Netdata: A Transmission Format for Network Data.

7 May 1981

John D. Palmer
Robert P. O'Hara

TABLE OF CONTENTS

NETDATA FORMAT

Netdata is a format in which files can be transmitted from one computer user or system to another. The data transmitted are the original file, plus enough descriptive information to enable the file to be reconstituted at the receiving end. The transmitted data are defined as a continuous stream of bytes. While our current network imposes on us the nostalgic notion of "card images", the "records" described herein are logical records (of varying length), and card boundaries are ignored.

The format of the transmitted file is shown in Figure 1 on page 4. Several control records begin the file, followed by the data records. A trailer record brings up the rear. An acknowledgement file is composed of only control records. If the transmission contains more than one file, the data records of the multiple files will be separated by a control record.

```
A Transmission of One File:

+-----------------------------------------------------------+
|                      CONTROL RECORDS                      |
+-----------------------------------------------------------+
|                                                           |
|                        DATA RECORDS                       |
|                                                           |
+-----------------------------------------------------------+
|                       TRAILER RECORD                      |
+-----------------------------------------------------------+

A Transmission of Two Files:

+-----------------------------------------------------------+
|                      CONTROL RECORDS                      |
+-----------------------------------------------------------+
|                                                           |
|                  DATA RECORDS - FILE 1                    |
|                                                           |
+-----------------------------------------------------------+
|                       CONTROL RECORD                      |
+-----------------------------------------------------------+
|                                                           |
|                  DATA RECORDS - FILE 2                    |
|                                                           |
+-----------------------------------------------------------+
|                       TRAILER RECORD                      |
+-----------------------------------------------------------+
```

Figure 1. File format

The control records and data records have the same format, which is shown in Figure 2 on page 5. The data portion of a control record begins with a 6 byte identifier. Following the identifier are 0 or more "text units", which are described below. An exception to this is the INMR02 control record, in which the data portion begins with a 4 byte file count field. This field is then followed by the text units.

Each record (both control and data) begins in the byte immediately following the end of the previous record. The records of the file to be transmitted are broken up into segments, whose format is shown in Figure 2 on page 5. A segment has a maximum length of 255 bytes, including the 2 byte header. If the length of a record in the file is greater than 253 bytes, it is sent as multiple segments.

```
+------+--------+--------------------------------------------+
| LEN  | FLAGS  | DATA...                                    |
|  1   |   2    | 3-len                                      |
+------+--------+--------------------------------------------+
```

Byte	Length	Contents
1	1	Length of segment including 2-byte header (2<=len<=255).
2	1	Segment descriptor flags: x'80' - First segment of original record. x'40' - Last segment of original record. x'20' - (part of a) Control record. x'10' - Record number of next record. x'0F' - Reserved for future use.
3	n	User data segment (0<=n<=253).

Figure 2. Data Record Format

CONTROL RECORD FORMATS

A control record will contain a length field in byte 1, a flag field in byte 2, and the control record name in bytes 3-8 of the record. The data begin in byte 9. These data are composed of a varying number of text units, except for the INMR02 control unit, in which the text units are preceded by a 4 byte file number field.

INMR01 -- HEADER RECORD

The INMR01 record is always the first record of a transmission. The identifier of the record is "INMR01" in bytes 3-8. The remainder of the record (beginning with byte 9) is composed of text units.

Required text units are:

INMFNODE	Origin node name.
INMFTIME	Origin time stamp.
INMFUID	Origin userid.
INMLRECL	Length of physical control record segments.
INMTNODE	Target node name.
INMTUID	Target userid.

Optional text units are:

INMFACK	Receipt notification requested. If not present, no receipt notification is requested.
INMFVERS	Origin version number. If not present, version "1" is assumed.
INMNUMF	Number of files in this transmission. If not present, there can only be 1 file in the transmission.
INMUSERP	user parameter string.

INMR02 -- FILE UTILITY CONTROL RECORD

Each INMR02 record controls a data transformation step. In a given transmission, one or more processes represented by a corresponding number of INMR02 records are required. Utility operations currently supported are:

INMCOPY	which converts sequential files to and from the netdata format.

IEBCOPY which converts partitioned files to and from sequential files (called "unloaded" files).

AMSCIPHR which invokes the Access Method Services REPRO command to encrypt and decrypt files.

