A Serial Link for Storage Subsystems

I D Judd

IBM (UK) Ltd
Mail Point 200
Hursley Park
Winchester
England SO21 2JN
Tel: 011-44-703-701421
Fax: 011-44-703-705106

15 January 1992

Table of Contents

IBM

- Serial Storage Architecture (SSA)

  SSA-0: General-purpose transport layer

  SSA-1: Intermediate-level disk orders

  SSA-2: SCSI mapping

- An enhancement of the IBM 9333 serial link:

  8B/10B code

  Choice of data rates

  Device-independence

  Daisy-chain and loop topologies (Proposed)

- Open architecture

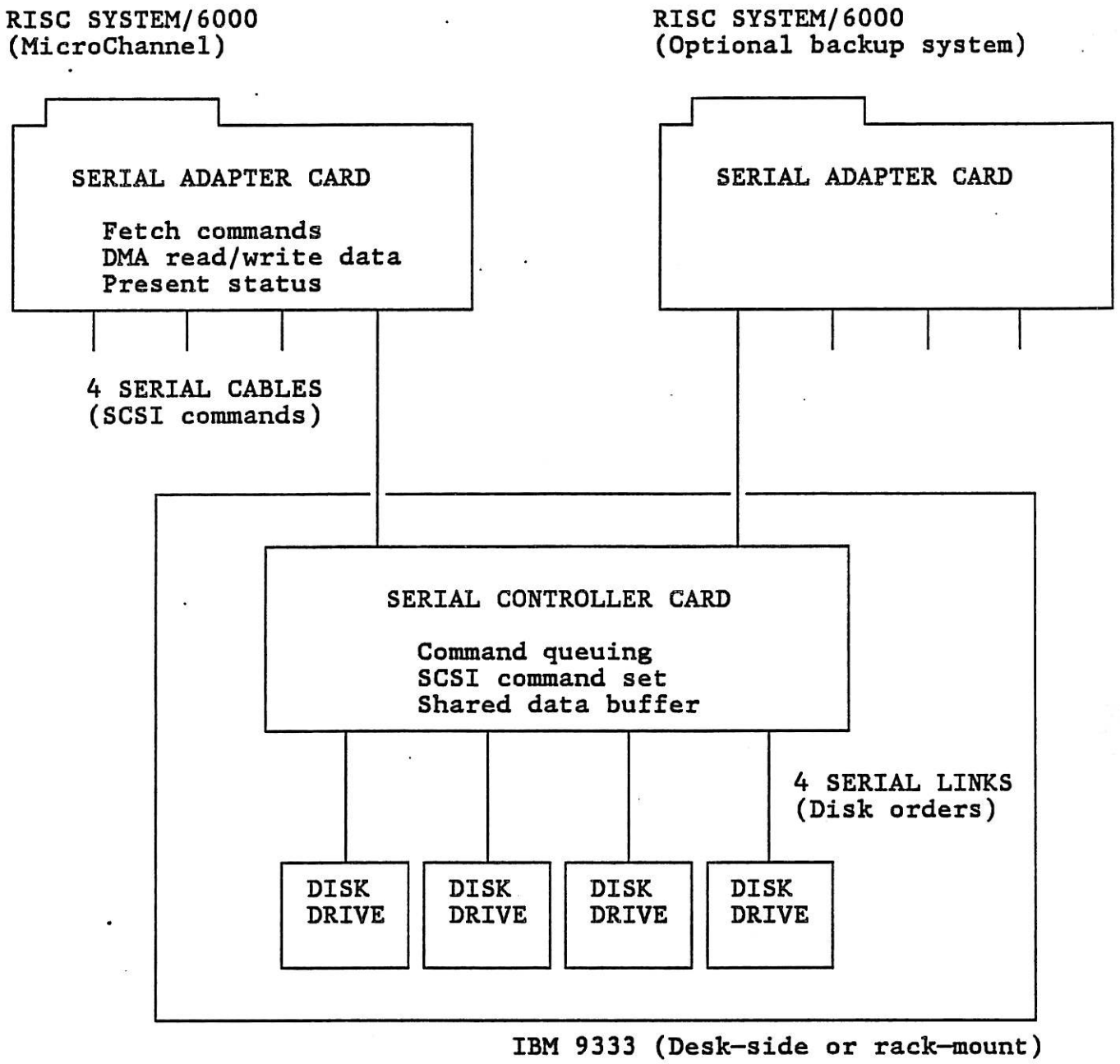
