AUTOMATIC PACKET REPORTING SYSTEM

APRS PROTOCOL REFERENCE

Protocol Version 1.2

Authors	The APRS Working Group
Document Version	**Draft 1.2 c**
Filename	APRS12c.pdf
Date of Issue	**November 2024**
Copyright	©2000 APRS Working Group All rights reserved
Technical Editor	v1.0 Ian Wade, G3NRW v1.2 John Langner, WB2OSZ

Version 1.2 -- Unofficial Compilation

APRS is still alive and well a quarter of a century later, with many new products, computer applications, and creative uses. Many corrections, clarifications, and additions have been made to the APRS protocol specification since APRS101.PDF was originally published. All of these updates were documented separately and informally, rather than being merged into this document.

- http://www.aprs.org/aprs11.html
- http://www.aprs.org/aprs12.html

Implementation is more difficult and error prone with the information scattered around. Some people, trying to implement APRS, might not even know there were any updates. This is a compilation of the original specification and all of the relevant updates since 2000.

This is not endorsed by the APRS Working Group. The position of the APRS Working Group is that the "official" version is the original APRS101.PDF and Bob's version of the errata on his site.

John Langner, WB2OSZ, editor.

April 2024

Version 1.0 Published by
Tucson Amateur Packet Radio Corp
8987–309 East Tanque Verde Road, #337
Tucson
AZ 85749-9399
United States of America

http://www.tapr.org

Version 1.2 edited by John Langner, WB2OSZ

FOREWORD

This APRS Protocol Reference document represents the coming-of-age of WB4APR's baby. Starting with a simple concept — a way to track the location of moving objects via packet radio — programs using the APRS protocol have grown into perhaps the most popular packet radio application in use today. It's also become one of the most complex; from the simple idea grew, and still grows, a tactical communications system of tremendous capability. Like many ham projects, the APRS protocol was designed as it was being implemented, and many of its intricacies have never been documented.

Until now. This specification defines the APRS on-air protocol with a precision and clarity that make it a model for future efforts. The work done by members of the APRS Working Group, as well as Technical Editor Ian Wade, G3NRW, should be recognized as a tremendous contribution to the packet radio art. With this document available, there is now no excuse for any developer to improperly implement the APRS protocol.

As an APRS Working Group member whose role was mainly that of observer, I was fascinated with the interplay among the APRS authors and the Technical Editor as the specification took form. Putting onto paper details that previously existed only in the minds of the authors exposed ambiguities, unconsidered consequences, and even errors in what the authors thought they knew. The discussion that followed each draft, and the questions Ian posed as he tried to wring out the uncertainties, gave everyone a better understanding of the protocol. I am sure that this process has already contributed to better interoperability among existing APRS applications.

Everyone who has watched the specification develop, from the initial mention in April 1999 until release of this Version 1.0 document in August 2000, knows that the process took much longer than was hoped. At the same time, they saw the draft transformed from a skeleton into a hefty book of over 110 pages. With the specification now in hand, I think we can all say the wait was worth it. Congratulations to the APRS Working Group and, in particular, to G3NRW, for a major contribution to the literature of packet radio.

John Ackermann, N8UR

TAPR Vice President and APRS Working Group Administrative Chair

August 2000

APRS PROTOCOL REFERENCE

TABLE OF CONTENTS

PREAMBLE

APRS Working Group

The APRS Working Group was an unincorporated association whose membersundertake to further the use and enhance the value of the APRS protocols by (a) publishing and maintaining a formal APRS Protocol Specification; (b) publishing validation tests and other tools to enable compliance with the Specification; (c) supporting an APRS Certification program; and (d) generallyworking to improve the capabilities of APRS within the amateur radio community.

Although the Working Group might have received support from TAPR and other organizations, it was an independent body and not affiliated with any organization. The Group had no budget, collects no dues, and owned no assets.

The members of the APRS Working Group were:

John Ackermann, N8UR	Administrative Chair & TAPR Representative
Bob Bruninga, WB4APR (SK)	Technical Chair, founder of APRS
Brent Hildebrand, KH2Z	Author of APRS+SA
Stan Horzepa, WA1LOU	Secretary
Mike Musick, N0QBF	Author of pocketAPRS
Keith Sproul, WU2Z	Co-Author of WinAPRS/MacAPRS/X-APRS
Mark Sproul, KB2ICI	Co-Author of WinAPRS/MacAPRS/X-APRS

Acknowledgements

This document is the result of contributions from many people. It includes much of the material produced by individual members of the Working Group.

In addition, the paper on the Mic-E data format by Alan Crosswell, N2YGK, and Ron Parsons, W5RKN was a useful starting point for explaining the complications of this format.

Document Version Number

Except for the very first public draft release of the APRS Protocol Reference, the document version number is a 3-part number “P.p.D” (for an approved document release) or a 4-part number “P.p.Dd” (for a draft release):

Document Version Number			
APRS Protocol Version		***Document Release***	***Draft***
Major Release	***Minor Release***		
P.	p.	D	d

Thus, for example:

- Document version number “1.2.3” refers to document release 3 covering APRS Protocol Version 1.2.
- Document version number “1.2.3c” is draft “c” of that document.

Release History

The release history for this document is listed in Appendix 7.

Document Conventions

This document uses the following conventions:

- `Courier font` ASCII characters in APRS data.
- ␣ Explicit ASCII space character.
- … (ellipsis) zero or more characters.
- `/$` Symbol from Primary Symbol Table.
- `\$` Symbol from Alternate Symbol Table.
- 0x hexadecimal (e.g. 0x1d).
- All callsigns are assumed to have SSID –0 unless otherwise specified.
- **Yellow marker** (appears as light gray background in hard copy). Marks text of interest — especially useful for highlighting single literal ASCII characters (e.g. **"**) where they appear in APRS data.
- Shaded areas in packet format diagrams are optional fields.

