

OASIS-CC

*Operations and Science Instrument Support
Command and Control*

Generic Communication User's Guide Unix and VMS Capabilities

For OASIS-CC V02.05.12
and subsequent versions
November 1994

Copyright 1986, 1992
By the Regents of the University of Colorado

OASIS-CC

*Operations and Science Instrument Support
Command and Control*

Generic Communication User's Guide Unix Capabilities

Permission is granted to make and distribute verbatim copies of this manual provided that the copyright notice and these paragraphs are preserved on all copies. The software and any accompanying written materials are provided “as is” without warranty of any kind. The University of Colorado disclaims all warranties of any kind, either expressed or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose.

Mention of any commercial company or product in this document does not constitute an endorsement by the University of Colorado.

Copyright 1986, 1992
By the Regents of the University of Colorado

Table of Contents

- 1. Introduction..... 1-1
 - Intended Audience..... 1-1
 - Document Structure..... 1-1
 - Related Documents..... 1-1
- 2. Overview..... 2-1
 - Supported Functions and Data Streams..... 2-1
 - Functions Provided..... 2-1
 - Supported Data Streams..... 2-1
 - Computer Platforms..... 2-1
 - Communication Services Available On Sparcstation Systems..... 2-1
 - Implementation Assumptions and Limitations..... 2-2
 - Database Commonalities..... 2-2
 - STREAMS table 2-2
- 3. Unix - Serial Communications (RS232 port)..... 3-1
 - Current Limitations..... 3-1
 - Specific Setup..... 3-2
- 4. Unix - TCP/IP Communications..... 4-1
 - Current Limitations..... 4-1
 - Specific Setup..... 4-2

This page is intentionally blank.

1. Introduction

Intended Audience

This document assists application developers planning to use the Generic Communication support provided by OASIS-CC.

Document Structure

Chapter 2 provides an overview of the Generic Communication support provided by OASIS-CC. The assumptions made during implementation and the limitations imposed on users are also provided. The remaining sections describe the available communications services.

Related Documents

The following documents provide additional detail on OASIS-CC and operating system software employed by the OASIS-CC Generic Communications Protocols:

OASIS-CC CSTOL Reference Manual: LASP

OASIS-CC Database Guide: LASP

OASIS-CC System Manager's Guide: LASP

OASIS-CC Quick Reference Manual: LASP

OASIS-CC Installation Guide: LASP

OASIS-CC Application Environment Reference Manual: LASP

Actions Workbench User's Guide: LASP

TAE+ Overview: Century Computing, Inc., Laurel, MD

TAE+ User Interface Developer's Guide: Century Computing, Inc., Laurel, MD.

VMS Networking Manual

Sun OS version 4. Network Programming

This page is intentionally blank.

2. Overview

Supported Functions and Data Streams

A set of generic I/O support capabilities is available in OASIS-CC running on Sparcstation platforms. These capabilities provide communications support without having to add communications code to the OASIS-CC software.

Functions Provided

The Generic Communications protocols provide both uplink and downlink functions. Uplink functions involve command message and routed data transmission. Downlink functions include the decomposition of data streams. The routing of data to control language processors (CLPs) is also provided.

Supported Data Streams

The Generic Communications protocols support telemetry data that are structured either as *packets* or *time division multiplexed (TDM) frames*. Packets are defined as congregate units of data that contain a unique identifier which implies the packet's content, and an identifier that reflects the packet's length. *TDM frames* are congregate units of data that may contain a unique identifier that implies the frame's content. All frames share the same length, and therefore do not require a length identifier within the frame.

Computer Platforms

The Generic Communications support provides several *protocols* that are tailored to take advantage of popular communications services found on Unix platforms.

Communication Services Available On Sparcstation Systems

The following capabilities are supported on Sparcstation platforms running SunOs 4.1.x and Solaris:

- Asynchronous communication via a serial terminal port
- Communication using TCP/IP

Implementation Assumptions and Limitations

The following rules apply to data the Generic Communications protocols will process (note: not all functions are currently implemented):

If sync patterns are present either in packets or TDM frames, they must be byte bounded, starting and ending on a byte boundary.

Packets must have commonly located packet identification and packet length data. Packets where the packet identification implicitly defines the packet length are not supported.

The implementation assumes error-free transmission. As a result, error checking of any kind is not supported.

Any byte reversing of downlink data, if necessary, must be performed by the sender of the downlink data. Any byte reversing of uplink data, if necessary, must be performed by the receiver of the uplink data. Byte reversing is not provided by any of the Generic Communications protocols.

Acknowledgment support is not provided.

