Release Date: _____/_____/_____/	Marshall Space Flight Center **SPECIFICATION/DOCUMENT CHANGE INSTRUCTION** Spec./Doc. No. MSFC-DOC-1949	Page 7 Copy No.:

Change No./Date	SCN/DCN No./Date	CCBD No./Date	Replacement Page Instructions
018/ 5/8/00	022/ 5/8/00	HM3-00-2463/ 3/17/00	Replace Volume 4, pages 4-15, 4-16, 4-23, 4-24, 4-35, 4-36, 4-39, and 4-40.
		HM3-00-2479/ 3/20/00	Revised Volume 5, pages 2-33, 2-34, 2-43, 2-44; 3-1 and 3-2; 4-3, 4-4, 4-9, 4-10, 4-15, and 4-16; A-3 thru A-6; H-3 and H-4; and I-9 thru I-12.
		HM3-00-2515/ 3/20/00	Add Volume 5, Appendix J, pages J-1 thru J-30.
			Replace Volume 5 Table of Contents, pages v thru viii; and xiii and xiv.
	026/ 5/8/00	HM3-00-2534/ 4/7/00	Replace Volume 6, pages 6-45 and 6-46; 6-121 thru 6-166; 8-27, 8-28, 8-35, 8-36; 8-49 thru 8-64.
			Replace Table of Contents, Volume 6, pages xi, xii, xxix, xxx, xxxi, xxxii, xxxiii, xxxiv, xxxv, and xxxvi.
			Add Form 4140 (S/DCI), page 7.
019/ 6/9/00	022R1/ 6/9/00	HM3-00-2490/ 12/3/99	Replace Volume 1, pages 1-1 and 1-2.
			Replace Volume 4, pages 1-1 and 1-2.
			Replace Form 4140 (S/DCI), page 7.
020/ 10/16/00	027/ 10/16/00	HM3-00-2570/ 8/4/00	Replace Volume 3, pages 3-1 thru 3-24.
		HM3-00-2602/ 8/14/00	Replace Volumes 4 and 6 in their entirety. These are being replaced without DCN bars.
	028/ 10/16/00	HM3-00-2616/ 9/8/00	Replace Volume 5, pages 2-6, 2-7; 2-17, 2-18; 2-49, 2-50, 4-5, 4-6, 4-9, 4-10; 4-13 thru 4-22, 4-25, 4-26, J-3, J-4; and J-9 thru J-12.
			Replace Form 4140 (S/DCI), page 7.
		HM3-00-2641/ 10/10/00	Add Form 4140 (S/DCI), page 8.

MSFC - Form 4140 (Revised September 1990) NOTE: After revising the document, file this sheet in document preceding table of contents.

MSFC-DOC-1949C
August 1998

MSFC HOSC Database Definitions

Volume 2

Command Database

Revision C

August 1998

MSFC Huntsville Operations Support Center (HOSC) Telemetry and Command Database Definition

Volume 2

Command Database

Revision C

August 1998

PREPARED FOR

Information Processing Branch (EO37)
Missions Systems Division
Mission Operations Laboratory
Science and Engineering Directorate
Marshall Space Flight Center

PREPARED UNDER:

NASA UMS Contract #NAS8-44000

Approved By:

_____ _____

Chris Sims
Database Engineering Branch
EO38

Date

_____ _____

Ann R. McNair, Assistant Director
Mission Operations Laboratory
EO01

Date

FORWARD

This document defines the detailed requirements for the George C. Marshall Space Flight Center (MSFC) Huntsville Operations Support Center (HOSC) command database that is used to support commanding.

Comments and questions concerning the content of this document should be addressed to Chris Sims, telephone (205) 544-8039, Office Code EO37, MSFC.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

SECTION **PAGE**

TABLE OF CONTENTS (CONTINUED)

SECTION **PAGE**

LIST OF FIGURES

FIGURE **PAGE**

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

LIST OF TABLES (CONTINUED)

TABLE **PAGE**

1.0 INTRODUCTION

This volume describes the Command Database that the Huntsville Operations Support Center (HOSC) command system requires for command processing. This database is used to supply flight-specific information to configure the HOSC command processing system. The Project/ Mission Manager for each support activity is required to provide this database.

Data can be supplied to the HOSC by tape, floppy disk, or electronically through a HOSC-supported network. The database must either be a set of relational database tables produced through HOSC-supported commercial Relational Database Management software or an American Standard Code for Information Interchange (ASCII) text file as specified by this volume in section 3.

1.1 PURPOSE

The purpose of this volume is to provide a description of the Marshall Space Flight Center (MSFC) HOSC Command Database that will be used to define commands for projects supported by the MSFC HOSC.

1.2 SCOPE

This volume defines the database structures required to support MSFC HOSC command processing. This volume is intended to replace the Command Database portion of the MSFC Payload Operations Control Center (POCC) Telemetry and Command Database Definition, MSFC-DOC-1149A, effective January 1, 1997. The MSFC HOSC will continue to accept data in the MSFC-DOC-1149A format after this date, but the internal HOSC database structures will be in compliance with MSFC-DOC-1949.

1.3 REFERENCE DOCUMENTS

The following documents were used as reference material for preparing this volume:

MSFC-DOC-1149	MSFC POCC Telemetry and Command Database Definition
MSFC-DOC-1949	MSFC HOSC Database Definitions, Volume 3
HOSC-EHS-065	Enhanced HOSC System (EHS) Concepts and Scenarios Document

THIS PAGE INTENTIONALLY LEFT BLANK

2.0 COMMAND DATABASE TABLES

The Command Database is a relational database that consists of the following tables:

- Command System
- Destination
- Owner
- Header
- Header Field
- Command
- Command Field
- Point Pair Decalibration
- Telemetry Verifier
- Chain
- Chain Contents
- Command Data Set
- Command Data Set Field

The following sections describe the columns of each command database table. Appendix B contains definitions of column types used in these descriptions, and Appendix C contains legal characters and special characters.

The following summary of the tables is provided to give the reader an overview before proceeding into the details of the contents of each table.

The Command System table is used to differentiate one project command database from another. This table is stored with the Command Database so that it can be retrieved from backups and restored in case of system faults.

The Destination table is used to define one or more onboard destinations for the commands in the database. It is also used to store project-specific constraints on lengths of command components for the specified destinations.

The Owner table is used to cross-reference an owner identifier (ID) to a text description of the owner. The authorized owner of a header, command, or command chain is the entity to which the ability to update the definition of the header, command, or chain has been assigned. This entity may be a spacecraft subsystem or payload, a HOSC console position, or anything else that the project wishes to use.

The Enhanced HOSC System (EHS) command concept is that each command is made up of a header and command data. The Header table contains the length and description of each header. The Command table is used to maintain general information about each command including special processing constraints. The header and command data are composed of fields which can be of different lengths and data types. The Header Field table contains the header fields and the Command Field table contains the command data fields. The contents of these tables, as supplied by the user, specify the field data and field positions required to construct each command.

Modifiable commands are commands which include modifiable data fields. During operational activity, the modifiable fields of such a command may be updated using a command update form. When a user enters command data into a command update form, the field data can be decalibrated into raw counts before uplink using either point pair or polynomial decalibration. Decalibration is the process of using calibration point pairs or polynomials to solve for a value in counts. For example, if a field is to represent a temperature value, the user may want to enter the value in degrees Celsius, but the onboard device expects the value in raw counts. The field data must be converted to raw counts before uplink. The calibration coefficients or point pairs to accomplish such a conversion are supplied in the Command Field table or Point Pair Decalibration table.

When a command is executed onboard a spacecraft, there is often some telemetry measurement that is associated with the command's execution. This measurement can be used to verify the command's success or failure. The Telemetry Verifier table is used to specify which measurements should be checked for each command and the ranges of values those measurements should be within to indicate positive command verification.

Commands can be grouped together into a sequence that is referred to as a command chain. Command chains can be used to automate the sequential uplinking of a group of commands or just to associate certain commands together. The Chain table is used to maintain general information about each command chain as a whole, including whether the chain is predefined or modifiable. The contents of a predefined chain, as defined in the Chain Contents table, are configuration controlled and cannot be modified without the approval of the Database Coordination Group (DBCG). The contents of a modifiable chain, however, are not configuration controlled. The commands within a modifiable chain may be changed or rearranged by an authorized user, although the information in the Chain table must remain under the configuration control of the DBCG.

The Chain Contents table is used to define the commands contained in each command chain and their positions in the chain.

Modifiable commands are commands which include modifiable data fields. For any given modifiable command, command data sets may be defined which specify values for the modifiable data fields in the command. Command data sets can be used to simplify the uplinking of modifiable commands by allowing users to define values for the modifiable data fields in advance. The Command Data Set table is used to maintain general information about each command data set as a whole. The Command Data Set Field table is used to define the contents of each field in the command data set. The contents of these tables are not configuration controlled by the HOSC.

Other constraints that apply to these tables are described in more detail in the sections devoted to each table.

Table 2.0-1 is a summary of the columns contained in the Command Database tables.

Table 2.0-1. Command Database Tables Summary

TABLE NAME	COLUMN NAME
Command System	PROJECT, MISSION, REV, DELIVER_DATE, PRE_RELEASE_DATE, BASELINE_DATE
Destination	DEST_TYPE, MAX_HEADER_LENGTH, MAX_WORDS, WORD_LENGTH
Owner	OWNER_ID, DESCRIPTION
Header	HEADER_ID, OWNER_ID, HEADER_TYPE, DEST_TYPE, LENGTH, DESCRIPTION
Header Field	HEADER_ID, FIELD_MNEMONIC, FIELD_TYPE, INPUT_DATA_TYPE, UPLINK_DATA_TYPE, LENGTH, START_WORD, START_BIT, INIT_DATA, CALC_TYPE, DESCRIPTION

Table 2.0-1. Command Database Tables Summary (Continued)

Command	CMD_MNEMONIC, MSID, OWNER_ID, CMD_TYPE, DEST_TYPE, CLASS, OP_CODE, LENGTH, VAR_LENGTH, INIT_STATE, THRU_TYPE, HAZARDOUS, CRITICAL, MASTER_COMMAND, SPECIAL_PROCESS, PARITY_PROCESS, TIME_CONSTRAINT, HEADER_ID, TLM_VERIF_FLAG, TLM_VERIF_TIMEOUT, TECHNICAL_NAME, DESCRIPTION
Command Field	CMD_MNEMONIC, FIELD_MNEMONIC, FIELD_TYPE, INPUT_DATA_TYPE, UPLINK_DATA_TYPE, ENG_UNIT, DIMENSION, DECAL_TYPE, LENGTH, VAR_LENGTH, START_WORD, START_BIT, INIT_DATA, RANGE_LOW, RANGE_HIGH, CAL_COEF_0, CAL_COEF_1, DESCRIPTION
Point Pair Decalibration	CMD_MNEMONIC, FIELD_MNEMONIC, COUNTS, VALUE
Telemetry Verifier	CMD_MNEMONIC, TLM_MSID, STATE_CODE, RANGE_LOW, RANGE_HIGH

Table 2.0-1. Command Database Tables Summary (Continued)

Chain	CHAIN_NAME, OWNER_ID, CHAIN_TYPE, INIT_STATE, DESCRIPTION
Chain Contents	CHAIN_NAME, CMD_POSITION, CMD_MNEMONIC
Command Data Set	CMD_MNEMONIC, DATA_SET_NAME
Command Data Set Field	CMD_MNEMONIC, DATA_SET_NAME, FIELD_MNEMONIC, DATA

The Command Database described by this document is based on the “relational” database model. In a relational database, the objects, or “things,” about which data are kept are called entities. Each table in the Command Database contains data about a single type of entity. Some of the tables contain data about command characteristics such as the Command table. Other tables contain data about the relationships between the entities that are represented by the other tables. For example, the Chain Contents table contains data about the relationship between the commands in the Command table and the chains in the Chain table. Regardless of whether the table contains data about command characteristics or relationships, all of the data for each occurrence of the entity represented by the table is located in a single row of the table.

Each table in the Command Database contains a set of columns that is called the “primary key” for that table. The set of values in the primary key columns of a table must be unique for each row in the table. Therefore, the primary key can be used to locate the data for any given occurrence of the corresponding entity. Some of the tables in the Command Database have a primary key that is only one column while other tables have several columns in the primary key. The columns in the primary key must not contain null values.

See Table 2.0-2 for the Command Database primary keys.

Table 2.0-2. Command Database Primary Keys

TABLE NAME	PRIMARY KEY
Command System	PROJECT, MISSION, and REV
Destination	DEST_TYPE
Owner	OWNER_ID
Header	HEADER_ID
Header Field	HEADER_ID and FIELD_MNEMONIC
Command	CMD_MNEMONIC
Command Field	CMD_MNEMONIC and FIELD_MNEMONIC
Point Pair Decalibration	CMD_MNEMONIC, FIELD_MNEMONIC, and COUNTS
Telemetry Verifier	CMD_MNEMONIC and TLM_MSID
Chain	CHAIN_NAME
Chain Contents	CHAIN_NAME and CMD_POSITION
Command Data Set	CMD_MNEMONIC and DATA_SET_NAME
Command Data Set Field	CMD_MNEMONIC, DATA_SET_NAME, and FIELD_MNEMONIC

In a relational database, a “foreign key” is one or more columns in one table that reference key columns in another table. To insert or update a row in a table that contains a foreign key, the set of new values for the foreign key columns must already exist in the referenced columns of the other table. However, some foreign key columns do not require values at all. In this case, these foreign key columns can contain “null” values. Foreign keys can be used to enforce data integrity by requiring a set of columns to contain valid values based on the values in the referenced table. Foreign keys can also be used to represent relationship between two tables. The columns that are referenced by a foreign key must contain unique sets of values for each row in the referenced table. Because of this restriction on foreign keys, some tables contain common columns that are not foreign keys, but which represent a relationship between the tables. Each table can contain any number of foreign keys and any number of columns that are common with other tables. Therefore, complicated relationships among a set of entities can be represented in the corresponding tables.

See Table 2.0-3 for the Command Database foreign keys.

Table 2.0-3. Command Database Foreign Keys

TABLE NAME	FOREIGN KEY(S)	REFERENCE TABLE	REFERENCE COLUMN(S)
Command System	None	None	None
Destination	None	None	None
Owner	None	None	None
Header	OWNER_ID	Owner	OWNER_ID
	DEST_TYPE	Destination	DEST_TYPE
Header Field	HEADER_ID	Header	HEADER_ID
Command	OWNER_ID	Owner	OWNER_ID
	DEST_TYPE	Destination	DEST_TYPE
	MASTER_COMMAND	Command	CMD_MNEMONIC
	HEADER_ID	Header	HEADER_ID
Command Field	CMD_MNEMONIC	Command	CMD_MNEMONIC
Point Pair Decalibration	CMD_MNEMONIC, FIELD_MNEMONIC	Command Field	CMD_MNEMONIC, FIELD_MNEMONIC
Telemetry Verifier	CMD_MNEMONIC	Command	CMD_MNEMONIC
Chain	OWNER_ID	Owner	OWNER_ID
Chain Contents	CHAIN_NAME	Chain	CHAIN_NAME
	CMD_MNEMONIC	Command	CMD_MNEMONIC
Command Data Set	CMD_MNEMONIC	Command	CMD_MNEMONIC

Table 2.0-3. Command Database Foreign Keys (Continued)

TABLE NAME	FOREIGN KEY(S)	REFERENCE TABLE	REFERENCE COLUMN(S)
Command Data Set Field	CMD_MNEMONIC, DATA_SET_NAME	Command Data Set	CMD_MNEMONIC, DATA_SET_NAME
	CMD_MNEMONIC, FIELD_MNEMONIC	Command Field	CMD_MNEMONIC, FIELD_MNEMONIC

Entity Relationship (E-R) diagrams provide a graphical method of representing the relationships among a set of entities and their corresponding tables in a relational database. Rectangles are used in an E-R diagram to represent an entity and its corresponding table. Arrows between the rectangles are used to represent the relationships. In the database tables, these relationships can be represented by foreign keys or by common columns in the tables. Only the primary relationships between entities are represented in the E-R diagrams.

The foreign keys that are only used to enforce data validation are not shown. Otherwise, the E-R diagrams would contain so many rectangles and arrows that the primary relationships would be obscured. Each relationship in an E-R diagram can be either one-to-one, one-to-many, or many-to-many. For example, a single owner can own many commands but each command can only have one owner. This is an example of a one-to-many relationship. In the E-R diagram in this document, a single arrowhead is used to represent a “one” side of a relationship and a double arrowhead is used to represent a “many” side.

The E-R diagram for the command database is shown in Figure 2.0-1.