  Specifications are freely available

  IBM would like SSA to be industry-standard

  IBM is seeking partners

Subsystem overview

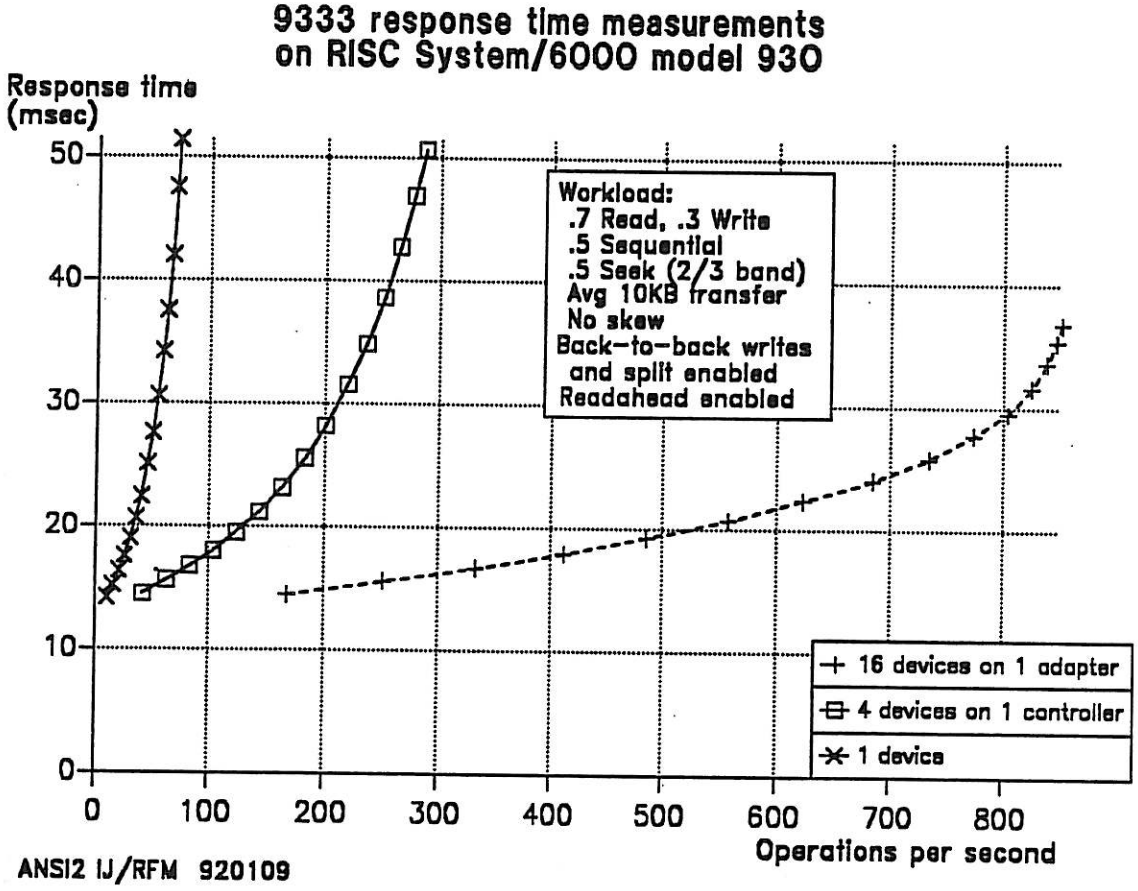
IBM 9333



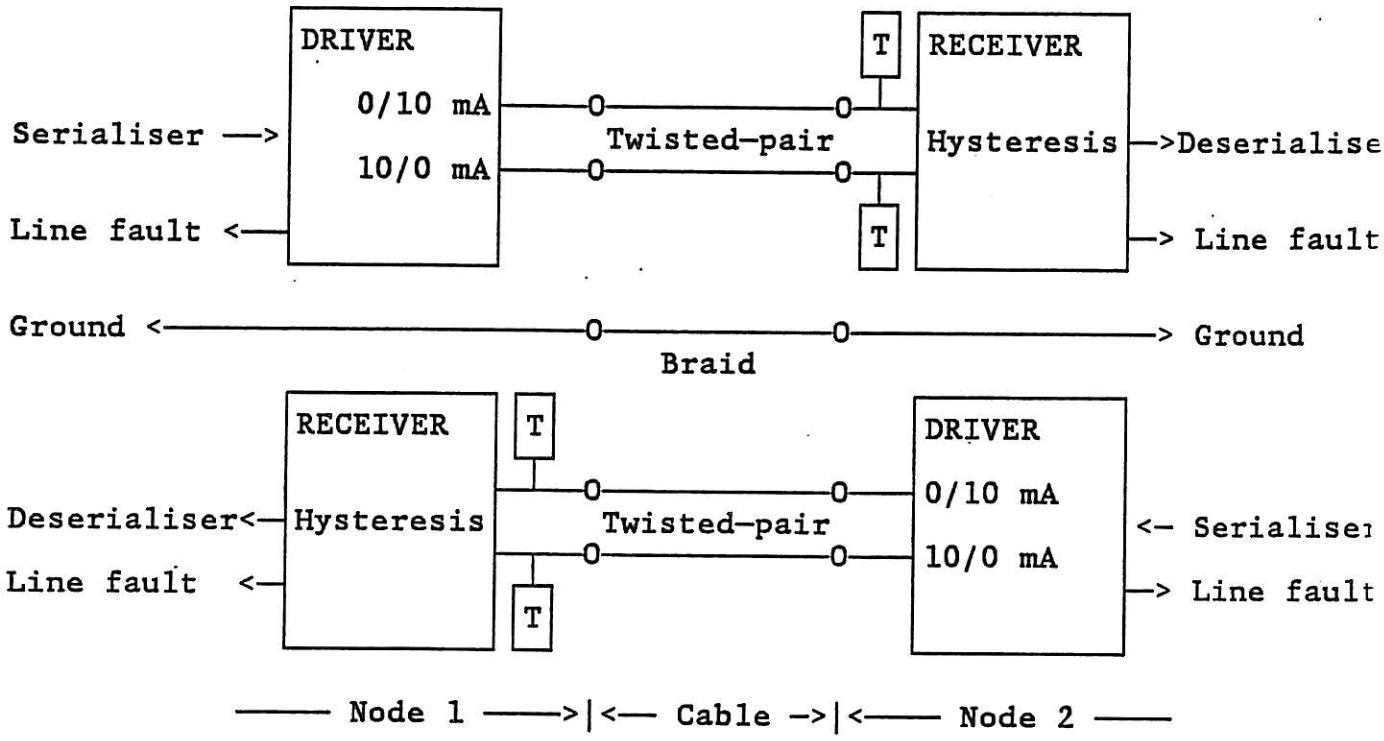
- IBM 9333 high-performance disk drive subsystem

