Monitor & Control (M&C) Interface Control Document (ICD) for the GSFC DTN Bundle Protocol Node (BPNode)

NASA GSFC DTN Project

Version 7.0, August 2025

Table of Contents

Preface

This document is under configuration management of the Goddard Space Flight Center (GSFC) DTN Project Configuration Control Board (CCB). A complete revision or Document Change Notice (DCN) will be issued to update the document for any future approved changes.

The GSFC DTN Project assumes responsibility for this document and updates it as required to meet its needs. Reviews of this document are performed at least annually, and updates to this document are made when appropriate.

Changes to this document require prior approval of the Change Authority listed on the signature page. Proposed changes shall be submitted to the DTN Configuration Management Engineer (CME) along with supporting material justifying the proposed change. Questions or comments concerning this document should be addressed to: GSFC DTN Systems Engineer or Product Development Lead (PDL).

Change History Log

Revision	Effective Date	Description of Changes (Reference the CCR & CCB/ERB Approval Date)
-	07/12/2024	Initial Release per ESC-CCR-0644 07/12/2024
1	08/25/2025	Build 7.0

Approvals

The signatories below indicate with their signature their commitment to the implementation of this Plan.

```
Prepared by:



_____________________________________ _______________
Sara Garcia-Beech                     Date
DTN FSW Development Team Lead
Flight Software Systems Branch
Code 582, NASA GSFC



Concurred by:



_____________________________________ _______________
Rebecca Besser                        Date
DTN Systems Engineering Lead
Software Systems Engineering Branch
Code 581, NASA GSFC



Approved by:



_____________________________________ _______________
Peyush Jain                           Date
Flight Segment Mission Manager
DTN Study and Pathfinder Lead
TEMPO, Code 450.2, NASA GSFC
```

Table of TBx

Section	ID	Description
5.6, 5.8, 5.12	3	Events TBD.

Chapter 1. Scope

1.1. Introduction

The Interface Control Document (ICD) is a controlled document that defines data interfaces and exchange formats between the Delay/Disruption-Tolerant Networking (DTN) Bundle Protocol (BP) nodes and Monitor and Control (M&C) ground systems. BP is a networking protocol that transmits data between nodes even when in the absence of continuous connectivity. It is designed to provide reliable end-to-end communication where other networking protocols may not be effective.

The current Bundle Protocol Node (BPNode) is a DTN Node application that implements the Bundle Protocol version 7 (BPv7). GSFC developed BPNode for a mixed-safety-criticality deployment environment, which means that the node may be hosted on:

- Class B architecture with symmetric multiprocessing, such as Linux, or
- Class A architecture with asymmetric multiprocessing, such as a real-time operating system.

This document is developed to ensure that each side of the interface is correctly designed and compatible.

1.2. Build 7.0 Release Notes

Although all the commands, tables, and telemetry packets in this ICD have hooks in the Build 7.0 code, some of them do not have the underlying functionality implemented.

The following directives have not been implemented:

- Startup directives
- Policy directives
- Storage directives
- reset-source-counters
- set-registration-state
- add-mib-array-key
- remove-mib-array-key

The following telemetry packets have not been implemented:

- Per-Source MIB Configuration
- Per-Source MIB Counters

The following tables have not been implemented:

- Compressed Reporting
- MIB Source Configuration

- Source Authorization Policy
- Custody Authorization Policy
- Custodian Authorization Policy
- Report-To-EID Authorization Policy
- Source Latency Policy
- Storage

For telemetry packets or tables where only certain fields have not been implemented, these fields will be indicated with *italics*.

1.3. Software Context

The BPNode software runs on Linux and real-time operating systems in a context where BP clients (BPApps from RFC 9171) communicate directly with BPNode. BPNode expects a Publish/Subscribe pattern where a BPApp publishes Application Data Units (ADUs) and BPNode subscribes to them. Likewise, a BPApp subscribes for expected ADUs and BPNode publishes them. The rest of the diagram below refers to:

1. Core Flight System (cFS) applications, such as
    a. Telemetry Output (TO)
    b. Scheduler (SCH)
    c. Command Ingest (CI)
2. Operating System Abstraction Layer (OSAL)
3. Convergence Layer Adapters (CLAs)
4. Mission Operations Center (MOC), bottom center, which represents the flight controllers and ground systems that communicate with the spacecraft through the M&C interface.

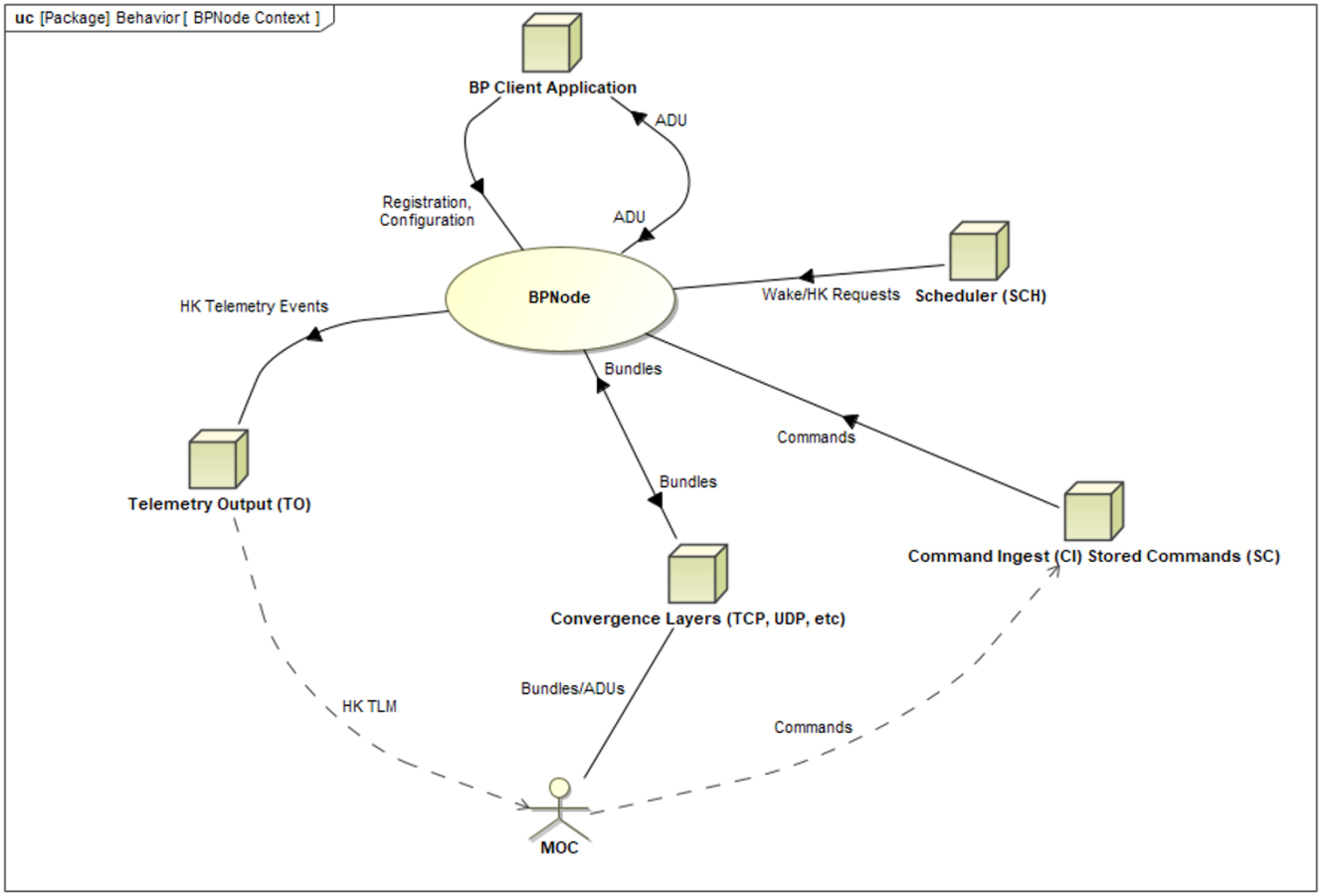

Figure 1. BPNode Context Diagram

1.4. Configuration Management

Interface configuration management is documented in the Configuration Management Plan (450.2-DTN-CMP) baselined in Technical Data Management System (TDMS). DTN CM includes software naming conventions and Discrepancy Reporting management.

1.5. Requirements Verification

Requirements are stored in the GSFC DOORS database and formally exported to TDMS after review and approval by a Configuration Control Board (CCB). The Requirements Verification Traceability Matrix (RVTM) provides verification traceability to testing.

DOORS synchronizes requirements to MagicDraw, which produces system and component diagrams as well as views of requirements that are saved in Teamwork Cloud. DOORS maps requirements to software modules, components, software versions, and source code files.

1.6. Interface Characteristics and Functions

DTN Nodes provide Management Information Base (MIB) objects to collect information required by Network Management. Mission M&C, DTN Network Management (DTNNM), and MOC use the same MIB-defined objects, which include directives, configuration containers, policy information, events, and telemetry. MIB object transport is mission-defined and has significant mission security, authority, and operational constraints.

DTN nodes support both policy and configuration management interfaces:

- Policy is the management interface that defines user access to network services and resources. It is negotiated between users and service providers to ensure adherence to user Service Level Agreements (SLA). DTNNM applies Node Policy based on protocol parameters and distributes it to node operators. Policy can change per contact schedule. The Internet Engineering Task Force (IETF) and Consultative Committee for Space Data Systems (CCSDS) develop policy distribution format and transport.
- Configuration is a management interface to configure node services and resources. It is more static but can change due to off-nominal operations, fault handling, or policy modifications.

DTNNM ensures that configurations are consistent across the network to support the user SLAs. MIB object scope can address sets of related bundles, all bundles, and node operations. Local and remote MIB object transports are necessary for robust operations and fault handling. All MIB objects have associated Level 6 requirements.

1.7. Space Packets

Once the BPNode starts, the M&C connects to it and establishes distinct connections for the exchange of control directives (command messages) and status messages (telemetry). Both types of messages are standard CCSDS Space Packets as defined in CCSDS 133.0.B-2. The User Data Field contains either the control directive sent by the M&C or the Status Data provided by the BPNode, differentiated by the Packet Type indicator, which is set to 0 for telemetry packets and 1 for telecommands.

Bit 4 of the Packet Primary Header contains the Secondary Header Flag, which indicates the presence or absence of the Packet Secondary Header within this Space Packet. It is set to '0' if a secondary header is absent and '1' if it is present. The latter is the case for both types of DTN packets:

- In a command message, the secondary header contains the Function Code (it is the first data point in the User Data Field) and the command checksum value. It also contains parameters associated with that function, encoded in accordance with their associated data type.
- In a telemetry message, it contains the time stamp.

