keywords, followed by a time field.
  All time fields are in the standard format YYYY/MM/DD HH:MM:SS.

- The originator ID (ORIG_ID) must be a valid E-Mail address where a validation report on the IAP will be sent. When it receives the IAP, the ECS planning software validates it and generates a report indicating errors if any are found. This report is sent in the form of an E-mail message to the address indicated in the ORIG_ID field. This address will also be used to send availability notifications for the EAP.

3.5.2 ECS ACTIVITY PLAN

The ECS Activity plan (EAP) is a file containing information provided by the SMOCC such as DSN contacts and FOT-reserved times as well as the merged activity requests that were specified by the instrumenters in the IAP files. The activity plan file is made available in two different formats: the fixed-field format which provides more readability but limits the amount of information provided, and the keyword format which offers more flexibitity.

3.5.2.1 ECS Activity Plan File Header

The activity plan file header is defined in table 3.15.

Table 3.15. File Header for the ECS Activity Plan

KEYWORD	DESCRIPTION
DATATYPE	"ECS ACTIVITY PLAN"
FILENAME	Name of this file: (see Appendix A) ECSYYYYMMDDvvv.EAP for files in the fixed-field format ECSYYYYMMDDvvv.KAP for files in the keyword format
OBSERVER	Person who generated this file
DATE_CRE	Date file was created YYYY/MM/DD HH:MM:SS (GMT)
STARTIME	Start time of period covered YYYY/MM/DD HH:MM:SS (GMT) where HH:MM:SS is 00:00:00 for a given day DD

ENDTIME	End time of period covered YYYY/MM/DD HH:MM:SS (GMT) where HH:MM:SS is 00:00:00 for day DD+1
OUT_FORM	Format indicator KEYWORD for the keyword format FIXED_FIELD for the fixed field format
COMMENT	Free text. This keyword may be repeated to allow several comment lines
END	

3.5.2.2 ECS Activity Plan Fixed-field Format

In the fixed-field format, the file body for the ECS activity plan is an ASCII-text formatted report. Each line corresponds to the description of one activity and contains the fixed-length fields defined in table 3.16.

Table 3.16. ECS Activity Plan Fixed-field Format

FIELD NAME	Byte Number	Length (Bytes)	DESCRIPTION
Activity	1	20	Name of the activity or resource (see Appendix B)
Blank-fill	20	2	
Activity Qualifier	22	6	Instrument performing the activity or Ground station ID for DSN contacts or Originating entity for the activity such a FOT or SOC
Blank-fill	28	2	
Start Time	30	19	Start time of the activity in the format YYYY/MM/DD HH:MM:SS
Blank-fill	49	2	
End Time	51	19	End time of the activity in the format YYYY/MM/DD HH:MM:SS
Blank-fill	70	2	
Duration	72		Duration of the activity in the format HH:MM:SS
Blank-fill	80	2	
Description/Comment (optional)	82	80	Textual description of or remarks applying to the activity
New-line separator	162	1	X'0A'

3.5.2.3 ECS Activity Plan Keyword Format

In the keyword format, the file body for the ECS activity plan is ASCII text. Refer to Appendix B for a list of valid keywords and their format specification.