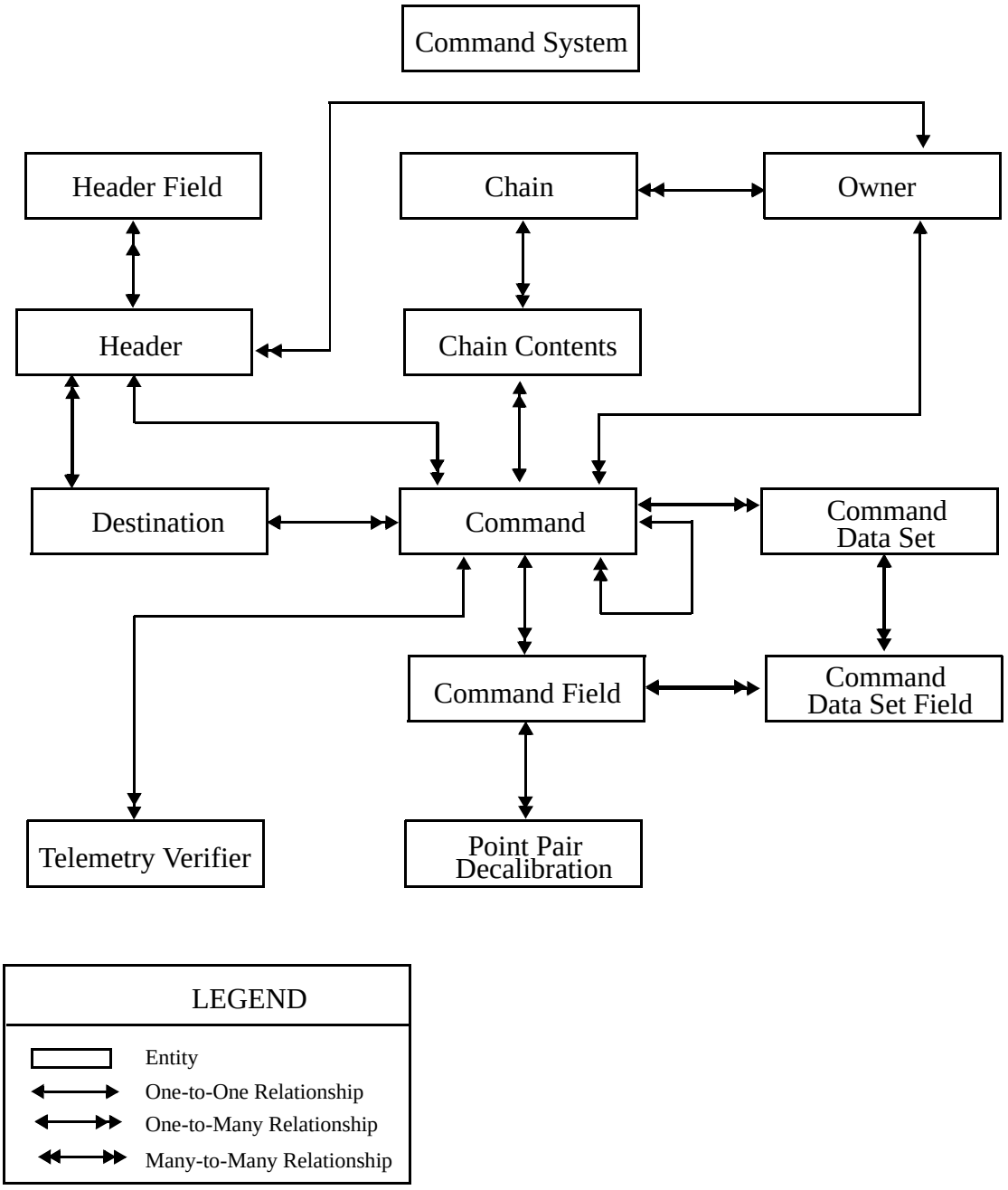

Figure 2.0-1. Command Database Entity-Relationship Diagram

2.1 COMMAND SYSTEM TABLE

The Command System table specifies the project, mission, and revision for the command database. The dates on which the database was delivered, pre-released, and baselined are also kept in this table. These data are used to accomplish configuration control of the command database.

See Table 2.1-1, Command System Table Definition, for more information.

Table 2.1-1. Command System Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
PROJECT	4	character	Organization/project name, acronym or abbreviation. "AXAF" - Advanced X-ray Astrophysics Facility "ISS" - INTERNATIONAL Space Station "SL" - Spacelab NOTE: Provided by HOSC personnel.
MISSION	4	character	Registered name, abbreviation, or acronym describing a specific mission or increment. Examples: AST1, INC2 NOTE: Provided by HOSC personnel.
REV	4	character	Revision identifier. NOTE: Provided by HOSC personnel.
DELIVER_DATE	20	timestamp	Date on which the revision is delivered. NOTE: Added by HOSC software when the database is delivered.
PRE_RELEASE_DATE	20	timestamp	Date on which the revision is pre-released. NOTE: Added by HOSC software when the database is pre-released.
BASELINE_DATE	20	timestamp	Date on which the revision is baselined. NOTE: Added by HOSC software when the database is baselined.

2.2 DESTINATION TABLE

The Destination table contains information about each onboard command destination for the system. This information includes the destination type, the maximum header length, maximum number of words per command, and word length for that destination. Note that a maximum of one of each type of destination may be defined in a single command database.

Each header defined in the Header table (see Section 2.4, Header Table) and each command defined in the Command table (see Section 2.6, Command Table) must be assigned to one of the destinations defined in the Destination table.

The destination types for the Spacelab project are attached payload (PL), Standard Serial Input/Output (SSI), and the Payload Signal Processor (PSP). For Advanced X-ray Astrophysics Facility (AXAF) and INTERNATIONAL Space Station (ISS) payloads, the destination types are AXAF and ISS, respectively.

See Table 2.2-1, Destination Table Definition, for more information.

Table 2.2-1. Destination Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
DEST_TYPE	5	character	Destination type that identifies the destination for a header or command. For AXAF: "AXAF" - Advanced X-ray Astrophysics Facility For ISS: "ISS" - INTERNATIONAL Space Station Payload For Spacelab: "PL" - Attached Payload "SSI" - Standard Serial Input/Output "PSP" - Payload Signal Processor NOTE: Must be provided by the project/ mission manager.
MAX_HEADER_LENGTH	3	integer	Maximum length in words of a command header for the specified destination. NOTE: Must be provided by the project/ mission manager.

Table 2.2-1. Destination Table Definition (Continued)

MAX_WORDS	5	integer	Maximum number of words a command may have for the specified destination. Includes header. NOTE: Must be provided by the project/ mission manager.
WORD_LENGTH	2	integer	Number of bits per word for the specified destination. NOTE: Must be provided by the project/ mission manager.

2.3 OWNER TABLE

The Owner table contains identifiers and text descriptions for each authorized owner of a header, command, or command chain. The owner of a header, command, or chain is the entity to which the ability to update the definition of the header, command, or chain has been assigned. This entity may be a spacecraft subsystem or payload, a HOSC console position, or anything else that the project wishes to use. The owner identifiers which represent each owner are used to control the ability to update the headers, commands, and chains with which they are associated. This is accomplished by associating users in the HOSC to the appropriate owner identifiers.

Suppose, for instance, that a particular mission includes a set of commands that are used to control the electrical subsystem on the spacecraft. The owner identifier “ELECTRICAL” is entered into the Owner table along with an appropriate description. Each of the commands used to control the electrical subsystem is assigned an owner identifier of “ELECTRICAL” in the Command table (see Section 2.6, Command Table). In the HOSC, groups of users that are to be given the authority to update these electrical subsystem commands are associated with this owner identifier for the mission. Only these authorized users can produce updated copies of the electrical subsystem command definitions, which may then be submitted for approval and incorporation into a subsequent revision of the database. Such approval is required since command definitions are configuration controlled.

As a second example, consider a modifiable command chain that is to be updated only by the payload commander (PAYCOM) for the mission. The owner identifier “PAYCOM” is entered into the Owner table along with an appropriate description. The command chain is assigned an owner identifier of “PAYCOM” in the Chain table (see Section 2.12, Chain Table). In the HOSC, the user who is the payload commander for the mission is associated with the “PAYCOM” owner identifier. In this way, the payload commander is granted exclusive privilege to update the contents of the modifiable command chain. No prior approval is required in this case since the contents of modifiable command chains are not configuration controlled.

Note that the owner identifier for a command chain does not necessarily have to be the same as the owner identifier for the commands contained in the chain. This allows the construction of chains containing commands that have different owner identifiers, as long as the user of the chain is associated with the owner identifiers of the chain and each of the commands in the chain.

See Table 2.3-1, Owner Table Definition, for more information.

Table 2.3-1. Owner Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
OWNER_ID	20	character	Unique identifier that indicates the owner of a header, command, or command chain. NOTE: Must be provided by the project/mission manager.
DESCRIPTION	100	character	Text description.

2.4 HEADER TABLE

The Header table contains the identifier, owner, header type, destination type, length, and text description of each command header. Each header must be identified by a unique header ID. The owner of the header is the entity to which the ability to update the header definition has been assigned (see Section 2.3, Owner Table). The header type indicates whether the header is predefined or modifiable, while the destination type identifies the onboard destination for the header. The length defined for a specific header ID must not exceed the maximum header length defined for the header's destination in the Destination table.

See Table 2.4-1, Header Table Definition, for more information.

Table 2.4-1. Header Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
HEADER_ID	20	character	Unique ID assigned to a command system header. NOTE: Must be provided by the project/mission manager.
OWNER_ID	20	character	Unique identifier that indicates the responsible and authorized owner of this header. NOTE: Must be provided by the project/mission manager.
HEADER_TYPE	1	character	Indicates whether this header is predefined or modifiable. A header is considered modifiable if at least one field in the header is modifiable. "P" - Predefined header. "M" - Modifiable header. NOTE: Must be provided by the project/mission manager
DEST_TYPE	5	character	Destination type that identifies the destination for the header. For AXAF: "AXAF" - Advanced X-ray Astrophysics Facility For ISS: "ISS" - INTERNATIONAL Space Station Payload For Spacelab: "PL" - Attached Payload "SSI" - Standard Serial Input/Output "PSP" - Payload Signal Processor NOTE: Must be provided by the project/mission manager.
LENGTH	3	integer	Indicates the total length of the header in words. NOTE: Must be provided by the project/mission manager
DESCRIPTION	100	character	Text description.

2.5 HEADER FIELD TABLE

The Header Field table contains the fields that make up the headers required by a given command system. Each header can be made up of multiple fields which are referenced back to a specific header ID, as defined in the Header table.

Header fields are all fixed length, but each field can be predefined or modifiable. Predefined fields are fields whose data values are defined in advance and may not be modified during an operational activity. All fields associated with a predefined header must be designated as predefined fields. Modifiable fields are fields whose data values may be modified or calculated during an operational activity. Some modifiable header field values are dependent upon the data defined for the command with which the header is uplinked and cannot, therefore, be defined until the uplink data for the command is built. In these cases, a calculation type must be specified for the field which indicates how the field's value is to be calculated.

Field data that are provided through the database should be entered according to the input data type specified for the field. The input data is then converted to uplink data according to the uplink data type. Fields for checksum data are not allowed in a header.

The INIT_DATA value is used as the predefined data for predefined fields and as initial data for modifiable fields. Note that the initial data which are defined for modifiable fields in this database is overwritten in the Operational Command Database (OCDB) the first time that the header is updated during an operational activity. Initial data are not required for header fields that are calculated by the HOSC software according to the specified calculation type.

Each header field must have a start position in the header. The total number of bits defined for the fields for a specific-header ID must not exceed the maximum header length defined for the header's destination in the Destination table.

See Table 2.5-1, Header Field Table Definition, for more information.

Table 2.5-1. Header Field Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
HEADER_ID	20	character	Unique ID assigned to a command system header. NOTE: Must be provided by the project/mission manager.
FIELD_MNEMONIC	20	character	Unique user-friendly mnemonic assigned to a header field in a command header. NOTE: Must be provided by the project/mission manager.
FIELD_TYPE	1	character	Indicates whether the field is predefined or modifiable. "P" - Predefined field. "M"- Modifiable field. NOTE: Must be provided by the project/mission manager.
INPUT_DATA_TYPE	1	character	Indicates the type of input data. NOTE: Must be provided by the project/mission manager.
UPLINK_DATA_TYPE	5	character	Indicates how the data will be converted before uplink (see MSFC-STD-2235). NOTE: Must be provided by the project/mission manager.
LENGTH	3	integer	Indicates the total length of the field. If the column UPLINK_DATA_TYPE indicates a string data type, then the LENGTH column contains the number of characters/bytes to be uplinked. Otherwise, The LENGTH column contains the number of bits to be uplinked. NOTE: Must be provided by the project/mission manager.
START_WORD	3	integer	The word within the header where the field begins. The first word of the header is word 1. NOTE: Must be provided by the project/mission manager.
START_BIT	2	integer	The bit within the start word where the field begins. The first bit of a word is bit 0. NOTE: Must be provided by the project/mission manager.

Table 2.5-1. Header Field Table Definition (Continued)

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
INIT_DATA	32	character	Actual command data to be located in the field specified by this record. Format must be consistent with the INPUT_ DATA_TYPE defined for the field. NOTE: Must be provided by the project/mission manager if the field is not calculated (i.e., CALC.TYPE is NULL). Not required for calculated fields (i.e., CALC.TYPE is not NULL).
CALC_TYPE	1	character	Indicates the method used to calculate the data value for a header field when that value is dependent upon the data defined for the command. For Spacelab: "L" - Value is set to the command length, as defined in the Command table (see Section 2.6). "O" - Value is set to the operations code, as defined in the Command table (see Section 2.6). NOTE: Must be provided by the project/mission manager for Spacelab if the value of the header field is dependent upon the data defined for the command and cannot, therefore, be defined until the uplink data for the command is built.
DESCRIPTION	100	character	Text description.

2.6 COMMAND TABLE

The Command table contains general information about each command in the command database. The command mnemonic uniquely identifies a command within the command database. The command mnemonic is a legal reference used to issue a command. A measurement/stimulus identifier (MSID) designates the wire destination of each command. Since multiple commands can be transmitted to a single wire, the relationship between an MSID and mnemonics is one to many. This table also specifies the owner of each command, which is the entity to which the ability to update the command definition has been assigned (see Section 2.3, Owner Table).

The Command table supports two types of commands: predefined and modifiable. Predefined commands are commands for which the data for all the component fields of the command are defined in advance and may not be modified in real time. All fields of a predefined command must be defined in the Field table of a delivered command database. Modifiable commands are commands which contain one or more component fields which may be modified during operational activities. Modifiable commands may be either fixed-length (i.e., lengths of the command fields and of the command itself are stored in the database) or variable-length. Data must be entered for predefined fields, and initial values may be entered for modifiable fields (see Section 2.7 for further information on command fields).

Each command must be assigned to a specific destination type, which identifies the onboard destination of the command. The destination types must be defined in the Destination table (see Section 2.2).

Commands may be assigned to command classes through the Command table. The command classes that are valid entries for the CLASS column are specified in Appendix H, Command Classes. An operations code (OP_CODE) is required for a Spacelab attached payload command.

The command length is also specified in the Command table. The command length is the total length in words of all the fields that make up the command. If a class is specified for a command, the length of the command cannot violate the length restrictions defined for the class (see Appendix H). If the command is designated as variable length in the VAR_LENGTH column, no length is defined.

The Command table also contains the initial state (enabled/disabled) of the command, as well as throughput type information. The throughput type helps define the processing requirements of the command. Special processing requirements of the command, such as checksum processing and cyclic redundancy code processing, may also be specified in this table, as may parity processing requirements.

Whether a command is critical or hazardous is also indicated in the Command table. A critical command is a command whose initiation and execution could possibly cause damage to a payload or spacecraft and impair the mission. A hazardous command is a command

whose initiation and execution could pose a threat to human life or the entire mission. The bit patterns of the predefined hazardous commands are compared with the bit patterns of all commands that are uplinked to ensure that hazardous commands are not being inadvertently sent.

Some commands may have identical structures with the exception of the data values for a few fields. Instead of entering the same command structure and its associated fields multiple times, a master command can be created and used as the shell for other commands of the same structure. The "copy" commands are referenced back to the master command by the MASTER_COMMAND column, which contains the command mnemonic of the master command. If a master command is specified for a command, the fields of the master defined in the Command Field table will be used to construct the copy command. Any field data that are specified for the copy command in the Command Field table will be used to override the master command fields of the same locations. Neither a master nor a copy command may be hazardous, critical, or variable length.

A timing constraint may be specified for each command which indicates how much time must elapse before the next command can be issued. This column is provided for information only and no EHS processing is done with this parameter.