If more than one INMR02 record is present, they will appear in inverse order. The first record in the data stream will represent the last utility operation performed by RECEIVE. Note that the INMCOPY utility is always the last (or only) utility invoked during a SEND (TRANSMIT) operation, and thus appears first in the file. The text units of the INMR02 record describe the result of the transformation operation. The input with which it must work is described by the previous INMR02-directed operation or by the INMR03 data description record.

If the transmission contains more than one file, one or more INMR02 records are required for each file in the transmission. The groups of INMR02 records are in the same order as the files in the transmission. The file number field (which precedes the text units in the INMR02 record) identifies to which of the multiple files in the transmission the particular INMR02 record applies.

The identifier for this record is "INMR02" in bytes 3-8. Bytes 9-12 contain the number of the file in this transmission to which the control record applies. Multiple files in a single transmission are numbered sequentially starting at 1. The text units begin in byte 13.

Required text units are:

INMDSORG	File organization.
INMLRECL	Logical record length.
INMRECFM	Record format.
INMSIZE	Approximate size of file in bytes.
INMUTILN	Utility program name.

Also required are either of the following:

INMDSNAM	File name.
INMTERM	Mail file.

Optional text units are:

INMBLKSZ	File block size.
INMCREAT	Creation date.
INMDIR	Number of directory blocks.
INMEXPDT	Expiration date.
INMFM	Filemode number.
INMLCHG	Last change date.
INMLREF	Last reference date.
INMMEMBR	Member name list.
INMUSERP	User parameter string.

INMR03 -- DATA CONTROL RECORD

The INMR03 record immediately precedes the transmitted data and identifies their format.

The identifier for this record is "INMR03" in bytes 3-8. The text units begin in byte 9.

Required text units are:

INMDSORG	File organization.
INMLRECL	Logical record length.
INMRECFM	Record format.
INMSIZE	Approximate size of file in bytes.

INMR04 -- USER CONTROL RECORD

The INMR04 record may appear anywhere among the control records. It contains user data to be passed to user exits during the RECEIVE operation, if such exits are implemented.

The identifier for this record is "INMR04" in bytes 3-8. The text units begin in byte 9.

Required text units are:

INMUSERP	User parameter string.

INMR06 -- TRAILER CONTROL RECORD.

The INMR06 record is always the last record in a transmission. This record is used to verify that the transmission is complete.

The identifier for this record is "INMR06" in bytes 3-8. The text units begin in byte 9, however no text units are presently defined for this record.

INMR07 -- ACKNOWLEDGEMENT CONTROL RECORD

The INMR07 record indicates acknowledgement of a previous transmission. When it appears, the transmission will consist of only the INMR01, INMR07, and INMR06 records (in that order).

The identifier for this record is "INMR07" in bytes 3-8. The text units begin in byte 9.

Required text units are:

INMFTIME	Origin time stamp.

Also required are either of the following:

INMDSNAM	File name.
INMTERM	Mail file.

Optional text units are:

INMERRCD	Error indication for the RECEIVE operation. If not present, a sucessful RECEIVE operation is assumed.
INMFACK	Acknowledgement id.
INMFFM	Filemode number.
INMUSERP	User parameter string.

TEXT UNITS

Text units are used to describe the information contained in the transmission control records. The format of a single text unit is:

Offset	Length	Value
+0	2	Text unit key. The key identifies the information in the text unit. Possible values for the key are given in the section "Text Unit Keys".
+2	2	Count. The count field contains the number of "length-value" fields which follow.
+4	2	Length field. The first of perhaps many length fields. The length value includes only the length of the following value field and not its own 2-byte length.
+6	n	Value field. The first of perhaps many value fields. The format of the field is specified for each text unit key. The length of the field is specified by the preceding length field.
6+n		Second length-value entry if the count field indicated more than one entry present.

Each text unit immediately follows the previous text unit.

Dates and Times

All dates and times are expressed in GMT. For all text units where a date or time is specified, the value field will have a standard format: EBCDIC characters for the year(4), month(2), day(2), hour(2), minute(2), second(2), and fraction of seconds(n). Only as much as is known of this time value need be specified. E.G. if only the year is known the value would be 'YYYY'. If microseconds are known, the value would be 'YYYYMMDDHHMMSSUUUUUU'.

Numeric Values

All numeric values may be specified with a length of 1 to 8 bytes. Current implementations use only the low-order 4 bytes if the field is longer than 8 bytes.

