Natural Net Data Interface NATNETTO

This document provides information on the Natural Net Data Interface and the `net data` protocol definition.

It covers the following topics:

- Natural Net Data Driver Functional Description
- General Message Layout
- Layout of Header
- Format Buffer Layout
- Value Buffer Layout
- Attribute Buffer

See also *Installing the Natural Net Data Interface* in the *Installation* documentation.

Natural Net Data Driver Functional Description

The Natural Net Data Driver `NATNETTO` is a component that was introduced to support the EntireX CICS 3270 Bridge and similar client/server solutions in message oriented server environments, that is, TP monitors.

`NATNETTO` implements a protocol driver, which allows program-to-program communication with Natural (legacy) applications from client applications, using a net-data protocol. One typical scenario is a desktop client (for example, built with Natural for Windows or VBA) accessing a Natural application that runs under a TP monitor such as CICS, IMS TM or UTM.

"Net data" means, that the protocol neither contains format data such as text constants nor any device-dependent control sequences. All data is communicated in printable format. This implies that eventually necessary marshaling and unmarshaling of non-alpha fields has to be done by the clients.

Basically, the protocol consists of two parts:

- A header or control block and a value buffer which contains the raw net data. This part is mandatory. The header contains control, environment and session information and maintains pointers to the other parts of the data buffer. The value buffer contains the actual net data which is to be exchanged between client and server.
- In addition, optional variable parts are available: format buffer and/or attribute buffer. The optional format buffer has an entry with descriptive data for each field in the value buffer. The attribute buffer consists of one byte with a preset value of 0 for each field in the value buffer. The client has to switch this value to 1 for each modified field, if the appropriate option is set, thus emulating the setting of mdt bits.

Header, value buffer and attribute buffer are parts of outbound and inbound messages; only the format buffer may occur in the outbound message only. The header maintains a transaction number which has to be mirrored by the client for flow-control purposes. Since legacy applications are mostly designed to be driven from block mode terminals, the protocol supports 3270 like functionality such as PF keys and cursor position.

General Message Layout

The following parts of the general message layout are mandatory:

- Header (the first two rows in the table below)
- Value buffer

The following parts are optional:

- Format buffer
- Attribute buffer

<table>
<tr><td><b>"FSCB"</b></td><td><b>Value Buffer Offset</b></td><td colspan="2"><b>Format Buffer Offset</b></td><td><b>...</b></td></tr>
<tr><td></td><td><b>Attribute Buffer Offset</b></td><td><b>Aid Char.</b></td><td><b>Cursor Pos.</b></td><td><b>...</b></td></tr>
<tr><td colspan="5">Value Buffer</td></tr>
<tr><td colspan="5">Format Buffer</td></tr>
<tr><td colspan="5">Attribute Buffer</td></tr>
</table>

For detailed information on the layout parts, refer to Table 1.

Layout of Header

Table 1: Control Block - Fixed Part

Field	Format	Scope	Meaning
Eyecatcher	A4	FSCB	Eyecatcher
Product code	A3	-	Product identification
Protocol version	N2	01 - 99	Version for specific product
Value buffer offset	N10	calculated	Value buffer offset from start of message
Format buffer offset	N10	calculated	Format buffer offset from start of message
Total message length	N10	calculated	Cumulated length of all buffers
Message number	N6	incremented by 1 every call	Echoed by communication partner
Block number	N5	01 - 99 (normaly 01)	For block splitting within one message

Field	**Format**	**Scope**	**Meaning**
Number of parameters	N5	calculated	Number of parameters in VB
Session token	A32		Security token
Message format	A1	see Table 2	Mode of field separation within value buffer
Delimiter character	A1	-	
Architecture	A2	see Table 3	Archictecture of sending partner
Call type	A2	see Table 4	Type of current call
Response code	N4	0001 - 9999	Response code from client
Block status	A1	L or N	Block is last one of msg or a next one follows
Server name / TAC	A8	-	TP transaction code or name of server
Aid character	A2	see Table 5	Aid character depressed or generated on client
Cursor line	N3	1 - max phys. line on client	Cursor line or 000 *)
Cursor column	N3	1 - max phys. col. on client	Cursor column or cursor field number *)
Attribute buffer offset	N10	calculated	AB offset from start of message
Timestamp	A16	generic	Store clock value: map stow time hex printable
DBID	N5	1 - 32767	DBID of FNAT on server
File number	N5	1 - 32767	File number of FNAT on server
Date form	A1	I, G, E, U	Date format according to Natural
Decimal character	A1	-	Natural delimiter character on server
Input delimiter char.	A1	-	Natural input delimiter character (server)
Control character	A1	-	Natural control character (server)
Language code	N2	01 - 99	Natural language code (server)
Application ID	A8	-	Natural application ID
Program name / map	A8	-	Program in execution / map or format name
Error number	N5	00001 - 99999	Natural error number
Line number	N4	0001 - 9999	Line number of current I/O statement
Error state	A1	-	Status byte
Error program	A8	-	Object causing an error