Each command may have a single header assigned to it. The header will be added to the body of the command before uplink. The header for a specific command is identified in the HEADER_ID column. The header ID must correspond to a header defined in the Header table (Section 2.4).

If a command has telemetry verifiers associated with it, these verifiers should be specified in the Telemetry Verifier table (Section 2.11) and the verifier flag should be set in the Command table. A timeout value may also be specified which defines how long to continue checking the telemetry verifiers after a command has been issued.

A technical name and description may also be specified for each command in the Command table. A technical name may be thought of as a long version of the command mnemonic. The description is simply a text description of the command. Neither of these may be used to initiate a command, but they are available when doing command look-ups and as labels on EHS displays, forms, etc.

See Table 2.6-1, Command Table Definition, for more information.

Table 2.6-1. Command Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
CMD_MNEMONIC	20	character	Unique user-friendly mnemonic assigned to a command. The command mnemonic is a legal reference used by the HOSC for issuing commands. NOTE: Must be provided by the project/ mission manager.
MSID	20	character	Wire destination identifier assigned to each command. No special characters allowed. NOTE: Must be provided by the project/ mission manager.
OWNER_ID	20	character	Unique identifier that indicates the responsible and authorized owner of this command. NOTE: Must be provided by the project/ mission manager.
CMD_TYPE	1	character	Indicates whether this command is predefined or modifiable. A command is considered modifiable if at least one field in the command is modifiable. "P" - Predefined command. "M" - Modifiable command. NOTE: Must be provided by the project/ mission manager.
DEST_TYPE	5	character	Destination type that identifies the destination for the command. For AXAF: "AXAF" - Advanced X-ray Astrophysics Facility For ISS: "ISS" - INTERNATIONAL Space Station Payload For Spacelab: "PL" - Attached Payload "SSI" - Standard Serial Input/Output "PSP" - Payload Signal Processor NOTE: Must be provided by the project/ mission manager.

Table 2.6-1. Command Table Definition (Continued)

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
CLASS	10	character	Class to which a command belongs. NOTE: Must be provided by the project/ mission manager for AXAF and Spacelab SSI and PSP commands (see Appendix H for acceptable values). Not required for Spacelab attached payload commands. Not required for non-AXAF and non-Spacelab projects.
OP_CODE	3	character	Orbiter header word operations code for a Spacelab attached payload command. 003” - Two-stage Realtime Command (RTC)/Multiple Realtime Command (MRTC) “004” - Stored Program Command (SPC) “045” - Single-stage RTC/MRTC NOTE: Must be provided by the project/ mission manager for Spacelab attached payloads. Not required for Spacelab SSI or PSP commands. Not required for non-Spacelab projects.
LENGTH	5	integer	Total length in words of all the fields for a given command without header fields. Null for variable length commands. NOTE: Must be provided by the project/ mission manager if the command is not variable length.
VAR_LENGTH	1	character	Indicates whether this command is variable length. "Y" - Variable length command. "N" - Fixed length command. NOTE: Must be provided by the project/ mission manager.
INIT_STATE	1	character	Indicates the initial state of this command. "E" - Enabled. "D" - Disabled. NOTE: Must be provided by the project/ mission manager.

Table 2.6-1. Command Table Definition (Continued)

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
THRU_TYPE	1	character	Indicates the throughput type associated with this command. "N" - No throughput type. "S" - Indicates Spacelab Single-Stage throughput. "T" - Indicates Spacelab Two-Stage downlinked for ground verification prior to throughput. NOTE: Must be provided by the project/ mission manager. Value should be "N" for non-Spacelab projects.
HAZARDOUS	1	character	Indicates whether this command is hazardous or not. "Y" - Hazardous command. "N" - Non-hazardous command. NOTE: Must be provided by the project/ mission manager.
CRITICAL	1	character	Indicates whether this command is critical or not. "Y" - Critical command. "N" - Non-critical command. NOTE: Must be provided by the project/ mission manager.
MASTER_COMMAND	20	character	User-friendly mnemonic of a command to be used as a command shell for this command. The master command is used as a shell to be copied with the exception of overridden fields.

Table 2.6-1. Command Table Definition (Continued)

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
SPECIAL_PROCESS	1	character	Indicates the type of special processing for this command. For Spacelab: "S" - No special processing. "X" - Exclusive-OR Checksum processing. "N" - Summation Checksum processing For AXAF: "S" - No special processing. "C" - Cyclic Redundancy Code (CRC) processing. NOTE: Must be provided by the project/ mission manager. Value should be "S" for projects other than Spacelab and AXAF.
PARITY_PROCESS	1	character	Indicates type of parity processing, if any, for this command. For AXAF: "1" Odd parity calculated over 31 bits (ground bits 9-39)
TIME_CONSTRAINT	14	floating point	Number of seconds that must elapse after issuing this command before another command can be issued. Includes decimal point, if applicable.
HEADER_ID	20	character	Identifier for header to be used for this command. NOTE: Must be provided by the project/ mission manager.
TLM_VERIF_FLAG	1	character	Indicates whether or not this command has telemetry verifiers associated with it. "Y" - Yes. "N" - No. NOTE: Must be provided by the project/ manager
TLM_VERIF_TIMEOUT	14	floating point	Number of seconds to continue checking telemetry verifiers after a command has been issued. Includes decimal point, if applicable. NOTE: Must be provided by the project/ mission manager.

Table 2.6-1. Command Table Definition (Continued)

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
TECHNICAL_NAME	89	character	A unique technical name assigned to the command.
DESCRIPTION	250	character	Text description.

2.7 COMMAND FIELD TABLE

A command is composed of one or more command fields. The Field table contains information defining those fields. A command field is uniquely identified by a field mnemonic and its corresponding command mnemonic. The relationship between command and command fields is one to many.

Command fields may be defined as either predefined or modifiable. Predefined fields are fields whose data values are defined in advance and may not be modified during an operational activity. All fields associated with a predefined command must be designated as predefined fields. Modifiable fields are fields whose data values may be modified during an operational activity. Modifiable fields may be of fixed or variable length. Only one variable length field may be defined for a single command, and that variable-length field must be positioned as the last field in the command. Initial data may be entered for modifiable fields.

The data contained in a command field are described by an input data type, an uplink data type, a length, and a location within the command. Field data that is provided through the database should be entered according to the input data type specified for the field. The input data type is then converted to uplink data according to the uplink data type. Legal input type to uplink type conversions are specified in Appendix I, Input and Uplink Data Types. Fields for checksum data will not have an input data type since these data are calculated prior to uplink and are never stored in the database. The length of the field is specified in either bytes or bits, depending on the uplink data type. If the field is designated as variable length in the VAR_LENGTH column, no length is defined. The location of the field within the command is defined by a start word and a start bit.

Each command field may have a dimension and engineering units associated with it. Values are also provided to define the valid range for the field in engineering units. The data provided for the field should be consistent with the engineering units and associated range of values.

Field data can be decalibrated into raw counts before uplink using either point pair or polynomial decalibration. Decalibration is the process of using calibration point pairs or polynomials to solve for a value in counts. For example, if a field is to represent a temperature value, the user may want to enter the value in degrees Celsius, but the onboard device expects the value in raw counts. The field data must be converted to raw counts before uplink. This conversion can be accomplished using either polynomial or point pair decalibration. The type of decalibration used is indicated by the decalibration type.

If point pair decalibration is used for the field, the point pairs used in the decalibration are defined in the Point Pair Decalibration table (see Section 2.8).

If polynomial decalibration is used, the calibration coefficients of the polynomial used for the decalibration are defined in the Command Field table. The coefficients are CAL_COEF_0 and CAL_COEF_1 where *FieldData = (CAL_COEF_1xCounts) + CAL_COEF_0.* Notice that the field data is a function of the raw counts value. However, the polynomial will be used

to decalibrate, or solve for counts. Therefore, only first-order polynomials are allowed to prevent ambiguous results in the decalibration process. As an example, consider a command that is used to set the temperature of a heater. The command might have a command mnemonic of SETHTR. The modifiable field for the SETHTR command might be named TEMP. A person may want to enter the value for TEMP in degrees Celsius on the display, but this value must be converted to raw counts before uplink. The polynomial defined by the calibration coefficients in this table would be used to decalibrate the degrees Celsius value into raw counts. The equation would be used as follows: *CelsiusDegrees = (CAL_COEF_1xCounts) + CAL_COEF_0*. The decalibration would be accomplished by solving for the *Counts* value. Only fields that have an input data type of "D" (decimal) or "G" (scientific notation) can be decalibrated.

The actual data to be placed in each command field are also maintained in the Command Field table. A data value is required for predefined fields (except for checksum fields) and may not be changed during operational activities. A data value is optional for modifiable fields and, if provided, is considered initial data and may be modified during operational activities. The data provided must be consistent with the input data type specified for the field, as well as the engineering units and valid range of values.

See Table 2.7-1, Command Field Table Definition, for more information.

Table 2.7-1. Command Field Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
CMD_MNEMONIC	20	character	Unique user-friendly mnemonic assigned to the command to which the field belongs. NOTE: Must be provided by the project/mission manager.
FIELD_MNEMONIC	20	character	User-friendly mnemonic for the command field. NOTE: Must be provided by the project/mission manager.
FIELD_TYPE	1	character	Indicates whether the field is predefined or modifiable. "P" - Predefined field. "M" - Modifiable field. NOTE: Must be provided by the project/mission manager.
INPUT_DATA_TYPE	1	character	Indicates type of input data. NOTE: Must be provided by the project/mission manager.

Table 2.7-1. Command Field Table Definition (Continued)

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
UPLINK_DATA_TYPE	5	character	Indicates how data will be converted before uplink (see MSFC-STD-2235). NOTE: Must be provided by the project/mission manager.
ENG_UNIT	10	character	Indicates the engineering units associated with the field.
DIMENSION	30	character	Indicates the physical property associated with the engineering units.
DECAL_TYPE	2	character	Indicates the type of decalibration associated with a command. "N" - No decalibration. "PC" - Polynomial Coefficient decalibration. "PP" - Point Pair decalibration. NOTE: Must be provided by the project/mission manager.
LENGTH	5	integer	Indicates length of the field. If the column UPLINK_DATA_TYPE indicates a string data type, then the LENGTH column contains the number of characters/bytes to be uplinked. Otherwise, the LENGTH column contains the number of bits to be uplinked. Null for variable length fields. NOTE: Must be provided by the project/mission manager.
VAR_LENGTH	1	character	Indicates whether this command field is variable length. "Y" - Variable length field. "N" - Fixed length field. NOTE: Must be provided by the project/mission manager.

Table 2.7-1. Command Field Table Definition (Continued)

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
START_WORD	3	integer	The word within the command where the field begins. The first word of the command is word 1. The command header is not included. NOTE: Must be provided by the project/mission manager.
START_BIT	2	integer	The bit within the start word where the field begins. The first bit of a word is bit 0. NOTE: Must be provided by the project/mission manager.
INIT_DATA	128	character	Actual command data to be located in the field specified by this record. Format must be consistent with the INPUT_ DATA_TYPE defined for the field. NOTE: Must be provided by the project/mission manager if the command field is predefined (i.e., FIELD_TYPE = “P”) and the UPLINK_DATA _TYPE is not “ICHK.” Optional for modifiable fields (i.e., FIELD_TYPE = “M”). Not required for checksum fields (i.e., UPLINK_DATA _TYPE = “ICHK.”
RANGE_LOW	16	floating point	Low end of range in engineering units for valid command field value. Includes sign and decimal point, if applicable.
RANGE_HIGH	16	floating point	High end of range in engineering units for valid command field value. Includes sign and decimal point, if applicable.
CAL_COEF_0	16	scientific notation	Calibration Coefficient 0. Used for polynomial decalibration. NOTE: Must be provided by the project/mission manager if the decalibration type is “PC.”
CAL_COEF_1	16	scientific notation	Calibration Coefficient 1. Used for polynomial decalibration. NOTE: Must be provided by the project/mission manager if the decalibration type is “PC.”
DESCRIPTION	100	character	Text description.

2.8 POINT PAIR DECALIBRATION TABLE

The Point Pair Decalibration table is used to specify calibration point pairs to be used to decalibrate an input field value into raw counts for uplink. For example, consider a field that is to represent a voltage value. The user on the ground may want to enter the value in volts AC, but the onboard device expects the value in raw counts. The field data must be converted to raw counts before uplink, but the necessary conversion is such that it cannot be represented by a first order polynomial (see Section 2.7). Such conversions can be accomplished by using known calibration point pairs to interpolate the uplink value from the input value. In our example, each point pair consists of a raw counts value and the corresponding volts AC value, such as are shown in Table 2.8-1. These values would be stored in the COUNTS and VALUE columns of the Point Pair Decalibration table, respectively.

Table 2.8-1. Point Pairs for the Volts AC to Counts Example

Counts	Volts AC
0	0
50	2.25
200	3.0
255	5.75

This set of data would be represented graphically as shown in Figure 2.8-1. Notice that the graph indicates that the volts AC value is a function of the raw counts value, since the point pairs provided are calibration point pairs. However, these points can be used to interpolate a counts value given a volts AC value, as described below.

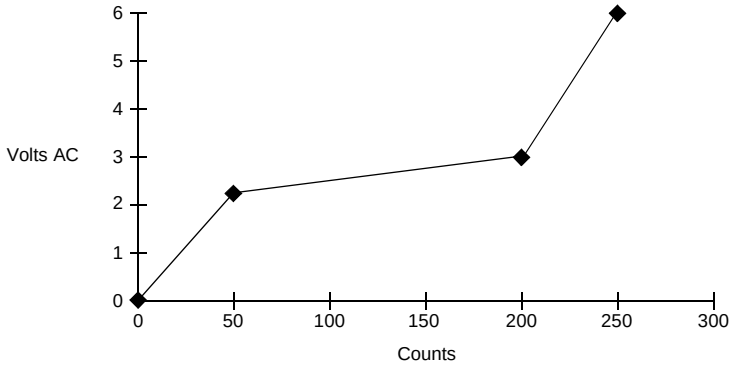

Figure 2.8-1. Graph of Points for the Volts AC to Counts Example

When a user enters a volts AC value for uplink, its corresponding value in raw counts is interpolated using the two point pairs between which it falls. For instance, using the values given above, the raw counts value corresponding with 2.7 volts AC would be calculated as follows:

1. The value of 2.7 volts AC falls between 2.25 and 3.0 volts AC, for which the raw count values are known to be 50 and 200, respectively. Therefore let:

$$V_1 = 2.25 \qquad V_2 = 3.0$$
$$C_1 = 50 \qquad C_2 = 200$$

2. Solve for the raw count value corresponding to 2.7 volts AC as follows:

$$C = C_1 + \frac{C_2 - C_1}{V_2 - V_1}(2.7 - V_1)$$
$$= 50 + \frac{200 - 50}{3.0 - 2.25}(2.7 - 2.25)$$
$$C = 140$$

Note that only fields that have an input data type of D (Decimal) or G (Scientific Notation) can be decalibrated. Furthermore, the graph described by the point pairs must be continuously increasing or decreasing to avoid ambiguous results in the decalibration process. That is, the slopes of all line segments described by the points must be of the same sign, and none of the line segments can have a slope of zero (see Figures 2.8-2 and 2.8-3). If the value to be decalibrated is in one of the defined point pairs, then the raw count value is equal to the defined raw count value, and solving for the raw count value is not necessary. Also, any points outside the range of the defined point pairs are considered undefined.

See Table 2.8-2, Point Pair Decalibration Table Definition, for more information.

