EDOC020

EUROGAM DATA ACQUISITION SYSTEM

Network Tape System

Edition 3.0
Sept 1995

Nuclear Physics Support Group
Central Laboratory of the Research Councils
Daresbury Laboratory

Introduction

This document describes the Network Tape System (NTS) protocol specification used by a Tape Server forming part of the NSF data acquisition system.

It is intended that this protocol be available over a number of communications media and hence the relevant *Communications Protocol* specification document should be consulted for detail in this area.

This protocol is an application carried by the RPC protocol which is described in the Eurogam document *Remote Procedure Call.*

The general structure of the protocol is described in the Eurogam document *RPC Application Programs.*

History

Edition 2.0 July 1990 - first entry in Eurogam document server

Edition 2.1 Jan 1991 - expand description of response to procedure 7 (inquire device status) and procedure 12 (inquire stream status).

Edition 2.2 Feb 1991 - further information for procedure 7

Edition 2.3 Apr 1991 - add Exabyte 8500 information for procedure 7

Edition 2.4 Oct 1991 - introduce magic number concept to replace block/record number in write data and read data procedures. Additional access modes in open file procedure. Introduce initialise volume, identify volume and associate stream procedures.

Edition 2.5 July 1992 - Add additional information to end of reply to procedure 12 (inquire stream status).

Edition 2.6 January 1994 - Add procedures 18-20 for Exabyte EXB-10i tape handler.

Edition 3.0 Janurary 1995 - Program version now 4. Minor changes to this document but program version changed for EDOC097.

Operation

Version 2.4 of the document corresponds to version 2 of the program protocol. Clients written to conform to version 1 of the program protocol will interwork with a Version 2 server.

Version 2.6 of the document corresponds to version 3 of the program protocol. Clients written to conform to version 1 or version 2 of the program protocol will interwork with a Version 3 server.

Version 3.0 of the document corresponds to version 4 of the program protocol.

Use

Throughout this document *capability* is an 8 byte (64 bit) opaque field generated by the server and used by the client as an access token.

Procedure Summary

1. allocate device
2. mount volume
3. open file
4. read data
5. write data
6. special device commands
7. inquire device status
8. close file
9. dismount volume
10. deallocate device
11. reset client
12. inquire stream status
13. initialise volume
14. identify volume
15. undefined
16. inquire available devices
17. reset tape
18. move tape
19. position tape changer
20. inquire element status

Program Specification

RPC Information

The NTS service uses only the Null authentication

NTS is supported on the UDP/IP transport only

NTS is RPC Program 28000200

This is Version 4 of the protocol

NTS uses UDP port number 10200

State Variables

dev_unalloc = 0

dev_alloc = 1

dev_mounting = 2

dev_mount = 3

dev_opening = 4

dev_open = 5

dev_closing = 6

dev_executing = 7

dev_initialising = 8

dev_identifying = 9

dev_putting = 10

dev_moving = 11

dev_err = 16

Procedure Status

ntserr_inuse = 1

ntserr_stale = 2

ntserr_nores = 3

ntserr_conx = 4

ntserr_dm = 5

ntserr_ral = 6

ntserr_state = 7

ntserr_cap = 8

ntserr_cmd = 10

ntserr_inv = 11

ntserr_nolabel = 12

ntserr_noc = 13

Procedure 0. Null procedure

This procedure does no work. There are no input parameters and no returned results. It is made available in all RPC services to allow server response testing and timing.

Procedure 1. Allocate Device

REQUEST

Allocates the specified device for exclusive use of the client to the supplied *client identifier*.

client identifier
device name length
device name

Client identifier is a 4 byte opaque field supplied by the client to identify this device. It must be different from any other *client identifier* currently in use by this client for previously allocated devices.
Device name specifies the server resource to be allocated. It may either be the name of an actual resource or a generic name given to a group of identical resources. The resources available are server dependant but a list may be obtained by use of the Inquire Available Devices procedure.

RESPONSE - success

success = 0
capability
real device name length
real device name

The response returns a 8 byte (64 bit) *capability* which is an opaque field generated by the server and used by the client as an access token in subsequent procedure requests for the allocated device.
The response also returns the actual device name allocated. This is useful when the request supplies a generic name requesting any one of a group of identical devices.
The current value of *stream state* for this stream changes to *dev_alloc*.
The current value of *procedure status* for this stream changes to *success* (=0).

RESPONSE - failure

procedure status

Procedure status is a 4 byte field which is non zero and indicates the cause of the failure.

