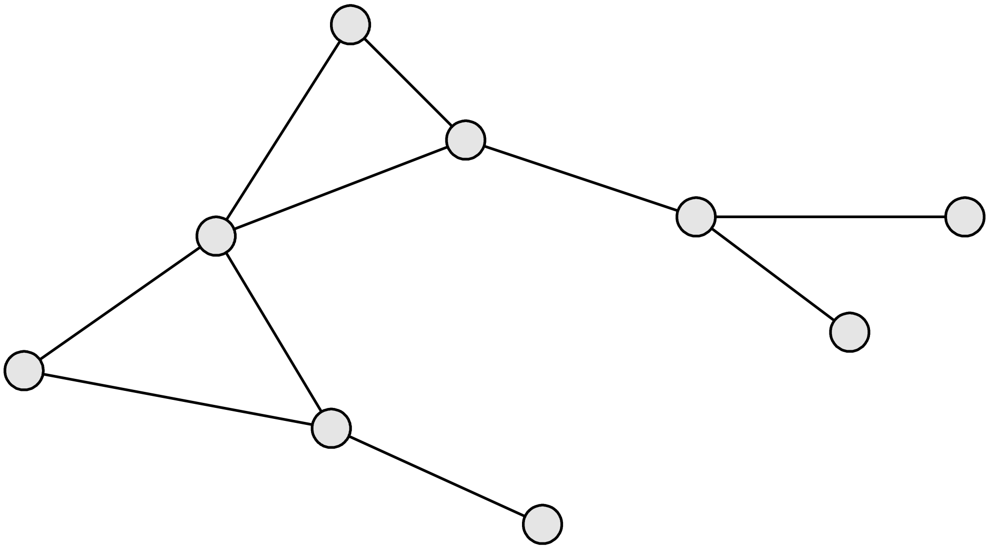
BBS SPECIFICATON

BY

W 0RLI



What is this document?

This version of the BBS specification has been cleaned up and updated from the version released in November of 1993. Additional information is included in several appendices.

Changes, Corrections and Open Questions

CTL-C not really restricted character?

LF not used as alternate EOL

Added information from AA4RE on "M in SID".

Added information from W0RLI on "X in SID".

Legal character set for BID?

Contributors to the original document

bob@arasmith.com (Bob Arasmith, N0ARY)

HOredson@aol.com (Hank Oredson, W0RLI)

wd6cmu@netcom.com (Eric Williams, WD6CMU)

enge@almaden.ibm.com (Roy Engehausen, AA4RE)

71151.720@CompuServe.COM (Victor Poor, W5SMM)

blloyd@axion.bt.co.uk (Brian Lloyd, G1NNA)

map@hopper.cba.csuohio.edu (Mike Pechura, WA8BXN)

durham@w2xo.pgh.pa.us (James Durham, W2XO)

johan@ece.orst.edu (Johannes Reinalda, WG7J)

trentin@enac.dgac.fr (to J-P Roubelat, F6FBB)

sproul@sproul.com (Mark Sproul, KB2ICI)

mike@km6px.arasmith.com (Mike Stickney, KM6PX)

70007.1365@compuserve.com (Ed Juge, W5TOO)

73126.3260@compuserve.com (Craig McCartney, WA8DRZ)

bbr77@aol.com (Brian Riley, KA2BQE)

Michael Jaggers, WB4TTZ 1377 Columbia Ave, Franklin, TN 37064

Contributors to this update

HOredson@aol.com (Hank Oredson, W0RLI)

71151.720@CompuServe.COM (Victor Poor, W5SMM)

(Hans Kessler,N8PGR)

Contents

Introduction

This document is meant to define the interaction between two BBS systems as they exchange messages. It is not the goal of the document to specify the interface of the future but rather to represent what a BBS system can expected to see at the current time.

In brief the following sequence of events takes place when the BBSs connect for forwarding:

Master System	Slave System
{Connect to SLAVE}	(Connect from MASTER)
	[ZXF-3.21-H$]
	N9ZZZ >
[RLI-19.18-HIX$]	
	>
SB ARES @ ALLCA < W7ZZZ $ARES0108	
	OK
message subject	
message routing headers	
message body	
Control-Z	
	>
SB WANT @ ALLUS < W8AAA $1029_N0XYZ	
	NO
	>
F>	
	SP WA2ABC @ N2AAA < N9AAA
OK	
	message subject
	message routing headers
	message body
	Control-Z
F>	
(Receive Disconnect)	{Issue Disconnect}

The above exchange represents a simple example of connecting, establishing a common protocol, forwarding and reverse forwarding. This document examines in detail each of these phases of the process of data interchange between packet BBS systems.

In the text the BBS initiating the connect is referred to as the MASTER and the remote BBS to which it connects is referred to as the SLAVE.

Each command and exchange is presented in BNF (Backus-Naur Form) which is a context-free grammar widely used to describe the syntax of formal languages. All commands and fields are case insensitive.

Throughout this document we present a verbose translation of the grammar highlighting any points that may not be obvious but could have significant affect. There are also examples illustrating what can actually be expected in practice.