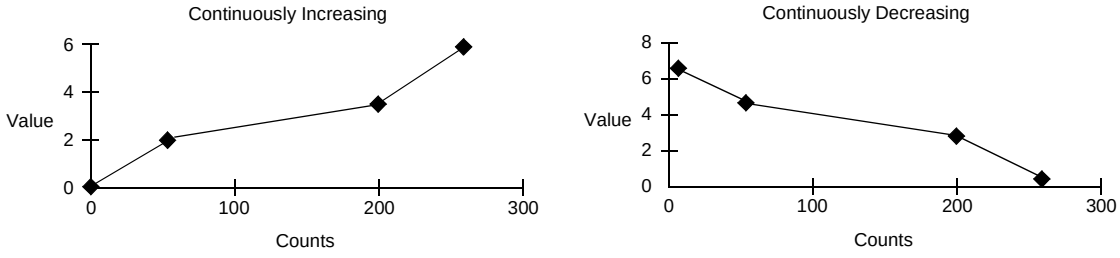

Figure 2.8-2. Examples of Points that are Valid for Decalibration

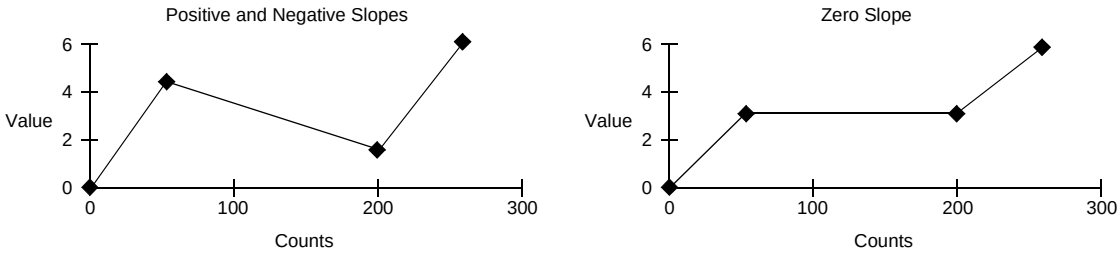

Figure 2.8-3. Examples of Points that are Invalid for Decalibration

Table 2.8-2. Point Pair Decalibration Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
CMD_MNEMONIC	20	character	Unique user-friendly mnemonic of a command to which field referenced by FIELD_ MNEMONIC belongs. NOTE: Must be provided by the project/mission manager.
FIELD_MNEMONIC	20	character	Unique user-friendly mnemonic of a command field to which point pair applies. NOTE: Must be provided by the project/mission manager.
COUNTS	12	integer	Decimal integer representing the raw count value of the point pair. NOTE: Must be provided by the project/mission manager.
VALUE	16	floating point	Engineering unit equivalent of the COUNTS column. Includes sign and decimal point, if applicable. NOTE: Must be provided by the project/mission manager.

2.9 TELEMETRY VERIFIER TABLE

The Telemetry Verifier table contains information used to specify what telemetry should be checked to verify that a command has executed correctly. If multiple measurements are to be verified for a command, the positive verifications for the measurements will be logically "ANDed" to determine success or failure of the command. The telemetry MSID of each measurement to be used for each command must be specified. The state code or range of values specified for each measurement is what the measurement is checked against to determine if the command completed successfully.

For each specified telemetry measurement either an expected state code or a range of engineering unit values can be specified, but not both.

See Table 2.9-1, Telemetry Verifier Table, for more information.

Table 2.9-1. Telemetry Verifier Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
CMD_MNEMONIC	20	character	Unique user-friendly mnemonic for a command. NOTE: Must be provided by the project/ mission manager.
TLM_MSID	20	character	Unique identifier assigned to the telemetry measurement to be checked for a command's verification. NOTE: Must be provided by the project/ mission manager.
STATE_CODE	12	character	State conversion code of the measurement identified by TLM_MSID which indicates positive command verification. NOTE: Must be provided by the project/ mission manager if range-low and range-high values are not provided.

Table 2.9-1. Telemetry Verifier Table Definition (Continued)

RANGE_LOW	16	floating point	Low end of range in engineering units for the measurement identified by TLM_MSID. If measurement value between this value and RANGE_HIGH, the verification is positive. Includes sign and decimal point, if applicable. NOTE: Must be provided by the project/ mission manager if a state code value is not provided.
RANGE_HIGH	16	floating point	High end of range in engineering units for the measurement identified by TLM_MSID. Includes sign and decimal point, if applicable. NOTE: Must be provided by the project/ mission manager if a state code value is not provided.

2.10 CHAIN TABLE

Commands can be grouped together into a sequence that is referred to as a command chain. Command chains can be used to automate the sequential uplinking of a group of commands or just to associate certain commands together. The Chain table defines the owner of the chain, whether a given chain is predefined or modifiable, and the initial state of the chain.

The owner of a chain is the entity to which the ability to update the chain definition has been assigned (see Section 2.3, Owner Table).

The contents of a predefined chain are configuration controlled and cannot be modified without approval of the DBCG. The contents of a modifiable chain, however, are not configuration controlled. The commands within a modifiable chain may be changed or rearranged by an authorized user, although the information in the Chain table must remain under the configuration control of the DBCG.

See Section 2.11 for more information on chain contents (see Table 2.10-1, Chain Table Definition, for more information).

Table 2.10-1. Chain Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
CHAIN_NAME	20	character	Unique name assigned to a command chain. NOTE: Must be provided by the project/mission manager.
OWNER_ID	20	character	Unique identifier that indicates the responsible and authorized owner of the chain. NOTE: Must be provided by the project/mission manager.
CHAIN_TYPE	1	character	Indicates whether this chain is predefined or modifiable. “P” - Predefined chain. “M” - Modifiable chain. NOTE: Must be provided by the project/mission manager
INIT_STATE	1	character	Indicates the initial state of this command chain. “E” - Enabled. “D” - Disabled. NOTE: Must be provided by the project/mission manager.
DESCRIPTION	100	character	Text description of the Chain.

2.11 CHAIN CONTENTS TABLE

Commands can be grouped together into a sequence that is referred to as a command chain. Command chains can be used to automate the sequential uplinking of a group of commands or to associate certain commands together. The Chain Contents table defines the commands contained in each command chain and their positions in the chain. A command in a chain is uniquely identified by the chain name and the command's position within the chain, thus allowing a single command to be included multiple times in a chain.

Since the contents of a modifiable chain are not configuration controlled, they may not be submitted from an external source for direct incorporation into the Command Database. This policy prevents conflicts between non-configuration-controlled data that is defined or modified by EHS users and that submitted from an external source. Since it is assumed that the data defined or modified by the users who will use the data is more recent and more relevant than data submitted from an external source, non-configuration-controlled data cannot be loaded directly from an external source.

Modifiable chain contents may be incorporated into an OCDB from a Ground Support Equipment (GSE) file, however, using the Command Operation application, as described in Section 6.2.5 of the EHS Concepts and Scenarios Document (HOSC-EHS-065). Once this is done, the chain contents may be stored into the Command Database from the OCDB using the capabilities of the Command Subsystem.

See Table 2.11-1, Chain Contents Table Definition, for more information.

Table 2.11-1. Chain Contents Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
CHAIN_NAME	20	character	Unique name assigned to a command chain. NOTE: Must be provided by the project/ mission manager.
CMD_POSITION	4	integer	Position of the command referenced by the command mnemonic within the command chain. NOTE: Must be provided by the project/ mission manager.
CMD_MNEMONIC	20	character	Unique user-friendly mnemonic of a command contained in the command chain. NOTE: Must be provided by the project/ mission manager.

2.12 COMMAND DATA SET TABLE

The Command Data Set table is used in combination with the Command Data Set Field table to define value sets for modifiable commands. For any given modifiable command, command data sets may be defined which specify values for modifiable fields in the command. Each data set is uniquely identified by a command mnemonic and a data set name.

A command data set and its fields are not configuration controlled. Command data sets may be created, modified, and deleted by an authorized user without approval of the DBCG. Since command data sets are not configuration controlled, they may not be submitted from an external source for direct incorporation into the Command Database. This policy prevents conflicts between non-configuration-controlled data that is defined or modified by EHS users and that submitted from an external source. Since it is assumed that the data defined or modified by the users who will use the data is more recent and more relevant than data submitted from an external source, non-configuration-controlled data cannot be loaded directly from an external source.

Command data sets may be incorporated into an OCDB from a GSE file, however, using the Command Operation application, as described in Section 6.2.5 of the EHS Concepts and Scenarios Document (HOSC-EHS-065). Once this is done, the command data set may be stored into the Command Database from the OCDB using the capabilities of the Command Subsystem.

See Table 2.12-1, Command Data Set Table Definition, for more information.

Table 2.12-1. Command Data Set Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
CMD_MNEMONIC	20	character	Unique user-friendly mnemonic of a command.
DATA_SET_NAME	20	character	Name assigned to a command data set. Unique for each data set for a particular command mnemonic.

2.13 COMMAND DATA SET FIELD TABLE

The Command Data Set Field table is used in combination with the Command Data Set table to define value sets for modifiable commands. For any given modifiable command, command data sets may be defined which specify values for modifiable fields in the command.

Each data set is uniquely identified by a command mnemonic and a data set name. The fields within each data set are identified by the corresponding command field mnemonics. Each modifiable field in the command must be represented in the command data set. The value for each field in the data set is defined in this table and must be consistent with the input data type specified for the corresponding command field in the Command Field table.

Since the fields of command data sets are not configuration controlled, they may not be submitted from an external source for direct incorporation into the Command Database. This policy prevents conflicts between non-configuration-controlled data that is defined or modified by EHS users and that submitted from an external source. Since it is assumed that the data defined or modified by the users who will use the data is more recent and more relevant than data submitted from an external source, non-configuration-controlled data cannot be loaded directly from an external source.

Command data set fields may be incorporated into an OCDB from a GSE file, however, using the Command Operation application, as described in Section 6.2.5 of the EHS Concepts and Scenarios Document (HOSC-EHS-065). Once this is done, the command data set fields may be stored into the Command Database from the OCDB using the capabilities of the Command Subsystem.

See Table 2.13-1, Command Data Set Field Table Definition, for more information.

Table 2.13-1. Command Data Set Field Table Definition

COLUMN NAME	MAXIMUM COLUMN LENGTH	COLUMN TYPE	DESCRIPTION
CMD_MNEMONIC	20	character	Unique user-friendly mnemonic of a command.
DATA_SET_NAME	20	character	Name assigned to a command data set. Unique for each data set for a particular command mnemonic.
FIELD_MNEMONIC	20	character	User-friendly mnemonic for a command field.
DATA	128	character	Actual command data to be located in the field. Format must be consistent with the INPUT_DATA_TYPE defined for the field in the Command Field table (see Section 2.7).

THIS PAGE INTENTIONALLY BLANK

3.0 HOSC STANDARD INPUT FORMATS

The following sections define the formats in which the data defined in Section 2.0 of this document may be submitted to the MSFC HOSC.

3.1 DELETED

3.2 ASCII TEXT FILES

If access to ORACLE or the Export utility is not available, ASCII text files may be submitted to populate the tables of the HOSC database. The rules for formatting these ASCII text files are as follows:

1. A separate file must be submitted for each table in Section 2.0 of this document, or a subset thereof, as agreed upon by the HOSC and the Project/Mission Manager.
2. The data for each row in a table must be terminated by a semicolon (;).
3. Each data value must be delimited by a comma (,).
4. *White space* is defined as one or more spaces, or horizontal tab characters. White space between data values is ignored.
5. If a data value contains white space, comma, or semicolon characters, then it must be enclosed in quotation marks (").
6. If a data value contains quotation marks, then double quotation marks (““) must be used. When the data value is inserted into the database, the extra quotation marks will be removed from the data.
7. The ASCII text file can contain carriage control and line-feed characters between data values and embedded within a data value. In other words, a text editor can be used to create an ASCII text file such that a single data value can *line wrap*. The carriage control and line-feed characters will be removed before the data are inserted into the database.
8. Blank lines between data values are ignored.

The Project/Mission Manager must also provide a list which specifies the actual name of the submitted file that corresponds to each applicable table in Section 2.0 of this document. This list should include a record for each table listed in Table 2.0-1 of this volume. Each record will consist of the table name as it appears in the “TABLE NAME” column of Table 2.0-1, followed by a comma (,) and the actual name of the ASCII file corresponding to that table. Spaces or tabs after the comma are optional. For optional tables that are not being provided, the value “n/a” should be used in place of the file name to indicate that the table is not included. For example:

```
Command Field, cmd_field.txt
Point Pair Decalibration, n/a
```

This list should be provided in an ASCII text file with the name CDBLIST.TXT.

THIS PAGE INTENTIONALLY LEFT BLANK

4.0 VALIDATION RULES

The following sections specify the validation rules for the database tables defined in Section 2.0 of this volume. These are the rules which will be applied by the HOSC software to any data that is submitted for a HOSC command database. The validation rules are divided into three categories: database level, table level, and column level.

Database level validation rules are those rules that involve data in more than one table. The database level rules for this volume are defined in Section 4.1.

Table level validation rules are those rules that involve the data in multiple columns of a single table. The table level validation rules for the tables defined in this volume are defined in Section 4.2.

Finally, column level validation rules are those rules that apply to the data in a specific column of a single table. The column level validation rules for the tables defined in this volume are defined in Section 4.3.

Notice that primary key and foreign key rules are specified in Section 2.0 of this volume and are not repeated in the following sections. These rules will also be applied by the HOSC software to any data that are submitted for a HOSC command database.

4.1 DATABASE LEVEL VALIDATION RULES

Database level validation rules are those rules that involve data in more than one table. The foreign key rules that are specified in Section 2.0 of this volume are database level rules. However, some database level validation rules cannot be expressed as foreign keys because they represent relationships between the non-key columns of two or more tables. Other rules may not represent a relationship between tables but may apply to many different columns that contain the same type of data. These columns may be in the same table or in different tables. A validation rule of this type can be specified as a database level rule to avoid the redundant expression of the rule for each column to which it applies. The database level rules for this volume are defined in Table 4.1-1.

Notice that foreign key rules are specified in Section 2.0 of this volume and are not repeated in Table 4.1-1. These rules will also be applied by the HOSC software to any data that are submitted for a HOSC command database.

Table 4.1-1. Database Level Validation Rules for the Command Database

	VALIDATION RULE
1.	Column type “character” must conform to the rules specified in Appendix B.
2.	Column type “integer” must conform to the rules specified in Appendix B.
3.	Column type “scientific notation” must conform to the rules specified in Appendix B.
4.	Column type “floating point” must conform to the rules specified in Appendix B.
5.	Column type “timestamps” must conform to the rules specified in Appendix B.
6.	If the PROJECT value in the Command System table is ‘SL,’ the DEST_TYPE value in the Destination table must be ‘PL,’ ‘PSP,’ or ‘SSI.’
7.	If the PROJECT value in the Command System table is ‘SL,’ the OWNER_ID value in the Owner table must be a two-digit numeric (00-99).
8.	The LENGTH value for a specific HEADER_ID in the Header table must be equal to the sum of the LENGTH values of all the header fields defined for that HEADER_ID in the Header Field table.
9.	The LENGTH value for a specific HEADER_ID in the Header table must be less than or equal to the MAX_HEADER_LENGTH value defined in the Destination table for the DEST_TYPE value assigned to the HEADER_ID in the Header table.
10.	The START_BIT value for a specific HEADER_ID in the Header Field table must be less than the WORD_LENGTH value defined in the Destination table for the DEST_TYPE that is assigned to the HEADER_ID in the Header table.
11.	If the PROJECT value in the Command System table is ‘SL,’ the CMD_MNEMONIC value in the Command table must conform to the following restrictions: a) May contain up to 10 characters consisting of alphanumeric characters (A-Z, 0-9), plus sign (+), and minus sign (-). Note that alphabetic characters must be uppercase. b) First two characters must be numeric (01-99). c) Must not contain embedded blanks.
12.	Deleted.
13.	The CMD_TYPE value for a specific CMD_MNEMONIC in the Command table must be ‘M’ if the FIELD_TYPE value of at least one of the command fields defined for that CMD_MNEMONIC in the Command Field table is ‘M.’
14.	If the HEADER_ID value for a specific CMD_MNEMONIC in the Command table is not NULL, the DEST_TYPE value for that CMD__MNEMONIC in the Command table must match the DEST_TYPE value in the Header table for the HEADER_ID that is assigned to the CMD_MNEMONIC in the Command table.
15.	If the VAR_LENGTH value is ‘N’ and the MASTER_COMMAND value is NULL for a specific CMD_MNEMONIC in the Command table, the LENGTH value for that CMD_MNEMONIC in the Command table must be equal to the sum (in words) of the LENGTH values of all the command fields defined for that CMD_MNEMONIC in the Command Field table.
16.	The VAR_LENGTH value for a specific CMD_MNEMONIC in the Command table must be ‘Y’ if the VAR_LENGTH value for the last command field of that CMD_MNEMONIC in the Command Field table is ‘Y’ (variable-length field). Otherwise, the VAR_LENGTH value for the CMD_MNEMONIC in the Command table must be ‘N.’
17.	If the PROJECT value in the Command System table is not ‘SL,’ the THRU_TYPE value in the Command table must be ‘N.’
18.	If the PROJECT value in the Command System table is ‘SL,’ the SPECIAL_PROCESS value in the Command table must be ‘S,’ ‘X,’ or ‘N.’
19.	If the PROJECT value in the Command System table is ‘AXAF,’ the SPECIAL_PROCESS value in the Command table must be ‘S’ or ‘C.’