1.8. DTN-Specific Field Types

Most of the mnemonic field types are standard data types (uint, string, etc), but the following data types will also be referenced in this ICD and defined here:

Table 1. Endpoint ID (EID)

Subfield	Description	Type
Scheme	Defines how to parse scheme-specific part (SSP), is either DTN (1) or IPN (2)	uint64 enum
IpnSspFormat	If the Scheme is IPN, defines whether it is in the 2 or 3 digit format	uint64 enum

Subfield	Description	Type
Allocator	Unique identifier for controlling organization	uint64
Node	Unique identifier for system that implements DTN communications protocol service	uint64
Service	Unique identifier for DTN communication protocol service	uint64

Table 2. Endpoint ID (EID) Pattern

Subfield	Description	Type
Scheme	Defines how to parse scheme-specific part (SSP), is either DTN (1) or IPN (2)	uint64 enum
IpnSspFormat	If the Scheme is IPN, defines whether it is in the 2 or 3 digit format	uint64 enum
MaxAllocator	Unique identifier for controlling organization, maximum inclusive value	uint64
MinAllocator	Unique identifier for controlling organization, minimum inclusive value	uint64
MaxNode	Unique identifier for system that implements DTN communications protocol service, maximum inclusive value	uint64
MinNode	Unique identifier for system that implements DTN communications protocol service, minimum inclusive value	uint64
MaxService	Unique identifier for DTN communication protocol service, maximum inclusive value	uint64
MinService	Unique identifier for DTN communication protocol service, minimum inclusive value	uint64

Chapter 2. Documentation

2.1. Reference Documents

The following documents serve as reference material for this document.

Table 3. Reference Documents

Document No.	Document Title
RFC-9171	Bundle Protocol Specification (Version 7)
CCSDS 734.2-B-1	CCSDS Bundle Protocol Specification
CCSDS 133.0.B-2	Space Packet Protocol
450.2-DTN-SRD-L6	DTN Level 6 Systems Requirements Document

Chapter 3. Directives

DTN subscribes to cFS directive-type CCSDS command messages from the cFS Software Bus (SB). DTN directives are listed below. The Space Packet secondary header for commands specifies the Function Code, which identifies the action being requested of the Node. It is the first data point in the User Data Field, which also contains parameters associated with that function, encoded in accordance with their associated data type.

Table 4. Directives Packet Secondary Header

Name	Description	Type	Start Byte	Start Bit	Length Bits
Reserved	Reserved	uint	6	0	1
FunctionCode	Unique function code for each directive	uint	6	1	7
Checksum	Command checksum value	uint	7	0	8

Directives followed by an asterisk are deferred to a future build.

Directive parameters can be formatted as a directive message or table. In general, the issuance of a **valid** control directive increments the **valid** command counter, the issuance of an *invalid* control directive increments the *invalid* command error counter. A directive may be invalid because of the wrong Function Code, length, format, or parameters. The subsequent sections discuss each directive in greater detail.

3.1. Startup Directives

Startup directives add and start all applications and control storage and metadata. None of the startup directives have parameters.

None of the functionality of the startup directives has been implemented as of this build.

3.1.1. Add All Applications*

Name add-all-applications

Function Code 1

Table
- Channel Configuration Table
- ADU Proxy Table

Parameter(s) none

Description Registers with the node all client applications configured to start on startup.

3.1.2. Start All Applications*

Name	start-all-applications
Function Code	2
Table	• Channel Configuration Table • ADU Proxy Table
Parameter(s)	none
Description	Starts all applications configured to start at startup. Begins accepting and delivering ADUs for all configured applications.

3.1.3. Verify Bundle Storage*

Name	verify-bundle-storage
Function Code	3
Table	no
Parameter(s)	none
Description	Verifies headers of bundles in persistent storage upon a cold restart.

3.1.4. Initialize Bundle Storage*

Name	initialize-bundle-storage
Function Code	4
Table	no
Parameter(s)	none
Description	Executes on node startup for a cold start and deletes all bundles in storage for a factory reset.

3.1.5. Verify Bundle Metadata*

Name	verify-bundle-metadata
Function Code	5
Table	no
Parameter(s)	none
Description	Checks whether the bundle metadata reflects the actual bundle storage.

3.1.6. Rebuild Bundle Metadata*

Name	rebuild-bundle-metadata
Function Code	6
Table	no
Parameter(s)	none
Description	Rebuilds bundle storage metadata from stored bundles, including their indices (source Endpoint ID (EID), destination EID, next action time).

3.1.7. Clear Volatile*

Name	clear-volatile
Function Code	7
Table	no
Parameter(s)	none
Description	Executes on warm restart and clears volatile data, including bundle metadata and bundles that are not in persistent storage.

3.1.8. Reload Saved Data*

Name	reload-saved-data
Function Code	8
Table	No
Parameter(s)	None
Description	Reload saved node configuration, bundle metadata from persistent storage, and saved MIB counters.

3.2. Counter Directives

These directives instruct the node to reset various counters.

3.2.1. Reset All Counters

Name	reset-all-counters
Function Code	9
Table	No

Parameter(s) None

Description Sets all resettable MIB counters to zero.

3.2.2. Reset Counter

Name reset-counter

Function Code 10

Table no

Parameter(s)

- uint16 *mibArrayIndex*: source MIB counter array index corresponding to counter (if the counter to reset is a node MIB counter, should be set to the maximum number of source MIB sets allowed, 10 by default)
- uint16 *spare*: spare bytes
- uint32 enum *counter*: Counter to reset. See the Field IDs of the Node and Source MIB Counters Packets for the enumeration values.

Description Sets to zero the MIB counter specified by the parameter.

3.2.3. Reset Source Counters*

Name reset-source-counters

Function Code 11

Table no

Parameter(s)

- uint16 *mibArrayIndex*: source MIB counter array index
- uint16 *spare*: spare bytes

Description Sets to zero all resettable MIB counters associated with a source EID pattern identified by the parameter.

3.2.4. Reset Bundle Counters

Name reset-bundle-counters

Function Code 12

Table no

Parameter(s) None

Description Sets all bundle-related counters to zero.

3.2.5. Reset Error Counters

Name	reset-error-counters
Function Code	13
Table	no
Parameter(s)	• uint16 *mibArrayIndex*: source MIB counter array index • uint16 *spare*: spare bytes
Description	Sets all error counters to zero.

3.3. Application Directives

Application directives add, start, stop, and remove specific applications, as well as set the channel's registration state.

3.3.1. Add Application

Name	add-application
Function Code	14
Table	• Channel Configuration Table • ADU Proxy Table
Parameter(s)	uint32 *ChanId*: The index in the Channel Configuration Table with this application's configurations, also functions as a unique identifier for this application.
Description	This directive adds a new application by: • Setting the channel configuration based on client application configuration • Establishing mapping between client application connection and channel • Opening an ADU channel

3.3.2. Remove Application

Name	remove-application
Function Code	15
Table	no

Parameter(s) uint32 *ChanId*: The index in the Channel Configuration Table with this application's configurations, also functions as a unique identifier for this application.

Description Terminates the connection, closes the ADU channel, and flushes the egress queue for the application specified by the parameter.

3.3.3. Set Registration State*

Name set-registration-state

Function Code 16

Table
- Channel Configuration Table
- ADU Proxy Table

Parameter(s)
- uint32 *ChanId*: The index in the Channel Configuration Table with this application's configurations, also functions as a unique identifier for this application
- uint8 enum *registrationState*: Active (0), PassiveDeferred (1), or PassiveAbandon (2)

Description Sets given application's registration state to specified state.

3.3.4. Start Application

Name start-application

Function Code 17

Table no

Parameter(s) uint32 *ChanId*: The index in the Channel Configuration Table with this application's configurations, also functions as a unique identifier for this application.

Description Verifies the channel configuration of the application indicated by the parameter and begins moving payloads bidirectionally between the node and the given application.

3.3.5. Stop Application

Name stop-application

Function Code 18

Table no

Parameter(s)	uint32 *ChanId*: The index in the Channel Configuration Table with this application's configurations, also functions as a unique identifier for this application.
Description	Stops moving bundles bidirectionally between the node and the given application.

3.4. Policy Directives

Policy Directives add and remove authorized EIDs.

None of the functionality of the policy directives has been implemented as of this build.

3.4.1. Add Authorized Sources*

Name	add-authorized-sources
Function Code	19
Table	Yes
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Adds the EID pattern to a set of authorized source EIDs if the pattern being added does not exceed the maximum size of authorized sources.

3.4.2. Remove Authorized Sources*

Name	remove-authorized-sources
Function Code	20
Table	No
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Removes the EID pattern from a set of authorized source EIDs.

3.4.3. Add Authorized Custody Sources*

Name	add-authorized-custody-sources
Function Code	21
Table	Yes
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Adds the EID pattern to a set of authorized custody source EIDs.

3.4.4. Remove Authorized Custody Sources*

Name	remove-authorized-custody-sources
Function Code	22
Table	No
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Removes the EID pattern from a set of authorized custody source EIDs.

3.4.5. Add Authorized Custodians*

Name	add-authorized-custodians
Function Code	23
Table	Yes
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Adds the EID pattern to a set of authorized custodian EIDs.

3.4.6. Remove Authorized Custodians*

Name	remove-authorized-custodians
Function Code	24
Table	No
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Removes the EID pattern from a set of authorized custodian EIDs.

3.4.7. Add Authorized Report-to EID*

Name	add-authorized-report-to-eid
Function Code	25
Table	No
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Adds the EID pattern to a set of authorized report-to EIDs.

3.4.8. Remove Authorized Report-to EID*

Name	remove-authorized-report-to-eid

Function Code	26
Table	No
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Removes the EID pattern from a set of authorized report-to EIDs.

3.4.9. Add Latency*

Name	add-latency
Function Code	27
Table	No Parameter(s)
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Adds the source EID pattern and latency information to the set of source latency policies.

3.4.10. Remove Latency*

Name	remove-latency
Function Code	28
Table	No
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Removes source EID pattern and latency information from the set of source latency priorities.

3.5. Contact Directives

Contact directives set up, start, stop, and teardown CLA contacts.

3.5.1. Set Up

Name	contact-setup
Function Code	29
Table	Contact Table
Parameter(s)	uint32 *contactId*: unique CLA contact ID, points to row in Contact Table with necessary configurations

Description	Establishes and configures CLA to obtain output queue bundles from Node storage to send to Convergence Layer (CL). It creates the output queue and configures the rate at which CLA sends and receives bundles to and from CL. If CLA type is: • LTP - the directive also configures the destination LTP engine ID • TCPCLP - it also establishes a session with peer entity.

This directive is rejected if setting up the contact would exceed the allowed maximum of simultaneous contacts.

3.5.2. Start

Name	contact-start
ID	30
Table	No
Parameter(s)	uint32 *contactId*: unique CLA contact ID
Description	Starts transferring bundles between the underlying network and the node.

3.5.3. Stop

Name	contact-stop
ID	31
Table	No
Parameter(s)	uint32 *contactId*: unique CLA contact ID
Description	This directive performs the following: • stops transferring bundles to and from CL • requests the CL to cancel the transfers in progress • sends any Compressed Reporting Signals (CRSs) (Multiple bundle status reports compressed into one ADU) and custody signals under construction.

If CL is LTP, the directive requests LTP to cancel active session(s) and notify the node of complete and incomplete bundle transmissions.

3.5.4. Tear Down

Name	contact-teardown
ID	32

Table	No
Parameter(s)	uint32 *contactId*: unique CLA contact ID
Description	Disestablishes CLA, frees all CLA resources, stores any bundles remaining in the egress queue, and deletes custody timers.

3.6. MIB Directives

MIB directives modify the MIB configurations.

3.6.1. Add MIB Array Key*

Name	add-MIB-array-key
Function Code	33
Table	Yes
Parameter(s)	EID Pattern[4] *eidPatterns*: EID patterns to add
Description	Adds the given EID patterns as key to the map of MIB configuration elements and counters accessed by the source EID.

3.6.2. Remove MIB Array Key*

Name	remove-MIB-array-key
Function Code	34
Table	No
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Removes the elements indexed by given EID pattern from the map of MIB configuration elements and counters accessed by source EID.