Meta-syntactic information

[x]	Optional: zero or one x
(x y)	Grouping: treat multiple tokens as a single token
<>	Set

Characters:

NUL	<0x00>
CNTRL_C	<0x03>
TAB	<0x09>
CR	<0x0D>
LF	<0x0A>
CNTRL_Z	<0x1A>
SP	<0x20>

Character sets:

ALPHA	<'A'-'Z'>
DIGIT	<'0'-'9'>
ALNUM	ALPHA | DIGIT
ASCII	<0x20-0x7F>
SPACE	SP | TAB
8BIT	<0x01 - 0xff, except CR, LF, CTRL_Z>

Character sequences:

EOL	CR | CR LF
NUMBER	DIGIT | NUMBER DIGIT
WHITE	SPACE | WHITE SPACE
ASCII_STR	ASCII | ASCII_STR ASCII
ALNUM_STR	ALNUM | ALNUM_STR ALNUM
8BIT_STR	8BIT | 8BIT_STR 8BIT

Section 1: Establishing connection

When connecting to a BBS the first line the BBS sends must be the SID (System Identifier). The SID is used to determine what features are supported by each BBS. The existence of the SID implies that the system supports reverse forwarding and OK/NO message rejection. There still exist old BBS programs that do not present a SID at connect. Refer to Section 4 for information on handling these old style BBS systems.

The SID as sent by both MASTER and SLAVE:

AUTHOR_ID	<ASCII_STR, except '[', ']' and '-'>
AUTHOR_DATA	<ASCII_STR, except '[' and ']'>
FEATURE	ALPHA [DIGIT]
FEATURE_LIST	FEATURE | FEATURE_LIST FEATURE
SID	"[" AUTHOR_ID "-" [AUTHOR_DATA "-"] FEATURE_LIST [$]"]" EOL

The AUTHOR_ID and FEATURE_SET must be present. If the optional AUTHOR_DATA is present it must be separated from the other two fields by a '-' character. AUTHOR_DATA may contain the '-' character. The AUTHOR_ID is typically a portion of the authors callsign or BBS name. The AUTHOR_DATA is typically the version number of the software. Each FEATURE consists of a letter, denoting the feature and an optional following number, denoting the feature level. The absence of a version number implies version zero. If BID's are supported, '$' must be the last symbol in the FEATURE_SET. The recommended minimal support is BID and hierarchical location identifiers. These are necessary for the proper operation of the network.

EXAMPLE: [ZFJ-2.3-H$]

The original prompt received from the SLAVE BBS may contain additional information intended for users. As soon as the SID is received by the SLAVE BBS it should respond with an abbreviated PROMPT, with the ASCII_STR field being empty.

Section 2: Forwarding

A message can be accepted of rejected by the SLAVE. There are many reasons that the SLAVE may choose to reject a message. The most common is a duplicate BID, indicating the SLAVE already has the message. Any use of a NO response other than to indicate a duplicate BID is discouraged since it will cause loss of messages. Here is an overview of the exchange.

ACCEPT EXCHANGE		REJECT EXCHANGE	
MASTER	SLAVE	MASTER	SLAVE
	PROMPT		PROMPT
SEND_CMD		SEND_CMD	
	RESPONSE (OK)		RESPONSE (NO)
MESSAGE			PROMPT
	COMMENT		
	PROMPT		

The SLAVE indicates it is prepared to accept a command by issuing a PROMPT. The PROMPT is also used as an acknowledgment of correct reception of the previous message. The MASTER initiates the message exchange by issuing a SEND_CMD. This command contains all the information necessary for the SLAVE to decide whether to ACCEPT or REJECT the message. It contains message type, destination, originator and possibly a bid.

```
COMMENT      <ASCII_STR, except '>'>
PROMPT       [COMMENT] ">" EOL
TYPE         'B' | 'P' | 'T'
CALLSIGN     <ASCII_STR, except '@', '.', max length 6>
LOC          <ASCII_STR, except '@', '.', max length 6>
HLOC         LOC | <HLOC "." LOC, max length 31>
BBS          CALLSIGN ["." HLOC]
TO           CALLSIGN [WHITE "@" WHITE BBS]
FROM          CALLSIGN
BID          <ASCII_STR, max length 12>
SEND_CMD     "S" TYPE WHITE TO [WHITE "<" WHITE FROM] [WHITE "$"
                BID]
```

Message Types:

Type B is a bulletin to be distributed to all BBS system. Type P is a personal message destined to one recipient. Type T is an NTS traffic message destined outside the Amateur Radio Service.

HLOC is a hierarchical location of the system. It is defined by the document "INTERNATIONAL ROUTING DESIGNATORS", ARRL Networking Conference Proceedings.

The TO field commonly takes the following forms:

callsign @ BBS.hloc	N6ZFJ @ N6QMY.CA.USA.NA
category @ distribution	WANT @ ALLCA
category @ BBS.hloc	WANT @ N6QMY.CA.USA.NA
category @ distibution.hloc	INFO @ ALLCA.CA.USA.NA

The first two are seen most often, but the other two forms are seen occasionally. The first two forms are also normally associated with the message type of PERSONAL and BULLETIN respectively. White space is required on either side of the '@' and between the '<' and the senders callsign. There cannot be white space between the '$' and the BID.

EXAMPLE

```
    SB WANT @ ALLSCV < N6ZFJ $2345_N6QMY
```

A "BID" (Bulletin Identifier) is associated with each Bulletin, and may be associated with any Personal message. If a BID is not given explicitly when the message is entered into the system, and the message type requires a BID, one is created automatically from the message number and callsign of the system into which the message was initially entered.

NOTE:

It is suggested that a generated BID have the form nnn_BBS, where nnn is the message number at the originating BBS system, and BBS is the callsign of the originating BBS system.

COMMON_BID	MSG_NUM "_" CALLSIGN

BBS programs are free to create whatever form they wish as long as the length does not exceed 12 characters.

Messages to a distribution, whether they are bulletins or personal messages, must have a BID. Personal messages to a specific user may carry a BID if the originator supplied one, but will not be given a default BID. NTS message never carry a BID.

The SLAVE BBS determines from the SEND_CMD whether or not to accept the message. The reason for rejection is usually that the message already exists on the SLAVE system (i.e. a duplicate BID). It may also indicate a system problem.