  Serial links for all internal connections

  Announced in July 1991 for RISC System/6000

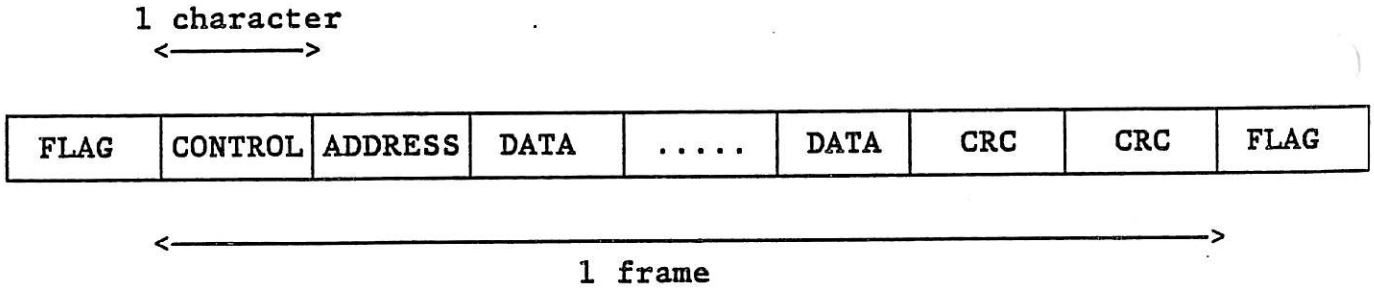

9333 response time measurements
on RISC System/6000 model 930
Response time
(msec)
50
40
30
20
10
0
0
100
200
300
400
500
600
700
800
Operations per second
Workload:
.7 Read, .3 Write
.5 Sequential
.5 Seek (2/3 band)
Avg 10KB transfer
No skew
Back-to-back writes
and split enabled
Readahead enabled
16 devices on 1 adapter
4 devices on 1 controller
1 device
ANSI2 IJ/RFM 920109

- High-performance transport layer
  - 10, 20 , ... MBytes/s (Automatically negotiated)
  - Full-duplex
  - Frame multiplexing
  - Spatial reuse with daisy-chain & loop
- Excellent Reliability, Availability & Serviceability
  - Wrap mode for Power-On-Self-Test
  - Comprehensive error detection
  - Transparent frame recovery
  - Concurrent maintenance
- Compatible with small form-factor devices
  - Fully integrated (10K equivalent gates per port)
  - Low power (0.3 watts per port)
  - Compact cables and connectors



- Internal and external connections
  - Dual twisted-pair or twinax cable
  - 10 Metres maximum (Point-to-point only)
  - +/- 2 Volt common mode range
  - Self-latching connectors
- 10-bit characters
  - 8B/10B code (DC-balanced)
  - 5 protocol characters
  - 100, 200, ... Mbits/s with NRZ modulation

Frame format



- FLAG PROTOCOL CHARACTER
  - Frame delimiter
  - Byte synchronisation
- CONTROL FIELD
  - 2-bit Frame_sequence_number
  - Resets
  - Extensions for daisy chain and loop
- ADDRESS FIELD
  - Used to route the frame at the destination node, eg.
    - A DMA channel (For data frames)
    - Microprocessor (For message frames)
- DATA FIELD
  - Variable length, typically 128 bytes maximum
- CRC FIELD
  - Checks Control, Address and Data fields

- Each frame expects 2 responses:

  ACK indicates the frame was received OK

  RR paces the next frame

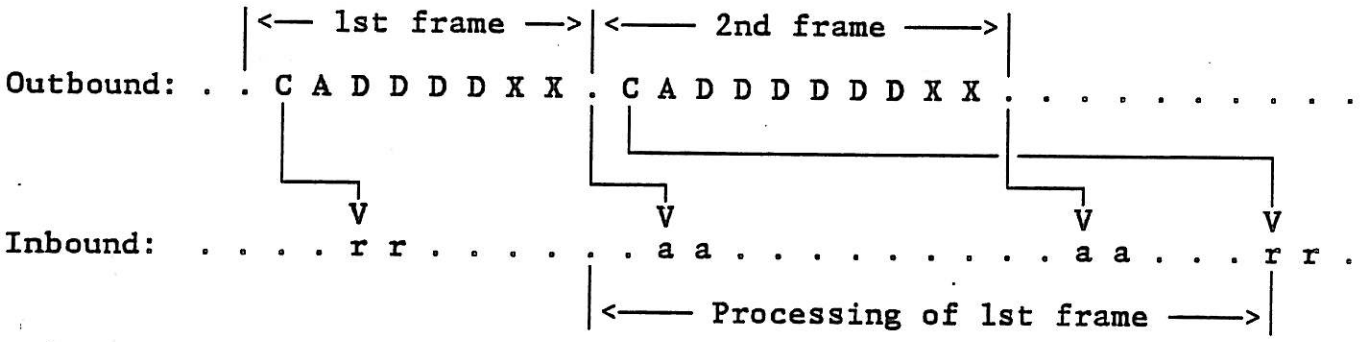
- ACK and RR are protocol characters, not frames