Feedback

Please address your feedback or other comments regarding the unofficial 1.2 DRAFT of this document to John Langner, WB2OSZ. (callsign at arrl.net)

AUTHORS' FOREWORD

This reference document describes what is known as *APRS Protocol Version 1.0*, and is essentially a description of how APRS operates today.

It is intended primarily for the programmer who wishes to develop APRS-compliant applications, but will also be of interest to the ordinary user who wants to know more about what goes on "under the hood".

It is not intended, however, to be a dry-as-dust, pedantic, RFC-style programming specification, to be read and understood only by the Mr Spocks of this world. We have included many items of general information which, although strictly not part of the formal protocol description, provide a useful background on how APRS is actually used on the air, and how it is implemented in APRS software. We hope this will put APRS into perspective, will make the document more readable, and will not offend the purists too much.

It is important to realize how APRS originated, and to understand the design philosophy behind it. In particular, we feel strongly that APRS is, and should remain, a light-weight tactical system — almost anyone should be able to use it in temporary situations (such as emergencies or mobile work or weather watching) with the minimum of training and equipment.

This document is the result of inputs from many people, and collated and massaged by the APRS Working Group. Our sincere thanks go to everyone who has contributed in putting it together and getting it onto the street. If you discover any errors or omissions or misleading statements, please let us know.

Finally, users throughout the world are continually coming up with new ideas and suggestions for extending and improving APRS. We welcome them.

The APRS Working Group

August 2000

Disclaimer

Like any navigation system, APRS is not infallible. No one should rely blindly on APRS for navigation, or in life-and-death situations. Similarly, this specification is not infallible.

The members of the APRS Working Group have done their best to define the APRS protocol, but this protocol description may contain errors, or there may be omissions. It is very likely that not all APRS implementations will fully or correctly implement this specification, either today or in the future.

We urge anyone using or writing a program that implements this specification to exercise caution and good judgement. The APRS Working Group, the editors, and all contributors to this specification, disclaim all liability for injury to persons or property that may result from the use of this specification or software implementing it.

THE STRUCTURE OF THIS SPECIFICATION

This specification describes the overall requirements for developing software that complies with APRS Protocol Version 1.0. The information flow starts with the standard AX.25 UI-frame, and progresses downwards into more and more detail as the use of each field in the frame is explored.

A key feature of the specification is the inclusion of dozens of detailed examples of typical APRS packets and related math computations.

Here is an outline of the chapters:

Introduction to APRS — A brief background to APRS and a summary of its main features.

The APRS Design Philosophy — The fundamentals of APRS, highlighting its use as a real-time tactical communications tool, the timing of APRS transmissions and the use of generic digipeating.

APRS and AX.25 — A brief refresher on the structure of the AX.25 UI-frame, with particular reference to the special ways in which APRS uses the Destination and Source Address fields and the Information field.

APRS Data in the AX.25 Destination and Source Address Fields — Details of generic APRS callsigns and callsigns that specify display symbols and APRS software version numbers. Also a summary of how Mic-E encoded data is stored in the Destination Address field, and how the Source Address SSID can specify a display icon.

APRS Data in the AX.25 Information Field — Details of the principal constituents of APRS data that are stored in the Information field. Contains the APRS Data Type Identifiers table, and a summary of all the different types of data that the Information field can hold.

Time and Position Formats — Information on formats for timestamps, latitude, longitude, position ambiguity, Maidenhead locators, NMEA data and altitude.

APRS Data Extensions — Details of optional data extensions for station course/speed, wind speed/direction, power/height/gain, pre-calculated radio range, DF signal strength and Area Object descriptor.

Position and DF Report Data Formats — Full details of these report formats.

Compressed Position Report Data Formats — Full details of how station position and APRS data extensions are compressed into very short packets.

Mic-E Data Format —Mic-E encoding of station lat/long position, altitude, course, speed, Mic-E position comment, telemetry data and APRS digipeater pathinto the AX.25 Destination Address and Information fields.

Object and Item Reports — Full information on how to set up APRS Objects and Items, and details of the encoding of Area Objects (circles, lines, ellipses etc).

Weather Reports — Full format details for weather reports from stand-alone (positionless) weather stations and for reports containing position information. Also details of storm data format.

Telemetry Data — A description of the MIM/KPC-3+ telemetry data format, with supporting information on how to tailor the interpretation of the raw data to individual circumstances.

Telemetry Metadata — A description of how to interpret the raw numeric telemetry data. This includes properties such as name, units, and scaling.

Messages, Bulletins and Announcements — Full format information.

Station Capabilities, Queries and Responses — Details of the ten different types of query and expected responses.

Status Reports — The format of general status messages, plus the special cases of using a status report to contain meteor scatter beam heading/power and Maidenhead locator.

Network Tunneling — The use of the Source Path Header to allow tunneling of APRS packets through third-party networks that do not understand AX.25 addresses, and the use of the third-party Data Type Identifier.

User-Defined Data Format — APRS allows users to define their own data formats for special purposes. This chapter describes how to do this.

Other Packets — A general statement on how APRS is to handle any other packet types that are not covered by this specification.

APRS Symbols —How to specify APRS symbols and symbol overlays, in position reports and in generic GPS destination callsigns.

APRS Data Formats — An appendix containing all the APRS data formats collected together for easy reference.

The APRS Symbol Tables —A complete listing of all the symbols in the Primary and Alternate Symbol Tables.

ASCII Code Table — The full ASCII code, including decimal and hex codes for each character (the decimal code is needed for compressed lat/long and altitude computations), together with the hex codes for bit-shifted ASCII characters in AX.25 addresses (useful for Mic-E decoding and general on-air packet monitoring).

Glossary — A handy one-stop reference for the many APRS-specific terms used in this specification.