Sent by SLAVE in response to SEND_CMD (or by MASTER if in reverse forward mode).

RESPONSE	ACCEPT | REJECT
ACCEPT	"OK" [WHITE COMMENT] EOL
REJECT	"N" ["O"] [WHITE COMMENT] EOL

NOTE:

The syntax of responses has grown in a random fashion. Many BBS programs expect to see the entire words "OK" and "NO" while others simply look at the first letter. It is suggested that new BBS programs should present the words "NO" and "OK", but look for only the first letter. Doing this should ensure compatibility with existing BBS programs.

```
  EXAMPLE
   Master:  SB WANT @ ALLCAN < N6ZFJ $4567_N0ARY
```

```
 Slave:   NO duplicate bid
 Slave:   >

 Master:  SB WANT @ ALLCAN < N6ZFJ $4567_N0ARY
 Slave:   OK #32190
 Master:  message body
 Master:  Control-Z
 Slave:   Message being held for sysop review
 Slave:   >
```

"OK" and "NO" must be followed by white space or EOL. If this is not done, some BBS programs will fail to recognize them.

```
EXAMPLE
 Master:  SB WANT @ ALLCAN < N6ZFJ $4567_N0ARY
 Slave:   NO, already have it
 Slave:   >
```

This would fail with some existing BBS programs due to the comma following the "NO" string.

If the SLAVE BBS responds with a REJECT condition the message should not be sent and the MASTER should ignore any response from the SLAVE until PROMPT is encountered. The MASTER can then submit a new SEND_CMD, initiate reverse forwarding or disconnect.

Once the SLAVE has issued the ACCEPT response, the MASTER BBS should immediately send the message subject, routing headers, message body and EOMA. The SLAVE will acknowledge with PROMPT once it receives an EOMA.

Message Routing Header

SUBJECT	<ASCII_STR, max length 79> EOL
YEAR	DIGIT DIGIT
MONTH	DIGIT DIGIT
DAY	DIGIT DIGIT
HOUR	DIGIT DIGIT
MINUTE	DIGIT DIGIT
STAMP	"R:" YEAR MONTH DAY "/" HOUR MINUTE ["Z"]
MSG_NUM	<NUMBER, range 1 - 65535 (0x0001 - 0xFFFF)>
HEADER1	STAMP WHITE MSG_NUM "@" BBS EOL
HEADER2	STAMP WHITE "@:" BBS WHITE [ASCII_STR] "#:" MSG_NUM [ASCII_STR]
HEADER	HEADER1 | HEADER2

Message structure

HEADERS	HEADER | HEADERS HEADER
EOMA	CNTRL_Z EOL
EOMB	EOL "/EX" EOL
BODY	8BIT_STR | BODY EOL 8BIT_STR

MESSAGE SUBJECT [HEADERS] EOL [BODY] (EOMA | EOMB)

Note:

Given the above definition of a message BODY, the BODY cannot contain either EOMA or EOMB. Common practice is to replace the string “/EX” if found in the BODY with the string “’/EX’” (enclose the text in single quotes). This restriction in the protocol has caused a few problems but is not felt to be serious.

EXAMPLE

Example message

```
R:930108/1729 @:N6QMY.#NOCAL.CA.USA.NA Fremont #:1114 Z:94536
R:930108/1259 1530@KA6FUB.#NOCAL.CA.USA.NA
R:930107/1638z @:WD6CMU.#NOCAL.CA.USA.NA "OSKbox" #:10576 Z:94805
R:930107/1045 50724@WX3K.#NOCAL.CA.USA.NA
```

The message body goes here. The blank line above separates the body from the routing headers. The BODY should be terminated with EOMA, but EOMB is allowed to support some existing codes for example WinLink and AA4RE.

Notes:

Both forms of the header described above are seen, and should be parsed. In either case there may be comments found at the end of the header line, or between the BBS and the MSG_NUM entries in HEADER2.

HEADER2 is derived from the header form used in early BBS programs. It allows additional information to be passed in the header and parsed if desired. Some examples of these additional optional fields are:

Z:zipcode

This is frequently seen, and is used by White Pages data capture. It’s utility is unclear.

O:orginator

This is probably obsolete.

$:bid

The meaning of this field is unclear. It appears on messages which have no BID, i.e. some personal messages, and also appears on NTS messages, which cannot have a BID.

C:name_of_sysops_cat

This has only been used by W0RLI and N6IYA, as a test.

B:bid-as-received

This is seen only from W0RLI systems. It shows the BID as the system received it in the S command or RFC-822 import.

Notes:

The time stamp in the routing header must be in GMT. Some existing BBS programs provide the time stamp in other than GMT, and may include a time zone indicator after the time stamp. This should be considered in the design of header parsers. The design of future BBS programs should conform to this specification.

Each system that handles the message places it's routing header at the top of the message headers before passing the message to the next BBS.

The headers must begin in column 1. A blank line separates the routing headers from the body of the message.

The first (oldest) routing header shows the BBS where the message originated. This header is the only source for this information. The information about the originating BBS is used to create a return address for reply messages.

Here are two examples of the same header, in the two different formats:

R:930101/0000 1530@KA6FUB.#NOCAL.CA.USA.NOAM

R:930101/0000 @:KA6FUB.#NOCAL.CA.USA.NOAM #:1530

These examples show the different syntax for the message number and originating BBS address. All fields not shown in the examples are optional at the discretion of the BBS author or the individual sysop.

Section 3: Reverse forwarding

Once all messages have either been successfully forwarded from the MASTER to the SLAVE, or been rejected by the SLAVE, the MASTER may accept messages from the SLAVE. The MASTER and SLAVE exchange roles. This is called "reverse forwarding".