Database Commonalities

Standard rules and controls apply to all of the Generic Communication support within OASIS-CC. These rules and controls are implemented through OASIS-CC database tables as follows:

STREAMS table

Asynchronous data streams (such as RS232) must contain a sync pattern field with a minimum length of 8 bits. Data streams delivered via TCP/IP sockets are inherently synchronous, however the data structures they transport may or may not be. OASIS-CC allows for either case. For example, if packets are transmitted via TCP/IP sockets in a synchronous manner, OASIS-CC need not provide synchronization. If however, the packets are transported asynchronously (say only half of the packet is sent per write to the socket), OASIS-CC must be directed to provide the synchronization. In general, OASIS-CC will perform synchronization for data streams if the SYNC_PATTERN_LENGTH is 8 (bits) or greater. If less than 8, then OASIS-CC assumes that the underlying communication protocol provides the synchronization.

The FRAME_LENGTH provides the length of the stream input buffer.

Each packet of a packetized stream must be defined as a secondary stream which can be decomposed and routed individually.

If the primary stream's STREAMS.SECONDARY_OFFSET is 0, then a pre-saved disk copy of the streams decomposition tree is used when the stream is switched on. If other than 0, the decomposition tree is built at the switch on. See Chapter 6 of the OASIS-CC System Manager's Guide for details. (This is implemented for the IP, IP_1, ASYNC_CHAR and ASYNC_CHAR_1 generic protocols.)

Specification of frame identification fields for TDM frames, and subcommutation of frame data is specified through SUBCOM IDs, as presented in Chapter 6 of the *OASIS-CC System Manager's Guide*.

This page is intentionally blank.

3. Unix - Serial Communications (RS232 port)

The implementation uses *termio(4)* with the following initialization:

Return streams are set to *read-only*, forward streams are *write-only*.

Return streams are set to *no-delay*.

Return and forward streams are set up in *raw* mode.

Current Limitations

The following limitations apply:

Only ILKs of type ATOM, SUBSTREAM, SUBCOMM or SUBCOMM_ID are supported;

Substreams and secondary streams should be byte bounded;

Formatted atoms (IS_FORMATTED = TRUE) should be byte bounded;

EU (DATA_TYPE = NEW_EU) unformatted atoms (IS_FORMATTED = FALSE) should be byte bounded;

Only the native floating point format is supported for EU unformatted atoms;

The BITS_ARE_REVERSED field applies only to DN (DATA_TYPE= NEW_DN) unformatted atoms (IS_FORMATTED= FALSE).

No decomposition occurs to the primary stream if the stream is packetized. All packets that must be decomposed or routed must have a corresponding secondary stream.

Specific Setup

The following records must be defined in the OASIS-CC database:

STREAMS	Field Descriptions
STREAM_NAME	Name of the Primary Stream
PROCESSOR	EXTCOMM_SUBSYSTEM
DIRECTION	FORWARD_STREAM or RETURN_STREAM
PROTOCOL	ASYNC_CHAR or ASYNC_CHAR_1* for RETURN_STREAM ASYNCH_ROUTER for FORWARD_STREAM
LINK_NAME	Name to reference the associated LINKS table record
TIME_OUT**	Delay between 2 reads (RETURN_STREAM only)
SECONDARY_OFFSET	If 0, OASIS-CC uses disk copy of the decomposition tree when switching on, otherwise the tree is built when switching on.
FRAME_LENGTH**	Maximum size in bits of FORWARD_STREAM messages
STREAM_TYPE	PRIMARY

* ASYNC_CHAR and ASYNC_CHAR_1 are identical, except ASYNC_CHAR_1 allows the sync_pattern to be expressed correctly – ASYNC_CHAR expects the sync word expressed in bit-reversed form.

** For packetized data streams, TIME_OUT and FRAME_LENGTH will characterize the processing load on the OASIS-CC acquiring the data. The minimum FRAME_LENGTH can be no less than the length of the largest packet, however users should consider an even larger value. The larger the FRAME_LENGTH, the larger the TIME_OUT value can be, thus reducing the workload. Ideally, FRAME_LENGTH/TIME_OUT >> bit rate.

For TDM data streams, The frame length must be set to the length of the minor frame. This results in less control over the processing load on OASIS-CC. This limitation will be rectified in a future release.

LINKS	Field Descriptions
LINK_NAME	The name referenced in the primary STREAMS record
DATA_LINE	port name (e.g. /dev/tty)

LATEST_DATA	Field Descriptions		
	Record 1	Record 2	Record 3
EXTERNAL_ELEMENT ITEM_NAME DATA_CLASS DN_LOW	Stream Name LINE_SPEED NUMERIC line speed*	Stream Name IS_PACKETIZED NUMERIC 0: TDM frames 1: Packets 2: Bit-wise packets**	Stream Name AWAITING_MSG_FRE NUMERIC Number of seconds separating the "Awaiting Data ..." messages in the absence of data.**

- * Legal values for *YOUR_STREAM* LINE_SPEED are: 75,110,150,134 (for 134.5), 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400.
- ** Bit-wise packets implies that the packet ID and packet size are not on byte boundaries. See Records 4-7 in the LATEST_DATA table below.
- *** If this record is not present, OASIS-CC will default to a 30 second message frequency..