3.6.3. Set MIB Item

Name	set-MIB-item
Function Code	35
Table	Yes

Parameter(s)	• EID Pattern *eid*: EID(s) to set MIB item for. Only the node MIB is supported for this build. • uint32 enum *mibId*: MIB item ID. See the Field IDs of the Node and Source MIB Configuration Packets/Tables for the enumeration values. • uint32 *value*: value to which the MIB item will be set
Description	Sets the value of a MIB configuration item specified by the parameter.

3.7. Storage Directives

Storage directives add and remove storage allocation. None of the functionality of the storage directives has been implemented as of this build.

3.7.1. Add Storage Allocation*

Name	add-storage-allocation
Function Code	36
Table	Yes Parameter(s)
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Adds the storage partition of a specified size for storing bundles whose source EID matches the given pattern.

3.7.2. Remove Storage Allocation*

Name	remove-storage-allocation
Function Code	37
Table	No
Parameter(s)	uint32 *Placeholder*: Placeholder parameter
Description	Removes the storage partition corresponding to a given EID pattern.

3.7.3. Perform Self-Test*

Name	perform-self-test
Function Code	38
Table	No
Parameter(s)	None
Description	Perform TBD tests, returning pass/fail.

3.8. Routine Directives

Routine directives test aliveness, trigger the node to wake up, and send telemetry packets.

3.8.1. Wakeup

Name	Wakeup
Function Code	39
Table	No
Parameter(s)	None
Description	Triggers BPNode tasks to wake up and start processing directives and bundles, as well as performing maintenance activities, such as time calculation, framework management, and garbage collection.

3.8.2. Send Node MIB Configuration HK

Name	send-node-mib-config-hk
Function Code	40
Table	No
Parameter(s)	None
Description	Sends the Per Node MIB Configuration telemetry packet.

3.8.3. Send Per Source MIB Configuration HK

Name	send-per-source-mib-config-hk
Function Code	41
Table	No
Parameter(s)	None
Description	Sends the Per Source MIB Configuration telemetry packet.

3.8.4. Send Node MIB Counters HK

Name	send-node-mib-counters-hk
Function Code	42
Table	No
Parameter(s)	None

Description	Sends the Per Node MIB Counters telemetry packet.

3.8.5. Send Per Source MIB Counters HK

Name	send-per-source-mib-counters-hk
Function Code	43
Table	No
Parameter(s)	None
Description	Sends the Per Source MIB Counters telemetry packet.

3.8.6. Send Storage HK

Name	send-storage-hk
Function Code	44
Table	No
Parameter(s)	None
Description	Sends the Storage telemetry packet.

3.8.7. Send Channel/Contact Status HK

Name	send-channel-contact-status-hk
Function Code	45
Table	No
Parameter(s)	None
Description	Sends the Channel/Contact Status telemetry packet.

3.8.8. Send Node MIB Reports HK

Name	send-node-mib-reports-hk
Function Code	46
Table	No
Parameter(s)	None
Description	Sends the Per Node MIB Reports telemetry packet.

3.8.9. Noop

Name	Noop
Function Code	0
Table	No
Parameter(s)	None
Description	Ensures that the connection is still alive. The BPNode version information is sent out in an event message.

Chapter 4. Telemetry

BPNode converts telemetry data received from its modules from internal library format to standard CCSDS Space Packets and publishes them to cFS SB to send them to M&C. The DTN Node generates the following telemetry packets:

- Node MIB Configuration
- Per-Source MIB Configuration
- Node MIB Counters
- Per-Source MIB Counters
- Node MIB Reports
- Storage HK
- Channel and Contact Status HK.

The Space Packet secondary header for telemetry indicates host time (absolute time value in an epoch defined by cFS) in seconds and subseconds.

Table 5. Telemetry Packet Secondary Header

Name	Description	Type	Start Byte	Start Bit	Length Bits
Seconds	Seconds since host epoch	uint	6	0	32
Subseconds	Subseconds (one subsecond = 2^{-16} seconds)	uint	8	0	16

In addition, DTN also applies timestamps within the body of generated telemetry packets so that operators can determine the DTN time:

Table 6. DTN Time Terminology

Term	Data Type	Units	Description
currentMonotonicTime	uint64	msec	The current time indicated by the monotonic clock, usually the time elapsed since the clock was last powered on
currentCorrelationFactor	uint64	msec	The offset needed to correlate the current monotonic time to the DTN epoch (DTN Time = currentMonotonicTime + currentCorrelationFactor)

Upon receiving one of the send HK directives, the node generates a CCSDS space packet, adds the relevant telemetry fields, acquires time data, and sends the packet to SB.

The telemetry packets generated by the node are listed in the following sections.

Although the per-source packets are generated by BPNode, in build 7.0 none of the

fields are implemented, so they will always be set to 0. For the remaining packets, any fields in *italics* are similarly not implemented and will always be set to 0.

4.1. Node MIB Configuration

This telemetry packet contains all node MIB configurations variables.

Table 7. MIB Node Configuration

Field ID	Field Name	Field Type	Description
N/A	instanceEID	EID	Endpoint ID of this node
0	*paramBundleSizeNoFragment*	uint32	Maximum size of bundles that can traverse DTN without additional bundle layer fragmentation (provided by DTNNM).
1	paramSetMaxSequenceNum	uint32	Maximum bundle sequence number.
2	*paramSetMaxPayloadLength*	uint32	Maximum payload length for fragmentation.
3	paramSetMaxBundleLength	uint32	Maximum bundle length.
4	*paramSetNodeDtnTime*	uint32	Time being tracked by the node
5	*paramSetBehaviorEventReporting*	uint32	Indication that only events at specified level or above are generated and reported via the M&C interface.
6	paramSetMaxLifetime	uint32	Maximum bundle lifetime allowed on node
N/A	spare	uint32	Spare for alignment

4.2. Per-Source MIB Configuration

This telemetry packet contains the MIB configuration variables for each set of sources.

None of the fields in this packet are implemented.

Field Name	Data Type	# Entries	Description
sourceConfigs	Source Configs Array	10 by default	Configurations for each source (see table below)

The Source Configurations Array contains the following fields for each entry:

Table 8. Source Configuration Array

Field ID	Field Name	Data Type	# Entries	Description
N/A	SrcEIDs	EID Pattern	Max MIB Per Source EID Patterns, 4 by default	Source EID Patterns
6	ParamSetMaxLifetime	uint32	1	Maximum bundle lifetime (in seconds) that ensures that bundle retention until its expiration time will not degrade operation of the receiving node. Used by node to determine when the bundle must be deleted to prevent network performance degradation.
7	ParamSetMaxBSRGenerationRate	uint32	1	Maximum number of bundles per minute that can be generated for each source
8	ParamSetMaxCBRGenerationRate	uint32	1	Maximum number of bundles per minute that can be generated for each source
9	BundleSetBehaviorReceivedBSRGenerate	uint32	1	Flag indicating bundle reception status reports should be generated
10	BundleSetBehaviorAcceptedBSRGenerate	uint32	1	Flag indicating bundle custody accepted status reports should be generated
11	BundleSetBehaviorForwardedBSRGenerate	uint32	1	Flag indicating bundle forwarded status reports should be generated
12	BundleSetBehaviorDeliveredBSRGenerate	uint32	1	Flag indicating bundle delivered status reports should be generated
13	BundleSetBehaviorDeletedBSRGenerate	uint32	1	Flag indicating bundle deleted status reports should be generated

Field ID	Field Name	Data Type	# Entries	Description
14	BundleSetBehaviorReceivedCBRGenerate	uint32	1	Flag indicating bundle reception status reports should be generated
15	BundleSetBehaviorAcceptedCBRGenerate	uint32	1	Flag indicating bundle custody accepted status reports should be generated
16	BundleSetBehaviorForwardedCBRGenerate	uint32	1	Flag indicating bundle forwarded status reports should be generated
17	BundleSetBehaviorDeliveredCBRGenerate	uint32	1	Flag indicating bundle delivered status reports should be generated
18	BundleSetBehaviorDeletedCBRGenerate	uint32	1	Flag indicating bundle deleted status reports should be generated

4.3. Node MIB Counters

This telemetry packet includes all node MIB counter variables.

Table 9. Node MIB Counters

Field ID	Field Name	Data Type	Description
0	aduCountDelivered	uint32	Number of ADUs delivered to an application.
1	aduCountReceived	uint32	Number of ADUs received
2	*bundleCountAbandoned*	uint32	Number of abandoned bundle payloads
3	*bundleCountCustodyRejected*	uint32	Number of unsuccessful custody transfers
4	*bundleCountCustodyRequest*	uint32	Number of bundles requesting custody transfer

Field ID	Field Name	Data Type	Description
5	*bundleCountCustodyReforwarded*	uint32	Number of bundles reforwarded for custody timeout
6	*bundleCountCustodyTransferred*	uint32	Number of successful custody transfers
7	bundleCountDeleted	uint32	Total number of bundle deletions
8	*bundleCountDeletedBadEid*	uint32	Number of bundles deleted to to having an unrecognized destination EID
9	*bundleCountDeletedCancelled*	uint32	Number of bundles deleted due to Transmission Cancelled condition
10	bundleCountDeletedExpired	uint32	Number of bundles deleted due to Lifetime Expired condition
11	*bundleCountDeletedForwardFailed*	uint32	Number of bundles deleted due to Forwarding Failed condition
12	bundleCountDeletedHopExceeded	uint32	Number of bundles deleted due to Hop Limit Exceeded condition
13	bundleCountDeletedInvalidPayload	uint32	Number of bundles deleted due to corrupted payload
14	bundleCountDeletedNoStorage	uint32	Number of bundles deleted due to insufficient storage
15	bundleCountDeletedTooLong	uint32	Number of bundles deleted due to being longer than the maximum bundle length
16	*bundleCountDeletedTrafficPared*	uint32	Number of bundles deleted due to Traffic Pared condition
17	*bundleCountDeletedUnauthorized*	uint32	Number of bundles deleted due to unrecognized source EID
18	bundleCountDeletedUnintelligible	uint32	Number of bundles deleted due to Block Unintelligible condition

Field ID	Field Name	Data Type	Description
19	bundleCount DeletedUnsupportedBlock	uint32	Number of bundles deleted due to Unsupported Block condition
20	bundleCount Delivered	uint32	Number of bundles delivered
21	*bundleCountDepleted*	uint32	Number of bundles whose rejected Custody Signals indicated lack of storage
22	bundleCount Discarded	uint32	Number of bundles discarded
23	bundleCount Forwarded	uint32	Number of bundles forwarded
24	*bundleCountForwardedFailed*	uint32	Number of bundles where forwarding failed
25	*bundleCountFragmented*	uint32	Number of bundles that needed fragmentation
26	*bundleCountFragmentError*	uint32	Number of fragments discarded due to error
27	*bundleCountGeneratedAccepted*	uint32	Number of accepted bundle transmission requests
28	*bundleCountGeneratedCustodySignal*	uint32	Number of custody signals generated
29	*bundleCountGeneratedFragment*	uint32	Number of generated bundle fragments
30	*bundleCountGeneratedRejected*	uint32	Number of rejected bundle transmission requests
31	*bundleCount MaxBsrRateExceeded*	uint32	Number of BSR bundles not sent due to rate limit
32	*bundleCount NoContact*	uint32	Number of bundles whose rejected Custody Signals indicated destination is not reachable before expiration
33	*bundleCount NoFurtherInfo*	uint32	Number of bundles whose rejected Custody Signals indicated No Further Info