ACCEPT EXCHANGE:

MASTER	SLAVE
	PROMPT ">"
REV_FWD "F>"	
	SEND_CMD
RESPONSE "OK"	
	MESSAGE
COMMENT	
REV_FWD "F>"	

REJECT EXCHANGE:

MASTER	SLAVE
	PROMPT ">"
REV_FWD "F>"	
	SEND_CMD
RESPONSE "NO"	
REV_FWD "F>"	

The MASTER indicates it is ready to accept a message by answering the Slave’s PROMPT, ">", with a REV_FWD command, "F>". The SLAVE then issues a SEND_CMD and the process continues as described in section 2, with the rolls of MASTER and SLAVE reversed. The only difference is that the MASTER responds with REV_FWD in place of PROMPT to indicate successful message transfer.

REV_FWD   "F>" EOL

The reverse forwarding phase is optional and is under the control of the MASTER. It can be terminated by any of the following:

The SLAVE disconnects

The SLAVE sends anything other than a legal SEND_CMD

The MASTER disconnects

The MASTER sends anything other than "F>".

Section 4: Forwarding to old BBSs

If the SLAVE does not return a SID then it is assumed that it will not support BIDs, hierarchical location designators or OK/NO message rejection.

The SLAVE indicates it is prepared to accept a command by issuing a PROMPT. This PROMPT is also an acknowledgment that the previous message has been received.

The MASTER initiates the message exchange by issuing a SEND_CMD with no HLOC or BID. The SLAVE has no way to reject the message so the MASTER proceeds to send the message subject and body without expecting acknowledgment from the SLAVE.

Since the SLAVE assumes a user is entering the message, it will display prompts to lead the user through the message sending process, asking for a subject, prompting for the message body, etc. The MASTER should consider this text to be part of the PROMPT and ignore it.

As in Section 2, the MASTER waits until PROMPT is received. This is the acknowledgment that the message has been received and the next message can be sent.

When the MASTER has no additional messages to send it disconnects from the SLAVE. Reverse forwarding is not supported in this situation.

Section 5: Half-Duplex links.

When two BBS systems communicate over a half duplex link, the following two conditions apply: 1) on link initiation, the SLAVE shall send the first data over the link. 2) The station sending data shall relinquish the link when it has finished sending. In the case of TOR, for example, this means the SLAVE is ISS when the link is established and ISS always issues the OVER.

Section 6: Error Conditions

When either the MASTER or the SLAVE detects an error, that station disconnects at once, without issuing a PROMPT or any other data.

Section 7: BNF Summary

Meta-syntactic information.

```
[x]              =   Optional Zero or one x.
(x y)            =   Grouping.  Treat multiple tokens as single token.
<>               =   Set.
```

Characters of interest.

```
TAB              =   <0x09>
LF               =   <0x0a>
CR               =   <0x0d>
CNTRL_Z          =   <0x1a>
SP               =   <0x20>
```

Character sets of interest.

```
ALPHA            =   <'A' - 'Z', 'a' - 'z'>
DIGIT            =   <'0' - '9'>
ALNUM            =   ALPHA | DIGIT
ASCII            =   <0x20 - 0x7f>
SPACE            =   SP | TAB
8BIT             =   <0x01 - 0xff, except CR, LF, CTRL_Z>
```

Character sequences of interest.

```
EOL              =   CR | CR LF
NUMBER           =   DIGIT | NUMBER DIGIT
WHITE            =   SPACE | WHITE SPACE
ASCII_STR        =   ASCII | ASCII_STR ASCII
ALNUM_STR        =   ALNUM | ALNUM_STR ALNUM
8BIT_STR         =   8BIT  | 8BIT_STR  8BIT
```

System Identifier syntax.

```
AUTHOR_ID        =   <ASCII_STR, except '[', ']', '-'>
AUTHOR_DATA      =   <ASCII_STR, except '[', ']'>
FEATURE          =   ALPHA [DIGIT]
FEATURE_LIST     =   FEATURE | FEATURE_LIST FEATURE
SID              =   "[" AUTHOR_ID ["-" AUTHOR_DATA] "-" FEATURE_LIST "]" EOL
```

System Prompt syntax.

```
COMMENT          =   <ASCII_STR, except '>'>
PROMPT           =    [COMMENT] ">" EOL
```

Send Command syntax.

```
TYPE             =   'B' | 'P' | 'T'
CALLSIGN         =   <ASCII_STR, except '@', '.', ' ', maximum length 6>
```

```
LOC           =   <ASCII_STR, except '@', '.', ' ', maximum length 6>
HLOC          =   <LOC | HLOC "." LOC, maximum length 31>
BBS           =   CALLSIGN ["." HLOC]
TO            =   CALLSIGN [("@" | WHITE "@" WHITE) BBS]
FROM          =   CALLSIGN
BID           =   <ASCII_STR, except ' ', maximum length 12>
SEND          =   "S" TYPE WHITE TO [WHITE "<" WHITE FROM] [WHITE "$" [BID]]
```

Message syntax.

```
SUBJECT       =   <ASCII_STR, maximum length 79> EOL
YEAR          =   DIGIT DIGIT
MONTH         =   DIGIT DIGIT
DAY           =   DIGIT DIGIT
HOUR          =   DIGIT DIGIT
MINUTE        =   DIGIT DIGIT
STAMP         =   "R:" YEAR MONTH DAY "/" HOUR MINUTE
MSG_NUM       =   <NUMBER, range 1 - 65535>
HEADER1       =   STAMP WHITE MSG_NUM "@" BBS EOL
HEADER2       =   STAMP WHITE "@:" BBS WHITE [ASCII_STR] "#:" MSG_NUM
                  [ASCII_STR]
HEADER        =   HEADER1 | HEADER2
HEADERS       =   HEADER | HEADERS HEADER
BODY          =   8BIT_STR | BODY EOL 8BIT_STR
EOMA          =   CTRL_Z EOL
EOMB          =   EOL "/EX" EOL
MESSAGE       =   SUBJECT [HEADERS] EOL [BODY] (EOMA | EOMB)
```

Standard message identification syntax.

```
MSG_ID        =    MSG_NUM "_" CALLSIGN
```

Section 8: Feature Letter Use

Feature $:

Supports BIDs. This feature must be supported.