Field	Format	Scope	Meaning
Error level	N2	01 - 15	Subroutine level of object in error
Message type	A1	see Table 6	Type of message
Option flag 1	A1	see Table 7	Control flag
Option flag 2	A1	see Table 7	Control flag
Option flag 3	A1	see Table 7	Control flag
Option flag 4	A1	see Table 7	Control flag
Option flag 5	A1	see Table 7	Control flag
Option flag 6	A1	see Table 7	Control flag
Option flag 7	A1	see Table 7	Control flag
Option flag 8	A1	see Table 7	Control flag

*) If the cursor field number notation is set in `NATCONFG`, the cursor line will always be `000` and the cursor column will contain the absolute number of the field, where the cursor shall be placed (outbound) or was located at send time (inbound).

Note:
Not all header fields are currently used!

Table 2: Modes of Field Separation

Format A1

Value	Meaning
`D`	Delimited mode
`F`	Fixed format mode
`L`	Length field precedes field (N3)

Table 3: Architecture of Sending Partner

Mask in Format A2

Value		Meaning
`-`	`1`	Mask for low order byte first (Vax)
`-`	`2`	Unused
`-`	`4`	Mask for EBCDIC architecture
`-`	`8`	Mask for ASCII 8 architecture
`1`	`-`	Mask for float representation VAX
`2`	`-`	Mask for float representation IEEE

Table 4: Call Type

Format A2

Type of Communication	Value	Meaning
Natural net data/3GL	`MD`	Map data (net data using format)
	`ND`	Net data
	`CM`	Command mode (server)
	`FD`	Map-format download
	`IP`	Normal input statement
	`CS`	Close session termination message

Table 5: Aid Character Table

Format A2

Aid Char.	PF Key
`EN`	Enter
`CL`	Clear
`P1`	PA1
`P2`	PA2
`P3`	PA3
`01`	PF1
`02`	PF2
`03`	PF3
`...`	...
`47`	PF47
`48`	PF48
`CS`	Close Session

Note:
`CS` - Close Session - allows clients to enforce an immediate close of the server session. Therefore, it is in fact not a real PF key, but a command code for the server.

Table 6: Message Type

Value	Meaning
D	Dialog message
A	Async. message
P	Printout message

Table 7: Option Flags for Natural Net-Data Communication

All flags are of format A1.

Flag	Values	Meaning
Option 1	F	Message includes format buffer (fb-option).
Option 2	S	Net data is generated from screen buffer.
	P	Net data is generated from page buffer.
Option 3	A	Message includes attribute buffer (ab-option).
Option 4	P	Data in VB is in presentation format (printable).
	I	Data in VB is in internal format of sender.
	A	Data is in internal format converted to alpha.
Option 5	M	Outbound message contains overlay part.
Option 6	1	Extended format buffer option 1.
	2	Extended format buffer option 2.
Option 7		For future use.
Option 8		For future use.

Format Buffer Layout

Base Part

Each format buffer entry is a variable length string consisting of four elements:

- Identifier
- Protection indicator
- Format indicator
- Printable field length

Table 8: Format Buffer Entry

Element	Value	Meaning
Identifier	F	Field
	S	Subfield
Protection	M	Modifiable field
	O	Output only field, protected field
Format	A	Alpha
	N	Numeric
Field length	L - LLL,L	Length specification according to Natural standard

Examples:

FMA20	Field, modifiable, format aplha 20
SMN12,4	Subfield, modifiable, format numeric 12.4
O	Output only field, protected field

Note:
The precision part of a numeric length is always separated by a comma (,), regardless of the current values of delimiter and decimal character profile parameters! For alpha type fields the precision part is omitted.