References — Pointers to other documents that are relevant to this specification.

1 INTRODUCTION TO APRS

What is APRS?

APRS is short for *Automatic Packet Reporting System*, which was designed by Bob Bruninga, WB4APR, and introduced by him at the 1992 TAPR/ARRL Digital Communications Conference.

Fundamentally, APRS is a packet communications protocol for disseminating live data to everyone on a network in real time. Its most visual feature is the combination of packet radio with the Global Positioning System (GPS) satellite network, enabling radio amateurs to automatically display the positions of radio stations and other objects on maps on a PC. Other features not directly related to position reporting are supported, such as weather station reporting, direction finding and messaging.

APRS is different from regular packet in several ways:

- It provides maps and other data displays, for vehicle/personnel location and weather reporting in real time.
- It performs all communications using a one-to-many protocol, so that everyone is updated immediately.
- It uses generic digipeating, with well-known callsign aliases, so that prior knowledge of network topology is not required.
- It supports intelligent digipeating, with callsign substitution to reduce network flooding.
- Using AX.25 UI-frames, it supports two-way messaging and distribution of bulletins and announcements, leading to fast dissemination of text information.
- It supports communications with the Kenwood TH-D7/D72/D74 and TM-D700/D710 radios, which have built-in TNC and APRS firmware.

Conventional packet radio is really only useful for passing bulk message traffic from point to point, and has traditionally been difficult to apply to real-time events where information has a very short lifetime. APRS turns packet radio into a real-time tactical communications and display system for emergencies and public service applications.

APRS provides universal connectivity to all stations, but avoids the complexity, time delays and limitations of a connected network. It permits any number of stations to exchange data just like voice users would on a voice net. Any station that has information to contribute simply sends it, and all stations receive it and log it.

APRS recognizes that one of the greatest real-time needs at any special event or emergency is the tracking of key assets. Where is the marathon leader? Where are the emergency vehicles? What’s the weather at various points in the county? Where are the power lines down? Where is the head of the parade? Where is the mobile ATV camera? Where is the storm?

To address these questions, APRS provides a fully featured automatic vehicle location and status reporting system. It can be used over any two-way radio system including amateur radio, marine band, and cellular phone. There is even an international live APRS tracking network on the Internet.

APRS Features

APRS applications are available for most platforms, including Windows, MacOS, Linux, and Android. Most implementations on these platforms support the main features of APRS:

- **Maps** — APRS station positions can be plotted in real-time on maps, with coverage from a few hundred yards to worldwide. Stations reporting a course and speed are dead-reckoned to their present position. Overlay databases of the locations of APRS digipeaters, US National Weather Service sites and even amateur radio stores are available. It is possible to zoom in to any point on the globe.
- **Weather Station Reporting** — APRS supports the automatic display of remote weather station information on the screen.
- **DX Cluster Reporting** — APRS an ideal tool for the DX cluster user. Small numbers of APRS stations connected to DX clusters can relay DX station information to many other stations in the local area, reducing overall packet load on the clusters.
- **Internet Access** — The Internet can be used transparently to cross-link local radio nets anywhere on the globe. It is possible to telnet into Internet APRS servers and see hundreds of stations from all over the world live. Everyone connected can feed their locally heard packets into the APRS server system and everyone everywhere can see them.
- **Messages** — Messages are two-way messages with acknowledgement. All incoming messages alert the user on arrival and are held on the message screen until killed.
- **Bulletins and Announcements** —Bulletins and announcements are addressed to everyone. Bulletins are sent a few times an hour for a few hours, and announcements less frequently but possibly over a few days.
- **Fixed Station Tracking** — In addition to automatically tracking mobile GPS/LORAN-equipped stations, APRS also tracks from manual reports or grid squares.
- **Objects** — Any user can place an APRS Object on his own map, and within seconds that object appears on all other station displays. This is particularly useful for tracking assets or people that are not equipped with trackers. Only one packet operator needs to know where things are (e.g. by monitoring voice traffic), and as he maintains the positions and movements of assets on his screen, all other stations running APRS will display the same information.

2 THE APRS DESIGN PHILOSOPHY

Net Cycle Time

It is important to note that APRS is primarily a *real-time, tactical* communications tool, to help the flow of information for things like special events, emergencies, Skywarn, the Emergency Operations Center and just plain in-the-field use under stress. But like the real world, for 99% of the time it is operating routinely, waiting for the unlikely serious event to happen.

Anything which is done to enhance APRS must not undermine its ability to operate in local areas under stress. Here are the details of that philosophy:

1. APRS uses the concept of a "net cycle time". This is the time within which a user should be able to hear (at least once) all APRS stations within range, to obtain a more or less complete picture of APRS activity. The net cycle time will vary according to local conditions and with the number of digipeaters through which APRS data travels.
2. The objective is to have a net cycle time of 10 minutes for local use. This means that within 10 minutes of arrival on the scene, it is possible to captured the entire tactical picture.
3. All stations, even fixed stations, should beacon their position at the net cycle time rate. In a stress situation, stations are coming and going all the time. The position reports show not only where stations are without asking, but also that they are still active.
4. It is not reasonable to assume that all APRS users responding to a stress event understand the ramifications of APRS and the statistics of the channel — user settings cannot be relied on to avoid killing a stressed net. Thus, to try to anticipate when the channel is under stress, APRS automatically adjusts its net cycle time according to the number of digipeaters in the UNPROTO path:
    - Direct operation (no digipeaters): 10 minutes (probably an event).
    - Via one digipeater hop: 10 minutes (probably an event).
    - Via two digipeater hops: 20 minutes.
    - Via three or more digipeater hops: 30 minutes.
5. Since almost all home stations set their paths to three or more digipeaters, the default net cycle time for routine daily operation is 30 minutes. This should be a universal standard that everyone can bank on — if you routinely turn on your radio and APRS and do nothing else, then in 30 minutes you should have virtually the total picture of all APRS stations within range.
6. Since knowing where the digipeaters are located is fundamental to APRS

connectivity, digipeaters should use multiple beacon commands to transmit position reports at different rates over different paths; i.e. every 10 minutes for sending position reports locally, and every 30 minutes for sending them via three digipeaters (plus others rates and distances as needed).