  Duplicated for checking

  Can be interleaved within a frame to reduce latency

- Typical transfer with A/B buffering

  NB. Half-duplex for clarity, but full-duplex is supported



Data characters:

C – Control

A – Address

D – Data

X – CRC

Protocol characters:

. – FLAG

a – ACK

r – RR

- Local 'wrap' provides an excellent power-on self-test
- The hardware provides comprehensive error detection

  Line faults

  Illegal characters

  CRC errors

  Non-sequential frame sequence numbers

  Missing ACK's

  Protocol errors

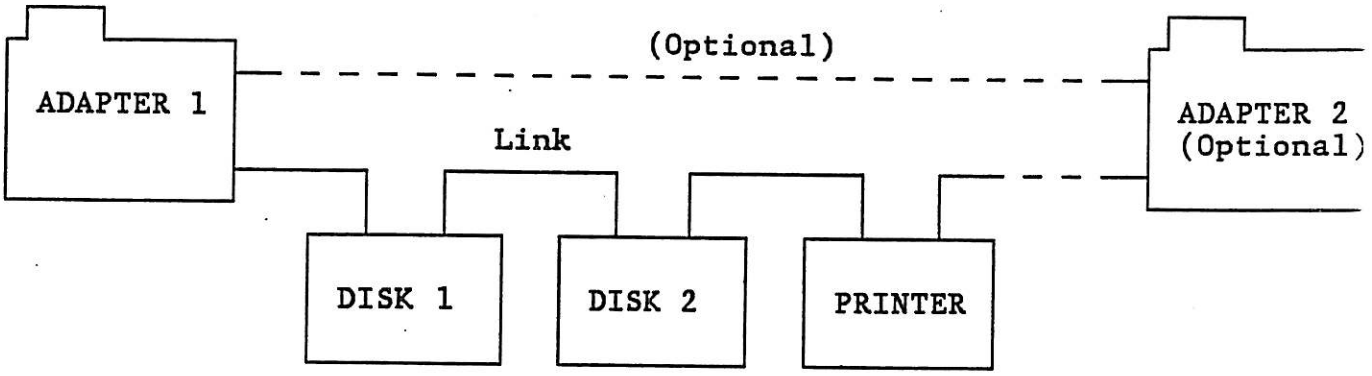
- There is an architected Error Recovery Procedure:

  Exchange Receive_sequence_numbers

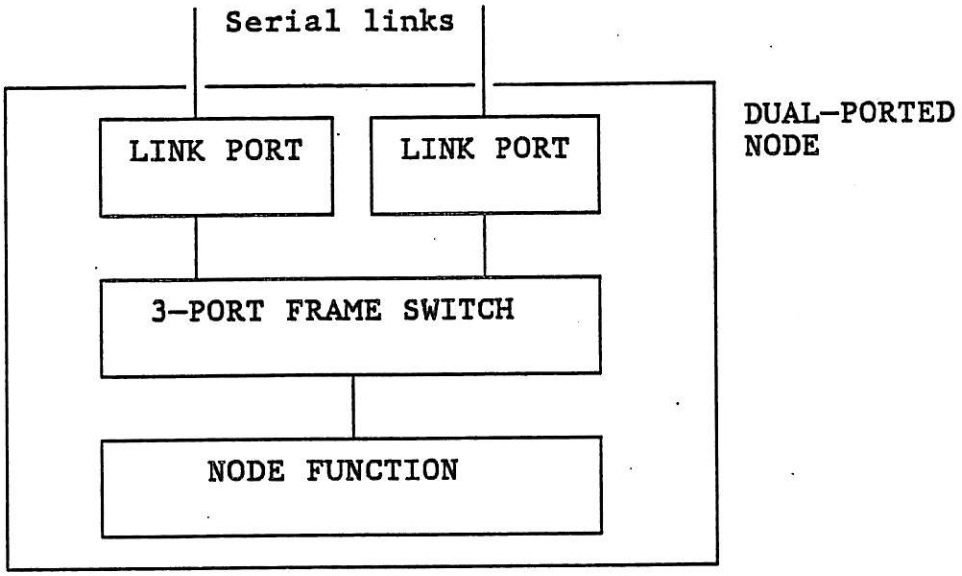
  Compare with Transmit_sequence_numbers