Subfields are used to determine fields which had been separated out of a base field using the Natural dynamic attribute facility. If a field is dynamically divided into various subfields, this is marked as follows:

The first subfield is marked with identifier F as usual, all other subfields are identified by S.

Extension 1

The following figure shows a part of a dsect, which describes layout of the Natural internal screen attribute buffer. If the format buffer extension option 1 is set, for each field those attribute bytes (PATTR1 - PATTR4) will be brought into printable format and added to the approriate fields format buffer entry. The extension is separated by a "/" (slash) from the base format entry.

PATTR1	DS	X		ATTRIBUTE BYTE 1
P1TMP	EQU	X’80’	1000 0000	TEMPORARY PROTECTED (ONLY PAGE)
P1EXTLNG	EQU	X’80’	1000 0000	EXTENDED LENGTH (ONLY SCREEN)
P1RPA	EQU	X’40’	0100 0000	FIELD CAN BE REPEATED
P1PROT	EQU	X’20’	0010 0000	FIELD IS PROTECTED
P1NUM	EQU	X’10’	0001 0000	FIELD IS NUMERIC

P1SKIP	EQU	P1PROT+P1NUM (X’30’)		FIELD WILL BE SKIPPED AUTOMATICALLY
P1HIGH	EQU	X’08’	0000 1000	FIELD IS HIGHLIGHTED
P1BLINK	EQU	X’04’	0000 0100	FIELD IS BLINKING
P1NOND	EQU	P1HIGH+P1BLINK (X’0C’)		FIELD IS NON-DISPLAY
P1NHC	EQU	X’02’	0000 0010	FIELD MAY NOT BE PRINTED
P1CURS	EQU	X’01’	0000 0001	SET CURSOR HERE (ONLY UNPROT)
	SPACE			
PATTR2	DS	X		ATTRIBUTE BYTE 2
P2ITAL	EQU	X’80’	1000 0000	ITALIC/CURSIVE
P2MAND	EQU	X’40’	0100 0000	INPUT MANDATORY
P2MFILL	EQU	X’20’	0010 0000	MANDATORY FILL
P2LC	EQU	X’10’	0001 0000	DO NOT TRANSLATE (LOWER CASE)
P2CS2	EQU	X’08’	0000 1000	SECOND CHARACTER SET
P2UL	EQU	X’04’	0000 0100	UNDERLINED
P2RVID	EQU	X’02’	0000 0010	REVERSED VIDEO
P2RL	EQU	X’01’	0000 0001	RIGHT-LEFT
	SPACE			
PATTR3	DS	X	COLOR ATTRIBUTE	ATTRIBUTE BYTE 3
P3TP	EQU	X’80’	1000 0000	TERMINAL PROGRAM AVAILABLE
P3PFK	EQU	X’40’	0100 0000	*COM FIELD
P3NUM	EQU	X’20’	0010 0000	NUMERIC FIELDS
P3HELPR	EQU	X’10’	0001 0000	HELP ROUTINE AVAILABLE
P3FRAME	EQU	X’08’	0000 1000	FRAME ATTRIBUTE

P3NEUTR	EQU	X’07’	0000 0111	NEUTRAL
P3YELL	EQU	X’06’	0000 0110	YELLOW
P3TURQ	EQU	X’05’	0000 0101	TURQUOISE
P3GREEN	EQU	X’04’	0000 0100	GREEN
P3PINK	EQU	X’03’	0000 0011	PINK
P3RED	EQU	X’02’	0000 0010	RED
P3BLUE	EQU	X’01’	0000 0001	BLUE
*		FBI (DB)		(FIELD PROCESSING INFORMATION)
	SPACE			
PATTR4	DS	X		INTERNAL PROCESSING ATTRIBUTES
P4TEXT	EQU	X’80’	1000 0000	FIELD IS TEXT CONSTANT
P4SAME	EQU	X’40’	0100 0000	SAME ATTRIBUTE AS BEFORE
P4NATTR	EQU	X’20’	0010 0000	FIELD NEW ATTRIBUTE
*				PAGE BUFFER, DYNAMIC ATTRIBUTE
P4OVL	EQU	X’10’	0001 0000	FIELD BELONGS TO OVERLAY BUFFER
P4MDT	EQU	X’08’	0000 1000	FIELD HAS BEEN MODIFIED
P4MDTH	EQU	X’04’	0000 0100	UPDATE FROM HELP (PAGE BUFFER)
P4NFLD	EQU	X’04’	0000 0100	FIELD NEW ON SCREEN
*				IF SET FOR OVL, NEW LINE
P4CONT	EQU	X’02’	0000 0010	FIELD IS CONT OF BEFORE
P4LAST	EQU	X’01’	0000 0001	LAST ATTRIBUTE IN BUFFER
**P4HELP	EQU	P4TEXT+P4MDT		HELP REQUEST FOR THIS FIELD

Example:

An extended format buffer entry `18820300` means, the field is numeric and shall be presented highlighted italic in reversed video mode. The color of the field is pink!