Table 4.1-1. Database Level Validation Rules for the Command Database (Continued)

	VALIDATION RULE
20.	If the PROJECT value in the Command System table is not 'SL' and not 'AXAF,' the SPECIAL_PROCESS value in the Command table must be 'S.'
21.	If at least one telemetry verifier is defined for a specific CMD_MNEMONIC in the Telemetry Verifier table, the TLM_VERIF_FLAG value for that CMD_MNEMONIC in the Command table must be 'Y.' Otherwise, the TLM_VERIF_FLAG value for the CMD_MNEMONIC in the Command table must be 'N.'
22.	If the DECAL_TYPE value for a specific CMD_MNEMONIC and FIELD_MNEMONIC in the Command Field table is 'PP,' then at least two corresponding point pairs must be defined for the specified CMD_MNEMONIC and FIELD_MNEMONIC in the Point Pair Decalibration table.
23.	The START_WORD value for a specific CMD_MNEMONIC in the Command Field table must be less than or equal to the MAX_WORDS value defined in the Destination table for the DEST_TYPE that is assigned to the CMD_MNEMONIC in the Command table.
24.	The START_BIT value for a specific CMD_MNEMONIC in the Command Field table must be less than the WORD_LENGTH value defined in the Destination table for the DEST_TYPE that is assigned to the CMD_MNEMONIC in the Command table.
25.	Deleted.
26.	Deleted.
27.	Deleted.
28.	If a data set is defined for a specific CMD_MNEMONIC in the Command Data Set table, the CMD_TYPE value for that CMD_MNEMONIC in the Command table must be 'M.'
29.	If a data set field is defined for a specific CMD_MNEMONIC and FIELD_MNEMONIC in the Command Data Set Field table, the FIELD_TYPE value for the specified CMD_MNEMONIC and FIELD_MNEMONIC in the Command Field table must be 'M.'
30.	The DATA value for a specific CMD_MNEMONIC and FIELD_MNEMONIC in the Command Data Set Field table must be consistent with the INPUT_DATA_TYPE defined for the specified CMD_MNEMONIC and FIELD_MNEMONIC in the Command Field table.
31.	The HEADER_TYPE value for a specific HEADER_ID in the Header table must be 'M' if the FIELD_TYPE value of at least one of the header fields defined for that HEADER_ID in the Header Field table is 'M.'
32.	If the PROJECT value in the Command System table is 'SL,' a CLASS value or OP_CODE value must be provided in the Command table.
33.	The HEADER_TYPE value for a specific HEADER_ID in the Header table must be 'P' if the FIELD_TYPE values of all of the header fields defined for that HEADER_ID in the Header Field table are 'P.'
34.	The CMD_TYPE value for a specific CMD_MNEMONIC in the Command table must be 'P' if the FIELD_TYPE values of all of the command fields defined for that CMD_MNEMONIC in the Command Field table are 'P.'
35.	Deleted.
36.	If any of the CMD_MNEMONICs defined for a CHAIN_NAME in the Chain Contents table has a HAZARDOUS value 'Y' in the Command table, then the INIT_STATE value for that CHAIN_NAME in the Chain table must be 'D.'
37.	If the PROJECT value in the Command System table is 'AXAF,' the DEST_TYPE value in the Destination table must be 'AXAF.'
38.	If the PROJECT value in the Command System table is 'ISS,' the DEST_ TYPE value in the Destination table must be 'ISS.'

Table 4.1-1. Database Level Validation Rules for the Command Database (Continued)

	VALIDATION RULE
39.	The converted DATA value for a specific CMD_MNEMONIC and FIELD_MNEMONIC in the Command Data Set Field table must fit into the LENGTH defined for that CMD_MNEMONIC and FIELD_MNEMONIC in the Command Field table.
40.	The converted DATA value for a specific CMD_MNEMONIC and FIELD_MNEMONIC in the Command Data Set Field table must be an integer value if the UPLINK_DATA_TYPE for that CMD_MNEMONIC and FIELD_MNEMONIC in the Command Field table is 'IMAG,' 'ITWO,' or 'IUNS.'
41.	If the VAR_LENGTH value in the Command table is 'Y' for a specific CMD_MNEMONIC, there can only be one modifiable field defined for that a CMD_MNEMONIC in the Command Field table, and that modifiable field must be the variable-length field.
42.	If the VAR_LENGTH value in the Command table is 'Y' for a specific CMD_MNEMONIC, there can be no command fields for that CMD_MNEMONIC in the Command Field table, with an UPLINK_DATA_TYPE value of 'ICHK.'
43.	If the VAR_LENGTH value is 'Y; and an INIT_DATA value is provided for a specific CMD_MNEMONIC and FIELD_MNEMONIC in the Command Field table, the length of the converted INIT_DATA value for that CMD_MNEMONIC and FIELD_MNEMONIC must be a multiple of the WORD_LENGTH defined in the Destination table for the DEST_TYPE specified for the CMD_MNEMONIC in the Command table.
44.	If the VAR_LENGTH value in the Command table is 'Y' for a specific CMD_MNEMONIC and the command is complete (i.e., an INIT_DATA value is provided for the variable-length field for the CMD_MNEMONIC in the Command Field table), the sum of the length of the converted INIT_DATA value for the variable-length field and the LENGTH values of the other fields for the CMD_MNEMONIC in the Command Field table must be a positive multiple of the WORD_LENGTH defined in the Destination table for the DEST_TYPE specified for the CMD_MNEMONIC in the Command table.
45.	If the VAR_LENGTH value is 'N' and the HEADER_ID value is not NULL for a specific CMD_MNEMONIC in the Command table, the sum of the LENGTH value for the CMD_MNEMONIC in the Command table and the LENGTH value in the Header table for the HEADER_ID specified in the Command table for the CMD_MNEMONIC must be less than or equal to the MAX_WORDS value specified in the Destination table for the DEST_TYPE defined for the CMD_MNEMONIC in the Command table.
46.	If the VAR_LENGTH value in the Command table is 'Y' for a specific CMD_MNEMONIC and the command is complete (i.e., an INIT_DATA value is provided for the variable-length field for the CMD_MNEMONIC in the Command Field table), the sum (in words) of the length of the converted INIT_DATA value for the variable-length field, the LENGTH values of other fields for the CMD_MNEMONIC in the Command Field table, and the LENGTH value in the Header table for the HEADER_ID specified in the Command table for the CMD_MNEMONIC must be less than or equal to the MAX_WORDS value specified in the Destination table for the DEST_TYPE defined for the CMD_MNEMONIC in the Command table.
47.	If the CALC_TYPE value for a specific HEADER_ID and FIELD_MNEMONIC in the Header Field table is 'L,' the difference (in bits) between the MAX_WORDS value in the Destination table for the DEST_TYPE value defined for the HEADER_ID in the Header table and the LENGTH value defined for the HEADER_ID in the Header table, when converted to uplink format, must fit into the length defined by the LENGTH value for the HEADER_ID and FIELD_MNEMONIC in the Header Field table.
48.	If the MASTER_COMMAND value is not NULL for a specific CMD_MNEMONIC in the Command table, the LENGTH, START_WORD, and START_BIT values of each FIELD_MNEMONIC for that CMD_MNEMONIC in the Command Field table must be equal to the LENGTH, START_WORD, and START_BIT values for the same FIELD_MNEMONIC for the MASTER_COMMAND in the Command Field table.

Table 4.1-1. Database Level Validation Rules for the Command Database (Continued)

	VALIDATION RULE
49.	If the CALC_TYPE value for a specific HEADER_ID and FIELD_MNEMONIC in the Header Field table is 'O,' an OP_CODE value must be specified for any CMD_MNEMONIC in the Command table that is associated with that HEADER_ID.
50.	If the CALC_TYPE value for a specific HEADER_ID and FIELD_MNEMONIC in the Header Field table is 'L,' the DEST_TYPE value for the specified HEADER_ID in the Header table must be 'SSI' or 'PSP.'
51.	If the CALC_TYPE value for a specific HEADER_ID and FIELD_MNEMONIC in the Header Field table is 'O,' the DEST_TYPE value for the specified HEADER_ID in the Header table must be 'PL.'
52.	If the UPLINK_DATA_TYPE value for a specific CMD_MNEMONIC and FIELD_MNEMONIC in the Command Field table is 'IPAR' and the PROJECT value in the Command System table is 'AXAF', then the PARITY_PROCESS value of that same command in the Command table must be '1'.
53.	If the DEST_TYPE value is 'AXAF' and the CLASS value is 'S' for a specific CMD_MNEMONIC in the Command table, that CMD_MNEMONIC must not be included in the contents of any command chain as defined in the Chain Contents table.
54.	If the DEST_TYPE value is 'AXAF' and the CLASS value is 'S' for a specific CMD_MNEMONIC in the Command table, no command data sets can be defined for that CMD_MNEMONIC in the Command Data Set table.
55.	If the PROJECT value in the Command System table is 'AXAF', the Header table must contain headers with the following HEADER_ID values: 'OCMDHDR' and 'HWCMDHDR'.
56.	If the PROJECT value in the Command System table is 'AXAF', the Header Field table must contain the following header fields: HEADER_ID = 'OCMDHDR', FIELD_MNEMONIC = 'OBC_A/B_SELECT'; HEADER_ID = 'HWCMDHDR', FIELD_MNEMONIC= 'BUS_A/B_SELECT'; and HEADER_ID = 'HWCMDHDR', FIELD_MNEMONIC = 'CTU_A/B_SELECT'.
57.	If the UPLINK_DATA_TYPE for a specific CMD_MNEMONIC and FIELD_MNEMONIC in the Command Field table is 'ICHK' and the PROJECT value in the Command System table is 'SL', then the SPECIAL_PROCESS value of that same command in the Command table must be 'X' or 'N'.
58.	If the UPLINK_DATA_TYPE for a specific CMD_MNEMONIC and FIELD_MNEMONIC in the Command Field table is 'ICHK' and the PROJECT value in the Command System table in 'AXAF', then the SPECIAL_PROCESS value of that same command in the Command table must be 'C'.
59.	If the HAZARDOUS value in the Command table is "Y" for a specific CMD_MNEMONIC, then the HEADER_TYPE value in the Header table must be "P" for the HEADER_ID assigned to the CMD_MNEMONIC in the Command table.
60.	For a specific CMD_MNEMONIC, if the CMD_TYPE value in the Command table is "P", or the CMD_TYPE value in the Command table is "M" and the command is complete (i.e., an INIT_DATA value is provided for each modifiable field for the CMD_MNEMONIC in the Command Field table), then the calculated uplink format for the command must not equal the calculated uplink format of any existing hazardous commands (i.e., commands with a HAZARDOUS value of "Y" in the Command table).
61.	The HAZARDOUS value in the Command table must be "N" for all the CMD_MNEMONICS in the Chain Contents table.
62.	If the PARITY_PROCESS value is not NULL for a specific CMD_MNEMONIC in the Command table, there must be exactly one command field defined for that CMD_MNEMONIC in the Command Field table with an UPLINK_DATA_TYPE value of "IPAR".
63.	If the SPECIAL_PROCESS value is not "S" for a specific CMD_MNEMONIC in the Command table, there must be exactly one command field defined for that CMD_MNEMONIC in the Command Field table with an UPLINK_DATA_TYPE value of "ICHK".

4.2 TABLE LEVEL VALIDATION RULES

Table level validation rules are those rules that involve the data in multiple columns of a single table. The table level validation rules for the tables defined in this volume are specified in the following sections.

4.2.1 Command System Table

The table level validation rules for the Command System table are defined in Table 4.2.1-1.

Table 4.2.1-1. Table Level Validation Rules for the Command System Table

	VALIDATION RULE
1.	The DELIVER_DATE value must not be NULL if a PRE_RELEASE_DATE value exists.
2.	The DELIVER_DATE value must be less than the PRE_RELEASE_DATE value.
3.	The PRE_RELEASE_DATE value must be NULL if the DELIVER_DATE value is NULL.
4.	The PRE_RELEASE_DATE value must not be NULL if a BASELINE_DATE value exists.
5.	The PRE_RELEASE_DATE value must be less than the BASELINE_DATE value.
6.	The BASELINE_DATE value must be NULL if the PRE_RELEASE_DATE value is NULL.

4.2.2 Destination Table

The table level validation rules for the Destination table are defined in Table 4.2.2-1.

Table 4.2.2-1. Table Level Validation Rules for the Destination Table

	VALIDATION RULE
1.	Deleted.
2.	If the DEST_TYPE value is ‘PL,’ ‘PSP,’ or ‘SSI,’ the MAX_HEADER_LENGTH value must be 1.
3.	If the DEST_TYPE value is ‘SSI,’ the MAX_WORDS value must be 33.
4.	If the DEST_TYPE value is ‘PL,’ the MAX_WORDS value must be 41.
5.	If the DEST_TYPE value is ‘PSP,’ the MAX_WORDS value must be 65.
6.	If the DEST_TYPE value is ‘PL,’ ‘PSP,’ or ‘SSI,’ the WORD_LENGTH value must be 16.
7.	Deleted.
8.	If the DEST_TYPE value is ‘AXAF,’ the MAX_HEADER_LENGTH value must be 3.
9.	If the DEST_TYPE value is ‘AXAF,’ the MAX_WORDS value must be 12.
10.	If the DEST_TYPE value is ‘AXAF,’ the WORD_LENGTH value must be 4.

4.2.3 Owner Table

No table level validation rules have been defined for the Owner table.

4.2.4 Header Table

The table level validation rules for the Header table are defined in Table 4.2.4-1.

Table 4.2.4-1. Table Level Validation Rules for the Header Table

	VALIDATION RULE
1.	Deleted.
2.	If the DEST_TYPE value is ‘PL,’ ‘PSP,’ or ‘SSI,’ the LENGTH value must be 1.
3.	If the DEST_TYPE value is ‘AXAF,’ the LENGTH value must be 3.

4.2.5 Header Field Table

The table level validation rules for the Header Field table are defined in Table 4.2.5-1.

Table 4.2.5-1. Table Level Validation Rules for the Header Field Table

	VALIDATION RULE
1.	Deleted.
2.	The LENGTH, START_WORD, and START_BIT values for all header fields with the same HEADER_ID must be defined such that the fields do not overlap and there are no gaps between fields.
3.	The INIT_DATA value must be consistent with the defined INPUT_DATA_TYPE.
4.	If the CALC_TYPE value is NULL, the INIT_DATA value must not be NULL.
5.	Deleted.
6.	If the CALC_TYPE value is not NULL, the FIELD_TYPE value must be ‘P.’
7.	The INPUT_DATA_TYPE and UPLINK_DATA_TYPE values must be a valid pair of values as defined in appendix I.
8.	If the CALC_TYPE value is not NULL, the INIT_DATA value must be NULL.
9.	If the INIT_DATA value is not NULL, the converted value must fit into the length defined for the header field by the LENGTH value.
10.	If the INIT_DATA value is not NULL and the UPLINK_DATA_TYPE value is ‘IMAG,’ ‘ITWO,’ or ‘IUNS,’ the converted INIT_DATA value must be an integer value.
11.	If the CALC_TYPE value is ‘L,’ the INPUT_DATA_TYPE value must be ‘D’ and the UPLINK_DATA_TYPE value must be ‘IUNS.’
12.	If the CALC_TYPE value is ‘O,’ the INPUT_DATA_TYPE value must be ‘H’ and the UPLINK_DATA_TYPE value must be ‘IUNS.’
13.	If the CALC_TYPE value is ‘O,’ the LENGTH value must be greater than or equal to 7 (to allow for the highest possible operations code, which is hexadecimal 045).

4.2.6 Command Table

The table level validation rules for the Command table are defined in Table 4.2.6-1.

Table 4.2.6-1. Table Level Validation Rules for the Command Table

	VALIDATION RULE
1.	If the VAR_LENGTH value is 'Y,' the LENGTH value must be NULL.
2.	If the HAZARDOUS value is 'Y,' the INIT_STATE value must be 'D.'
3.	If the TLM_VERIF_FLAG value is 'Y,' the TLM_VERIF_TIMEOUT value must not be NULL.
4.	Deleted.
5.	If the SPECIAL_PROCESS value in the Command table is 'X' or 'N,' then the OP_CODE value must not be NULL or the CLASS value must be 'SSI' or 'PSP.'
6.	If the CLASS value is not NULL, the OP_CODE value must be NULL.
7.	If the OP_CODE value is not NULL, the CLASS value must be NULL.
8.	Deleted.
9.	If the VAR_LENGTH value is 'Y,' the CMD_TYPE value must be 'M.'
10.	The TECHNICAL_NAME value must be unique within the Command table.
11.	If the VAR_LENGTH value is 'N' and the DEST_TYPE value is 'PL,' the LENGTH value must be less than or equal to 40.
12.	If the CLASS value is not NULL, and the VAR_LENGTH value is 'N,' the LENGTH value must be consistent with the valid lengths defined for the specified CLASS value in appendix H.
13.	The MASTER_COMMAND value for a given command must not be the same as the CMD_MNEMONIC value for that command.
14.	If the VAR_LENGTH value is 'Y,' the MASTER_COMMAND value must be NULL.
15	If the VAR_LENGTH value is 'Y' for a given command, the CMD_MNEMONIC value for that command must not be specified as the MASTER_COMMAND value for any other commands.
16.	If the MASTER_COMMAND value is not NULL for a specific CMD_MNEMONIC, the LENGTH value for that CMD_MNEMONIC must be equal to the LENGTH value for the MASTER_COMMAND,
17.	Deleted.
18.	If the PARITY_PROCESS value is not NULL, the DEST_TYPE value must be 'AXAF'.
19.	If the DEST_TYPE value is 'AXAF' and the CLASS value is 'S', the HEADER_ID value must be NULL. Otherwise, the HEADER_ID value must not be NULL.
20.	If the DEST_TYPE value is 'AXAF' and the CLASS value is 'S', the SPECIAL_PROCESS value cannot be 'C'.
21.	If the DEST_TYPE value is 'AXAF', the CLASS value must not be NULL.
22.	If the HAZARDOUS value is 'Y', the CMD_TYPE value must be 'P'.