Procedure status=	1	client identifier already in use for this client
	3	no free resources
	6	requested resource unavailable

Procedure 2. Mount Volume

REQUEST

Requests that the specified volume be loaded onto the device currently allocated to the client's *client identifier*.

client identifier
capability
volume name length
volume name

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.
Volume name specifies the volume to be loaded onto the allocated device.
The current *stream state* for the stream must be *dev_alloc* otherwise the request will fail.

RESPONSE

procedure status

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	accepted
	2	unknown client identifier for this client
	6	mount request rejected
	7	stream is in wrong state
	8	capability is invalid

If *procedure status* = accepted the current value of the *stream status* for this stream changes to *dev_mounting*.

When the mount operation has been completed successfully (which could take some time) the *stream state* for the stream changes to *dev_mount*.
Should the mount operation fail (for example the requested volume cannot be found) *stream state* reverts to *dev_alloc* and *procedure status* is set appropriately.

See also procedure 14 - Identify volume

Procedure 3. Open File

REQUEST

The device specified by *client identifier* is prepared for data transfer.
In the case of read access the file is located on the medium.
In the case of write access the medium is positioned and file labels written.

client identifier
capability
access mode
label type
record length
block length
file name length
file name

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.

Access mode=	0	update
	1	read only
	2	write only
	8	RAW (update)
	9	RAW (read only)
	10	RAW (write only)
label type=	0	undefined
	1	IBM labels
	2	ANSI labels
	8	unlabelled
record length=	>0	fixed length records ie F format
	0	variable length records ie VB format - IBM D format - ANSI
block length=	0	undefined

If record format = F then *block length* must be a multiple of *record length*.

File name specifies the name of the file to be accessed.
Label type = 0 may be used if the server system determines this information from the medium volume label.
Record length and *block length* need not be supplied for read access if the server system determines this information from the device file labels.
Access mode = RAW may be used by clients writting data in undefined formats. In this case the remaining fields may be set as undefined. The server system may place restrictions on the use of this option for reasons of data security.
The current *stream state* for the stream must be *dev_mount* otherwise the request will fail.

RESPONSE

procedure status

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	accepted
	2	unknown client identifier for this client
	4	connect error
	5	data management error
	7	stream is in wrong state
	8	capability is invalid

If *procedure status* = accepted the current value of the *stream status* for this stream changes to *dev_openning*.

When the open operation has been completed successfully (which could take some time) the *stream state* for the stream changes to *dev_open*.
Should the open operation fail (for example the requested file cannot be found) *stream state* reverts to *dev_mount* and *procedure status* is set appropriately.

Procedure 4. Read Data

REQUEST

Transfers data from the device in use for this client's *client identifier*.
Returns up to *count* bytes of data from the current (or next) record (or block) of data.

client identifier
capability
magic number
offset
count

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.
Magic number is an opaque field supplied by the client to identify this request. It is used by the server to identify retried requests for which the client may have missed the response. The client must use different *magic numbers* for successive requests.
Offset = 0 implies that data should be taken starting at the beginning of the next record (or block) otherwise the data is taken starting at *offset* bytes into the current record (or block).
Access is in terms of logical records on the medium in all cases except for access modes = 8,9 & 10 (ie RAW) in which case it is in terms of physical blocks.
The current *stream state* for the stream must be *dev_open* otherwise the request will fail.

RESPONSE - success

success $= 0$
mode
data length
data

Mode $= 0$ indicates that the end of the current record (or block) has been reached.
Mode $= 1$ indicates that further data exists in the current record (or block).
Mode $= 2$ indicates that end of file has been reached. In this case *data length* $= 0$ and no data is returned.
Mode $= 3$ indicates that a file mark has been found. In this case *data length* $= 0$ and no data is returned.

RESPONSE - failure

procedure status

Procedure status is a 4 byte field which is non zero and indicates the cause of the failure.

Procedure status=	2	unknown client identifier for this client
	5	data management error
	7	stream is in wrong state
	8	capability is invalid

Procedure 5. Write Data

REQUEST

Data is transferred to the device allocated to the client's *client identifier* starting at *offset* bytes into the current record (or block).

client identifier
capability
magic number
offset
spare
data length
data

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.
Magic number is an opaque field supplied by the client to identify this request. It is used by the server to identify retried requests for which the client may have missed the response. The client must use different *magic numbers* for successive requests.
Offset = 0 implies that the following data starts a new record (or block) otherwise the data is a continuation of the current record (or block).
Access is in terms of logical records on the medium in all cases except for access modes = 8,9 & 10 (ie RAW) in which case it is in terms of physical blocks.
The current *stream state* for the stream must be *dev_open* otherwise the request will fail.