7. If the net cycle time is too long, users will be tempted to send queries for APRS stations. This will increase the traffic on the channel unnecessarily. Thus the recommended extremes for net cycle time are 10 and 30 minutes — this gives network designers the fundamental assumptions for channel loading necessary for good engineering design.

Packet Timing

Since APRS packets are error-free, but are not guaranteed delivery, APRS transmits information redundantly. To assure rapid delivery of new or changing data, and to preserve channel capacity by reducing interference from old data, APRS should transmit new information more frequently than old information.

There are several algorithms in use to achieve this:

- **Decay Algorithm** — Transmit a new packet once and n seconds later. Double the value of n for each new transmission. When n reaches the net cycle time, continue at that rate. Other factors besides “doubling” may be appropriate, such as for new message lines.
- **Fixed Rate —** Transmit a new packet once and n seconds later. Transmit it x times and stop.
- **Message-on-Heard** — Transmit a *new* packet according to either algorithm above. If the packet is still valid, and has not been acknowledged, and the net cycle time has been reached, then the recipient is probably not available. However, if a packet is then subsequently heard from the recipient, try once again to transmit the packet.
- **Time-Out** — This term is used to describe a time period beyond which it is reasonable to assume that a station no longer exists or is off the air if no packets have been heard from it. A period of 80 minutes is suggested as the nominal default timeout to account for stations via satellites. This time-out is not used in any transmittingalgorithms, but is useful in some programs to decide when to cease displaying stations as “active”. Note that on HF, signals come and go, so decisions about activity may need to be more flexible.

Generic Digipeating

The power of APRS in the field derives from the use of *generic* digipeating, in that packets are propagated without a priori knowledge of the network.

Do not use RELAY, TRACE, or plain WIDE, as seen in outdated literature, because they have been obsolete since around 2004.

1. Digipeaters look at the **first unused** address in the digipeater via path of a packet. They always respond to their own name.

2. **WIDEn-N** — Rather than using specific digipeater names, the normal procedure is to use generic aliases. A typical digipeater via path is "WIDE1-1,WIDE2-1". Typically, low range "fill-in" digipeaters will respond to only "WIDE1-*". Those with very wide area coverage will respond to "WIDE2-*".

   In this context, the SSID is actually a remaining use count. For example, "WIDE2-2" is equivalent to "WIDE2-1,WIDE2-1".

   If the digipeater configuration contains the prefix pattern, the processing depends on the value of the SSID, represented as N here.

   N = 0 Do not digipeat.
   N =1 Digipeat and replace generic form with digipeater callsign. Digipeaters should always identify themselves, and mark their address used, so the actual path taken is known.
   N ≥ 2 Decrement N and leave address in path. Leave it marked as unused. Insert digipeater callsign before it and mark digipeater callsign as used.

   The list of stations, in the used part of the digi via path (before the "*" character in the human readable monitoring format), reveals the path the packet has taken.

   All digipeaters retain a copy of each packet retransmitted for a short amount of time, typically about 28 seconds. An identical packet, ignoring the via path, will not be transmitted again during this time. For efficiency, many implementations will keep a checksum, rather than the entire packet.

   A digipeater must not change anything other than the via path. If it changed anything else, it would prevent duplicate detection.

3. **SSn-N** — Digipeaters can also be configured to respond to other prefixes representing geographical regions or special events. In the U.S., regions might correspond to a state, county or ARRL section. For example, MDn-N might be used for Maryland. In Europe, the regions are based on a combination of two iso-standards. ISO 3166-1 country

code und ISO 3166-2 subdivision code.

4. **GATE** — This generic callsign is used by HF-to-VHF Gateway digipeaters. Any packet heard on HF via `GATE` will be digipeated locally on VHF. This permits local networks to keep an eye on the national and international picture.

Recommended Digipeater Symbol Overlays.

A	An Alt. Freq. input digi (Typically on 144.99 MHz)
~~R~~	~~RELAY only (Obsolete in 2005.)~~
~~W~~	~~WIDE and RELAY (Obsolete in 2005.)~~
~~T~~	~~PacComm RELAY,WIDE and TRACE (Obsolete in 2005.)~~
~~N~~	~~WIDEn-N and NOID is set (Obsolete in 2005 - all digipeaters should identify themselves.)~~
I	Digipeater is also an IGate.
S	**The New n-N Paradigm digi with State SSn-N support (and ID is on)**
P	PacComm digi supporting New n-N Paradigm
~~U~~	~~UI-DIGI firmware (should now be upgraded to S)~~
~~D~~	~~DIGI_NED (should now be upgraded to S)~~

Communicating Map Views Unambiguously

APRS is a tactical geographical system. To maximize its operational effectiveness and minimize confusion between operators of different systems, users need to have an unambiguous way to communicate to others the "location" and "size" (or area of coverage) of any map view.

The APRS convention is by reference to a *center* and *range* which specify the geographical center and approximate radius of a circle that will fit in the map view independent of aspect ratio. The radius of the circle (in nautical miles, statute miles or km) is known as the "range scale". This convention gives all users a simple common basis for describing any specific map view to others over any communications medium or program.

If you are on a 64 mile "range Scale" that means that everything within 64 miles of the center is visible. If you zoom to the 8 mile range scale, then anyone else that is on an "8 mile range scale" (no matter what version of APRS he is running) he will see the same "area". It is only one line of code to convert from an arbitrary "zoom factor" or "map size" to a meaningful "range scale". And APRS would be so much the better for it...