Field ID	Field Name	Data Type	Description
34	*bundleCount NoRoute*	uint32	Number of bundles whose rejected Custody Signals indicated the destination is not reachable
35	*bundleCountR eassembled*	uint32	Number of bundles delivered that were reassembled fragments
36	bundleCount Received	uint32	Number of bundles received
37	*bundleCountR eceivedAdmin Record*	uint32	Number of admin records received
38	*bundleCountR eceivedBsrAcc epted*	uint32	Number of Accepted BSRs received
39	*bundleCountR eceivedBsrDel eted*	uint32	Number of Deleted BSRs received
40	*bundleCountR eceivedBsrDel ivered*	uint32	Number of Delivered BSRs received
41	*bundleCountR eceivedBsrFor warded*	uint32	Number of Forwarded BSRs received
42	*bundleCountR eceivedBsrRec eived*	uint32	Number of Received BSRs received
43	*bundleCountR eceivedCrsAcc epted*	uint32	Number of Accepted CRSs received
44	*bundleCountR eceivedCrsDel eted*	uint32	Number of Deleted CRSs received
45	*bundleCountR eceivedCrsDel ivered*	uint32	Number of Delivered CRSs received
46	*bundleCountR eceivedCrsFor warded*	uint32	Number of Forwarded CRSs received
47	*bundleCountR eceivedCrsRec eived*	uint32	Number of Received CRSs received

Field ID	Field Name	Data Type	Description
48	*bundleCountReceivedCustodySignal*	uint32	Number of Custody Signals received
49	*bundleCountReceivedFragment*	uint32	Number of fragment bundles received
50	*bundleCountRedundant*	uint32	Number of bundles where Custody Signals indicated redundancy
51	*bundleCountRejectedCustody*	uint32	Number of bundles whose custody the node rejected
52	*bundleCountReturned*	uint32	Number of bundles returned to sender
53	*bundleCountUnintelligibleBlock*	uint32	Number of bundles for which Custody Signals indicated the bundle contained an unknown block type
54	*bundleCountUnintelligibleEid*	uint32	Number of bundles rejected for unknown EIDs
55	bundleCount UnprocessedBlocks	uint32	Number of unprocessed blocks removed
56	bundleAgent AcceptedDirectiveCount	uint32	Number of accepted control directives received from the M&C interface.
57	bundleAgent RejectedDirectiveCount	uint32	Number of rejected invalid control directives received from the M&C interface.
58	*bundleCountCustodySignalReceived*	uint32	Number of Custody Signal bundles received.
59	*bundleCountGeneratedAnonymous*	uint32	Number of anonymous bundles created
60	*bundleCountGeneratedBsrAccepted*	uint32	Number of BSRs of bundle accepted for custody since the last counter reset.
61	*bundleCountGeneratedBsrDeleted*	uint32	Number of BSRs of bundles deleted since the last counter reset.

Field ID	Field Name	Data Type	Description
62	*bundleCountGeneratedBsrDelivered*	uint32	Number of BSRs of bundles delivered since the last counter reset.
63	*bundleCountGeneratedBsrForwarded*	uint32	Number of BSRs of bundles forwarded since the last counter reset.
64	*bundleCountGeneratedBsrReceived*	uint32	Number of BSRs of bundles received generated since the last counter reset.
65	*bundleCountGeneratedCrs*	uint32	Number of CRSs generated since last counter reset.
66	*bundleCountGeneratedCrsAccepted*	uint32	Number of accepted bundle reports in each CRS since the last counter reset.
67	*bundleCountGeneratedCrsDeleted*	uint32	Number of deleted bundle reports in each CRS since the last counter reset.
68	*bundleCountGeneratedCrsDelivered*	uint32	Number of delivered bundle reports in each CRS since the last counter reset.
69	*bundleCountGeneratedCrsForwarded*	uint32	Number of forwarded bundle reports in each CRS since the last counter reset.
70	*bundleCountGeneratedCrsReceived*	uint32	Number of received bundle reports in each CRS since the last counter reset.
71	*bundleCountGeneratedCustody*	uint32	Number of custody signal bundles generated since the last counter reset.
72	*bundleCountInvalidPrimaryBlock*	uint32	Number of unprocessed bundles received with invalid primary blocks.
73	*bundleCountInCustody*	uint32	Number of bundles in custody
74	*bundleCountMaxCrsRateExceeded*	uint32	Number of CRS bundles not sent to avoid exceeding maximum rate.
75	*bundleCountReceivedCrs*	uint32	Number of CRSs received since last counter reset.

4.4. Per-Source MIB Counters

This telemetry packet includes all MIB counter variables for each set of sources.

None of the fields in this packet are implemented.

Table 10. Source Counters

Field Name	Data Type	# Entries	Description
sourceCounters	Source Counters Array	10 by default	Counters for each source (see table below)

The Source Counters Array contains the following fields for each entry:

Table 11. Source MIB Counters

Field ID	Field Name	Data Type	Description
N/A	sourceEIDs	Array of 4 EID patterns	Source EID patterns for this entry
N/A	activeKeys	uint8	Number of keys are active in the EID patterns
N/A	spare	uint8[3]	Spare for alignment
0	aduCountDelivered	uint32	Number of ADUs delivered to an application.
1	aduCountReceived	uint32	Number of ADUs received
2	*bundleCountAbandoned*	uint32	Number of abandoned bundle payloads
3	*bundleCountCustodyRejected*	uint32	Number of unsuccessful custody transfers
4	*bundleCountCustodyRequest*	uint32	Number of bundles requesting custody transfer
5	*bundleCountCustodyReforwarded*	uint32	Number of bundles reforwarded for custody timeout
6	*bundleCountCustodyTransferred*	uint32	Number of successful custody transfers
7	bundleCountDeleted	uint32	Total number of bundle deletions
8	*bundleCountDeletedBadEid*	uint32	Number of bundles deleted to to having an unrecognized destination EID

Field ID	Field Name	Data Type	Description
9	*bundleCountDeletedCancelled*	uint32	Number of bundles deleted due to Transmission Cancelled condition
10	bundleCountDeletedExpired	uint32	Number of bundles deleted due to Lifetime Expired condition
11	*bundleCountDeletedForwardFailed*	uint32	Number of bundles deleted due to Forwarding Failed condition
12	bundleCountDeletedHopExceeded	uint32	Number of bundles deleted due to Hop Limit Exceeded condition
13	bundleCountDeletedInvalidPayload	uint32	Number of bundles deleted due to corrupted payload
14	bundleCountDeletedNoStorage	uint32	Number of bundles deleted due to insufficient storage
15	bundleCountDeletedTooLong	uint32	Number of bundles deleted due to being longer than the maximum bundle length
16	*bundleCountDeletedTrafficPared*	uint32	Number of bundles deleted due to Traffic Pared condition
17	*bundleCountDeletedUnauthorized*	uint32	Number of bundles deleted due to unrecognized source EID
18	bundleCountDeletedUnintelligible	uint32	Number of bundles deleted due to Block Unintelligible condition
19	bundleCountDeletedUnsupportedBlock	uint32	Number of bundles deleted due to Unsupported Block condition
20	bundleCountDelivered	uint32	Number of bundles delivered
21	*bundleCountDepleted*	uint32	Number of bundles whose rejected Custody Signals indicated lack of storage
22	bundleCountDiscarded	uint32	Number of bundles discarded

Field ID	Field Name	Data Type	Description
23	bundleCount Forwarded	uint32	Number of bundles forwarded
24	*bundleCountForwardedFailed*	uint32	Number of bundles where forwarding failed
25	*bundleCountFragmented*	uint32	Number of bundles that needed fragmentation
26	*bundleCountFragmentError*	uint32	Number of fragments discarded due to error
27	*bundleCountGeneratedAccepted*	uint32	Number of accepted bundle transmission requests
28	*bundleCountGeneratedCustodySignal*	uint32	Number of custody signals generated
29	*bundleCountGeneratedFragment*	uint32	Number of generated bundle fragments
30	*bundleCountGeneratedRejected*	uint32	Number of rejected bundle transmission requests
31	*bundleCount MaxBsrRateExceeded*	uint32	Number of BSR bundles not sent due to rate limit
32	*bundleCount NoContact*	uint32	Number of bundles whose rejected Custody Signals indicated destination is not reachable before expiration
33	*bundleCount NoFurtherInfo*	uint32	Number of bundles whose rejected Custody Signals indicated No Further Info
34	*bundleCount NoRoute*	uint32	Number of bundles whose rejected Custody Signals indicated the destination is not reachable
35	*bundleCountReassembled*	uint32	Number of bundles delivered that were reassembled fragments
36	bundleCount Received	uint32	Number of bundles received
37	*bundleCountReceivedAdmin Record*	uint32	Number of admin records received

Field ID	Field Name	Data Type	Description
38	*bundleCountReceivedBsrAccepted*	uint32	Number of Accepted BSRs received
39	*bundleCountReceivedBsrDeleted*	uint32	Number of Deleted BSRs received
40	*bundleCountReceivedBsrDelivered*	uint32	Number of Delivered BSRs received
41	*bundleCountReceivedBsrForwarded*	uint32	Number of Forwarded BSRs received
42	*bundleCountReceivedBsrReceived*	uint32	Number of Received BSRs received
43	*bundleCountReceivedCrsAccepted*	uint32	Number of Accepted CRSs received
44	*bundleCountReceivedCrsDeleted*	uint32	Number of Deleted CRSs received
45	*bundleCountReceivedCrsDelivered*	uint32	Number of Delivered CRSs received
46	*bundleCountReceivedCrsForwarded*	uint32	Number of Forwarded CRSs received
47	*bundleCountReceivedCrsReceived*	uint32	Number of Received CRSs received
48	*bundleCountReceivedCustodySignal*	uint32	Number of Custody Signals received
49	*bundleCountReceivedFragment*	uint32	Number of fragment bundles received
50	*bundleCountRedundant*	uint32	Number of bundles where Custody Signals indicated redundancy
51	*bundleCountRejectedCustody*	uint32	Number of bundles whose custody the node rejected

Field ID	Field Name	Data Type	Description
52	*bundleCountReturned*	uint32	Number of bundles returned to sender
53	*bundleCountUnintelligibleBlock*	uint32	Number of bundles for which Custody Signals indicated the bundle contained an unknown block type
54	*bundleCountUnintelligibleEid*	uint32	Number of bundles rejected for unknown EIDs
55	bundleCount UnprocessedBlocks	uint32	Number of unprocessed blocks removed

4.5. Node MIB Reports

This telemetry packet contains all node MIB reports variables.

Table 12. Node MIB Reports

Field Name	Data Type	Description
systemNodeName	char[32]	Textual name of the entity with DTN functionality. This name is human readable and used to unambiguously identify a node in the network.
systemNodeOwner	char[32]	Textual identifier for the primary manager of the node, who allocates node resources or functions.
systemSoftwareExec	char[32]	Textual identification of the underlying operating system or executive that controls the resources upon which the DTN functionality is running.
systemSoftwareExecVersion	char[32]	Textual representation of the version and patch-level of the software defined via nodeExecutive. Need to know the full OS with version and patch numbers.
bundleAgentSoftwareVersion	char[32]	Version of the BPA
bundleAgentOperationalState	char[32]	Operational state of the BPA
bundleAgentConfiguration	char[32]	Indication of the BPA configuration
paramSupportedCLAs	char[32]	List of supported CLAs
nodeActiveEndpoints	char[32]	List of active endpoints
systemNodeUpTime	uint32	Time in seconds since node has been reinitialized
bundleAgentAvailableStorage	uint32	Amount of memory initially allocated for bundle storage.