  Retransmit any lost frames

Distributed switch proposal



- Each device has dual ports and a frame switch
  - Links operate independently with the existing protocol
- Increases connectivity, similar to SCSI bus
  - 16 nodes maximum
  - Peer-to-peer communication
- Can be configured as a loop by using dual-port adapters
  - No single point of link failure
  - Increases bandwidth 2X
- Spatial reuse increases the effective bandwidth, eg.
  - DISK1 <–> ADAPTER1 and DISK2 <–> ADAPTER2
  - Provides up to 4X bandwidth of a token ring



- The frame control field is extended to include:

  Hop_count (4 bits), decremented by each switch

  Inbound frames are forwarded until Hop_count = 0

  Multi-cast flag for multiple destinations

  Priority flag for synchronous traffic (eg. multi-media)

- Switch uses 'worm-hole' routing to reduce latency
- Fairness algorithm shares bandwidth equally

  Each node has a quota for originating frames

  Quotas are refreshed by a circulating **SAT** character

Disk orders

- Intermediate-level orders optimised for clusters & arrays
  - Device independence
  - Low overheads
  - Tight device control
  - Buffered or unbuffered devices
  - Dual-ported devices for high availability
  - Integrated spindle synchronisation
- **Read** and **Write** orders access up to 64K sectors
  - Automatic head and cylinder seeks
  - Device manages track format, defects & header fields
  - Device checks for split transfer
  - Device performs error recovery (eg. ECC)
- **Extend** order can be over-lapped with Read or Write:
  - Back-to-back writes
  - Continuation of read-ahead
- **Format** & **Reassign Block** for media maintenance
  - Device manages the defect list

- 2 types of frame:

  Message frames have x'FF' in the address field

  Data frames specify a **channel** in the address field

  Channels are allocated dynamically using messages

  NB. The destination node is selected by the control field

- Typical message frames:

  SCSI_command (Initiator, LUNTRN, Tag, CTL, CDB) I ->T

  Data_ready (Target, Tag, Offset, Length) T ->I

  Data_reply (Initiator, Tag, Channel) I ->T

  Data_request (Target, Tag, Offset, Length, Channel) T ->I

  SCSI_status (Target, Tag, status) T ->

- Example of a read command:

  INITIATOR **TARGET**

  SCSI_command message —>

  <— Data_ready message

  Allocate DMA channel

  Data_reply message ———>

  <— Data frames

  <— SCSI_status message

Read order

READ (Ret_addr, LBA, Count, Options)

- Implies initial seek and termination of read-ahead
  - Device does logical-to-physical conversion
- **Ret_addr** specifies address for data frames
- **LBA** specifies next starting block
- **Count** allows up to 64 K sectors
  - Device performs head & cylinder seeks as necessary
  - Device skips defects as necessary.
- **Options**
  - Report all errors
  - Limit recovery
  - Read continuous
  - Disable split

- Read
- Write
- Extend
- Motor_control
- Set_position
- Read_status
- Code_download
- Release
- Format
- Reassign_block
- Read_device_characteristics
- Read_Vital_Product_Data
- Diagnostic