APRS Voice Alert

APRS Voice Alert allows you to hear voice transmissions on the APRS frequency by using an agreed upon CTCSS tone when sending voice.

3 APRS AND AX.25

Protocols

At the data link layer, APRS uses AX.25 frames, as defined in *AX.25 Link Access Protocol for Amateur Packet Radio* (see Appendix 6 for details), utilizing Unnumbered Information (UI) frames exclusively. This means that APRS runs in *connectionless* mode, whereby AX.25 frames are transmitted withoutexpecting any response, and reception at the other end is not guaranteed.

At a higher level, APRS supports a messaging protocol that allows users to send short messages (one line of text) to nominated stations, and expects to receive acknowledgements from those stations.

The AX.25 and APRS Specifications do not mandate a particular type of modem for the physical layer.

The AX.25 Frame

All APRS transmissions use AX.25 UI-frames, with 9 fields of data:

AX.25 UI-FRAME FORMAT

	Flag	*Destination Address*	*Source Address*	*Digipeater Addresses (0-8)*	*Control Field (UI)*	*Protocol ID*	*INFORMATION FIELD*	*FCS*	*Flag*
Bytes:	1	7	7	0–56	1	1	1–256	2	1

- **Flag** — The flag field at each end of the frame is the bit sequence 01111110 that separates each frame. A single flag can be both the end of one frame and the beginning of the next. Whenever it is necessary for a TNC to keep its transmitter on while not actually sending frames, the time should be filled with contiguous flags.
- **Destination Address** — Unlike Packet Radio, this is not used for the destination for the packet. This field is used in multiple ways for the different APRS data types. APRS data is encoded to ensure that the field conforms to the standard AX.25 callsign format (i.e. up to 6 upper case alphanumericcharacters plus SSID). Most often it will be of the form AP*xxxx* to identify the type of system that generated the packet. If the SSID is non-zero, it specifies a generic APRS digipeater path.

  In accordance with the AX.25 specification, for UI frames, the C bit of the Destination SSID octet should be 1. Receiving applications must ignore it because this has not been done consistently.
- **Source Address** — This field contains the callsign and SSID of the transmitting station. In some cases, if the SSID is non-zero, the SSID may specify an APRS display Symbol Code.

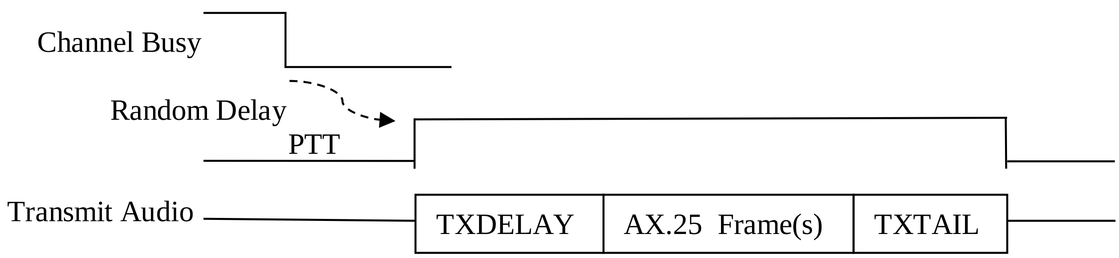
  In accordance with the AX.25 specification, for UI frames, the C bit of the Source SSID octet should be 0. Receiving applications must ignore it because this has not been done consistently.
- **Digipeater Addresses** — From zero to 8 digipeater callsigns may be

included in this field. These can be actual station names but are are usually aliases so you don't need to know about the network topology. **Note**: These digipeater addresses may be overridden by a generic APRS digipeater path (specified in the Destination Address SSID).

- **Control Field** — This field is set to 0x03 (UI-frame).
- **Protocol ID** — This field is set to 0xf0 (no layer 3 protocol).
- **Information Field** — This field contains more APRS data. The first character of this field is the APRS Data Type Identifier (DTI) that specifies thenature of the data that follows.
- **Frame Check Sequence** — The FCS is a sequence of 16 bits used for checking the integrity of a received frame.

Channel Access

APRS is designed for many stations to share a single frequency and transmit at unpredictable times. Obviously you should wait for the channel to be clear before transmitting. If everyone transmitted as soon as the channel becomes clear, there would be many collisions. These collisions can be minimized by waiting a minimum amount of time and then waiting a random amount of time. The technical term for this is p-persistent Carrier Sense Multiple Access (CSMA).



Transmit timing is determined by 4 parameters with these traditional names and typical default values.

SLOTTIME 10	x 10 mSec per unit = 100 mSec.
PERSIST 63	probability for transmitting after each slottime.
TXDELAY 30	x 10 mSec per unit = 300 mSec.
TXTAIL 10	x 10 mSec per unit = 100 mSec.

Some implementations use milliseconds, rather than units of 10 milliseconds, so read the documentation if not using the defaults provided by the application.

- For digipeated frames the transmission should begin immediately after the channel is clear. Having multiple digipeaters transmit at the same time is intentional to minimize the amount of time taken for the same packet. The strongest signal will win due to the FM capture effect.

- For other frames, SLOTTIME and PERSIST are used to generate a random delay to minimize the chances of two different stations starting to transmit at the same time. The process is:

(a) Wait for channel to be clear (i.e. no signal detected by modem)

(b) Wait for SLOTTIME.

(c) If the channel is busy (i.e. signal detected by the modem), go back to step (a).

(d) Generate a random number in the range of 0 to 255. If it is less than or equal to PERSIST (Typically 25% probability) , start to transmit. Otherwise go back to step (b).