Field Name	Data Type	Description
kbytesCountStorageAvailable	uint32	Kilobytes of storage that is free
bundleCountStored	uint32	Number of bundles currently in storage
BundleIngressRateBytesPerSec	uint32	Rate of bundles received from CLAs in bytes per second
BundleIngressRateBundlesPerSec	uint32	Rate of bundles received from CLAs in bundles per second
BundleEgressRateBytesPerSec	uint32	Rate of bundles forwarded to CLAs in bytes per second
BundleEgressRateBundlesPerSec	uint32	Rate of bundles forwarded to CLAs in bundles per second
BundleIngestedRateBytesPerSec	uint32	Rate of bundles received locally in bytes per second
BundleIngestedRateBundlesPerSec	uint32	Rate of bundles received locally in bundles per second
BundleDeliveryRateBytesPerSec	uint32	Rate of bundles delivered locally in bytes per second
BundleDeliveryRateBundlesPerSec	uint32	Rate of bundles delivered locally in bundles per second
BundleIngressRejectedRateBytesPerSec	uint32	Rate of bundles received and rejected from CLAs in bytes per second
BundleIngressRejectedRateBundlesPerSec	uint32	Rate of bundles received and rejected from CLAs in bundles per second
Spare	uint32	Spare for alignment
nodeStartupCounter	uint32	Number of times node has started up

4.6. Storage Telemetry

This packet contains telemetry values related to storage or memory usage.

Table 13. Storage Telemetry

Field Name	Data Type	Description
BytesMemInUse	size_t	Bytes of memory currently in use
BytesMemFree	size_t	Bytes of memory that are free
BytesMemHighWater	size_t	Memory high water mark in bytes
KbStorageInUse	size_t	Kilobytes of storage currently in use
KbBundlesInStor	size_t	Kilobytes of storage currently occupied by bundles

4.7. Channel/Contact Status Telemetry

The telemetry packet contains the following components:

Table 14. Channel/Contact Status Telemetry

Field Name	Data Type	# Entries	Description
channelStatus	Channel Status Array	2 by default	Status for each channel (see table below)
contactStatus	Contact Status Array	1 by default	Status for each contact (see table below)

4.7.1. Channel Status Data

The Channel Status Array contains the following fields for each entry:

Table 15. Channel Status Data

Field Name	Data Type	Description
localServiceNum	uint32	Service number for the application sending/receiving ADUs on this channel
state	uint8 enum	Removed (0), Stopped (1), Added (2), or Started (3)
registrationState	uint8 enum	Active (0), PassiveDeferred (1), or PassiveAbandon (2)
spare	uint32	Spare for alignment

4.7.2. Contact Status Data

The Contact Status Array contains the following fields for each entry:

Table 16. Contact Status Data

Field Name	Data Type	Description
state	uint32 enum	Torndown (0), Setup (1), Started (2), or Stopped (3)
spare	uint32	Spare for alignment
EIDs	Array of 3 EID patterns	List of destination EIDs for this contact

Chapter 5. Tables

BPNode contains a built-in table translator and reader of cFS tables. The node receives table parameters through cFE Table Service, parses, translates, and passes them to the relevant internal components, e.g., for node configuration. The tables can be modified by commands to cFE Table Services and/or the specified directive(s). BPNode has the following tables and directives for their implementation:

Table Name	Directive or Ground Table Load
Compressed Reporting	Ground Table Load
Channel Configuration	Ground Table Load
ADU Proxy Configuration	Ground Table Load
Contact Configuration	Ground Table Load
MIB Configuration per Node	`set-MIB-item`
MIB Configuration per Source	`set-MIB-item`, `add/remove-MIB-array-key`
Storage	`add/remove-storage-allocation`
Source Authorization Policy	`add/remove-authorized-sources`
Custody Authorization Policy	`add/remove-authorized-custody-sources`
Custodian Authorization Policy	`add/remove-authorized-custodians`
Report-To-EID Authorization Policy	`add/remove-authorized-report-to-eid`
Source Latency Policy	`add/remove-latency`

Build 7.0 implements only the MIB configuration per node, ADU proxy, channel configuration, and contact configuration tables. Non-implemented fields in those tables are *italicized*. The remaining tables are part of BPNode but not yet used.

5.1. Compressed Reporting

This table can be updated by a ground table load. A `contact-stop` directive as well as CRS time or size triggers will prompt the processor to send an in-progress CRS. This table's functionality has not been implemented as of this build. This table contains the following components:

Table 17. CRS Trigger Data

Field Name	Data Type	# Entries	Description
crsTriggerData	CRS Trigger Array	10 by default	Configurations for each CRS trigger

Table 18. CRS Trigger Data

Field Name	Field Type	Description
destinationEID	EID	Destination EID.

Field Name	Field Type	Description
timeTrigger	uint32	A timeout value that triggers sending a CRS after no longer than the specified value.
sizeTrigger	uint32	Maximum CSR size before it is encoded and sent.

5.2. Channel Configuration

This table captures the configurations for flowing bundles to a client application. After loading the table through cFE Table Services, channel configurations in the table can be added or removed from the node with add-application/remove-application directives. The table contains the following components:

Table 19. Channel Configuration

Field Name	Data Type	# Entries	Description
channelConfigs	Channel Configs Array	Maximum number of channels, 2 by default	Configurations for each channel (see table below)

The Channel Configurations Array contains the following fields for each entry:

Table 20. Channel Configuration Array Entry Fields

Field Name	Data Type	Description
AddAutomatically	bool	Load this configuration upon node startup.
RequestCustody	bool	Whether to request custody.
AduWrapping	bool	Whether to wrap an ADU in a CCSDS header upon egress.
AduUnwrapping	bool	Whether to unwrap an ADU in a CCSDS header upon ingress.
RegistrationState	uint8 enum	Active (0), PassiveDeferred (1), or PassiveAbandon (2).
HopLimit	uint8	Maximum number of forwards/hops.
CrcType	uint8 enum	Primary block CRC type: None (0), CRC-16 (1), or CRC-32 (2).
Spare	uint8	Spare for alignment
IngressBitsPerCycle	size_t	Maximum number of bits to ingress per wakeup cycle. Note: defaut wakeup rate is 10Hz.
EgressBitsPerCycle	size_t	Maximum number of bits to ingress per wakeup cycle. Note: defaut wakeup rate is 10Hz.
LocalServiceNumber	uint32	Local service number (node number assumed).
MaxBundlePayloadSize	uint32	Maximum bundle payload size.

Field Name	Data Type	Description
BundleProcessingCtrlFlags	uint64	RFC-9171 Bundle Processing Control Flags
Lifetime	uint64	Lifetime for all bundles.
DestEID	EID	Destination EID
ReportToEID	EID	EID to send status reports to.
PrevNodeBlkConfig	Canonical Block Config	Configurations for the previous node block.
AgeBlkConfig	Canonical Block Config	Configurations for the age block.
HopCountBlkConfig	Canonical Block Config	Configurations for the hop count block.
PayloadBlkConfig	Canonical Block Config	Configurations for the payload block.

Table 21. Canonical Block Configuration

Field Name	Field Type	Description
includeBlock	bool	Whether to include this block (must be true for the payload block)
CrcType	uint8 enum	CRC type: None (0), CRC-16 (1), or CRC-32 (2).
Spare	uint16	Spare for alignment
BlockNum	uint32	Block number
BlockProcessingCtrlFlags	uint64	Canonical block processing control flags

5.3. ADU Proxy Configuration

This table is the only table not managed by BPLib since it is specific to cFS. It defines the cFS-specific configurations for each ADU Proxy channel. The table contains the following components:

Table 22. ADU Configuration

Field Name	Data Type	# Entries	Description
aduConfigs	ADU Configs Array	Maximum number of channels, 2 by default	Configurations for each channel (see table below)

The ADU Configurations Array contains the following fields for each entry:

Table 23. ADU Configurations Array Entry fields

Field Name	Data Type	# Entries	Description
SendToMID	uint32	1	Outgoing ADU message ID (if packetization is enabled).
NumRecvFromMIDs	uint32	1	Number of valid MIDs in RecvFromMIDs array.
MsgLims	uint32	10	Array of message limits corresponding to each of the message IDs
RecvFromMIDs	uint32	10	Array of MIDs to which the node will subscribe.

5.4. Contact Configuration

This table captures configurations for flowing bundles to a CLA. After loading the table through cFE Table Services, contact configurations in the table can be added or removed from the node with the contact-setup/contact-teardown directives. The table contains the following components:

Table 24. Contact Configuration

Field Name	Data Type	# Entries	Description
contactConfigs	Contact Configs Array	Max contacts TBD	Configurations for each contact (see table below)

The Contact Configurations Array contains the following fields for each entry. The following Convergence Layer Protocols are planned to be supported:

- UDP - User Datagram Protocol
- TCP - Transmission Control Protocol (with a TCP Convergence Layer Protocol (TCPCLP) header
- EPP - Encapsulation Packet Protocol
- LTP - Licklider Transmission Protocol.

As of build 7.0, only UDP is presently supported.

Table 25. Contact Configurations Array Entry Fields

Field Name	Field Type	Description
destEIDs	EID pattern array of size 3 by default	Destination EID patterns corresponding to this contact
claType	uint32 enum	Type of CLA. UDP (0), TCP (1), EPP (2), or LTP (4)
claInAddr	char[10]	CLA ingress IP address

Field Name	Field Type	Description
claOutAddr	char[10]	CLA egress IP address
claInPort	uint16	CLA ingress port number
claOutPort	uint16	CLA egress port number
retransmitTimeout	uint32	Bundle reforwarding timeout.
csTimeTrigger	uint32	Custody signal time trigger in seconds.
csSizeTrigger	uint32	Custody signal size trigger.
ingressBitsPerCycle	size_t	Maximum bits to ingress per wakeup cycle. Note: defaut wakeup rate is 10Hz.
egressBitsPerCycle	size_t	Maximum bits to egress per wakeup cycle. Note: defaut wakeup rate is 10Hz.

5.5. MIB Node Configuration

This table defines the MIB configurations for the node:

Table 26. MIB Node Configuration

Field ID	Field Name	Field Type	Description
N/A	instanceEID	EID	Endpoint ID of this node
0	*paramBundleSizeNoFragment*	uint32	Maximum size of bundles that can traverse DTN without additional bundle layer fragmentation (provided by DTNNM).
1	paramSetMaxSequenceNum	uint32	Maximum bundle sequence number.
2	*paramSetMaxPayloadLength*	uint32	Maximum payload length for fragmentation.
3	paramSetMaxBundleLength	uint32	Maximum bundle length.
4	*paramSetNodeDtnTime*	uint32	Time being tracked by the node
5	*paramSetBehaviorEventReporting*	uint32	Indication that only events at specified level or above are generated and reported via the M&C interface.
6	paramSetMaxLifetime	uint32	Maximum bundle lifetime allowed on node
N/A	spare	uint32	Spare for alignment

5.6. MIB Source Configuration

This table defines the MIB configurations for each specified source. This table's functionality has

not been implemented as of this build.