4.2.7 Command Field Table

The table level validation rules for the Command Field table are defined in Table 4.2.7-1.

Table 4.2.7-1. Table Level Validation Rules for the Command Field Table

	VALIDATION RULE
1.	Deleted.
2.	If the INPUT_DATA_TYPE value is not ‘D’ or ‘G,’ the DECAL_TYPE value must be ‘N.’
3.	If the DECAL_TYPE value is ‘PC,’ the CAL_COEF_0 and CAL_COEF_1 values must not be NULL.
4.	The LENGTH, START_WORD, and START_BIT values for all command fields with the same CMD_MNEMONIC must be defined such that the fields do not overlap and there are no gaps between fields.
5.	If the VAR_LENGTH value is Y (variable-length field), the command field must be the last field in the command.
6.	If the FIELD_TYPE value is ‘P,’ and the INPUT_DATA_TYPE is not NULL, the INIT_DATA value must not be NULL.
7.	The INIT_DATA value must be consistent with the defined INPUT_DATA_TYPE.
8.	The RANGE_LOW value must be less than or equal to the RANGE_HIGH value.
9.	If the INPUT_DATA_TYPE value is NULL, the INIT_DATA value must be NULL.
10.	The INPUT_DATA_TYPE and UPLINK_DATA_TYPE values must be a valid pair of values as defined in Appendix I.
11.	The ENG_UNIT and DIMENSION values must be a valid pair of values as defined in Appendix D.
12.	If the VAR_LENGTH value is ‘N,’ the LENGTH value must be consistent with the valid lengths defined in Appendix I for the UPLINK_DATA_TYPE value.
13.	The INIT_DATA value must be greater than or equal to the RANGE_LOW value and less than or equal to the RANGE_HIGH value.
14.	If RANGE_LOW or RANGE_HIGH values are defined, the DECAL_TYPE value must be ‘PC’ or ‘PP.’
15.	If the VAR_LENGTH value is ‘Y,’ the LENGTH value must be NULL.
16.	If the VAR_LENGTH value is ‘Y,’ the INPUT_DATA_TYPE value must be ‘H’ and the UPLINK_DATA_TYPE value must be ‘IUNS.’
17.	If the INIT_DATA value is not NULL and the VAR_LENGTH value is ‘N,’ the converted value must fit into the length defined for the command field by the LENGTH value.
18.	If the INIT_DATA value is not NULL and the UPLINK_DATA_TYPE value is ‘IMAG,’ “ITWO,’ or ‘IUNS,’ the converted INIT_DATA value must be an integer value.
19.	If the INPUT_DATA_TYPE value is NULL, the FIELD_TYPE value must be ‘P.’
20.	The ENG_UNIT and DIMENSION values must be a valid pair of values as defined in Appendix D.
21.	If the UPLINK_DATA_TYPE value is ‘IPAR’, the FIELD_TYPE value must be ‘P’.
22.	If the UPLINK_DATA_TYPE value is ‘ICHK’, the command field must be the last field in the command.
23.	If the INPUT_DATA_TYPE value is NULL, the UPLINK_DATA_TYPE value must be ‘ICHK’.

4.2.8 Point Pair Decalibration Table

The table level validation rules for the Point Pair Decalibration table are defined in Table 4.2.8-1.

Table 4.2.8-1. Table Level Validation Rules for the Point Pair Decalibration Table

	VALIDATION RULE
1.	The COUNTS and VALUE values for a given command field must be defined such that the graph described by the point pairs of the command field is continuously increasing or decreasing.

4.2.9 Telemetry Verifier Table

The table level validation rules for the Telemetry Verifier table are defined in Table 4.2.9-1.

Table 4.2.9-1. Table Level Validation Rules for the Telemetry Verifier Table

	VALIDATION RULE
1.	If the STATE_CODE value is NULL, the RANGE_LOW or RANGE_HIGH value must not be NULL. Otherwise, the RANGE_LOW and RANGE_HIGH values must be NULL.
2.	If the RANGE_LOW and RANGE_HIGH values are NULL, the STATE_CODE value must not be NULL. Otherwise, the STATE_CODE value must be NULL.
3.	The RANGE_LOW value must be less than or equal to the RANGE_HIGH value.

4.2.10 Chain Table

No table level validation rules have been defined for the Chain table.

(Table 4.2.10-1 has been deleted.)

4.2.11 Chain Contents Table

No table level validation rules have been defined for the Chain Contents table.

4.2.12 Command Data Set Table

No table level validation rules have been defined for the Command Data Set table.

4.2.13 Command Data Set Field Table

No table level validation rules have been defined for the Command Data Set Field table.

4.3 COLUMN LEVEL VALIDATION RULES

Column level validation rules are those rules that apply to the data in a specific column of a single table. These rules are divided into standard column level rules and specific column level rules.

The standard column level validation rules are a fixed set of standard validation rules which can be defined for any column of any table. An individual standard rule may not apply to a particular column and there may be many columns that have no standard rules. Standard column level rules include minimum and maximum values, default values, and valid values.

The specific column level validation rules for a specific column are those column level rules that cannot be expressed as standard rules. A single column may have many specific validation rules or there may not be any specific rules for a given column.

The column level validation rules for the tables defined in this volume are specified in the following sections.

4.3.1 Command System Table

The standard column level validation rules for the Command System table are defined in Table 4.3.1-1. The specific column level validation rules for the Command System table are defined in Table 4.3.1-2.

Table 4.3.1-1. Standard Column Level Validation Rules for the Command System Table

COLUMN NAME	MINIMUM VALUE	MAXIMUM VALUE	DEFAULT VALUE	VALID VALUES
PROJECT				"AXAF," "ISS," "SL"

Table 4.3.1-2. Specific Column Level Validation Rules for the Command System Table

COLUMN NAME	VALIDATION RULE
MISSION	All letters in the value must be uppercase with no embedded blanks.
REV	All letters in the value must be uppercase with no embedded blanks

4.3.2 Destination Table

The standard column level validation rules for the Destination table are defined in Table 4.3.2-1. The specific column level validation rules for the Destination table are defined in Table 4.3.2-2.

Table 4.3.2-1. Standard Column Level Validation Rules for the Destination Table

COLUMN NAME	MINIMUM VALUE	MAXIMUM VALUE	DEFAULT VALUE	VALID VALUES
MAX_HEADER_LENGTH	1			
MAX_WORDS	1			
WORD_LENGTH				4, 8, 16, 32

Table 4.3.2-2. Specific Column Level Validation Rules for the Destination Table

COLUMN NAME	VALIDATION RULE
DEST_TYPE	All letters in the value must be uppercase.
MAX_HEADER_LENGTH	Value must not be NULL.
MAX_WORDS	Value must not be NULL.
WORD_LENGTH	Value must not be NULL.

4.3.3 Owner Table

No standard column level validation rules have been defined for the Owner table. The specific column level validation rules for the Owner table are defined in Table 4.3.3-1.

Table 4.3.3-1. Specific Column Level Validation Rules for the Owner Table

COLUMN NAME	VALIDATION RULE
OWNER_ID	All letters in the value must be uppercase with no embedded blanks.

4.3.4 Header Table

The standard column level validation rules for the Header table are defined in Table 4.3.4-1. The specific column level validation rules for the Header table are defined in Table 4.3.4-2.

Table 4.3.4-1. Standard Column Level Validation Rules for the Header Table

COLUMN NAME	MINIMUM VALUE	MAXIMUM VALUE	DEFAULT VALUE	VALID VALUES
HEADER_TYPE			“P”	“P,” “M”
LENGTH	1			

Table 4.3.4-2. Specific Column Level Validation Rules for the Header Table

COLUMN NAME	VALIDATION RULES
HEADER_ID	All letters in the value must be uppercase with no embedded blanks.
OWNER_ID	Value must not be NULL.
DEST_TYPE	Value must not be NULL.
HEADER_TYPE	Value must not be NULL.
LENGTH	Value must not be NULL.

4.3.5 Header Field Table

The standard column level validation rules for the Header Field table are defined in Table 4.3.5-1. The specific column level validation rules for the Header Field table are defined in Table 4.3.5-2.

Table 4.3.5-1. Standard Column Level Validation Rules for the Header Field Table

COLUMN NAME	MINIMUM VALUE	MAXIMUM VALUE	DEFAULT VALUE	VALID VALUES
FIELD_TYPE			"P"	"P," "M"
INPUT_DATA_TYPE				See Appendix I
UPLINK_DATA_TYPE				See Appendix I
LENGTH	1			
START_WORD	1			
START_BIT	0			
CALC_TYPE				"L," O,"

Table 4.3.5-2. Specific Column Level Validation Rules for the Header Field Table

COLUMN NAME	VALIDATION RULES
FIELD_MNEMONIC	All letters in the value must be uppercase with no embedded blanks.
FIELD_TYPE	Value must not be NULL.
INPUT_DATA_TYPE	Value must not be NULL.
UPLINK_DATA_TYPE	Value must not be NULL.
LENGTH	Value must not be NULL.
START_WORD	Value must not be NULL.
START_BIT	Value must not be NULL.

4.3.6 Command Table

The standard column level validation rules for the Command table are defined in Table 4.3.6-1. The specific column level validation rules for the Command table are defined in Table 4.3.6-2.

Table 4.3.6-1. Standard Column Level Validation Rules for the Command Table

COLUMN NAME	MINIMUM VALUE	MAXIMUM VALUE	DEFAULT VALUE	VALID VALUES
CMD_TYPE			“P”	“P,” “M”
CLASS				See Appendix H
OP_CODE				“003,” “004,” “045”
LENGTH	0			
VAR_LENGTH			“N”	“Y,” “N”
INIT_STATE			“E”	“E,” “D”
THRU_TYPE			“N”	“N,” “S,” “T”
HAZARDOUS				“Y,” “N”
CRITICAL				“Y,” “N”
SPECIAL_PROCESS			“S”	
TIME_CONSTRAINT	0			
TLM_VERIF_FLAG			“N”	“Y,” “N”
TLM_VERIF_TIMEOUT	0			

Table 4.3.6-2. Specific Column Level Validation Rules for the Command Table

COLUMN NAME	VALIDATION RULES
CMD_MNEMONIC	All letters in the value must be uppercase with no embedded blanks. Beginning letter in the value must not be *"#"*.
MSID	Value must not be NULL. All letters in the value must be uppercase with no embedded blanks.
OWNER_ID	Value must not be NULL.
CMD_TYPE	Value must not be NULL.
DEST_TYPE	Value must not be NULL.
VAR_LENGTH	Value must not be NULL.
INIT_STATE	Value must not be NULL.
THRU_TYPE	Value must not be NULL.
HAZARDOUS	Value must not be NULL.
CRITICAL	Value must not be NULL.
SPECIAL_PROCESS	Value must not be NULL.
TLM_VERIF_FLAG	Value must not be NULL.
TECHNICAL_NAME	All letters in the value must be uppercase.

4.3.7 Command Field Table

The standard column level validation rules for the Command Field table are defined in Table 4.3.7-1. The specific column level validation rules for the Command Field table are defined in Table 4.3.7-2.

Table 4.3.7-1. Standard Column Level Validation Rules for the Command Field Table

COLUMN NAME	MINIMUM VALUE	MAXIMUM VALUE	DEFAULT VALUE	VALID VALUES
FIELD_TYPE			"P"	"P," "M"
INPUT_DATA_TYPE				See Appendix I
UPLINK_DATA_TYPE				See Appendix I
DECAL_TYPE			"N"	"N," "PC," "PP"
LENGTH	0			
VAR_LENGTH			"N"	"Y," "N"
START_WORD	1			
START_BIT	0			

Table 4.3.7-2. Specific Column Level Validation Rules for the Command Field Table

COLUMN NAME	VALIDATION RULES
FIELD_MNEMONIC	All letters in the value must be uppercase with no embedded blanks.
FIELD_TYPE	Value must not be NULL.
UPLINK_DATA_TYPE	Value must not be NULL.
DECAL_TYPE	Value must not be NULL.
START_WORD	Value must not be NULL.
START_BIT	Value must not be NULL.
CAL_COEF 1	Value must not be 0.

4.3.8 Point Pair Decalibration Table

No standard column level validation rules have been defined for the Point Pair Decalibration table. The specific column level validation rules for the Point Pair Decalibration table are defined in Table 4.3.8-1.

Table 4.3.8-1. Specific Column Level Validation Rules for the Point Pair Decalibration Table

COLUMN NAME	VALIDATION RULES
VALUE	Value must not be NULL.

4.3.9 Telemetry Verifier Table

No standard column level validation rules have been defined for the Telemetry Verifier table. The specific column level validation rules for the Telemetry Verifier table are defined in Table 4.3.9-1.

Table 4.3.9-1. Specific Column Level Validation Rules for the Telemetry Verifier Table

COLUMN NAME	VALIDATION RULES
TLM_MSID	All letters in the value must be uppercase with no embedded blanks.
STATE_CODE	All letters in the value must be uppercase.

4.3.10 Chain Table

The standard column level validation rules for the Chain table are defined in Table 4.3.10-1. The specific column level validation rules for the Chain table are defined in Table 4.3.10-2.

Table 4.3.10-1. Standard Column Level Validation Rules for the Chain Table

COLUMN NAME	MINIMUM VALUE	MAXIMUM VALUE	DEFAULT VALUE	VALID VALUES
CHAIN_TYPE			“P”	“P,” “M”
INIT_STATE			“E”	“E,” “D”

Table 4.3.10-2. Specific Column Level Validation Rules for the Chain Table

COLUMN NAME	VALIDATION RULES
CHAIN_NAME	All letters in the value must be uppercase with no embedded blanks. Beginning letter in the value must not be “#”.
OWNER_ID	Value must not be NULL.
CHAIN_TYPE	Value must not be NULL.
INIT_STATE	Value must not be NULL.

4.3.11 Chain Contents Table

The standard column level validation rules for the Chain Contents table are defined in Table 4.3.11-1. The specific column level validation rules for the Chain Contents table are defined in Table 4.3.11-2.

Table 4.3.11-1. Standard Column Level Validation Rules for the Chain Contents Table

COLUMN NAME	MINIMUM VALUE	MAXIMUM VALUE	DEFAULT VALUE	VALID VALUES
CMD_POSITION	1			

Table 4.3.11-2. Specific Column Level Validation Rules for the Chain Contents Table

COLUMN NAME	VALIDATION RULES
CMD_MNEMONIC	Value must not be NULL.

4.3.12 Command Data Set Table

No standard column level validation rules have been defined for the Command Data Set table. The specific column level validation rules for the Command Data Set table are defined in Table 4.3.12-1.

Table 4.3.12-1. Specific Column Level Validation Rules for the Command Data Set Table

COLUMN NAME	VALIDATION RULES
DATA_SET_NAME	All letters in the value must be uppercase with no embedded blanks.

4.3.13 Command Data Set Field Table

No standard column level validation rules have been defined for the Command Data Set Field table. The specific column level validation rules for the Command Data Set Field table are defined in Table 4.3.13-1.

Table 4.3.13-1. Specific Column Level Validation Rules for the Command Data Set Field Table

COLUMN NAME	VALIDATION RULES
DATA	Value must not be NULL.