For the typical default values, we have delays with the following probabilities:

Delay, mSec	**Probability**	
0	0	= 0%
100	.25	= 25%
200	.75 * .25	= 19%
300	.75 * .75 * .25	= 14%
400	.75 * .75 * .75 * .25	= 11%
500	.75 * .75 * .75 * .75 * .25	= 8%
etc.	...	

The Push to Talk (PTT) control line is asserted.

The data can't be sent immediately because the transmitter takes a little while to stabilize and reach full power after being activated.

The HDLC "flag" pattern (01111110) is sent for a time period of TXDELAY. For historical reasons, going back to the 1980s, the configuration numbers are often expressed in 10 millisecond (mS) units so 30 actually means 300 milliseconds.

When sending with a hardware modem, you know exactly when the audio for the frame is finished so the transmitter can be turned off fairly soon after that. TXTAIL can be very small.

When sending audio out through a "soundcard" there is latency between sending an audio waveform to the audio output device and when the sound comes out. We can't be sure precisely when the queued up sound has been completed so we need to keep the PTT on a little longer. The HDLC "frame" pattern is also sent during this time to keep the channel busy.

4 APRS DATA IN THE AX.25 DESTINATION AND SOURCE ADDRESS FIELDS

The AX.25 Destination Address Field

The name "destination" causes a lot of confusion to many people. The term "destination" comes from AX.25 where it is the destination for the packet. APRS is sent to everyone, rather than being directed to a particular station.

APRS uses this field for other multiple purposes:

- An APRS software version (or device) number.
- Mic-E encoded data.
- A generic APRS address.
- A generic APRS address with a symbol.
- A Maidenhead Grid Locator (obsolete).
- An Alternate Net (ALTNET) address.

In all of these cases, the Destination Address SSID may specify a generic APRS digipeater path.

APRS Software Version Number

The AX.25 Destination Address field will most often contain the version number of the APRS software or device that is running at the station. Knowledge of the version number can be useful when troubleshooting.

Here are some early examples. The variable part (xxx) often contains a software version number.

APCxxx	APRS/CE, Windows CE
APDxxx	Linux aprsd server
APExxx	PIC-Encoder
APIxxx	Icom radios
APKxxx	Kenwood radios
APMxxx	MacAPRS
APPxxx	pocketAPRS
APRxxx	APRSdos
APSxxx	APRS+SA
APWxxx	WinAPRS
APXxxx	X-APRS
APYxxx	Yaesu radios
APZxxx	Experimental

For example, a station using version 3.2.6 of MacAPRS could use the version APM326.

Developers of new products should obtain a unique software/device identifier to identify the product which is creating the packet. The traditional list at http://aprs.org/aprs11/tocalls.txt is no longer being maintained. The current location is https://github.com/aprsorg/aprs-deviceid

The APZxxx Experimental destination is designated for *temporary* use only while a product is being developed, before a special APRS Software Version addressis assigned to it.

Mic-E Encoded Data

Another alternative use of the AX.25 Destination Address field is to contain Mic-E encoded data. This data includes:

- The latitude of the station.
- A West/East Indicator and a Longitude Offset Indicator (used in longitude computations).
- A Position Comment.
- The APRS digipeater path.

This data is used with associated data in the AX.25 Information field to provide a complete Position Report and other information about the station (see Chapter 10: Mic-E Data Format).

The rest of these are already declared obsolete or probably should be.

Generic APRS Address with Symbol

APRS can also use this field for the display symbol. This was for the very early trackers, with extremely limited resources. They would transmit raw GPS NMEA sentences rather than formatting the data into position reports.

These special addresses are `GPSxyz`, `GPSCnn`, `GPSEnn`, `SPCxyz` and `SYMxyz`, and areintended for use where it is not possible to include the symbol in the AX.25 Information field.

The `GPS` addresses above are for general use.

The `SPC` addresses are intended for special events.

The `SYM` addresses are reserved for future use.

The characters `xy` and `nn` refer to entries in the APRS Symbol Tables. The character `z` specifies a symbol overlay. See Chapter 20: APRS Symbols and Appendix 2 for more information.

Generic APRS Digipeater Path

(Obsolete)

The SSID in the Destination Address field of all packets is can be used to specifythe APRS digipeater path.

If the Destination Address SSID is –0, the packet follows the standard AX.25 digipeater ("VIA") path contained in the Digipeater Addresses field of the AX.25 frame.

If the Destination Address SSID is non-zero, the packet follows one of 15 generic APRS digipeater paths.

The SSID field in the Destination Address (i.e. in the 7th address byte) is encoded as follows:

APRS Digipeater Paths in Destination Address SSID

SSID	Path	SSID	Path
-0	Use VIA path	-8	North path
-1	WIDE-1	-9	South path
-2	WIDE-2	-10	East path
-3	WIDE-3	-11	West path
-4	WIDE-4	-12	North path + WIDE
-5	WIDE-5	-13	South path + WIDE
-6	WIDE-6	-14	East path + WIDE
-7	WIDE-7	-15	West path + WIDE

This was hack for a very early APRS tracker with extremely limited compute resources, and 4 DIP switches to set the destination SSID. This was many years before the WIDEn-N paradigm was created.

We are not aware of any contemporary implementations. The community consensus is that this should be considered obsolete.

Maidenhead Grid Locator inDestination Address (Obsolete)

The AX.25 Destination Address field may contain a 6-character Maidenhead Grid Locator. For example: **IO91SX**. This format is typically used by meteor scatter and satellite operators who need to keep packets as short as possible.

This format is now obsolete.

Can we drop this to clean up some of the clutter?