TEXT UNIT KEYS

The following text units are currently defined:

Text Unit Key (hex)	Mnemonic	Function
0030	INMBLKSZ	Block size
1022	INMCREAT	Creation date
0001	INMDDNAM	DDname for the file
000C	INMDIR	Number of directory blocks
0002	INMDSNAM	Name of the file
003C	INMDSORG	File organization
1027	INMERRCD	RECEIVE command error code
0022	INMEXPDT	Expiration date
1026	INMFACK	Originator requested notification
102D	INMFFM	Filemode number
1011	INMFNODE	Origin node name
1024	INMFTIME	Origin time stamp
1012	INMFUID	Origin userid
1023	INMFVERS	Origin version number
1021	INMLCHG	Last change date
0042	INMLRECL	Logical record length
1020	INMLREF	Last reference date
0003	INMMEMBR	Member name list
102F	INMNUMF	Number of files
102A	INMRECCT	Transmitted record count
0049	INMRECFM	Record format
102C	INMSIZE	File size in bytes
0028	INMTERM	Mail file
1001	INMTNODE	Target node name
1025	INMTTIME	Destination time stamp
1002	INMTUID	Target userid
1029	INMUSERP	User parameter string
1028	INMUTILN	Name of utility function

Blocksize specification - Key = X'0030'

INMBLKSZ specifies the block size of the file. When this key is specified, NUM is 1, LEN is 1 to 8, and PARM contains the blocksize of the file.

Example: specification of blocksize 32768.

KEY	NUM	LEN	PARM
0030	0001	0004	00008000

Creation date specification - Key = X'1022'

INMCREAT specifies the creation date of the file. When this key is specified, NUM is 1, LEN is 4 or more, and PARM contains the creation date in standard format.

Example: specification of August 28, 1969.

KEY	NUM	LEN	PARM
1022	0001	0008	F1F9F6F9F0F8F2F8

Ddname specification - Key = X'0001'

INMDDNAM specifies a ddname associated with the file. When this key is specified, NUM is 1, LEN is the length of the ddname field, and PARM contains the ddname.

Example: specification of the ddname DD1.

KEY	NUM	LEN	PARM
0001	0001	0003	C4C4F1

Directory Block specification - Key = X'000C'

INMDIR specifies the number of blocks in the file. When this is key is specified, NUM is 1, LEN is 1 to 8, and PARM contains the number of directory blocks.

Example: specification of 15 directory blocks.

KEY	NUM	LEN	PARM
000C	0001	0003	00000F

Dsname specification - Key = X'0002'

INMDSNAM specifies the file name of the file. File names are divided into "fields".

* In MVS a file is called a <u>data set</u>. In its name there one or more "fields", and each field has a maximum length of 8 characters. The fields are separated by periods; for example, AA.BB.CC.DD has 4 fields. The maximum length of the data set name, including all periods, is 44 characters.

* In CMS a file name is called a fileid. It always has three fields that are separated by blanks (filename, filetype and filemode, having a maximum of 8, 8 and 2 characters respectively).

When transmitting a CMS file, the filemode number is not specified as part of the fileid. Instead, it is specified on the INMFFM text unit. The filemode letter is considered to be the "highest qualifier" of the fileid, so it is always transmitted as the first field.

When this key is specified, NUM is the number of fields in the file name, and the LEN and PARM fields contain the length and name of each field of the file name.

Example: specification of the file name "A.B".

KEY	NUM	LEN1	PARM1	LEN2	PARM2
0002	0002	0001	C1	0001	C2

Example: specification of the fileid "ABC EXEC A2".

KEY	NUM	LEN1	PARM1	LEN2	PARM2	LEN3	PARM3
0002	0003	0001	C1	0002	C1C2C3	0004	C5E7C5C3

File organization specification - Key = X'003C'

INMDSORG specifies the file organization. When this key is specified, NUM is 1, LEN is 2, and PARM contains:

X'0008'	for VSAM
X'0200'	for partitioned organization
X'4000'	for physical sequential

Example: specification of a physical sequential file.

KEY	NUM	LEN	PARM
003C	0001	0002	4000

Error code specification - Key = X'1027'

INMERRCD indicates the result of the RECEIVE operation. When this key is specified, NUM is 1, LEN is 1 or more, and PARM will contain a string indicating the result of the receive operation.

Example: specification that the sent file was "RECEIVED".