Table 27. Source Configuration

Field Name	Data Type	# Entries	Description
SourceConfigs	Source Configs Array	Max sources, 10 by default	Configurations for each source (see table below)

The Source Configurations Array contains the following fields for each entry:

Table 28. Source Configuration Array

Field ID	Field Name	Data Type	# Entries	Description
N/A	SrcEIDs	EID Pattern	Max MIB Per Source EID Patterns, 4 by default	Source EID Patterns
6	ParamSetMaxLifetime	uint32	1	Maximum bundle lifetime (in seconds) that ensures that bundle retention until its expiration time will not degrade operation of the receiving node. Used by node to determine when the bundle must be deleted to prevent network performance degradation.
7	ParamSetMaxBSRGenerationRate	uint32	1	Maximum number of bundles per minute that can be generated for each source
8	ParamSetMaxCBRGenerationRate	uint32	1	Maximum number of bundles per minute that can be generated for each source
9	BundleSetBehaviorReceivedBSRGenerate	uint32	1	Flag indicating bundle reception status reports should be generated
10	BundleSetBehaviorAcceptedBSRGenerate	uint32	1	Flag indicating bundle custody accepted status reports should be generated
11	BundleSetBehaviorForwardedBSRGenerate	uint32	1	Flag indicating bundle forwarded status reports should be generated

Field ID	Field Name	Data Type	# Entries	Description
12	BundleSetBehaviorDeliveredBSRGenerate	uint32	1	Flag indicating bundle delivered status reports should be generated
13	BundleSetBehaviorDeletedBSRGenerate	uint32	1	Flag indicating bundle deleted status reports should be generated
14	BundleSetBehaviorReceivedCBRGenerate	uint32	1	Flag indicating bundle reception status reports should be generated
15	BundleSetBehaviorAcceptedCBRGenerate	uint32	1	Flag indicating bundle custody accepted status reports should be generated
16	BundleSetBehaviorForwardedCBRGenerate	uint32	1	Flag indicating bundle forwarded status reports should be generated
17	BundleSetBehaviorDeliveredCBRGenerate	uint32	1	Flag indicating bundle delivered status reports should be generated
18	BundleSetBehaviorDeletedCBRGenerate	uint32	1	Flag indicating bundle deleted status reports should be generated

5.7. Policy by EID Pattern Authorization Tables

The Policy by EID Pattern Authorization Tables are: Source Authorization Policy, Custodian Authorization Policy, Report-To-EID Authorization Policy, Source Latency Policy, and Storage Allocation. These tables' functionality have not been implemented as of this build.

Note: When a table has only one field its entry is simplified to two columns.

5.8. Source Authorization Policy

This table contains the following components:

authorizedSources (EID Pattern[10]) Source EIDs from which the node is authorized to receive data.

5.9. Custodian Authorization Policy

This table contains the following components:

authorizedCustodians (EID Pattern[10]) Custodian block source node IDs to which the node is authorized to acknowledge custody.

5.10. Report-To-EID Authorization Policy

This table contains the following components:

authorizedReportToEIDs (EID Pattern[10]) Node EIDs to which this node is authorized to send reports.

5.11. Source Latency Policy

This table contains the following components:

Table 29. Latencies

Field Name	Data Type	# Entries	Description
Latencies	Latency Array	Max source latencies, 10 by default	Latencies for each defined source EID (see table below)

The Latency Array contains the following fields for each entry:

sourceEID (EID Pattern) Source EID pattern

latency (uint32 enum) Source latency: Low, Medium, High

spare (uint32) Spare for alignment

5.12. Storage

The storage allocation defines how to partition the storage database by source EIDs. This table's functionality has not been implemented as of this build. This table contains the following components:

Table 30. Partition Configuration

Field Name	Data Type	# Entries	Description
partitionConfigs	Partition Configs Array	Max partitions, 10 by default	Partition sizes for each source EID (see table below)

The Partition Configurations Array contains the following fields for each entry:

sourceEID (EID Pattern array of length 10 by default) Source EID patterns

partitionSize (uint32) Storage partition size

Chapter 6. Events

Events are a subset of telemetry with a real-time transport Quality of Service for local fault handling. When a BPNode event occurs, the node logs messages and reports them to operators in the form of the CCSDS space packets.

BPLib_EM_EventType_DEBUG	(1) Intended only for debugging, not nominal operations
BPLib_EM_EventType_INFORMATION	(2) Identify a state change or action that is nominal
BPLib_EM_EventType_WARNING	(3) Identify a state change or action that is not an error but is off-nominal
BPLib_EM_EventType_ERROR	(4) Identify an error that is not catastrophic (e.g., a bad command)
BPLib_EM_EventType_CRITICAL	(5) Identify errors that are unrecoverable autonomously.

In a cFS-based system, these event types map onto the equivalent cFS event types, with the exception of BPLib_EM_EventType_WARNING, which maps onto CFE_EVS_EventType_INFORMATION.

The sections below group events by subsystem or module that generates them. All event messages are time-stamped.

6.1. BPNode Main Task Events

These events are within the scope of the cFS BPNode app and are issued by the main task.

Table 31. BPNode Main Task Events

Event Mnemonic	Event ID	Event Type	Description
BPNODE_INIT_INF_EID	1	Informational	Issued after a successful app initialization
BPNODE_CC_ERR_EID	2	Error	An invalid command code was received
BPNODE_NOOP_INF_EID	3	Informational	A no-op command was received
BPNODE_MID_ERR_EID	5	Error	An invalid message ID was received
BPNODE_CMD_LEN_ERR_EID	6	Error	A command with an invalid length was received
BPNODE_PIPE_ERR_EID	7	Error	A SB pipe error was received when attempting to read a pipe

Event Mnemonic	Event ID	Event Type	Description
BPNODE_CR_CMD_PIPE_ERR_EID	8	Error	Error creating the SB command pipe
BPNODE_CR_WKP_PIPE_ERR_EID	9	Error	Error creating the SB wakeup pipe
BPNODE_SUB_CMD_ERR_EID	10	Error	Error subscribing to the SB command pipe
BPNODE_SUB_WKP_ERR_EID	11	Error	Error subscribing to the SB wakeup pipe
BPNODE_TBL_REG_ERR_EID	12	Error	Error registering a table
BPNODE_TBL_LD_ERR_EID	13	Error	Error loading a table
BPNODE_TBL_ADDR_ERR_EID	14	Error	Error getting the address for a table
BPNODE_TBL_MNG_ERR_EID	15	Error	Error checking for table updates
BPNODE_FWP_INIT_ERR_EID	16	Error	Error initializing the Framework Proxy
BPNODE_EXIT_CRIT_EID	17	Critical	App is shutting down
BPNODE_AUTO_ADD_APP_INF_EID	18	Informational	Automatically adding and starting an application at initialization
BPNODE_ADU_START_SUB_DBG_EID	23	Debug	Error subscribing to a message ID on a start-application directive
BPNODE_ADU_STOP_UNSUB_DBG_EID	26	Debug	Error unsubscribing to a message ID on a stop-application directive
BPNODE_TIME_INIT_ERR_EID	51	Error	Error initializing Time Management
BPNODE_TIME_WKP_ERR_EID	52	Error	Error performing time maintenance operations
BPNODE_CLA_IN_CFG_PORT_ERR_EID	62	Error	Error setting UDP port for CLA ingress
BPNODE_CLA_IN_CFG_IP_ERR_EID	63	Error	Error setting IP address for CLA ingress
BPNODE_CLA_IN_CFG_SET_RUN_ERR_EID	65	Error	Error setting I/O run state for CLA ingress

Event Mnemonic	Event ID	Event Type	Description
BPNODE_CLA_OUT_CFG_PORT_ERR_EID	84	Error	Error setting UDP port for CLA egress
BPNODE_CLA_OUT_CFG_IP_ERR_EID	85	Error	Error setting IP address for CLA egress
BPNODE_CLA_OUT_CFG_SET_RUN_ERR_EID	87	Error	Error setting I/O run state for CLA egress
BPNODE_NC_AS_INIT_ERR_EID	114	Error	Error initializing NC/AS
BPNODE_DEL_HANDLER_ERR_EID	118	Error	Error installing delete handler
BPNODE_QM_INIT_ERR_EID	122	Error	Error initializing QM
BPNODE_MEM_INIT_ERR_EID	123	Error	Error initializing memory
BPNODE_ADU_OUT_PI_OUT_ERR_EID	124	Error	Error egressing an ADU
BPNODE_NC_CFG_UPDATE_ERR_EID	126	Error	Error managing configuration updates
BPNODE_INIT_NOTIF_CR_ERR_EID	128	Error	Error creating the start work child task notification

6.2. BPNode ADU In Task Events

These events are within the scope of the cFS BPNode app and are issued by the ADU In Tasks. Since there can be multiple ADU In tasks (one for each allowed channel), the channel ID of the task is specified at the beginning of every event message.

Table 32. BPNode ADU In Task Events

Event Mnemonic	Event ID	Event Type	Description
BPNODE_ADU_IN_TOO_BIG_ERR_EID	27	Error	Received an ADU that is larger than the channel configuration table allows
BPNODE_ADU_IN_INIT_SEM_ERR_EID	28	Error	Error creating the initialization semaphore
BPNODE_ADU_IN_NOTIF_ERR_EID	29	Error	Error pending on start work notification
BPNODE_ADU_IN_EXIT_SEM_ERR_EID	30	Error	Error creating the exit semaphore

Event Mnemonic	Event ID	Event Type	Description
BPNODE_ADU_IN_CREATE_ERR_EID	31	Error	Error creating the child task
BPNODE_ADU_IN_RUN_ERR_EID	32	Error	Error taking the initialization semaphore
BPNODE_ADU_IN_INIT_SEM_TK_ERR_EID	33	Error	Error giving the initialization semaphore
BPNODE_ADU_IN_INIT_INF_EID	34	Informational	Child task initialized
BPNODE_ADU_IN_EXIT_CRIT_EID	35	Critical	Child task is shutting down
BPNODE_ADU_IN_NO_ID_ERR_EID	36	Critical	Failed to get task ID
BPNODE_ADU_IN_INV_ID_ERR_EID	37	Error	Unable to match task ID to a channel ID
BPNODE_ADU_IN_CR_PIPE_ERR_EID	38	Error	Error creating ADU in SB pipe
BPNODE_ADU_IN_UNK_EXIT_CRIT_EID	39	Critical	Unable to obtain channel ID of task and shutting down

6.3. BPNode ADU Out Task Events

These events are within the scope of the cFS BPNode app and are issued by the ADU Out Tasks. Since there can be multiple ADU Out tasks (one for each allowed channel), the channel ID of the task is specified at the beginning of every event message.

Table 33. BPNode ADU Out Task Events

Event Mnemonic	Event ID	Event Type	Description
BPNODE_ADU_OUT_INIT_SEM_ERR_EID	40	Error	Error creating the initialization semaphore
BPNODE_ADU_OUT_NOTIF_ERR_EID	41	Error	Error pending on start work notification
BPNODE_ADU_OUT_EXIT_SEM_ERR_EID	42	Error	Error creating the exit semaphore
BPNODE_ADU_OUT_CREATE_ERR_EID	43	Error	Error creating the child task
BPNODE_ADU_OUT_RUN_ERR_EID	44	Error	Error taking the initialization semaphore

Event Mnemonic	Event ID	Event Type	Description
BPNODE_ADU_OUT_INIT_SEM_TK_ERR_EID	45	Error	Error giving the initialization semaphore
BPNODE_ADU_OUT_INIT_INF_EID	46	Informational	Child task initialized
BPNODE_ADU_OUT_EXIT_CRIT_EID	47	Critical	Child task is shutting down
BPNODE_ADU_OUT_NO_ID_ERR_EID	48	Critical	Failed to get task ID
BPNODE_ADU_OUT_INV_ID_ERR_EID	49	Error	Unable to match task ID to a channel ID
BPNODE_ADU_OUT_UNK_EXIT_CRIT_EID	50	Critical	Unable to obtain channel ID of task and shutting down

6.4. BPNode CLA In Task Events

These events are within the scope of the cFS BPNode app and are issued by the CLA In Tasks. Since there can be multiple CLA In tasks (one for each allowed contact), the contact ID of the task is specified at the beginning of every event message.