Generic APRS Destination Addresses

APRS uses the following generic beacon-style destination addresses:

DGPS*	**ALL***	**AP***	**BEACON**	**CQ***	**GPS***	**DF***
QST*	**DRILL***	**DX***	**ID***	**JAVA***	**MAIL***	**MICE***
	QTH*	**RTCM***	**SKY***	**SPACE***	**SPC***	**SYM***
TEL*	**TEST***	**TLM***	**WX***			

The asterisk is a wildcard, allowing the address to be extended (up to a total of 6 alphanumeric characters). Thus, for example, `WX1`, `WX12` and `WX12CD` are all valid APRS destination addresses.

All of these addresses have an SSID of –0. Non-zero SSIDs are reserved for generic APRS digipeating.

These addresses are copied by everyone. All APRS software must accept packets with these destination addresses.

The addresses **DGPS** and **RTCM** are used by differential GPS correction stations. Most software will not make use of packets using this address, other than to pass them on to an attached GPS unit.

The address **SKY** is used for Skywarn stations.

Packets addressed to SPCL are intended for special events. APRS software can display such packets to the exclusion of all others, to minimize clutter on from other stations not involved in the special event. The addresses TEL and TLM is used for telemetry stations.

The use of "APRS," in the destination field, is obsolete. Use an APxxxx software version number as explained earlier.

Alternate Nets

Any other destination address not included in the specific generic list or the other categories mentioned above may be used in Alternate Nets (ALTNETs) by groups of individuals for special purposes. Thus they can use the APRS infrastructure for a variety of experiments without cluttering up the maps and lists of other APRS stations. Only stations using the same ALTNET address should see their data. HUH? Can we elaborate or declutter this?

The AX.25 Source Address SSID to specify Symbols

The AX.25 Source Address field contains the callsign and SSID of the originating station. If the SSID is –0, APRS does not treat it in any special way.

If, however, the Source Address SSID is non-zero, APRS interprets it as a display icon. This is intended for use only with stand-alone trackers where there is no other method of specifying a display symbol or a destination address (e.g. MIM trackers or NMEA trackers).

For more information, see Chapter 20: APRS Symbols.

5 APRS DATA IN THE AX.25 INFORMATION FIELD

Generic Data Format

In general, the AX.25 Information field can contain some or all of the following information:

- APRS Data Type Identifier
- APRS Data
- APRS Data Extension
- Comment

Generic APRS Information Field

	Data Type ID	APRS Data	APRS Data Extension	Comment
Bytes:	1	n	7	n

APRS Data Type Identifier

Every APRS packet contains an APRS Data Type Identifier (DTI). This determines the format of the remainder of the data in the Information field, as follows:

APRS Data Type Identifiers

Ident	Data Type
0x1c	Current Mic-E Data (Rev 0 beta)
0x1d	Old Mic-E Data (Rev 0 beta)
!	Position without timestamp (no APRS messaging), or Ultimeter 2000 WX Station
"	*[Unused]*
#	Peet Bros U-II Weather Station
$	Raw GPS data or Ultimeter 2000
%	Agrelo DFJr / MicroFinder
&	*[Reserved — Map Feature]*
'	Old Mic-E Data (but *Current* data for TM-D700)
(	*[Unused]*
)	Item
*	Peet Bros U-II Weather Station
+	*[Reserved — Shelter data with time]*
,	Invalid data or test data
-	*[Unused]*
.	*[Reserved — Space weather]*
/	Position with timestamp (no APRS messaging)
0–9	*[Do not use]*
:	Message
;	Object

Ident	Data Type
<	Station Capabilities
=	Position without timestamp (with APRS messaging)
>	Status
?	Query
@	Position with timestamp (with APRS messaging)
A–S	*[Do not use]*
T	Telemetry data
U–Z	*[Do not use]*
[	Maidenhead grid locator beacon (obsolete)
\	*[Unused]*
]	*[Unused]*
^	*[Unused]*
_	Weather Report (without position)
`	Current Mic-E Data (*not used* in TM-D700)
a–z	*[Do not use]*
{	User-Defined APRS packet format
|	*[Do not use]*
}	Third-party traffic
~	*[Do not use]*

Note: The Kenwood TM-D700 radio uses the ' DTI for *current* Mic-E data. The radio does not use the ` DTI.

APRS Data and Data Extension

There are 10 main types of APRS Data:

- Position
- Direction Finding
- Objects and Items
- Weather
- Telemetry
- Messages, Bulletins and Announcements
- Queries
- Responses
- Status
- Other

Some of this data may also have an APRS Data Extension that provides additional information.

The APRS Data and optional Data Extension follow the Data Type Identifier.

The table on the next page shows a complete list of all the different possible types of APRS Data and APRS Data Extension.

	Possible APRS Data	*Possible APRS Data Extension*
Position	Time (DHM or HMS) Lat/long coordinates Compressed lat/long/course/speed/radio range/altitude Symbol Table ID and Symbol Code Mic-E longitude, speed and course, telemetry or status Raw GPS NMEA sentence Raw weather station data	Course and Speed Power, Effective Antenna Height/Gain/Directivity Pre-Calculated Radio Range Omni DF Signal Strength Storm Data (in Comment field)
Direction Finding	Time (DHM or HMS) Lat/long coordinates Compressed lat/long/course/speed/radio range/altitude Symbol Table ID and Symbol Code	Course and Speed Power, Effective Antenna Height/Gain/Directivity Pre-Calculated Radio Range Omni DF Signal Strength Bearing and Number/Range/Quality (in Comment field)
Objects and Items	Object name Item name Time (DHM or HMS) Lat/long coordinates Compressed lat/long/course/speed/radio range/altitude Symbol Table ID and Symbol Code Raw weather station data	Course and Speed Power, Effective Antenna Height/Gain/Directivity Pre-Calculated Radio Range Omni DF Signal Strength Area Object Storm Data (in Comment field)
Weather	Time (MDHM) Lat/long coordinates Compressed lat/long/course/speed/radio range/altitude Symbol Table ID and Symbol Code Raw weather station data	Wind Direction and Speed Storm Data (in Comment field)
Telemetry	Telemetry (non Mic-E)	
Messages, Bulletins and Announcements	Addressee Message Text Message Identifier Message Acknowledgement Bulletin ID, Announcement ID Group Bulletin ID	
Queries	Query Type Query Target Footprint Addressee (Directed Query)	
Responses	Position Object/Item Weather Status Message Digipeater Trace Stations Heard Heard Statistics Station Capabilities	Course and Speed Power, Effective Antenna Height/Gain/Directivity Pre-Calculated Radio Range Omni DF Signal Strength Area Object Wind Direction and Speed
Status	Time (DHM zulu) Status text Meteor Scatter Beam Heading/Power Maidenhead Locator (Grid Square) Altitude (Mic-E) E-mail message	
Other	Third-Party forwarding Invalid Data/Test Data	