KEY	NUM	LEN	PARM
1027	0001	0008	D9C5C3C5C9E5C5C4

Example: specification that the sent file was "DELETED".

KEY	NUM	LEN	PARM
1027	0001	0007	C4C5D3C5E3C5C4

Expiration date specification - Key = X'0022'

INMEXPDT specifies the expiration date of the file. When this key is specified, NUM is 1, LEN is 4 or more, and PARM contains the expiration date value in standard format.

Example: specification of an expiration date of January 1, 1981.

KEY	NUM	LEN	PARM
0022	0001	0008	F1F9F8F1F0F1F0F1

Acknowledge specification - Key = X'1026'

INMFACK specifies that the origin has requested an acknowledgement of receipt of the transmitted file. When this key is specified, NUM is 0 or 1. If NUM is 1, LEN is 1 to 64 and PARM contains a transmission identifier to be returned with the notification. The return is also made via this text unit.

Example: specification notification request without id.

KEY	NUM	LEN	PARM
1026	0000		

Example: specification notification request with id FRED.

KEY	NUM	LEN	PARM
1026	0001	0004	C6D9C5C4

Filemode number specification - Key = X'102D'

INMFFM specifies the filemode number of the file. When this key is specified, NUM is 1, LEN is 1, and PARM contains the filemode number of the file.

Example: specification of filemode number of 0.

KEY	NUM	LEN	PARM
102D	0001	0001	F0

Origin node name specification - Key = X'1011'

INMFNODE specifies the node name of the origin of this transmission. When this key is specified, NUM is 1, LEN is the length of the node name, and PARM is the name of the origin node.

Example: specification of node VENICE.

KEY	NUM	LEN	PARM
1011	0001	0006	E5C5D5C9C3C5

Origin time stamp specification - Key = X'1024'

INMFTIME specifies the time at which the transmission was sent. When this key is specified NUM is 1, LEN is 4 or more, and PARM is the time the transmission was created, specified in standard format.

Example: Specification of July 19, 1951 at 3:20 PM.

KEY	NUM	LEN	PARM
1024	0001	000C	F1F9F5F1F0F7F1F9F1F5F2F0

Origin userid specification - Key = X'1012'

INMFUID specifies the userid of the origin of this transmission. When this key is specified, NUM is 1, LEN is the length of the userid, and PARM is the userid of the origin of the transmission.

Example: specification of userid IBMUSER.

KEY	NUM	LEN	PARM
1012	0001	0007	C9C2D4E4E2C5D9

Origin version specification - Key = X'1023'

INMFVERS specifies the version of the data format used for this transmission. When this key is specified NUM is 1, LEN is 1 to 8, and PARM is the version number of the data format used.

Example: specification of version 1.

KEY	NUM	LEN	PARM
1023	0001	0004	00000001

Last change date specification - Key = X'1021'

INMLCHG specifies the last change date of the file. When this key is specified, NUM is 1, LEN is 4 or more, and PARM contains the last change date in standard format.

Example: specification of 04/01/81 8:12 PM.

KEY	NUM	LEN	PARM
1021	0001	000E	F1F9F8F1F0F4F0F1F2F0F1F2

Logical record length specification - Key = X'0042'

INMLRECL specifies the actual or maximum length, in bytes, of a logical record in the file. When this key is specified, NUM is 1, LEN isl to 8, and PARM contains either the actual or maximum record length.

Example: specification of 80-byte records.

KEY	NUM	LEN	PARM
0042	0001	0001	50

Last reference date specification - Key = X'1020'

INMLREF specifies the last reference date of the file. When this key is specified, NUM is 1, LEN is 4 or more, and PARM contains the last reference date in standard format.

Example: specification of February 14, 1981.

KEY	NUM	LEN	PARM
1020	0001	0008	F1F9F8F1F0F2F1F4

Member specification - Key = X'0003'

INMMEMBR specifies a list of member names of the file. When this key is specified, NUM is 1 or more and the LEN and PARM fields will contain the length and name of each member.

Example: specification of the member IEASYS00.

KEY	NUM	LEN	PARM
0003	0001	0008	C9C5C1E2E8E2F0F0

Example: specification of the members ABC and DEFG.