Table 34. BPNode CLA In Task Events

Event Mnemonic	Event ID	Event Type	Description
BPNODE_CLA_IN_FIND_NAME_ERR_EID	61	Error	Error finding UDP I/O driver
BPNODE_CLA_IN_CFG_DIR_ERR_EID	64	Error	Error setting I/O direction to input
BPNODE_CLA_IN_INIT_SEM_ERR_EID	66	Error	Error creating the initialization semaphore
BPNODE_CLA_IN_INIT_INF_EID	67	Informational	Child task initialized
BPNODE_CLA_IN_NOTIF_ERR_EID	68	Error	Error pending on start work notification
BPNODE_CLA_IN_EXIT_SEM_ERR_EID	69	Error	Error creating the exit semaphore
BPNODE_CLA_IN_CREATE_ERR_EID	70	Error	Error creating the child task
BPNODE_CLA_IN_RUN_ERR_EID	71	Error	Error giving/taking the initialization semaphore

Event Mnemonic	Event ID	Event Type	Description
BPNODE_CLA_IN_EXIT_CRIT_EID	72	Critical	Child task is shutting down
BPNODE_CLA_IN_NO_ID_ERR_EID	73	Critical	Failed to get task ID
BPNODE_CLA_IN_INV_ID_ERR_EID	74	Error	Unable to match task ID to a contact ID
BPNODE_CLA_IN_UNK_EXIT_CRIT_EID	77	Critical	Unable to obtain contact ID of task and shutting down
BPNODE_CLA_IN_IO_READ_ERR_EID	81	Error	Error reading from the UDP port

6.5. BPNode CLA Out Task Events

These events are within the scope of the cFS BPNode app and are issued by the CLA Out Tasks. Since there can be multiple CLA Out tasks (one for each allowed contact), the contact ID of the task is specified at the beginning of every event message.

Table 35. BPNode CLA Out Task Events

Event Mnemonic	Event ID	Event Type	Description
BPNODE_CLA_OUT_LIB_LOAD_ERR_EID	82	Error	Error getting bundle for egress
BPNODE_CLA_OUT_FIND_NAME_ERR_EID	83	Error	Error finding UDP I/O driver
BPNODE_CLA_OUT_CFG_DIR_ERR_EID	86	Error	Error setting I/O direction to output
BPNODE_CLA_OUT_INIT_SEM_ERR_EID	88	Error	Error creating the initialization semaphore
BPNODE_CLA_OUT_INIT_INF_EID	89	Informational	Child task initialized
BPNODE_CLA_OUT_NOTIF_ERR_EID	90	Error	Error pending on start work notification
BPNODE_CLA_OUT_EXIT_SEM_ERR_EID	91	Error	Error creating the exit semaphore
BPNODE_CLA_OUT_CREATE_ERR_EID	92	Error	Error creating the child task
BPNODE_CLA_OUT_RUN_ERR_EID	93	Error	Error giving/taking the initialization semaphore
BPNODE_CLA_OUT_INV_ID_ERR_EID	98	Error	Unable to match task ID to a contact ID

Event Mnemonic	Event ID	Event Type	Description
BPNODE_CLA_OUT_UNK_EXIT_CRIT_EID	100	Critical	Child task is shutting down

6.6. BPNode Generic Worker Task Events

These events are within the scope of the cFS BPNode app and are issued by the Generic Worker tasks. Since there can be multiple Generic Worker tasks, the worker ID of the task is specified at the beginning of every event message.

Table 36. BPNode Generic Worker Events

Event Mnemonic	Event ID	Event Type	Description
BPNODE_GEN_WRKR_SEM_CR_ERR_EID	101	Error	Error creating the initialization semaphore
BPNODE_GEN_WRKR_EXIT_SEM_ERR_EID	102	Error	Error creating the exit semaphore
BPNODE_GEN_WRKR_CREATE_ERR_EID	103	Error	Error creating the child task
BPNODE_GEN_WRKR_RUN_ERR_EID	104	Error	Error taking the initialization semaphore
BPNODE_GEN_WRKR_SEM_INIT_ERR_EID	105	Error	Error giving/taking the initialization semaphore
BPNODE_GEN_WRKR_INIT_INF_EID	106	Informational	Child task initialized
BPNODE_GEN_WRKR_EXIT_CRIT_EID	107	Critical	Child task is shutting down
BPNODE_GEN_WRKR_NO_ID_ERR_EID	108	Error	Failed to get task ID
BPNODE_GEN_WRKR_INV_ID_ERR_EID	109	Error	Unable to match task ID to a worker ID
BPNODE_GEN_WRKR_UNK_EXIT_CRIT_EID	110	Critical	Unable to obtain worker ID of task and shutting down
BPNODE_GEN_WRKR_NOTIF_ERR_EID	111	Error	Error pending on start work notification

Event Mnemonic	Event ID	Event Type	Description
BPNODE_GEN_WRKR_REGISTER_ERR_EID	112	Error	Error registering worker with BPLib
BPNODE_GEN_WRKR_TASKRUN_ERR_EID	113	Error	Error running worker job

6.7. BPLib Node Configuration Events

These events are issued directly by BPLib and relate to directive processing and node configuration. They can be issued by any of the specified tasks, although the task name should be specified at the beginning of every event message if the issuing task is not the main task.

Table 37. BPLib Node Configuration Events

Event Mnemonic	Event ID	Event Type	Description
BPLIB_NC_NOOP_SUCCESS_EID	501	Informational	Success receiving a noop directive
BPLIB_NC_ADD_ALL_APPS_SUCCESS_EID	502	Informational	Success receiving an add-all-applications directive
BPLIB_NC_ADD_APP_SUCCESS_EID	503	Informational	Success receiving an add-appication directive
BPLIB_NC_ADD_AUTH_CUSTODIANS_SUCCESS_EID	504	Informational	Success receiving an add-authorized-custodians directive
BPLIB_NC_ADD_AUTH_CUST_SRCS_SUCCESS_EID	505	Informational	Success receiving an add-authorized-custody-sources directive
BPLIB_NC_ADD_AUTH_RPT_EID_SUCCESS_EID	506	Informational	Success receiving an add-authorized-report-to-eid directive
BPLIB_NC_ADD_AUTH_SRCS_SUCCESS_EID	507	Informational	Success receiving an add-authorized-sources directive
BPLIB_NC_ADD_LATENCY_SUCCESS_EID	508	Informational	Success receiving an add-latency directive
BPLIB_NC_ADD_MIB_ARR_KEY_SUCCESS_EID	509	Informational	Success receiving an add-mib-array-key directive

Event Mnemonic	Event ID	Event Type	Description
BPLIB_NC_ADD_STOR_ALLOC_SUCCESS_EID	510	Informational	Success receiving an add-storage-allocation directive
BPLIB_NC_CLR_VOLATILE_SUCCESS_EID	511	Informational	Success receiving a clear-volatile directive
BPLIB_NC_CONTACT_SETUP_SUCCESS_EID	512	Informational	Success receiving a contact-setup directive
BPLIB_NC_CONTACT_START_SUCCESS_EID	513	Informational	Success receiving a contact-start directive
BPLIB_NC_CONTACT_STOP_SUCCESS_EID	514	Informational	Success receiving a contact-stop directive
BPLIB_NC_CONTACT_TEARDOWN_SUCCESS_EID	515	Informational	Success receiving a contact-teardown directive
BPLIB_NC_INIT_BNDL_STOR_SUCCESS_EID	516	Informational	Success receiving an initialize-bundle-storage directive
BPLIB_NC_PERFORM_SELF_TEST_SUCCESS_EID	517	Informational	Success receiving a perform-self-test directive
BPLIB_NC_REBUILD_BNDL_META_SUCCESS_EID	518	Informational	Success receiving a rebuild-bundle-metadata directive
BPLIB_NC_RELOAD_SVD_DATA_SUCCESS_EID	519	Informational	Success receiving a reload-saved-data directive
BPLIB_NC_RESET_ALL_CTRS_SUCCESS_EID	520	Informational	Success receiving a reset-all-counters directive
BPLIB_NC_RESET_BNDL_CTRS_SUCCESS_EID	521	Informational	Success receiving a reset-bundle-counters directive
BPLIB_NC_RESET_CTR_SUCCESS_EID	522	Informational	Success receiving a reset-counter directive
BPLIB_NC_RESET_ERR_CTRS_SUCCESS_EID	523	Informational	Success receiving a reset-error-counters directive

Event Mnemonic	Event ID	Event Type	Description
BPLIB_NC_RESET_SRC_CTRS_SUCCESS_EID	524	Informational	Success receiving a reset-source-counters directive
BPLIB_NC_RM_APP_SUCCESS_EID	525	Informational	Success receiving a remove-application directive
BPLIB_NC_RM_AUTH_CUSTODIANS_SUCCESS_EID	526	Informational	Success receiving a remove-authorized-custodians directive
BPLIB_NC_RM_AUTH_CUST_SRCS_SUCCESS_EID	527	Informational	Success receiving a remove-authorized-custody-sources directive
BPLIB_NC_RM_AUTH_RPT_EID_SUCCESS_EID	528	Informational	Success receiving a remove-authorized-report-to-eid directive
BPLIB_NC_RM_AUTH_SRCS_SUCCESS_EID	529	Informational	Success receiving a remove-authorized-sources directive
BPLIB_NC_RM_LATENCY_SUCCESS_EID	530	Informational	Success receiving a remove-latency directive
BPLIB_NC_RM_MIB_ARR_KEY_SUCCESS_EID	531	Informational	Success receiving a remove-mib-array-key directive
BPLIB_NC_RM_STOR_ALLOC_SUCCESS_EID	532	Informational	Success receiving a remove-storage-allocation directive
BPLIB_NC_SET_MIB_ITEM_SUCCESS_EID	533	Informational	Success receiving a set-mib-item directive
BPLIB_NC_SET_REGI_STAT_SUCCESS_EID	534	Informational	Success receiving a set-registration-state directive
BPLIB_NC_START_ALL_APPS_SUCCESS_EID	535	Informational	Success receiving a start-all-applications directive
BPLIB_NC_START_APP_SUCCESS_EID	536	Informational	Success receiving a start-application directive
BPLIB_NC_STOP_APP_SUCCESS_EID	537	Informational	Success receiving a stop-application directive