Feature A:

F6FBB Acknowledge for personal messages.

Feature B:

F6FBB Binary messages supported.

Feature C:

Supports automatic distribution of current date / time. (W0RLI, K0ZXF) This feature is obsolete:

Feature F:

F6FBB Batch forwarding protocol supported.

Feature H:

Supports Hierarchical Location designators. This feature must be supported.

Feature I:

Some transport mechanisms do not identify the calling station in the protocol, as does AX.25. To handle this situation, a "null command" is defined as a line beginning with ';'. This allows the sending station to send callsigns to satisfy the regulations for station identification.

EXAMPLE

```
; W5SMM de W0RLI
```

The receiving station must ignore such lines.

Feature L:

G1NNA – Compression supported.

Feature M:

For AA4RE systems:

Basic level ("M0")

MID = message Identifier

Every message has a MID. The MID looks like a "generated" BID (example 12345_AA4RE). The "M" letter in the SID indicates that a MID is to be transmitted with messages that have no BID. Processing: If a send command is received by a BBS with a MID, the program should determine if it is a duplicate. If it is, the program should respond with the usual "NO" as if it was matching a BID.

Note:

This protocol will on occasion cause the loss of messages. Its use is discouraged.

Feature R:

AA4RE - Extended reject responses.

OK

> Yes, send to me. Note that no number is required. This may help eliminate phantom message numbers. (Ed. Note: What are “phantom message numbers?”)

NO

> This is a real dupe

REJECT

> I can't handle this

LATER

> Send to me later

The LATER is useful when you are already receiving a bulletin on one connect and someone else tries to send you the same bulletin.

Rejected messages should be called to the attention of the SYSOP to see why the reject occurs.

The difference between NO and REJECT is whether or not SYSOP attention is needed.

Abbreviations are allowed: 1 Letter only ... O N L R

There must be a blank after the word or letter. This one is fine:

> N Already have it.

This one is not correct:

> REJECT, Something wrong

The comma is not allowed.

Feature S:

AA4RE - Extended "S" commands supported.

Feature T:

WinLink.

Feature U:

WinLink.

Feature X:

Supports ihave/iwant compressed batch forwarding. Defines the syntax and use of commands SS, SX, SY. Uses lzh compression

Command Definitions and BNF description.

N	INTEGER, size 32 bits.
file_size	N
msg_count	<range 0 - 25>
checksum	<file checksum>

id	BID | MID CR
ids	id | ids id
byte	<0x00 - 0xff)
bytes	byte | bytes byte
ihave	"SX " msg_count CR [ids]
iwant	"SY " msg_count CR [ids]
isend	"SS " file_size [" " checksum] CR bytes

For iwant:

SY means want only these.

Note that the SS command must terminate with CR, without LF.

The optional checksum is computed by summing all the bytes in the file as unsigned integers. Example code:

```
{
  int byte;
  unsigned int sum;

  sum = 0;
  for (all bytes in file) sum += byte;
}
```

RFC-822 header fields used are:

To:

From:

Subject:

Message-ID:

X-msgtype:

X-BID:

The basic interchange (example):

bn are BIDs (or MIDs if the message has no BID). m: is master, s: is slave.

m: SX 5

m: b1

m: b2

m: b3

m: b4

m: b5

s: SY 3

s: b1

s: b3

s: b5

m: SS 26545
m: <sends 26545 byte file>

In the case of any error, the station noticing the error must disconnect immediately.

Section 9: Import / Export files.

Import/Export files may contain any number of messages. Each message is terminated by EOMB

RFC-822 header use in import/export files.

The standard RFC-822 keywords, which must be supported, are:
To:

From:

Subject:

Message-ID:

cc:

Extension keywords, which must be supported, are:
Produced by W0RLI systems
X-msgtype:

X-BID:

Produced by JNOS systems
X-BBS-Msg-Type:

X-BBS-Hold:

X-Forwarded-To:

Appendix A: Discussion of BIDs and MIDs.

Response by W0RLI to a query from KB7OGD about BIDs and MIDs Wed Dec 20 19:17:59 1995:

A MID is the identifier of a message, consisting of the message number and BBS callsign. It does not have any particular display format. That is, you could say, "Message 2387 at W0RLI" or you could say, "2387_W0RLI" or you could say "W0RLI's#2387" It is just a way to identify a particular instance of a message. By instance is meant a specific copy of a message: your message to me had it's first instance on your system, as your number xxx, then it was forwarded to me, where a second instance was created on my system. That second instance was my number yyyy. Through the headers, the message carries information about each instance as it forwards from system to system. That's what a MID is about.

A BID is a bulletin identifier. It is just a string of characters, not longer than 12, which stays with every instance of a message. It is not quite that simple however. A BID actually identifies the *contents* of a message. Thus, a message containing the orbital data for the Oscar birds might be entered in several different BBS at about the same time. When entered into each BBS, it is given the same BID (ORBS-237.O for example) by the person entering it. The BID is used to assure that only one instance of this message will appear at any one BBS. Which original this instance derives from is unimportant. Every bulletin must carry a BID, so that there is a way to assure that only one instance of the bulletin appears at each BBS, no matter how many different paths that bulletin might take on it's way to the particular BBS. So when a user enters a bulletin and does not specify a BID, the system must generate one for him. That is often done by using the information contained in the MID (message number, BBS callsign). Thus we have the "usual" bulletin BID of nnnn_call or call_nnnn or nnnncall, etc. Some system simply generate a unique string of characters possibly unrelated to the system callsign or message number, for example my list server does exactly that.