KEY	NUM	LEN1	PARM1	LEN2	PARM2
0003	0002	0003	C1C2C3	0004	C4C5C6C7

Number of Files - Key = X'102F'

INMNUMF specifies the number of files that make up this transmission. It is required on the INMR01 control record if there are more than zero files in the transmission. (If this text unit is missing, number of files is assumed to be zero. This makes sense only on an acknowledgement transmission.) When this key is specified, NUM is 1, LEN is 1 to 8, and PARM contains the number of files that make up this transmission.

Example: Specification of 2 files in this transmission.

KEY	NUM	LEN	PARM
102F	0001	0004	00000002

Transmitted record count specification - Key = X'102A'

INMRECCT specifies a count of transmitted records. When this key is specified, NUM is 1, LEN is 1 to 8, and PARM contains the number of records transmitted.

Example: specification of 129 records.

KEY	NUM	LEN	PARM
102A	0001	0001	81

Record format specification - Key = X'0049'

INMRECFM specifies the record format of the file. When this key is specified, NUM is 1, LEN is from 2 to 8 (although the current implementation only examines the first 2 bytes). and PARM is 1 or more of the following "OR"ed together:

X'0001'	Shortened VBS format used for transmission records
X'xx02'	Varying length records without the 4-byte header
X'xx04'	Packed file (multiple blanks removed)
X'xx08'	Compressed file (Huffman encoded)
X'0200'	Data includes machine code printer control characters
X'0400'	Data contains ASA printer control characters
X'0800'	Standard fixed records or spanned variable records
X'1000'	Blocked records
X'2000'	Track overflow or variable ASCII records
X'4000'	Variable length records
X'8000'	Fixed length records
X'C000'	Undefined records

Example: specification of fixed blocked records.

KEY	NUM	LEN	PARM
0049	0001	0002	9000

File size in bytes - Key = X'102C'

INMSIZE specifies the size of the file in bytes. When this key is specified, NUM is 1, LEN may be 1 to 8, and PARM contains the size of the file in bytes.

Example: Specification of a 1 megabyte file.

KEY	NUM	LEN	PARM
102C	0001	0004	000F4240

Mail file specification - Key = X'0028'

INMTERM specifies that the file is a "mail" file. When this key is specified, NUM is 0 and LEN and PARM omitted.

Example: specification of a mail file.

KEY	NUM	LEN	PARM
0028	0000		

Target node name specification - Key = X'1001'

INMTNODE specifies the node name of the target of this transmission. When this key is specified, NUM is 1, LEN is the length of the node name, and PARM is the name of the target node.

Example: specification of node ROME.

KEY	NUM	LEN	PARM
1001	0001	0004	D9D6D4C5

Destination time stamp specification - Key = X'1025'

INMFTIME specifies the time at which the transmission was received. When this key is specified NUM is 1, LEN is 4 or more, and PARM is the time the transmission was received, specified in standard format.

Example: Specification of March 14, 1981 8:30:25 AM

KEY	NUM	LEN	PARM
1025	0001	000E	F1F9F8F1F0F3F1F4F0F8F3F0F2F5

Target userid specification - Key = X'1002'

INMTUID specifies the userid of the target of this transmission. When this key is specified NUM is 1, LEN is the length of the userid, and PARM is the userid of the target of the transmission.

Example: specification of userid IBMUSER.

KEY	NUM	LEN	PARM
1002	0001	0007	C9C2D4E4E2C5D9

User parameter string specification - Key = X'1029'

INMUSERP specifies a user parameter string which should be passed to appropriate user exits. When this key is specified NUM is 1, LEN is 1 to 251, and PARM contains the user parameter string.

Example: specification of user string PARM1.

KEY	NUM	LEN	PARM
1029	0001	0005	D7C1D9D4F1

Utility program name specification - Key = X'1028'

INMUTILN specifies the name of the utility program which is used as part of the process of restoring the transmitted data to its original format. When this key is specified NUM is 1, LEN is the length of the utility name, and PARM contains the name of the utility. Currently defined names are:

INMCOPY Invoke internal utility to convert from the transmission format to a sequential file.

IEBCOPY Invoke the IEBCOPY utility to reload a partitioned file.

AMSREPRO Invoke the Access Method Services REPRO command to decrypt a file.

Example: specification of utility program INMCOPY.

KEY	NUM	LEN	PARM
1028	0001	0007	C9D5D4C3D6D7E8

Figure ID's

id	File	Page	Figure References
file	NETDATA	4	1: 3
data	NETDATA	5	2: 4, 4