The following LATEST_DATA and STREAMS records are only necessary for packetized data streams:

LATEST_DATA	Field Descriptions		
	Record 4	Record 5	Record 6
EXTERNAL_ELEMENT ITEM_NAME DATA_CLASS DN_LOW	Stream Name ID_OFFSET* NUMERIC Offset of the packet ID field	Stream Name ID_SIZE* NUMERIC Size of packet the ID field	Stream Name LENGTH_OFFSET* NUMERIC Offset of the packet length field

LATEST_DATA	Field Descriptions		
	Record 7	Record 8	Record 9
EXTERNAL_ELEMENT ITEM_NAME DATA_CLASS DN_LOW	Stream Name LENGTH_SIZE* NUMERIC Size of the packet length field	Stream Name LENGTH_UNIT NUMERIC Number of bytes per unit length	Stream Name LENGTH_CORRECTIO** NUMERIC Number of bytes to add to packet size*LENGTH_UNIT to get full packet length

- * If *YOUR_STREAM* IS_PACKETIZED=1, then the values contained in these fields are interpreted as bytes. For example, if *YOUR_STREAM* ID_OFFSET=20, and *YOUR_STREAM* ID_SIZE=4, OASIS-CC will look for 4*8, or 32 bits worth of packet identification data 20*8, or 160 bits from the start of the packet. If instead, *YOUR_STREAM* IS_PACKETIZED=2, then the values contained in these fields are interpreted as bits. For example, if *YOUR_STREAM* ID_OFFSET=20, and *YOUR_STREAM* ID_SIZE=4, OASIS-CC will look for 4 bits worth of packet identification data 20 bits from the start of the packet.
- ** Some packetized systems have packets in which the self contained length describes the length of the data portion of the packets only. OASIS-CC must know the full packet length: LENGTH_CORRECTIO is used to specify the number of bytes to add to the value supplied by the packet to arrive at an overall length.

LATEST_DATA (Optional)	Field Descriptions
	Record 9
EXTERNAL_ELEMENT ITEM_NAME DATA_CLASS DN_LOW	Stream Name DEBUG_LEVEL NUMERIC 0: (or if this record is not present): No debug messages are generated. 1: Message is output to the message window each time a packet is recognized.

One of the following secondary stream records is required for each packet that is to be routed or decomposed. The secondary offset and frame length fields are somewhat arbitrary. If OASIS-CC is to route the secondary streams, and the packet header information is to be routed as well, choose a SECONDARY_OFFSET of 1, and a FRAME_LENGTH that corresponds to the actual packet length.

STREAMS	Field Descriptions
STREAM_NAME	Name of the secondary stream
ID_VALUE	The ID value contained in the packet ID field that identifies this secondary stream
SECONDARY_OFFSET	The offset in bits from the beginning of the packet that's desired
PARENT	Name of the associated primary stream
FRAME_LENGTH	The length desired for the secondary stream
STREAM_TYPE	SECONDARY

4. Unix - TCP/IP Communications

The implementation uses the virtual circuit protocol (called *stream socket* in Berkeley terminology) in the Internet Domain. The implementation does not support the datagram protocol. OASIS-CC may be a client or server process. This implementation may be used for inter-process communication with both processes running on the same or different nodes.

Current Limitations

The following limitations apply:

- Only ILKs of type ATOM, SUBSTREAM, SUBCOMM or SUBCOMM_ID are supported;

- Substreams and secondary streams should be byte bounded;

- Formatted atoms (IS_FORMATTED = TRUE) should be byte bounded;

- EU (DATA_TYPE = NEW_EU) unformatted atoms (IS_FORMATTED = FALSE) should be byte bounded;

- Only the native floating point format is supported for EU unformatted atoms;

- The BITS_ARE_REVERSED field applies only to DN (DATA_TYPE= NEW_DN) unformatted atoms (IS_FORMATTED= FALSE).

No decomposition occurs to the primary stream if the stream is packetized. All packets that must be decomposed or routed must have a corresponding secondary stream.

Specific Setup

The following records must be defined in the OASIS-CC database:

STREAMS	Field Descriptions
STREAM_NAME	Name of the Primary Stream
PROCESSOR	EXTCOMM_SUBSYSTEM
DIRECTION	FORWARD_STREAM or RETURN_STREAM
PROTOCOL	IP or IP_1*
LINK_NAME	Name to reference the associated LINKS table record (This is only required if the stream is a client to a server process.)
TIME_OUT**	Delay between 2 reads (RETURN_STREAM only)
SECONDARY_OFFSET	If 0, OASIS-CC uses disk copy of the decomposition tree when switching on, otherwise the tree is built when switching on.
FRAME_LENGTH**	Maximum size in bits of FORWARD_STREAM messages
STREAM_TYPE	PRIMARY

* IP_1 is identical to IP, except that the sync pattern can be expressed correctly, rather than in the bit-reversed form required by IP.