Personal messages may also carry a BID. This will occur if the user gives the message a BID (sp w0rli@k1ugm.ma.usa.noam $mybid003), or if the message is to be treated like a bulletin and distributed.

NTS messages may never have a BID.

What happens if a personal message has a BID? Normally, nothing unusual happens. However, if that message has to return to a system where an instance of the message already exists, it will be refused. That is the purpose of the BID. This behavior can cause the loss of personal messages, if they are given a BID.

There is also great confusion between the MID and the representation of the MID. The MID "is" the message number and callsign at the system where the particular instance of the message resides. It can be represented any way the author of the software chooses to represent it. I use <12345@W0RLI> for example (complete with the angle brackets). I do it this way to help the sysop avoid confusing the MID with the generated BID, which "is" the string of characters and not some particular number/callsign etc. The confusion arises because the characters normally shown for a MID are the same characters normally shown for a generated BID, even though the two have nothing whatsoever to do with each other. Some BBS programs confuse the two, which causes serious problems when forwarding. This is where duplicates and lost personal messages occur.

There is one more term we need to be able to discuss all the instances of a particular message. A LID is the ID of an instance of a message. The term “MID” is normally used only to mean the LID of a message at the originating bbs. Messages have lots of different "identities".

A final example: that AMSAT bulletin if it originated at your system, and was then forwarded to mine.

At your system it could be "Number 1234 at KB7OGD". At mine it could be "Number 4321 at W0RLI". Those are both LIDs (they identify the particular instance of the message on each system). It's MID is "Number 1234 at KB7OGD" because it originated there. That information is in your header, which is the first header. Its BID is ORBS-237.O, which was assigned by the user who entered it into your system. The BID is the same for every instance.

Now assume the same bulletin was also entered at WA7SJN. At Bill's system it could be "Number 2345 at WA7SJN." That is it's LID at Bill’s system, and also it's MID at any system to which it is forwarded. It has a BID of ORBS-237.O, just like the other two instances above. It cannot be forwarded to me or to you (that would be a duplicate.)

Now let's see what happens at Duane’s system. It could happen that either message (MID of "Number 1234 at KB7OGD" or MID of "2345 at WA7SJN") gets there first. Whichever one gets there second will be rejected. However, Duane will get exactly one copy of the bulletin, and it will have a BID of ORBS-237.O.

Why do we need to consider LIDs and MIDs? Because we need to be able to talk about HOW the instances of a message arrived at a given BBS, so we can track down forwarding problems, problems of data corruption, etc.

Questions like "Why is the ORBS-237.O at WA7SJN different from the one at W0RLI?" crop up from time to time. Usually we just follow the headers back to the source (follow the LIDs back to the MID) and eventually discover the problem. The same technique can be used to discover the source of duplicates as well. Duplicates occur when some system changes the BID it was given, thus producing a "new bulletin" which happens to have the same contents as an existing bulletin.

This should give some idea why it is important to say what that “1234_KB7OGD" "is". It could be a LID, or a MID, or a BID. Until one specifies which one it is, it is simply a string of characters, with no meaning at all. It could be your Visa card balance for all anyone knows.

Comments by W0RLI to WA8BXN on Wed Dec 20 23:53:39 1995:

If it is a P message, and not to a distribution, then it can only have a BID if the user intentionally entered one. If there is no M in the SID, then FBB will behave properly, and only send a BID with the S command. It is that old confusion about message identity (specifically the difference in identity of message instances depending on message type) that Roy started with the "M in the SID" thing. Not having an "M" in one's SID allows things to work properly (rejecting on duplicate BID, marking for sysop attention on duplicate MID.)

The X-BID: line in the rcc-822 header can only be present if there is a BID. If the message does not have a BID, then the X-BID: line must *not* be present. The X-BID: line is there to pass the BID if the message has one. The MID is passed in the header, as usual. It is the lack of an X-BID: line that tells you the message has no BID, just as the lack of a trailing "$" field on the S command tells you the user did not assign a BID. (Note that this means you can tell the difference between a "real BID" and a "generated BID" and we could pass this info along also. We don't yet, and the protocol has no means to support this feature. My code keeps track of this difference internally.

This has nothing to do with the proposal. The proposal uses a BID if one exists, and a standardized representation of the MID if no BID exists. One must check only against BIDs already seen. The message cannot be rejected on already seen MID. Note that there will not be confusion between a generated BID and a MID as seen in the proposal. BBS programs store them in different databases so they can tell whether a string of characters one has seen before is a MID or is a BID.

So what you do is check for duplicate BID *before* you get the actual message, because you can reject the message due to a duplicate BID. You check for duplicate MID *after* you get the message because you cannot reject the message due to a duplicate MID. You can only mark a message, which has a duplicate MID, for attention of the sysop, since there is probably some routing problem. Doing things any other way will result in lost personal messages, as they will be rejected instead of accepted if they happen to pass through a system more than one time. True, this should "never" happen, but it often does, due to routing changes, HF systems which pass messages back and forth depending on time of day, routing loops, mistakes made in addressing messages, etc. It is far better to allow personal messages to go back where they came from instead of just tossing them into the bit bucket. Thus "reject *before* you get it on duplicate BID, *mark* after you get it on duplicate MID."