RESPONSE

procedure status

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	success
	2	unknown client identifier for this client
	5	data management error
	7	stream is in wrong state
	8	capability is invalid

Procedure 6. Special Device Commands

REQUEST

Requests a device specific command be issued to the device allocated to the client's *client identifier*.

client identifier
capability
magic number
option
parameter

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.
Magic number is an opaque field supplied by the client to identify this request. It is used by the server to identify retried requests for which the client may have missed the response. The client must use different *magic numbers* for successive requests.

Option=	1	rewind file or medium
	2	write *parameter* file marks
	3	space forwards *parameter* records (or blocks)
	4	space backwards *parameter* records (or blocks)
	5	space forwards *parameter* file marks
	6	space backwards *parameter* file marks

Certain commands are only acceptable in Access Mode = RAW.
The current *stream state* for the stream must be *dev_open* otherwise the request will fail.

RESPONSE

procedure status

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	accepted
	2	unknown client identifier for this client
	5	data management error
	7	stream is in wrong state
	8	capability is invalid

If *procedure status* = accepted the current value of the *stream state* for this stream changes to *dev_executing*.

When the operation has been completed (which could take some time) the *stream state* for the stream reverts to dev_open. Should the operation fail then *procedure status* is set appropriately.

Procedure 7. Inquire Device Status

REQUEST

Returns information on the state of the device in use for this client's *client identifier.*

client identifier
capability

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.

RESPONSE - success

The field *device type* is the generic name assigned to the device and may be used to interpret the data which follows.
The fields *tape length* and *tape remaining* are in units which are dependent on *device type.* The field *tape i/o error percentage* is expressed in units of 0.1%. The value -1 indicates that the information cannot be supplied for this device.
Information is further device specific status which may have length = 0 if none is available or can be obtained.

success = 0
device type length
device type
tape length
tape remaining
tape i/o errors
tape i/o error percentage
information length
information

Format of *information* for *device type* EXB-8200.
Information length = 112.
The format of data in these fields is define in the Interface User's Manual for the Exabyte 8200 Tape Subsystem.
Tape length and *tape remaining* are given in units of blocks of length 1 Kbyte.

length = 26
device sense data
length = 17
device mode sense data
length = 52
device inquiry data

Format of *information* for *device type* EXB-8500.
Information length = 172.
The format of data in these fields is define in the User's Manual for the Exabyte 8500 Tape Subsystem.
Tape length and *tape remaining* are given in units of blocks of length 1 Kbyte.

length = 29
device sense data
length = 17
device mode sense data
length = 106
device inquiry data

Format of *information* for *device type* CIPHER-M990.
Information length = 68.
The format of data in these fields is defined in the Technical Manual for the Cipher GCR CacheTape Series M990.
Tape written is given in units of feet.

length = 13
machine status
length = 5
configuration status
length = 11
error history status
length = 14
current status

RESPONSE - failure

procedure status

Procedure status is a 4 byte field which is non zero and indicates the cause of the failure.

Procedure status=	2	unknown client identifier for this client
	8	capability is invalid

Procedure 8. Close File

REQUEST

client identifier
capability

For files open in write mode causes file trailer labels to be written. No further data transfer requests may be issued until a further *Open file* request has been received for this client's *client identifier.*

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.
The current *stream state* for the stream must be *dev_open* otherwise the request will fail.

RESPONSE

procedure status

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	accepted
	2	unknown client identifier for this client
	5	data management error
	7	stream is in wrong state
	8	capability is invalid

If *procedure status* = accepted the current value of the *stream state* for this stream changes to dev_closing.
When the operation has been completed (which could take some time) the *stream state* for the stream changes to *dev_mount*. Should the operation fail then *procedure status* is set appropriately and *stream state* reverts to dev_open.

Procedure 9. Dismount Volume

REQUEST

client identifier
capability

Reverses the action of the mount command.

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.
The current *stream state* for the stream must be *dev_mount* otherwise the request will fail.
Any medium loaded on the device should if possible be unloaded.

RESPONSE

procedure status

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	success
	2	unknown client identifier for this client
	7	stream is in wrong state
	8	capability is invalid

If *procedure status* = success the current value of the *stream state* for this stream changes to *dev_alloc*.

Procedure 10. Deallocate Device

REQUEST

Deallocates the device currently allocated to the client's *client identifier*

client identifier
capability

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.
The current *stream state* for the stream must be *dev_alloc* otherwise the request will fail.