APPENDIX A

ABBREVIATIONS AND ACRONYMS

APPENDIX A - ABBREVIATIONS AND ACRONYMS

ASCII	American Standard Code for Information Interchange
AXAF	Advanced X-ray Astrophysics Facility
CAL	Calibration
CDB	Command Database
CHK	Checksum
CMD	Command
COEF	Coefficient
CRC	Cyclic Redundancy Code
DBCG	Database Coordination Group
DECAL	Decalibration
DOC	Document
EBCDIC	Extended Binary Coded Decimal Interchange Code
ECOS	Experiment Computer Operating System
EHS	Enhanced HOSC System
ENG	Engineering
E-R	Entity Relationship
EXP	Expected
GMT	Greenwich Mean Time
GSE	Ground Support Equipment
HEX	Hexadecimal
HOSC	Huntsville Operations Support Center
IBM	International Business Machines
ID	Identifier

IEEE	Institute of Electrical and Electronics Engineers
INIT	Initial
ISS	INTERNATIONAL Space Station
MIN	Minimum
MAX	Maximum
MRTC	Multiple Real-time Command
MSFC	Marshall Space Flight Center
MSID	Measurement/Stimulus Identifier
NASA	National Aeronautics and Space Administration
OCDB	Operational Command Database
OI	Operational Instrumentation
PAYCOM	Payload Commander
PC	Polynomial Coefficient
PL	Payload
POCC	Payload Operations Control Center
PP	Point Pair
PSP	Payload Signal Processor
RI	Record Identification
RDBMS	Relational Database Management System
REV	Revision
RTC	Real-time Command
SL	Spacelab
SPC	Stored Program Command
SSI	Standard Serial Input/Output

TBD	To Be Determined
THRU	Throughput
TLM	Telemetry
VAR	Variable
VERIF	Verifier
ZWC	Zero Word Count

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B

COLUMN TYPE DEFINITIONS

APPENDIX B - COLUMN TYPE DEFINITIONS

This appendix defines the column types used in the table descriptions of this volume.

<table>
<tr><th>COLUMN TYPE</th><th>DESCRIPTION</th></tr>
<tr><td>character</td><td>ASCII text characters as specified in Appendix C.</td></tr>
<tr><td>integer</td><td>Member of the set of decimal whole numbers (.. -2, -1, 0, 1, 2...). No decimal point allowed.</td></tr>
<tr><td>scientific notation</td><td>A number of the following format:

<table>
<tr><td>±X.XXXXXXX</td><td>E±XX</td></tr>
<tr><td>A</td><td>B</td></tr>
</table>

A: Mantissa
• sign is optional (positive is assumed)
• decimal point may float
• length may vary
• no embedded blanks

B: Exponent (optional)
• sign is required
• immediately follows the last digit of the mantissa
• no embedded blanks</td></tr>
<tr><td>floating point</td><td>A number of the following format:

<table>
<tr><td>±X.XXXXXXXX</td></tr>
<tr><td>A</td></tr>
</table>

A: Mantissa
• sign is optional (positive is assumed)
• decimal point may float
• length may vary
• no embedded blanks</td></tr>
<tr><td>timestamp</td><td>Valid date/time value of the following format:

<table>
<tr><td>DD-MMM-YYYY</td><td>HH:MM:SS</td></tr>
<tr><td>A</td><td>B</td></tr>
</table>
A: Date
DD two-digit day of the month (01-31)
MMM first three letters of the month (e.g., JAN, FEB, MAR)
YYYY four-digit year (e.g., 1995)

B: Time
HH two-digit hour of the day (00-23)
MM two-digit minute of the hour (00-59)
SS two-digit second of the minute (00-59)

There should be one blank space between the date and the time.

Examples: 03-JAN-1995 10:05:34
21-MAY-2003 22:58:03</td></tr>
</table>

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C

LEGAL CHARACTERS

APPENDIX C - LEGAL CHARACTERS

This appendix defines the legal characters that can be input into the tables defined in this volume when the user supplies database information.

UPPER CASE	LOWER CASE	DIGITS	SPECIAL CHARACTERS	
A	a	0		blank/NULL
B	b	1	**:**	colon
C	c	2	**.**	period
D	d	3	**,**	comma
E	e	4	"	quotation mark
F	f	5	'	apostrophe
G	g	6	**&**	ampersand
H	h	7	**;**	semicolon
I	i	8	**!**	exclamation point
J	j	9	**_**	underline
K	k		$	dollar sign
L	l		=	equal sign
M	m		-	minus sign
N	n		+	plus sign
O	o			
P	p		/	slash
Q	q		>	greater than
R	r		<	less than
S	s			
T	t		#	number sign
U	u		**@**	commercial at
V	v		**(**	left parenthesis
W	w		**)**	right parenthesis
X	x			
Y	y			
Z	z			

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D

ENGINEERING UNIT ABBREVIATIONS

APPENDIX D - ENGINEERING UNIT ABBREVIATIONS

This appendix defines the legal engineering unit values that may be assigned to the command fields in the Command Field table.

DIMENSION	UNIT OF MEASURE	ENGINEERING UNIT
ABSORBANCE	ABSORPTION UNITS	AU
ABSORBED DOSE	GRAY	GRAY
	RADIATION ABSORBED DOSES	RADAD
ABSORBED DOSE RATE	MILLI-RADIANS PER MINUTE	MRADPM
ACCLERATION (ANGULAR)	DEGREE/SEC/SEC	DEGSS
	RADIANS/SEC/SEC	RADSS
ACCELERATION (LINEAR)	FEET/SEC/SEC	FTSS
	GRAVITY	G
	INCH PER SECOND SQUARED	INPSEC2
	METERS/SEC/SEC	MSS
ACOUSTIC IMPEDANCE	PASCAL SECONDS PER CUBIC METER	PASPM3
ACTIVITY (OF A RADIONUCLIDE)	BECQUEREL	BQ
AMOUNT	KILOMOLE	KMOL
	MICROMOLE	UMOL
	MILLIMOLE	MMOL
	MOLE	MOL
ANGLE OR ELECT PHASE	ARC MINUTES	AMIN
	ARC SECONDS	ASEC
	DEGREES	DEG
	MICRORADIANS	URAD
	MILLIRADIANS	MRAD
	RADIANS	RAD
ANGULAR MOMENTUM	FOOT POUND SECONDS	FTLBSEC
ANGULAR/ ATTITUDE RATE ERROR	DEGREE PER HOUR	DEGPHR

DIMENSION	UNIT OF MEASURE	ENGINEERING UNIT
ANGULAR VELOCITY	DEGREES/SEC	DEGPS
	RADIANS/SEC	RADPS
	REVOLUTIONS/MIN	RPM
	REVOLUTIONS/SEC	RPS
AREA	SQUARE CENTIMETERS	CM2
	SQUARE DECIMETERS	DM2
	SQUARE KILOMETERS	KM2
	SQUARE METERS	M2
	SQUARE MILLIMETERS	MM2
ATTITUDE RATE NOISE	DEGREES SQUARED PER SECOND CUBED	DEG2PSEC3
BANDWIDTH	BITS PER SECOND	BPS
BRIGHTNESS	MAGNITUDE INTENSITY	MI
CAPACITANCE	FARAD	FAR
	MICROFARAD	UFAR
	MILLIFARAD	MFAR
	NANOFARAD	NFAR
	PICOFARAD	PFAR
CHARGE	AH	AH
	COULOMB	C
	KILOCOULOMB	KC
	MEGACOULOMB	MGC
	MICROAH	UAH
	MICROCOULOMB	UC
	MILLIAH	MAH
	MILLICOULOMB	MC
	NANOCOULOMB	NC
	PICOCOULOMB	PC
CHARGE DENSITY	COULOMB/SQUARE METER	CPM2
COEFFICIENT OF HEAT TRANSFER	WATT/SQUARE METER KELVIN	WPM2K
CONCENTRATION	MICROGRAMS PER MILLIMETER	UGPMM
	PARTS PER BILLION	PPB

DIMENSION	UNIT OF MEASURE	ENGINEERING UNIT
CONCENTRATION OF SUBSTANCE	MOLE/CUBIC METER	MOLPM3
CONDUCTANCE	KILOSIEMENS	KSIE
	MICROSIEMENS	USIE
	MILLISIEMENS	MSIE
	SIEMENS	SIE
CONDUCTIVITY	KILOSIEMENS/METER	KSIEPM
	MICRO-MHOS PER CENTIMETER	UMHOSPCM
	SIEMENS/METER	SIEPM
COUNTS/QUANTITY	COUNTS	CNTS
CURRENT	AMPERES	AMP
	MICROAMPERES	UAMP
	MILLIAMPERES	MAMP
	NANOAMPERES	NAMP
	PICOAMPERES	PAMP
CURRENT DENSITY	AMPS/SQUARE METER	AMPPM2
DATA RATE	KILOBITS PER SECOND	KBPS
DENSITY	GRAMS PER CUBIC CENTIMETER	GPC3
	KILOGRAMS PER CUBIC METER	KGPM3
DIFFERENTIAL PRESS/SEAL LEAK	POUNDS PER SQUARE INCH DIFFERENTIAL	PSID
DOSE EQUIVALENT	ROTOGEN EQUIVALENT MAN	REM
	SIEVERTS	SV
	THOUSAND POUNDS FORCE	KLBF
	THOUSAND REV PER MINUTE	KRPM
	THOUSANDS FEET	KFT
EFFECTIVE DRAG AREA	FEET SQUARED	FT2
ELECTRIC DIPOLE MOMENT	COULOMB METER	CM
ELECTRIC FIELD STRENGTH	VOLT/METER	VPM
ELECTRIC POLARIZATION	COULOMB/SQUARE METER	CPM2
	KILOCOULOMB/SQUARE METER	KCPM2
	MICROCOULOMB/SQUARE METER	UCPM2
	MILLICOULOMB/SQUARE METER	MCPM2

DIMENSION	UNIT OF MEASURE	ENGINEERING UNIT
ELECTROMAGNETIC MOMENT	AMPS SQUARE METERS	AMPM2
ELECTROMOTIVE FORCE	MILLIVOLTS DIRECT CURRENT	MVDC
ENERGY	ELECTRON VOLT	EV
	GIGA ELECTRON VOLT	GEV
	GIGAJOULES	GJ
	JOULES	J
	KILO ELECTRON VOLT	KEV
	KILOJOULES	KJ
	KILOWATT HOURS	KWH
	MEGA ELECTRON VOLT	MGEV
	MEGAJOULES	MGJ
	MILLIJOULES	MJ
	TERAJOULES	TJ
FLOW RATE	CUBIC METERS/SEC	M3PS
	GALLONS/MIN	GPM
	LITERS/SEC	LPS
	MILLILITER PER HOUR	MLPHR
	POUND PER HOUR	LBPHR
FORCE	KILONEWTONS	KNEW
	MEGANEWTONS	MGNEW
	MICRONEWTONS	UMEW
	MILLINEWTONS	MNEW
	NEWTONS	NEW
	POUNDS FORCE	LBF
FREQUENCY	GIGAHERTZ	GHZ
	HERTZ (CYCLES/SEC)	HZ
	KILOHERTZ	KHZ
	KILOPULSES/SEC	KPPS
	MEGAHERTZ	MHZ
	PULSES/SEC	PPS
	TERAHERTZ	THZ
HEAT CAPACITY	JOULE/KELVIN	JPK
HUMIDITY	PERCENT	PCT

DIMENSION	UNIT OF MEASURE	ENGINEERING UNIT
ILLUMINANCE	LUX	LX
INDUCTANCE	HENRY	H
	MICROHENRY	UH
	MILLIHENRY	MH
	NANOHENRY	NH
	PICOHENRY	PH
IRRADIANCE	WATT/SQUARE METER	WPM2
LENGTH OR DISTANCE	ASTRONOMIC UNIT	AU
	CENTIMETERS	CM
	FEET	FT
	INCHES	IN
	KILOMETERS	KM
	METERS	M
	MICROMETERS	UM
	MILE	MI
	MILLIMETERS	MM
	NANOMETERS	NM
	NAUTICAL MILES	NAUTM
	PARSEC	PS
	PICOMETERS	PM
	YARD	YD
LIGHT EXPOSURE	LUX SECOND	LUXS
LINEAR CURRENT DENSITY	AMPS/METER	AMPPM
	AMPS/MILLIMETER	AMPPMM
LINEAR DIMENSIONS	INTENSITY	INTENS
LINEAR ENERGY TRANS	KILO-ELECTRON VOLTS PER MICRON	KEVPMICRON
LINEAR EXPANSION COEFFICIENT	1/KELVIN	ONEPK
LUMINANCE	CANDELS/SQUARE METER	CDPM2
LUMINOSITY	CANDELA	CD
LUMINOUS EFFICACY	LUMENS/WATT	LUMPW
LUMINOUS EXCITANCE	LUMENS/SQUARE METER	LUMPM2
LUMINOUS FLUX	LUMENS	LUM

DIMENSION	UNIT OF MEASURE	ENGINEERING UNIT
MAGNETIC DIPOLE MOMENT	WEBER METER	WBM
MAGNETIC FLUX	MILLIWEBER	MWB
	WEBER	WB
MAGNETIC INDUCTION	MICROTESLA	UTES
	MILLITESLA	MTES
	NANOTESLA	NTES
	TESLA	TES
MAGNETIC VECTOR POTENTIAL	WEBER/METER	WBPM
MAGNETIZATION	AMPS/METER	AMPPM
MASS	ATOMIC MASS UNIT	U
	GRAIN	GR
	GRAMS	GM
	HUNDRED-WEIGHT	CWT
	KILOGRAMS	KG
	MICROGRAMS	UG
	MILLIGRAMS	MG
	OUNCE	OZ
	POUNDS MASS	LBM
	TON	TON
	TONNE	TONNE
MECHANICAL IMPEDANCE	NEWTON SECOND/METER	NWSPM
MOLALITY OF SOLUTE	MOLE/KILOGRAM	MOLPKG
MOLAR HEAT CAPACITY	JOULE/MOLE KELVIN	JPMOLK
MOLAR INTERNAL ENERGY	JOULE/MOLE	JPMOL
	KILOJOULE/MOLE	KJPMOL
MOLAR MASS	GRAM/MOLE	GPMOL
	KILOGRAM/MOLE	KGPMOL
MOLAR VOLUME	CUBIC METER/MOLE	M3PMOL
	LITER/MOL	LPMOL
MOMENTUM	KILOGRAMS METER PER SECOND	KGMPS
MOMENTUM (ANGULAR)	KILGRAMS SQUARE METERS PER SEC	KGM2PS

DIMENSION	UNIT OF MEASURE	ENGINEERING UNIT
MOMENT OF INERTIA	KILOGRAMS SQUARE METERS	KGM2
PERMEABILITY	HENRY/METER	HPM
	MICROHENRY/METER	UHPM
	NANOHENRY/METER	NHPM
PERMITIVITY	FARAD/METER	FARPM
	MICROFARAD/METER	UFARPM
	NANOFARAD/METER	NFARPM
	PICOFARAD/METER	PFARPM
POUNDS SECOND	POUNDS SECOND	LBSEC
POWER	GIGAWATTS	GW
	KILOWATTS	KW
	MEGAWATTS	MGW
	MICROWATTS	UW
	MILLIGRAVITY	MGEE
	MILLIWATTS	MW
	WATTS	W
PRESSURE	ATMOSPHERES	ATM
	BAR	BAR
	GIGAPASCAL	GPA
	INCHES OF MERCURY	INHG
	KILOPASCAL	KPA
	MEGAHERTZ	MGHZ
	MEGAPASCAL	MGPA
	MICROBAR	UBAR
	MICROPASCAL	UPA
	MILLIBAR	MBAR
	MILLIBARS	MB
	MILLIPASCAL	MPA
	MILLIRADIANS PER SECOND	MRADPSEC
	MM OF MERCURY	MMHG
	PASCAL	PA
	POUNDS PER SQUARE INCH ABSOLUTE PER MINUTE	PSIAPMIN
	POUNDS/SQUARE FOOT	PSF

DIMENSION	UNIT OF MEASURE	ENGINEERING UNIT
PRESSURE, continued	POUNDS/SQUARE INCH	PSI
	PSI ABSOLUTE	PSIA
	PSI GAUGE	PSIG
	TORR	TORR
	VOLTS PEAK TO PEAK	VPP
QUANTITY OF LIGHT	LUMENS SEC	LUMS
RADIANCE	WATTS/STERADIAN SQUARE METER	WPSRM2
RADIANT INTENSITY	WATTS/STERADIANS	WPSR
RADIATION	CURIES	CI
	ROENTGENS	R
RATIOS	DECIBELS	DB
	FOCAL STOP	F
	PARTS PER MILLION	PPM
	PERCENT	PCT
	UNITS	UNITS
RELUCTANCE	1/HENRY	ONEPH
RESISTANCE	GIGAOHM	GOHM
	KILOOHM	KOHM
	MEGAOHM	MGOHM
	MILLIOHM	MOHM
	OHM	OHM
RESISTIVITY	GIGAOHM METERS	GOHMM
	KILOOHM METERS	KOHMM
	MEGAOHM METERS	MGOHMM
	MILLIOHM METERS	MOHMM
	OHM METERS	OHMM
SOLID ANGLE	STERADIAN	SR
SPECIFIC ACOUSTIC IMPEDANCE	PASCAL SECONDS PER METER	PASPM
SPECIFIC HEAT CAPACITY	JOULE/KILOGRAM KELVIN	JPKGK