Appendix B: Hierarchical Addressing Protocol

This is an extract and update from the original document. The specific regional and area designators have been removed, since they are determined on a local basis.

Authors:

Dave Wolf, WO5H
Barry Buelow, WA0RJT
Roy Engehausen, AA4RE
Hank Oredson, W0RLI
Greg Jones, WD5IVD

Date:

August 30th, 1994

Adopted by TAPR:

March 3rd, 1995

Send Updates to Attached Tables to:
Barry Buelow, WA0RJT, TAPR BBS SIG Chair
Packet: wa0rjt@wa0rjt.ia.usa.noam
Internet: barry@ia.net

Introduction

The TAPR BBS Special Interest Group recommends the adoption of the x.3.4 hierarchical address protocol.

After discussion of previous articles on hierarchical addressing standards (1,2) and taking into account international issues of regional/state name sizes, the TAPR BBS Special Interest Group recommends the adoption of the x.3.4 standard on an international basis. x is defined as 2, 3, or 4 letter region names as defined by the country.

Examples of x.3.4:

@WA6GVD.CA.USA.NOAM

@EA2CMO.EAZ.ESP.EURO

@F5JGK.FAQI.FRA.EURO

Regional identifiers may be duplicated in different countries (i.e. AK, Alaska, USA, could be used in another country as a regional identifier); however, Country and Continental identifiers SHOULD NOT be used as regional names.

It is important to note that there is a distinct and significant difference between HIERARCHICAL ADDRESSES and FLOOD DESIGNATORS. Hierarchical address elements are common to all messages (bulletins, P and T types) and are the foundation of the digital forwarding system. Flood designators are used for routing and filtering bulletins. Geographical flood designators are likely based upon hierarchical address elements. It is therefore important that any attempt to establish standards concentrate first on hierarchical address elements. Standards for flood designators can follow.

It is the purpose of this document to generate a changing recommendation that reflects current hierarchical routing. Reference Tables at the end will be changed as necessary to reflect current configurations within the international BBS network. These tables will need to be changed and updated in order to meet future needs of user and sysops.

Hierarchical Routing Syntax Summary

This summary uses a modified Backus-Naur form to summarize the syntax for hierarchical addressing. Optional fields are shown within square braces.

Hierarchial_address = bbs.[“#”area.]region.country.continent

bbs = Callsign as defined by the local communications authority.

Area = additional identifier depending on local requirements.

Region = As defined by the local country. This is normally the abbreviation for the state, province, prefecture, etc.

Country = 3-character country identifier as defined by ANSI X.12 and EDIFACT. Published in ISO 3166-1981(E/F).

Continent = 4-character continental identifier from the following table.

TABLE 1: Continent Identifiers

EURO	Europe
MEDR	Mediterranean
INDI	Indian Ocean including the Indian subcontinent
MDLE	Middle East
SEAS	South-East Asia
ASIA	The Orient
NOAM	North America (Canada, USA, Mexico)
CEAM	Central America
CARB	Caribbean
SOAM	South America
AUNZ	Australia/New Zealand
EPAC	Eastern Pacific
NPAC	Northern Pacific
SPAC	Southern Pacific
WPAC	Western Pacific
NAFR	Northern Africa
CAFR	Central Africa
SAFR	Southern Africa
ANTR	Antarctica

Examples:

F6CNB.#SETX.TX.USA.NOAM

KB7WE.#WWA.WA.USA.NOAM

OH6RBV.#VAA.FIN.EURO

SK2AT.AC.SWE.EURO

OH6RBG.FIN.EURO

KE7KD.#NONEV.NV.USA.NOAM

WX3K.#NOCAL.CA.USA.NOAM

References:

1. Jenkins, Lew (N6VV), Dave Toth (VE3GYQ), and Hank Oredson (W0RLI). International Routing Designators. Proceedings of the ARRL 7th Computer Networking Conference. Columbia Maryland. October 1, 1988. pp. 91-93.

2. Clark, Tom (W3IWI). Some comments on the Hierarchical Continent Address Designator. Proceedings of the ARRL 9th Computer Networking Conference. London, Ontario Canada. September 22, 1990. Pp. 278-279.

Appendix C: Distribution Designators

FROM: N6IYA @ N6IYA.#WCCA.CA.USA.NOAM
SUBJ: Forwarding and Distribution

**** BBS FORWARDING STANDARDS - DRAFT ****

This document is submitted as a foundation for the forwarding of messages in the Personal Message Forwarding System (PMFS) and Bulletin Distribution Forwarding System (BDFS) as used by the Amateur Radio Packet Network as presently constituted in the United States of America. Its main focus is on establishing a set of standards for the distribution of bulletin type messages meant for the general consumption of all licensed Amateur Radio Stations in a message forwarding system. It assumes an understanding of Hierarchical Addressing in the PMFS and the difference between a routing and a distribution.

This proposal is base on the premise that the unused domains in Hierarchical addressing can be re-used in a bulletin forwarding scheme. It also assumes that the current generation of software can differentiate between a type P and type B message. The proposal has been tested and is in use in areas against the various types of software that are currently in use. Older versions of software may not be able to support this and the users of this software should be encouraged to upgrade to more current versions. Bulletins are considered to be distributions and are distributed according to the token as placed in the @BBS field. Currently the tokens used operate using a modified form of the continental or state code with ALL or BBS added.

The initial usage of this form of token was to differentiate bulletins in the infancy of packet message handling. This was because the early versions of the code could not differentiate between @XX and .XX. But these were the days when we all kept callsigns for all of our forwarding schemes no matter where the message was destined. To understand how this scheme would work it is necessary to understand the personal messaging system and its addressing.

