EDOC315

MIDAS DATA ACQUISITION SYSTEM

Event Data Transfer - RPC Program Specification

Edition 1.0
April 1997

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

Introduction

This document describes the Event Data Transfer RPC procedure used within the Midas Data Acquisition System. The procedure is used to transfer formatted event data between an Event Builder System and the Midas Tape Server. It is also used to broadcast formatted event data when the Event Builder and Tape Server are combined into a single system.

A single procedure is defined within the Tape Server RPC program which should be consulted for further details. See documents EDOC020, EDOC097, EDOC304 and EDOC305.

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

History

Edition 1.0 April 1997 - first entered into document database.

Operation

Formatted event data blocks are 4 Kbytes or 16 Kbytes in length (dependent on event builder implementation). These are sent as a sequence of up to 12 data fragments dependent on the network medium type being used.

The server operates a "window" for each data stream which is the number of data fragments which the client may send. The data fragments are sent using the Event Data Transfer RPC procedure but unlike standard RPC procedures the server normally does not send a response unless requested by the client. The procedure reply may be sent by the server at any time and returns to the client the status of all data streams which includes an update to the permitted transfer window.

This transfer protocol does not claim to be a "reliable" data transfer protocol in that no attempt is made to ensure that all data from the client is received correctly by the server. There is however a significant improvement in the data rates achievable by the method over the standard "reliable" methods. The server maintains statistics on the number of data fragments known to have been lost using the procedure sequence number. Implementations can tune the data transfer "window" sizes used to the hardware characteristics of the network interfaces. In this way actual data lose can be made zero or very close to zero over long periods of time.

Program Specification

RPC Information

The EGTS service uses only the Null authentication

This is supported on the UDP/IP transport only

This is RPC Program 28000205

This is Version 4 of the RPC program

Uses UDP port numbers 10205, 10206 and 10207

The UDP port number is selected by the data acquisition instance in use. Several data acquisition systems using different port numbers can be supported by one tape server system.

Procedure 31. Transfer Event Data

REQUEST

Transfers a fragment of the current event data block from the client (event builder) to the server (tape server).

EG data stream
capability
transfer options
sequence number
offset
event data length
event data

EG data stream identifies the logical data stream for this data fragment. It should be in the range 1 to 4. Many event builder implementations only support one data stream in which case *EG data stream* always has the value 1.
If *EG data stream* = 0 then no further parameters or data follow and the server will send a reply. This can be used by the client to force the server to return its stream status information.

capability is an 8 byte (64 bit) opaque field generated by the server and used by the client as an access token. For the event data transfer procedure the capability field always contains nulls.

transfer options
Least significant bit (End bit) = 1 - The following data is the last fragment of the current data block. The next request should have offset = 0.
Least but one significant bit (Reply bit) = 1 - The client does not require that the server reply to this request.

sequence number should be incremented for each procedure call. At any time the client may reset the sequence to zero. The server will not consider it to be a fatal error if it detects a sequence number error and will reset its expected sequence number using the value received.

offset is the start address of this data fragment in the current data block. Offset = 0 implies that the following data fragment starts a new data block.
event data length and *event data* define the data fragment.

RESPONSE

The standard reply from the server is sent only for those requests for which a reply is required. The state of all 4 EG data streams is sent regardless of the number used by the client implementation.

procedure status = 0
EGTS state
value=true
EG stream
EG stream state
sequence
window
repeat for all 4 EG streams
value=false

EGTS state=	1	halted
	2	going

The fields *EG stream*, *EG stream state*, *sequence* and *window* are repeated for all 4 EG data streams.

EGTS stream state=	1	disabled
	2	enabled

sequence - last sequence received for this EG data stream

window - number of data fragment requests permitted for this EG data stream