Value Buffer Layout

Three modes of value buffer structure are possible:

- **Fixed Format**
  All parameters are simply concatenated without any delimitation. This means, that the single parameters have to be separated either according to the format description in the format buffer or by covering them with a C-structure, a data area or a dsect.

- **Delimited Format**
  The parameters are separated by an configurable delimiter character.

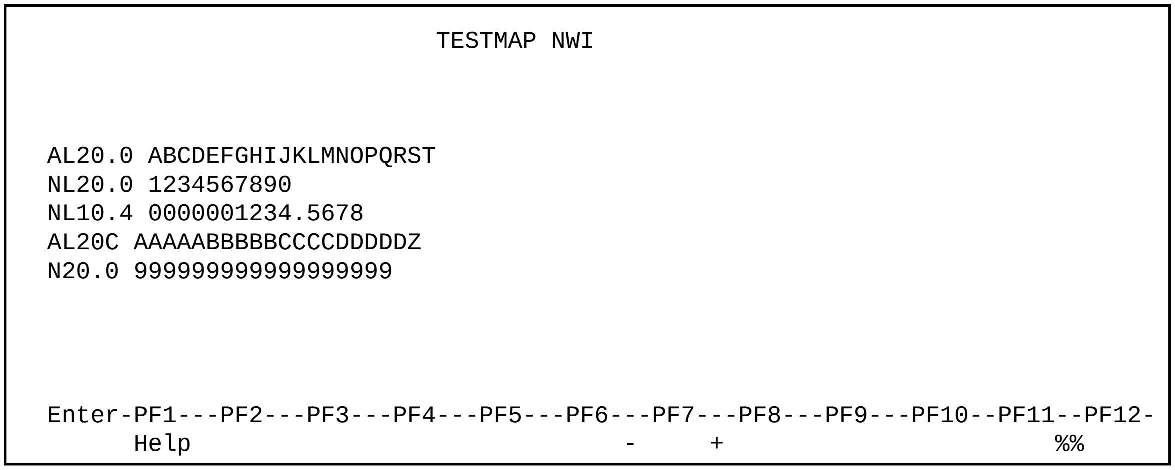
- **Length Preceded Format**
  Each parameter is preceded by a length field of format N3. The length notation is explicit.

Attribute Buffer

The attribute buffer is optional. It consists of a one-byte entry for each parameter field, which represents the mdt flag. The mdt has to be set by the client for each modified field. The value of this flag is "0" or "1". A value of 1 means the mdt is set.

Example:

This example shows the screen image of a 3270 format in Figure 1 and the generated net-data stream for the same format in Figure 2. The name of the Natural map is `NETM002`.



```
                           TESTMAP NWI



AL20.0 ABCDEFGHIJKLMNOPQRST
NL20.0 1234567890
NL10.4 0000001234.5678
AL20C AAAAABBBBBCCCCDDDDDZ
N20.0 999999999999999999




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12-
      Help                              -     +                       %%
```

Figure 1: `NETM002` on a 3270 Device

```
FSCBNAT010000000206000000038000000004710000120000100006
  F 04MD0000LNATvrsXSEN0000010000000465B3E0C25A1A1DE4000000000000I.,%01NETT
O NETM002 000000170   D  FSAP 1 ABCDEFGHIJKLMNOPQRST1234567890
  0000001234.5678AAAAABBBBBCCCCDDDDDZ999999999999999999
                          FMA20/08100024FOA20/
38102024FMA15/08102024FOA20/38101624FOA20/38102024FOA79/70000035.000000
```

Figure 2: Net-Data Stream Generated from `NETM002` Execution

Where `NAT`*`vrs`* stands for *`version`*, *`release`*, *`system maintenance level`* of the current Natural version.

Configuration Settings: Fixed format, format buffer + extended format buffer, attribute buffer option, cursor position represented as field number.