** For packetized data streams, TIME_OUT and FRAME_LENGTH will characterize the processing load on the OASIS-CC acquiring the data. The minimum FRAME_LENGTH can be no less than the length of the largest packet, however users should consider an even larger value. The larger the FRAME_LENGTH, the larger the TIME_OUT value can be, thus reducing the workload. Ideally, FRAME_LENGTH/TIME_OUT >> bit rate.

For TDM data streams, The frame length must be set to the length of the minor frame. This results in less control over the processing load on OASIS-CC. This limitation will be rectified in a future release.

LINKS	Field Descriptions
LINK_NAME	The name referenced in the primary STREAMS record. (This record is only necessary if the primary stream is defined as a client process.)
DATA_LINE	Internet address of the server process sending data to or receiving commands from OASIS-CC.

LATEST_DATA	Field Descriptions			
	Record 1	Record 2	Record 3	Record 4
EXTERNAL_ELEMENT ITEM_NAME DATA_CLASS DN_LOW	Stream Name IS_PACKETIZED* NUMERIC 0: TDM frames 1: Packets 2: Bit-wise packets	Stream Name SOCKET_ADDRESS NUMERIC Socket address of telemetry source or command receiver.	Stream Name MODE NUMERIC 0: OASIS-CC is a client process 1: OASIS-CC is a server Process	Stream Name AWAITING_MSG_FRE NUMERIC Number of seconds separating the "Awaiting Data ..." messages in the absence of data.**

* Bit-wise packets implies that the packet ID and packet size are not on byte boundaries. See Records 5-8 in the LATEST_DATA table below.

** If this record is not present, OASIS-CC will default to a 30 second message frequency..

The following LATEST_DATA and STREAMS records are only necessary for packetized data streams:

LATEST_DATA	Field Descriptions		
	Record 5	Record 6	Record 7
EXTERNAL_ELEMENT ITEM_NAME DATA_CLASS DN_LOW	Stream Name ID_OFFSET* NUMERIC Offset of the packet ID field	Stream Name ID_SIZE* NUMERIC Size of the packet ID field	Stream Name LENGTH_OFFSET* NUMERIC Offset of the packet length field

LATEST_DATA	Field Descriptions		
	Record 8	Record 9	Record 10
EXTERNAL_ELEMENT ITEM_NAME DATA_CLASS DN_LOW	Stream Name LENGTH_SIZE* NUMERIC Size of the packet length field	Stream Name LENGTH_UNIT NUMERIC Number of bytes per unit length	Stream Name LENGTH_CORRECTIO** NUMERIC Number of bytes to add to packet_size*LENGTH_UNIT to get full packet length

* If YOUR_STREAM IS_PACKETIZED=1, then the values contained in these fields are interpreted as bytes. For example, if YOUR_STREAM ID_OFFSET=20, and YOUR_STREAM ID_SIZE=4, OASIS-CC will look for 4*8, or 32 bits worth of packet identification data 20*8, or 160 bits from the start of the packet. If instead, YOUR_STREAM IS_PACKETIZED=2, then the values contained in these fields are interpreted as bits. For example, if YOUR_STREAM ID_OFFSET=20, and YOUR_STREAM ID_SIZE=4, OASIS-CC will look for 4 bits worth of packet identification data 20 bits from the start of the packet.

** Some packetized systems have packets in which the self contained length describes the length of the data portion of the packets only. OASIS-CC must know the full packet length: LENGTH_CORRECTIO is used to specify the number of bytes to add to the value supplied by the packet to arrive at an overall length.

LATEST_DATA (Optional)	Field Descriptions
	Record 10
EXTERNAL_ELEMENT ITEM_NAME DATA_CLASS DN_LOW	Stream Name DEBUG_LEVEL NUMERIC 0: (or if this record is not present): No debug messages are generated. 1: Message is output to the message window each time a packet is recognized.

One of the following secondary stream records is required for each packet that is to be routed or decomposed. The secondary offset and frame length fields are somewhat arbitrary. If OASIS-CC is to route the secondary streams, and the packet header information is to be routed as well, choose a SECONDARY_OFFSET of 1, and a FRAME_LENGTH that corresponds to the actual packet length.

STREAMS	Field Descriptions
STREAM_NAME	Name of the secondary stream
ID_VALUE	The ID value contained in the packet ID field that identifies this secondary stream
SECONDARY_OFFSET	The offset in bits from the beginning of the packet that's desired
PARENT	Name of the associated primary stream
FRAME_LENGTH	The length desired for the secondary stream
STREAM_TYPE	SECONDARY