THE PERSONAL MESSAGING FORWARDING SYSTEM (PMFS)

With more sophisticated code we can now differentiate between @XX and.XX and type P and type B traffic to the point that our personal message forwarding scheme has been reduced to the use of the state, country and continental H addresses except within our immediate domain.

The domains are defined in a personal messaging system in format of bbs.region.state.country.continent

Continent: Four characters, e.g. NOAM.

The largest geographic region used only to move personal traffic in/out of this continent. Once the message is within this domain the continent is ignored.

Country: Three characters, e.g. USA.

This domain is the next smallest. Once the message is inside of the domain that domain is ignored.

State/Province: Two characters, usually a postal code e.g. NY.

The next once inside of this domain it is ignored.

Region: No more than six characters starting with “#”, e.g. #WCCCA

This domain is optional, at the discretion of the local region. In use where the next highest domain is split smaller units. Once inside of the domain it is ignored.

Note: This is an optional domain the octothorpe is required where the regional domain token might conflict with and existing token or postal code. Example #97 and Zip Code routing 97*.

BBS: The callsign of the target BBS system, e.g. N6IYA.

Used to move traffic inside of the region. Only ignored at the destination system.

The system using the above would contain in its forwarding files the minimum amount of information required to move personal messages to their intended addressee. The examples are based on my BBS where the H address is Callsign @ N6IYA.#NOCAL.CA.USA.NOAM

All of the continental codes except for its own domain address.

For example: within North America the NOAM designator would not be present and is not considered for forwarding.

All of the Country codes within its continental domain except for its own domain address.

For example: within North America and the United States the USA designator would not be present and not considered for forwarding.

All of the state or provincial codes within its country domain except for its own domain address.

For example: within the United States the CA designator would not be present and not considered for forwarding.

All of the regional codes within its state domain except for its own regional domain address. This is optional and not required. All others are required.

For example: within CA the #NOCAL designator would not be present and not considered for forwarding within its own domain. It would be used from the other adjacent domains to move traffic into the area on an optional basis.

All of the BBS callsigns with its regional or State (if regional forwarding is not used) domains except its own.

For example: at the N6IYA system the N6IYA callsign would not be present and not considered for forwarding.

BULLETIN DISTRIBUTION FORWARDING SYSTEM (BDFS)

All of the schemes in use currently adapt the state or continental code by using ALL or BBS as a prefix or suffix in the format of ALLXX, XXALL XXBBS or BBSXX and places this token in the @BBS field. This limits the 6-character field to 3 useful information-carrying characters. It also limits the flexibility and expendability of the current system because there are no effective standards for use of this field. The system in place severely limits the ability to distribute bulletin traffic either in a wide scale scheme or to specific target distributions. All of the tokens in place have at their root the continental, state or regional designators that are used in the personal messaging forwarding system detailed above.

This is the least common denominator and by dropping the ALL and BBS will allow network expansion and flexibility for bulletin distribution by freeing the three character fields currently occupied by ALL and BBS for use in determining bulletin distribution tokens. This would give the current system a greater degree of flexibility in creating addition distribution tokens.

STATE DISTRIBUTION

Bulletins that are to be sent to a state are to be addressed @XX where XX is the state USPS Code. A second token in the form of the standard H address can be used to further route the traffic to its intended distribution target. So a bulletin destined to OK would be addressed as SB <Topic> @ OK.OK. The .OK routes the message and the @ OK determines its distribution. In the case where a distribution is required outside of the current country domain it will be necessary to add the appropriate country domain to the address. (See Below for details) A simplified method would be to add the whole Hierarchical string to every message. This would remove any confusion.

State addressing can be further broken down around the cardinal points of the compass to subdivide a state for more precise targeting. These precede the XX state code and should not exceed 4 characters in length

N = North
S = South
E = East
W = West
C = Central
NE = Northeast
NW = Northwest
SE = Southeast
SW = Southwest
Examples:

SB <Topic> @ TX Targets whole state

SB <Topic> @ NTX Targets North Texas

SB <Topic> @ NETX Targets Northeast Texas

If further subdivision is desired there is plenty of ample space for additional designators. I don't anticipate too much breakdown beyond the ones listed. Conflicts are always possible and will have to be considered on a case by case basis. If a regional type distributor token is in use that conflicts with the above rules the offending designator should be changed. For example, just dropping the "all" from the local distributor would result in a distributor of CAN (California North). This would conflict with the country distributor of CAN for Canada. The local distributor should be changed to conform with the above rules to NCA (Northern California).

COUNTRY DISTRIBUTION

Bulletins for countrywide distribution would be addressed as SB <Topic> @ XXX where XXX is the country code. This is the current practice in Europe and works quite well. Again the second token would be required to route the bulletin to its intended target distribution if outside of the local domain.

(See State section above.)

SB <Topic> @ USA for the United States

SB <Topic> @ CAN for Canada

SB <Topic> @ MEX for Mexico

If the bulletin is to be targeted to a country outside of your own Continental domain then:

SB <Topic> @ XXX.YYYY.

SB <Topic> @ FRA.EURO Target a bulletin to France

SB <Topic> @ FRA Target a bulletin to France *

* If there are 2 different paths for message movement one for personals and one for bulletins the second address would be used in this situation in addition to distribution within the country domain.

Bulletins to be sent to a Continental area are to be sent as @XXXX (NOAM) as dictated by the receiving parties. NOAM is to include CAN, MEX and USA.

SB <Topic> @ NOAM would distribute all of North America.

SB <Topic> @ EURO would distribute to Europe.

Worldwide distribution would still be @WW.

Regional distribution remains at the discretion of each region but could also be addressed under the same scheme. What is presented above will take care of the vast majority of the bulletin addressing and give the users some flexibility.