DIMENSION	UNIT OF MEASURE	ENGINEERING UNIT
SPECIFIC INTERNAL ENERGY	JOULE/KILOGRAM	JPKG
	KILOJOULE/KILOGRAM	KJPKG
	MEGAJOULE/KILOGRAM	MGJPKG
SURFACE DENSITY OF CHARGE	COULOMB/SQUARE METER	CPM2
SURFACE TENSION	MILLINEWTON/METER	MNWPM
	NEWTON/METER	NWPM
TEMPERATURE	DEGREES CENTIGRADE	DEGC
	DEGREES FAHRENHEIT	DEGF
	DEGREES KELVIN	DEGK
	DEGREES RANKIN	DEGR
TEMPERATURE RATE	DEGREE FAHRENHEIT PER MINUTE	DEGFPMIN
THERMAL CONDUCTIVITY	WATT/METER KELVIN	WPMK
TIME	DAYS	DAY
	HOURS	HR
	MICROSECONDS	USEC
	MILLISECONDS	MSEC
	MINUTES	MIN
	SECONDS	SEC
	YEARS	YEARS
TORQUE	FOOT POUNDS	FTLB
	INCH POUNDS	INLB
	KILONEWTON METER	KNWM
	MEGANEWTON METER	MGNWM
	MICRONEWTON METER	UNWM
	MILLINEWTON METER	MNWM
	NEWTON METER	NWM
	POUNDS FEET	LBFT

DIMENSION	UNIT OF MEASURE	ENGINEERING UNIT
VELOCITY	FEET/SEC	FPS
	INCH PER SECOND	INPSEC
	KILOMETERS/HOUR	KMPHR
	KILOMETERS/SEC	KMPS
	KNOTS	KT
	METERS/SEC	MPS
VELOCITY/VELOCITY (STATE VEC)	FEET PER SECOND	FTPSEC
VISCOSITY	MILLIPASCAL SECONDS	MPAS
	PASCAL SECONDS	PAS
VOLTAGE	KILOVOLTS	KV
	MICROVOLTS	UV
	MILLIVOLTS	MV
	VOLT	V
	VOLTS AC	VAC
	VOLTS DC	VDC
VOLUME	CUBIC CENTIMETERS	CM3
	CUBIC DECIMETERS	DM3
	CUBIC FEET	FT3
	CUBIC MILLIMETERS	MM3
	CUBIC METERS	M3
	DAY PLUS HOURS PLUS MINUTES PLUS SECONDS	DHMS
	DECIBEL REFERENCED TO ONE MIL-LIWATT	DBM
	GALLONS	GAL
	LITERS	L
	MILLILITERS	ML

APPENDIX E

GLOSSARY

APPENDIX E - GLOSSARY

Critical Command	A critical command is a command whose initiation and execution could possibly cause damage to a payload or spacecraft and impair the mission.
Decalibration	Decalibration is the process of using calibration point pairs or polynomials to solve for a value in counts.
Hazardous Command	A hazardous command is a command whose initiation and execution could pose a threat to human life or the entire mission.
Owner	The owner of a header, command, or command chain, is the entity to which the ability to update the definition of the header, command, or chain has been assigned. This entity may be a spacecraft subsystem or payload, a HOSC console position, or anything else that the project wishes to use.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F

EXAMPLES

APPENDIX F - EXAMPLES

This appendix provides examples of how to populate the Command Database tables in selected situations. Note that the data values presented here are purely fictitious and are provided only as examples.

Example 1: Predefined Command

The following data define a predefined command with three (3) fields. The word length for the specified destination is 16 bits.

Command Table

CMD_MNEMONIC	= CMD001
MSID	= MSID032
OWNER_ID	= OWNER012
CMD_TYPE	= P
DEST_TYPE	= ISS
LENGTH	= 2
VAR_LENGTH	= N
HAZARDOUS	= N
CRITICAL	= N
HEADER_ID	= HEADER01
TLM_VERIF_FLAG	= N

Command Field Table

CMD_MNEMONIC	= CMD001
FIELD_MNEMONIC	= FIELD01
FIELD_TYPE	= P
INPUT_DATA_TYPE	= A
UPLINK_DATA_TYPE	= SASC
LENGTH	= 2
START_WORD	= 1
START_BIT	= 0
INIT_DATA	= ST

Example 1: Predefined Command (Continued)

Command Field Table, continued	
CMD_MNEMONIC	= CMD001
FIELD_MNEMONIC	= FIELD02
FIELD_TYPE	= P
INPUT_DATA_TYPE	= B
UPLINK_DATA_TYPE	= IUNS
LENGTH	= 8
START_WORD	= 2
START_BIT	= 0
INIT_DATA	= 10010011
CMD_MNEMONIC	= CMD001
FIELD_MNEMONIC	= FIELD03
FIELD_TYPE	= P
INPUT_DATA_TYPE	= B
UPLINK_DATA_TYPE	= IUNS
LENGTH	= 8
START_WORD	= 2
START_BIT	= 8
INIT_DATA	= 00111101

Example 2: Modifiable, Fixed-Length Command

The following data define a modifiable, fixed-length command with one (1) predefined field and two (2) modifiable fields. The word length for the specified destination is 16 bits.

Command Table

CMD_MNEMONIC	= CMD002
MSID	= MSID024
OWNER_ID	= OWNER010
CMD_TYPE	= M
DEST_TYPE	= ISS
LENGTH	= 3
VAR_LENGTH	= N
HAZARDOUS	= N
CRITICAL	= N
HEADER_ID	= HEADER01
TLM_VERIF_FLAG	= N

Command Field Table

CMD_MNEMONIC	= CMD002
FIELD_MNEMONIC	= FIELD01
FIELD_TYPE	= P
INPUT_DATA_TYPE	= B
UPLINK_DATA_TYPE	= IUNS
LENGTH	= 8
START_WORD	= 1
START_BIT	= 0
INIT_DATA	= 1001111
CMD_MNEMONIC	= CMD002
FIELD_MNEMONIC	= FIELD02
FIELD_TYPE	= M
INPUT_DATA_TYPE	= H
UPLINK_DATA_TYPE	= IUNS
LENGTH	= 8
START_WORD	= 1
START_BIT	= 8

Example 2: Modifiable, Fixed-Length Command (Continued)

Command Field Table, continued	
CMD_MNEMONIC	=CMD002
FIELD_MNEMONIC	=FIELD03
FIELD_TYPE	=M
INPUT_DATA_TYPE	=D
UPLINK_DATA_TYPE	=FIBM
LENGTH	=32
START_WORD	=2
START_BIT	=0
INIT_DATA	=5.12

Example 3: Modifiable, Variable-Length Command

The following data define a modifiable, variable-length command with one (1) predefined field and one (1) modifiable fields.

Command Table

CMD_MNEMONIC	= CMD003
MSID	= MSID012
OWNER_ID	= OWNER005
CMD_TYPE	= M
DEST_TYPE	= ISS
LENGTH	= 0
VAR_LENGTH	= Y
HAZARDOUS	= N
CRITICAL	= N
HEADER_ID	= HEADER02
TLM_VERIF_FLAG	= N

Command Field Table

CMD_MNEMONIC	= CMD003
FIELD_MNEMONIC	= FIELD01
FIELD_TYPE	= P
INPUT_DATA_TYPE	= B
UPLINK_DATA_TYPE	= IUNS
LENGTH	= 16
START_WORD	= 1
START_BIT	= 0
INIT_DATA	= 1100111100011100
CMD_MNEMONIC	= CMD003
FIELD_MNEMONIC	= FIELD02
FIELD_TYPE	= M
INPUT_DATA_TYPE	= H
UPLINK_DATA_TYPE	= IUNS
LENGTH	= 0
START_WORD	= 2
START_BIT	= 0

Example 4: Command Field with Polynomial Coefficient Decalibration

Command Field Table	
CMD_MNEMONIC	= CMD004
FIELD_MNEMONIC	= FIELD01
FIELD_TYPE	= M
INPUT_DATA_TYPE	= D
UPLINK_DATA_TYPE	= FIBM
ENG_UNIT	= PSI
DIMENSION	= PRESSURE
DECAL_TYPE	= PC
LENGTH	= 32
START_WORD	= 1
START_BIT	= 0
INIT_DATA	= 1.00
RANGE_LOW	= 0
RANGE_HIGH	= 4.5
CAL_COEF_0	=1.64E-05
CAL_COEF_1	=3.4523E-07

Example 5: Command Field with Point Pair Decalibration

The following data define a command field with point pair (PP) decalibration.

Command Field Table

CMD_MNEMONIC	= CMD005
FIELD_MNEMONIC	= FIELD01
FIELD_TYPE	= M
INPUT_DATA_TYPE	= D
UPLINK_DATA_TYPE	= IUNS
ENG_UNIT	= VAC
DIMENSION	= VOLTAGE
DECAL_TYPE	= PP
LENGTH	= 32
START_WORD	= 1
START_BIT	= 0
INIT_DATA	= 3.0
RANGE_LOW	= 0
RANGE_HIGH	= 5.75

Point Pair Decalibration Table

CMD_MNEMONIC	= CMD005
FIELD_MNEMONIC	= FIELD01
COUNTS	= 0
VALUE	= 0
CMD_MNEMONIC	= CMD005
FIELD_MNEMONIC	= FIELD01
COUNTS	= 50
VALUE	= 2.25
CMD_MNEMONIC	= CMD005
FIELD_MNEMONIC	= FIELD01
COUNTS	= 200
VALUE	= 3.0

Example 5: Command Field With Point Pair Decalibration

Point Pair Delcalibration Table (Continued)	
CMD_MNEMONIC	= CMD005
FIELD_MNEMONIC	= FIELD01
COUNTS	= 255
VALUE	= 5.75

APPENDIX G

ASCII AND EBCDIC CHARACTER TRANSLATIONS

APPENDIX G - ASCII AND EBCDIC CHARACTER TRANSLATIONS

This appendix indicates the ASCII and EBCDIC values for each of the legal characters that can be input into the tables defined in this volume, as defined in Appendix C.

TEXT	ASCII[1]	EBCDIC[2]	TEXT	ASCII[1]	EBCDIC[2]
A	41	C1	s	73	A2
B	42	C2	t	74	A3
C	43	C3	u	75	A4
D	44	C4	v	76	A5
E	45	C5	w	77	A6
F	46	C6	x	78	A7
G	47	C7	y	79	A8
H	48	C8	z	7A	A9
I	49	C9			
J	4A	D1	0	30	F0
K	4B	D2	1	31	F1
L	4C	D3	2	32	F2
M	4D	D4	3	33	F3
N	4E	D5	4	34	F4
O	4F	D6	5	35	F5
P	50	D7	6	36	F6
Q	51	D8	7	37	F7
R	52	D9	8	38	F8
S	53	E2	9	39	F9
T	54	E3			
U	55	E4	[space]	20	40
V	56	E5	!	21	5A
W	57	E6	“	22	7F
X	58	E7	#	23	7B
Y	59	E8	$	24	5B
Z	5A	E9	&	26	50
a	61	81	‘	27	7D
b	62	82	(	28	4D
c	63	83	)	29	5D
d	64	84	+	2B	4E
e	65	85	,	2C	6B
f	66	86	-	2D	60
g	67	87	.	2E	4B
h	68	88	/	2F	61
i	69	89	:	3A	7A
j	6A	91	;	3B	5E
k	6B	92	=	3D	7E
l	6C	93	>	3E	6E
m	6D	94	@	40	7C
n	6E	95	_	5F	6D
o	6F	96			
p	70	97			
q	71	98			
r	72	99			

1 The ASCII representation of each character is shown in hexadecimal.

2 The EBCDIC representation of each character is shown in hexadecimal.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX H

COMMAND CLASSES

APPENDIX H - COMMAND CLASSES

This appendix defines the command classes for each HOSC-supported project that utilizes command classes. It also defines the valid lengths for commands of each class, either as a range of valid lengths or as one or two discrete valid lengths. A separate table is provided for each project.

The command classes for the Spacelab project are specified in Table H-1

The command classes for the AXAF project are specified in Table H-2.

Table H-1. Spacelab Command Classes

CLASS	DESCRIPTION	MINIMUM LENGTH	MAXIMUM LENGTH	VALID LENGTH 1	VALID LENGTH 2
PSP	Route commands to payload signal processor.	1	64	--	--
SSI	Route commands to standard serial input/output.	1	32	--	--

Table H-2. AXAF Command Classes

CLASS	DESCRIPTION	MINIMUM LENGTH	MAXIMUM LENGTH	VALID LENGTH 1	VALID LENGTH 2
R	Realtime commands	--	--	9	--
S	OBC stored commands	--	--	7	--

APPENDIX I

INPUT AND UPLINK DATA TYPES

APPENDIX I - INPUT AND UPLINK DATA TYPES

This appendix defines the input and uplink data types that can be assigned to header fields in the Header Field table and to command fields in the Command Field table. Table I-1 lists the valid input data types. Table I-2 indicates the valid combinations of input and uplink data types that can be used when defining a header or command field.

Table I-1. Input Data Types

INPUT DATA TYPE	DESCRIPTION	NOTE
A	Alphanumeric String	---
B	Binary	---
D	Decimal	Non-exponential input shall have a maximum length of 16 characters. Sign and decimal point are optional.
G	Scientific Notation	Exponential inputs will have a maximum length of 22 characters. Decimal point is required as is the exponent with its sign. Exponent magnitude may not exceed 75.
H	Hexadecimal	---
O	Octal	---

Table I-2. Uplink Data Types

UPLINK DATA TYPE	DESCRIPTION	MINIMUM LENGTH	MAXIMUM LENGTH	VALID LENGTH 1	VALID LENGTH 2	NOTE
FIBM	IBM Floating Point	---	---	32	64	IBM floating point data will be converted from the input format to either a single-precision (32 bits) or double-precision (64 bits) IBM 370 floating point value, stored in the defined field and rounded as necessary. No field padding is performed.
ICHK	Checksum	---	---	8	---	Checksum data will be calculated prior to uplink, converted to binary integer representation, and stored right-justified in the least significant 8 bits of the last command data word of the command, padded with leading zeros as necessary to fill the 8 bits.
IDIS	Discrete	---	---	1	---	Discrete data will be converted to binary integer representation and stored in the defined field. No field padding is performed.
IMAG	Signed Integer	2	32	---	---	Signed integers will be converted from the input format to binary integer representation, ignoring the sign, and stored right-justified within the defined field. If the input value is positive, it will be padded with leading zeros and the high-order bit of the parameter will be set to zero. If the input value is negative, it will be padded with leading zeros and the high-order bit will set to one.
IPAR	Parity Bit	---	---	1	---	Parity bit data will be calculated prior to uplink, converted to binary integer representation, and stored in the defined field. No field padding is performed.

Table I-2. Uplink Data Types (Continued)

UPLINK DATA TYPE	DESCRIPTION	MINIMUM LENGTH	MAXIMUM LENGTH	VALID LENGTH 1	VALID LENGTH 2	NOTE
ITWO	Two's Complement Signed Integer	2	32	---	---	Two's complement signed integers will be converted from the input format to binary integer representation and stored right-justified within the defined field. If the input value is positive, it will be padded with leading zeros as necessary to fill the field. The high-order bit of the field is then set to zero to indicate a positive value. If the value is negative, it will be converted from the input format to two's complement binary integer representation with the sign bit propagated left to the high-order bit of the field. The high-order bit of the field is then set to one to indicate a negative value.
IUNS	Unsigned Integer	2	32	---	---	Unsigned integers will be converted to binary integer representation and stored right-justified within the defined field, padded with leading zeros as necessary to fill the field.
SASC	ASCII Characters	1	60	---	---	ASCII character data will be translated as required, stored left-justified within the defined field, and padded with trailing blanks as necessary to fill the field. ASCII and EBCDIC character translations will be performed according to appendix G.
SEBC	EBCDIC Characters	1	60	---	---	EBCDIC character data will be translated as required, stored left-justified within the defined field, and padded with trailing blanks as necessary to fill the field. ASCII and EBCDIC character translations will be performed according to appendix G.

Table I-3. Input vs. Uplink Data Types

INPUT DATA TYPE	UPLINK DATA TYPE
A	SASC
A	SEBC
B	IDIS
B	IPAR
B	IUNS
D	FIBM
D	IMAG
D	ITWO
D	IUNS
H	IUNS
O	IUNS
G	FIBM