RESPONSE

procedure status

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	success
	2	unknown client identifier for this client
	7	stream is in wrong state
	8	capability is invalid

If *procedure status* = success the current value of the *stream state* for this stream becomes undefined. The server invalidates the *capability* and no further actions are permitted on the device.

Procedure 11. Reset client

Closes all tape files, dismounts all tapes and deallocates all devices for this client.

Procedure 12. Inquire Stream Status

REQUEST

Returns information for this client's *client identifier*.

client identifier
capability

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.

RESPONSE - success

success = 0
last procedure status
current stream state
information length
information
device name length
device name
real device name length
real device name
volume name length
volume name
file name length
file name
access mode
label type
record length
block length
spare
data length
magic number
magic number (write)
magic number (read)
block count
byte count
data rate

This procedure does not change the value of *procedure status* and is permitted for any *stream state*.

Information may contain a text message to explain further the reason for any error state indicated by *procedure status*. *Information length* must be zero if no message is available.

The field *device name* is that specified by the *allocate device* procedure and the field *real device name* is that returned by the procedure.

The field *volume name* is that specified by the *mount volume* procedure.

The fields *file name*, *access mode*, *label type*, *record length* and *block length* are those specified by the *open file* procedure.

The field *data length* is the length of the current block/record (read) or the length of data currently in the block/record buffer (write).

The field *magic number* is that specified by the last *special device commands* procedure.

The field *magic number (read)* is that specified by the last *read data* procedure.

The field *magic number (write)* is that specified by the last *write data* procedure.

RESPONSE - failure

procedure status

Procedure status is a 4 byte field which is non zero and indicates the cause of the failure.

Procedure status=	2	unknown client identifier for this client
	8	capability is invalid

Procedure 13. Initialise Volume

REQUEST

Requests that the volume currently loaded on the device specified by *client identifier* be initialised. Standard volume labels of the type specified by *label type* are written onto the tape. Note that this operation will cause all data currently on the tape to be lost.

client identifier
capability
volume name length
volume name
label type
density
current volume name length
current volume name

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.
Volume name specifies the name to be used when writting the volume labels.
Label type specifies the format type of the volume labels to be written.

label type=	1	IBM labels
	2	ANSI labels

Density specifies the option to be selected on the device when writting the labels. This field is only used by those devices which support more than one data density.

density =	0	device default
	1	low -1600 bpi or 8200 mode on Exabyte 8500
	2	high - 6250 bpi or 8500 mode on Exabyte 8500
	3	compressed mode on Exabyte 8500C/8505

Current volume name must be supplied if the tape already contains a valid volume label. If the tape is currently unlabelled this field should be supplied as a null length string or the name NONE supplied.
The current *stream state* for the stream must be either *dev_alloc* or *dev_mount* otherwise the request will fail.

RESPONSE

procedure status

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	accepted
	2	unknown client identifier for this client
	7	stream is in wrong state
	8	capability is invalid

If *procedure status* = accepted the current value of the *stream status* for this stream changes to *dev_initialising*.

When the initialise operation has been completed successfully (which could take some time) the *stream state* for the stream reverts to dev_alloc.
Should the initialise operation fail *stream state* reverts to *dev_alloc* and *procedure status* is set appropriately.

Procedure 14. Identify Volume

REQUEST

client identifier
capability

Requests that the volume loaded on the device currently allocated to the client's *client identifier* be examined for a valid standard volume label and a Mount Volume operation be automatically performed if one is found.

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.
The current *stream state* for the stream must be *dev_alloc* otherwise the request will fail.

RESPONSE

procedure status

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	accepted
	2	unknown client identifier for this client
	7	stream is in wrong state
	8	capability is invalid

If *procedure status* = accepted the current value of the *stream status* for this stream changes to *dev_identifying*.

When the identify operation has been completed successfully (which could take some time) the *stream state* for the stream changes to *dev_mount* and the label type and volume name found can be obtained by using the Inquire Stream Status procedure.
Should the identify operation fail (for example the tape is not actually labelled) *stream state* reverts to *dev_alloc* and *procedure status* is set appropriately.

Procedure 16. Inquire Available Devices

REQUEST

Requests a list of the device names available which may be specified in the *allocate device* procedure.

There are no procedure specific parameters.

RESPONSE

The available devices are returned in a variable length list. For each device its real device name and generic device name are returned with the current status of the device.

Status=	0	free
	1	allocated
	2	currently not usable

success = 0
valid = *true*
status
real device name length
real device name
device name length
device name
:
value = *false*

Procedure 17. Reset Stream

REQUEST

Closes the stream, dismounts the tape and deallocates the device.

client identifier
capability

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.