Event Mnemonic	Event ID	Event Type	Description
BPLIB_NC_VERIF_BNDL_META_SUCCESS_EID	538	Informational	Success receiving a verify-bundle-metadata directive
BPLIB_NC_VERIF_BNDL_SUCCESS_EID	539	Informational	Success receiving a verify-bundle-storage directive
BPLIB_NC_ADD_ALL_APPS_ERR_EID	540	Error	Error receiving an add-all-applications directive
BPLIB_NC_ADD_APP_ERR_EID	541	Error	Error receiving an add-application directive
BPLIB_NC_ADD_AUTH_CUSTODIANS_ERR_EID	542	Error	Error receiving an add-authorized-custodians directive
BPLIB_NC_ADD_AUTH_CUST_SRCS_ERR_EID	543	Error	Error receiving an add-authorized-custody-sourcesdirective
BPLIB_NC_ADD_AUTH_RPT_EID_ERR_EID	544	Error	Error receiving an add-authorized-report-to-eid directive
BPLIB_NC_ADD_AUTH_SRCS_ERR_EID	545	Error	Error receiving an add-authorized-sources directive
BPLIB_NC_ADD_LATENCY_ERR_EID	546	Error	Error receiving an add-latency directive
BPLIB_NC_ADD_MIB_ARR_KEY_ERR_EID	547	Error	Error receiving an add-mib-array-key directive
BPLIB_NC_ADD_STOR_ALLOC_ERR_EID	548	Error	Error receiving an add-storage-allocation directive
BPLIB_NC_CLR_VOLATILE_ERR_EID	549	Error	Error receiving a clear-volatile directive
BPLIB_NC_CONTACT_SETUP_ERR_EID	550	Error	Error receiving a contact-setup directive
BPLIB_NC_CONTACT_START_ERR_EID	551	Error	Error receiving a contact-start directive
BPLIB_NC_CONTACT_STOP_ERR_EID	552	Error	Error receiving a contact-stop directive
BPLIB_NC_CONTACT_TEARDOWN_ERR_EID	553	Error	Error receiving a contact-teardown directive

Event Mnemonic	Event ID	Event Type	Description
BPLIB_NC_INIT_BNDL_STOR_ERR_EID	554	Error	Error receiving a initialize-bundle-storage directive
BPLIB_NC_PERFORM_SELF_TEST_ERR_EID	555	Error	Error receiving a perform-self-test directive
BPLIB_NC_REBUILD_BNDL_META_ERR_EID	556	Error	Error receiving a rebuild-bundle-metadata directive
BPLIB_NC_RELOAD_SVD_DATA_ERR_EID	557	Error	Error receiving a reload-saved-data directive
BPLIB_NC_RESET_CTR_ERR_EID	558	Error	Error receiving a reset-counter directive
BPLIB_NC_RESET_SRC_CTRS_ERR_EID	559	Error	Error receiving a reset-source-counters directive
BPLIB_NC_RESET_BNDL_CTRS_ERR_EID	560	Error	Error receiving a reset-bundle-counters directive
BPLIB_NC_RESET_ERR_CTRS_ERR_EID	561	Error	Error receiving a reset-error-counters directive
BPLIB_NC_RM_APP_ERR_EID	562	Error	Error receiving a remove-application directive
BPLIB_NC_RM_AUTH_CUSTODIANS_ERR_EID	563	Error	Error receiving a remove-authorized-custodians directive
BPLIB_NC_RM_AUTH_CUST_SRCS_ERR_EID	564	Error	Error receiving a remove-authorized-custody-sources directive
BPLIB_NC_RM_AUTH_RPT_EID_ERR_EID	565	Error	Error receiving a remove-authorized-report-to-eid directive
BPLIB_NC_RM_AUTH_SRCS_ERR_EID	566	Error	Error receiving a remove-authorized-sources directive
BPLIB_NC_RM_LATENCY_ERR_EID	567	Error	Error receiving a remove-latency directive
BPLIB_NC_RM_MIB_ARR_KEY_ERR_EID	568	Error	Error receiving a remove-mib-array-key directive
BPLIB_NC_RM_STOR_ALLOC_ERR_EID	569	Error	Error receiving a remove-storage-allocation directive

Event Mnemonic	Event ID	Event Type	Description
BPLIB_NC_SET_MIB_ITEM_ERR_EID	570	Error	Error receiving a set-mib-item directive
BPLIB_NC_SET_REGI_STAT_ERR_EID	571	Error	Error receiving a set-registration-state directive
BPLIB_NC_START_ALL_APPS_ERR_EID	572	Error	Error receiving a start-all-applications directive
BPLIB_NC_START_APP_ERR_EID	573	Error	Error receiving a start-application directive
BPLIB_NC_STOP_APP_ERR_EID	574	Error	Error receiving a stop-application directive
BPLIB_NC_VERIF_BNDL_ERR_EID	575	Error	Error receiving a verify-bundle-storage directive
BPLIB_NC_VERIF_BNDL_META_ERR_EID	576	Error	Error receiving a verify-bundle-metadata directive
BPLIB_NC_SEND_NODE_CONFIG_ERR_EID	577	Error	Error receiving a send-node-mib-config-hk directive
BPLIB_NC_SEND_SRC_CONFIG_ERR_EID	578	Error	Error receiving a send-source-mib-config-hk directive
BPLIB_NC_SEND_NODE_CNTRS_ERR_EID	579	Error	Error receiving a send-node-mib-counters-hk directive
BPLIB_NC_SEND_SRC_CNTRS_ERR_EID	580	Error	Error receiving a send-source-mib-counters-hk directive
BPLIB_NC_SEND_STORAGE_ERR_EID	581	Error	Error receiving a send-storage-hk directive
BPLIB_NC_SEND_CONTACTS_ERR_EID	582	Error	Error receiving a send-channel-contact-status-hk directive
BPLIB_NC_SEND_REPORTS_ERR_EID	583	Error	Error receiving a send-node-mib-reports-hk directive
BPLIB_NC_TBL_UPDATE_INF_EID	606	Informational	Updated a configuration table successfully

6.8. BPLib Admin Statistics (AS) Events

These events are issued directly by BPLib and relate to AS operations. They can be issued by any of the specified tasks, although the task name should be specified at the beginning of every event message if the issuing task is not the main task.

Table 38. BPLib Admin Statistics Events

Event Mnemonic	Event ID	Event Type	Description
BPLIB_AS_TAKE_MUTEX_ERR_EID	590	Error	Error taking counters mutex
BPLIB_AS_GIVE_MUTEX_ERR_EID	591	Error	Error giving counters mutex
BPLIB_AS_SET_CTR_ERR_EID	592	Error	Error incrementing/decrementing a counter

6.9. BPLib Storage Events

These events are issued directly by BPLib and relate to Storage operations. They can be issued by any of the specified tasks, although the task name should be specified at the beginning of every event message if the issuing task is not the main task.

Table 39. BPLib Storage Events

Event Mnemonic	Event ID	Event Type	Description
BPLIB_STOR_SQL_LOAD_ERR_EID	608	Error	Error loading bundles from storage for egress
BPLIB_STOR_SQL_STORE_ERR_EID	609	Error	Error storing bundle
BPLIB_STOR_SQL_GC_ERR_EID	610	Error	Error discarding egressed/expired bundles
BPLIB_STOR_DB_FULL_INF_EID	611	Informational	Storage is full, dropping bundles
BPLIB_STOR_DB_GET_SIZE_ERR_EID	612	Error	Error getting database size

6.10. BPLib Bundle Interface Events

These events are issued directly by BPLib and relate to BI and CLA operations. They can be issued by any of the specified tasks, although the task name should be specified at the beginning of every event message if the issuing task is not the main task.

Table 40. BPLib BI/CLA Events

Event Mnemonic	Event ID	Event Type	Description
BPLIB_BI_INGRESS_CBOR_DECODE_INF_EID	650	Error	Error ingressing a bundle
BPLIB_CLA_CONTACT_NO_STATE_CHG_DBG_EID	661	Debug	A contact directive was received that didn’t change the contact state

Event Mnemonic	Event ID	Event Type	Description
BPLIB_CLA_INVALID_CONTACT_ID_DBG_EID	662	Debug	Invalid contact ID provided by a contact directive
BPLIB_CLA_REMOVE_QUEUE_FLUSH_DGB_EID	663	Debug	Error storing bundles in egress queue upon receiving a contact-teardown directive

6.11. BPLib Payload Interface (PI) Events

These events are issued directly by BPLib and relate to PI operations. They can be issued by any of the specified tasks, although the task name should be specified at the beginning of every event message if the issuing task is not the main task.

Table 41. BPLib PI Events

Event Mnemonic	Event ID	Event Type	Description
BPLIB_PI_EGRESS_ERR_EID	682	Error	Error copying ADU out for egress
BPLIB_PI_INGRESS_ERR_EID	683	Error	Error ingressing an ADU
BPLIB_PI_REMOVE_STATE_DBG_EID	684	Debug	Invalid app state upon receiving a remove-application directive
BPLIB_PI_REMOVE_ID_DBG_EID	685	Debug	Invalid channel ID in a remove-application directive
BPLIB_PI_REMOVE_FWP_DBG_EID	686	Debug	Framework-specific error upon receiving a remove-application directive
BPLIB_PI_ADD_ID_DBG_EID	687	Debug	Invalid channel ID in an add-application directive
BPLIB_PI_ADD_STATE_DBG_EID	688	Debug	Invalid app state upon receiving an add-application directive
BPLIB_PI_ADD_FWP_DBG_EID	689	Debug	Framework-specific error upon receiving an add-application directive
BPLIB_PI_START_ID_DBG_EID	690	Debug	Invalid channel ID in a start-application directive
BPLIB_PI_START_STATE_DBG_EID	691	Debug	Invalid app state upon receiving a start-application directive
BPLIB_PI_START_FWP_DBG_EID	692	Debug	Framework-specific error upon receiving a start-application directive
BPLIB_PI_STOP_ID_DBG_EID	693	Debug	Invalid channel ID in a stop-application directive

Event Mnemonic	Event ID	Event Type	Description
BPLIB_PI_STOP_STATE_DBG_EID	694	Debug	Invalid app state upon receiving a stop-application directive
BPLIB_PI_STOP_FWP_DBG_EID	695	Debug	Framework-specific error upon receiving a stop-application directive
BPLIB_PI_REMOVE_QUEUE_FLUSH_DBG_EID	696	Debug	Error storing bundles in egress queue upon receiving a remove-application directive

6.12. BPLib Queue Management (QM) Events

These events are issued directly by BPLib and relate to queue and job operations. They can be issued by any of the specified tasks, although the task name should be specified at the beginning of every event message if the issuing task is not the main task.

Table 42. BPLib QM Events

Event Mnemonic	Event ID	Event Type	Description
BPLIB_QM_EBP_OUT_ERR_EID	700	Error	Error updating extension blocks

Appendix A: Acronyms and Abbreviations

Acronym or Abbreviation	Description
ADU	Application Data Unit
AS	Admin Statistics
BI	Bundle Interface
BP	Bundle Protocol
BPNode	Bundle Protocol Node
BPLib	Bundle Protocol Library
BSR	Bundle Status Report
CBOR	Concise Binary Object Representation
CBR	Compressed Bundle Reporting
CCB	Configuration Control Board
CCSDS	Consultative Committee for Space Data Systems
cFS	core Flight System
CL	Convergence Layer
CLA	Convergence Layer Adapter
CM	Configuration Management
CRC	Cyclic Redundancy Check
CREB	Compressed Reporting Extension Block
CRS	Compressed Reporting Signal
CTDB	Custody Transfer Database
CTEB	Custody Transfer Extension Block
CTI	Custody Transfer Items
DOORS	Dynamic Object-Oriented Requirements System
DTN	Delay/Disruption-Tolerant Networking
DTNNM	DTN Network Management
EID	Endpoint Identifier
FSW	Flight Software
GSFC	Goddard Space Flight Center
HK	Housekeeping
ICD	Interface Control Document
IETF	Internet Engineering Task Force
LTP	Licklider Transmission Protocol

Acronym or Abbreviation	Description
M&C	Monitor and Control
MIB	Management Information Base
MOC	Mission Operations Center
NC	Node Configuration
PI	Payload Interface
QM	Queue Management
RVTM	Requirements Verification Traceability Matrix
SB	Software Bus
SLA	Service Level Agreement
TDMS	Technical Data Management System