RESPONSE

procedure status

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	success
	2	unknown client identifier for this client
	8	capability is invalid

If *procedure status* = success the current value of the *stream state* for this stream changes to *dev_unalloc*.

Procedure 18. Move Tape

REQUEST

Requests the device to move a tape from one internal location to another. This procedure is available for multi-tape handling systems.

client identifier
capability
magic number
source location
destination location

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.
Magic number is an opaque field supplied by the client to identify this request. It is used by the server to identify retried requests for which the client may have missed the response. The client must use different *magic numbers* for successive requests.
Source location and *destination location* specify the source and destination locations of the tape. The values are device specific.
Values of *locations* for Exabyte EXB-10i Cartridge Handling Subsystem.

Location =	0	Cartridge Tape Subsystem (CTS), the tape drive
	1-10	Tape storage slots
	11	Cartridge Handling Mechanism (CHM)

The current *stream state* for the stream must be *dev_alloc* otherwise the request will fail.

RESPONSE

procedure status

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	accepted
	2	unknown client identifier for this client
	5	data management error
	7	stream is in wrong state
	8	capability is invalid

If *procedure status* = accepted the current value of the *stream status* for this stream changes to *dev_moving*.

When the move operation has been completed (which could take some time) the *stream state* reverts to *dev_alloc*.

Procedure 19. Position Tape Changer

REQUEST

Requests the device to move the changer mechanism to a specified internal location. This procedure is available for multi-tape handling systems.

client identifier
capability
destination

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.
Destination specifies the location that the mechanism is to be moved to. The value is device specific.
Values of *destination* for Exabyte EXB-10i Cartridge Handling Subsystem.

Location =	0	Cartridge Tape Subsystem (CTS), the tape drive
	1-10	Tape storage slots
	11	Cartridge Handling Mechanism (CHM)

This procedure can be used to pre-position the mechanism before a subsequent Move Tape request. If the location is specified as the CHM itself, the mechanism will be parked to allow access to the cartridge storage slots and the CTS.

Location = -1. Used to enable the Handling Subsystem to update its information on storage occupancy if it wishes. The element status information can be retrieved by procedure 20. The procedure to test all locations in the stacker for occupancy takes about 1 minute.

RESPONSE

procedure status

Procedure status is a 4 byte field which indicates the result of the request.

Procedure status=	0	accepted
	2	unknown client identifier for this client
	5	data management error
	7	stream is in wrong state
	8	capability is invalid

If *procedure status* = accepted the current value of the *stream status* for this stream changes to *dev_moving*.

When the move operation has been completed (which could take some time) the *stream state* reverts to *dev_alloc*.

Procedure 20. Inquire Element Status

REQUEST

Requests information about an internal element of the device allocated to the client's *client identifier*. This procedure is available for multi-tape handling systems.

client identifier
capability
location

Client identifier identifies the device for the client and *capability* is the corresponding access key provided by the server for this device.
Location specifies the location for which the status is requested. The value is device specific.
Value of *location* for Exabyte EXB-10i Cartridge Handling Subsystem.

Location =	0	Cartridge Tape Subsystem (CTS), the tape drive
	1-10	Tape storage slots
	11	Cartridge Handling Mechanism (CHM)
	-1	Cartridge Subsystem

This procedure is permitted for any *stream state*.

The actual hardware it not accessed by this procedure to ensure consistent response time for this procedure. (See procedure 19)

RESPONSE - success

The values are device specific.

success = 0
element state
occupancy
source
information length
information

Values for Exabyte EXB-10i Cartridge Handling Subsystem.

Element state=	0	normal
	1	cartridge holder missing
	2	status unknown
	3	CTS door open
Occupancy=	0	empty
	1	full

Source identifies the last location from which the cartridge was moved.

Source=	0	Cartridge Tape Subsystem (CTS), the tape drive
	1-10	Tape storage slots
	11	Cartridge Handling Mechanism (CHM)

If on request location = -1 then the status of the Cartridge Subsystem hardware is returned. The status *busy* is returned if the Handling Subsystem is currently updating its information on storage occupancy as a result of a previous request to procedure 20 with location = -1.

Information is a text string which may contain further information about the state of the Cartridge Handling Subsystem.

element state =	0	idle
	10	busy

RESPONSE - failure

procedure status

Procedure status is a 4 byte field which is non zero and indicates the cause of the failure.

Procedure status=	2	unknown client identifier for this client
	8	capability is invalid