Comment Field

In general, any APRS packet can contain a plain text comment (such as a beacon message) in the Information field, immediately following the APRS Data or APRS Data Extension.

There is no separator between the APRS data and the comment unless otherwise stated.

The freeform text part of the comment may contain any printable ASCII or UTF-8 character.

Do not put any carriage return (0x0d) or line feed (0x0a) at the end. IGate stations will remove them, resulting in slightly different contents.

The very small and seemingly arbitrary maximum length of the comment field depends on the report — detailsare included in the description of each report.

In special cases, the Comment field can also contain further APRS data:

- **Altitude** in comment text (see Chapter 6: Time and Position Formats), or in Mic-E status text (see Chapter 10: Mic-E Data Format).
- **Maidenhead Locator** (grid square), in a Mic-E status text field (see Chapter 10: Mic-E Data Format) or in a Status Report (see Chapter 16: Status Reports).
- **Bearing and Number/Range/Quality** parameters (`/BRG/NRQ`), in DF reports (see Chapter 7: APRS Data Extensions).
- **Area Object Line Widths** (see Chapter 11: Object and Item Reports).
- **Signpost Objects** (see Chapter 11: Object and Item Reports).
- **Weather and Storm Data** (see Chapter 12: Weather Reports).
- **Beam Heading and Power**, in Status Reports (see Chapter 16: Status Reports).

Base-91 Notation

Two APRS data formats use base-91 notation: lat/long coordinates in compressed format (see Chapter 9) and the altitude in Mic-E format (see Chapter 10).

Base-91 data is compressed into a short string of characters. All the characters are printable ASCII, with character codes in the range 33–124 decimal (i.e. **!** through **|**).

To compute the base-91 ASCII character string for a given data value, the value is divided by progressively reducing powers of 91 until the remainder is less than 91. At each step, 33 is added to the modulus of the division process to obtain the corresponding ASCII character code.

For example, for a data value of 12345678:

$12345678 / 91^3$ = modulus **16**, remainder 288542
$288542 / 91^2$ = modulus **34**, remainder 6988
$6988 / 91^1$ = modulus **76**, remainder **72**

The four ASCII character codes are thus 49 (i.e. **16**+33), 67 (i.e. **34**+33), 109 (i.e. **76**+33) and 105 (i.e. **72**+33), corresponding to the ASCII string **1Cmi**.

APRS Precision and Datum Option

This enhancement to the APRS spec is to allow for greater precision to be transmitted between users. But since precision without the accuracy of the datum is meaningless, this format also includes the identification of the datum used to gain this precision.

DATUM: The default datum for GPS and for APRS is WGS84. This is specified in the APRS spec that all on-air positions are assumed to be WGS84 unless otherwise indicated. This "option" then documents two such methods for "otherwise indicating":

BY COUNTRY: or Continental local agreement. Two such agreements have been declared: WGS84 in North America and OSGB36 in the UK. If others cam be agreed, they will be listed.

BY DATUM BYTE: the DATUM byte in this "PRECISION and DATUM Option".

FORMAT: The format of this APRS Precision-and-Datum option is the presence of a 5 byte field !DAO! appearing anywhere in the position comment field. This option is backwards compatible because the basic DDMM.HH/DDDMM.HH format is retained and will still be decoded by all existing applications. The !DAO! simply provides the additional precision down to a foot or so and also identifies the datum:

!DAO! - is fixed length anywhere in the position commment
D - is the datum identifier (base-91)
A - is the added LAtitude precision (base-91)
O - is the added LOngitude precision (base-91)

RECOMMENDATION: Recommend placing this option on the END of all other position comment text. This way it does not displace any human readable comment text that is otherwise desired to display on older systems. This added option may extend beyond the existing "viewable" limit of 57 bytes currently in the spec since these added bytes are not used by legacy systems anyway.

PRECISION: There are three degrees of precision offered by this option.

1) HUMAN READABLE, Thousandths of a minute. This is good to the nearest 6 feet or so and being human readable gives even users of the D7 and D700 or any other existing application the ability

to human read position to 3 decimal digits of Minutes. This format is identified if the Datum byte is in uppercase.

2) BASE-91. This adds additional precision to the nearest 91/ten thousandth of a minute, or about 4 decimal fractional digits of a minute or about one foot. This format is identified if the datum byte is in lowercase.

3) NULL. If the A and O bytes are SPACE characters, then they are only there to fulfill the !DAO! format and imply NO ADDED precision. This is used when one wants to send DATUM info but without claming added precision. This use of space characters to imply lacking digits of precision is consistent with the existing APRS ambiguity system.

DATUMS: There are several categories of datums that can be used.

PRE-DEFINED: The Letters A-Z and a-z indicate one of 26 common datums. The case of these 26 letters indicates which precision is used. Capital letters indicate human readable decimal digits for A and O, and lower case indicates base 91 encoding for A and O. There are 26 pre-defined DATUMS. A table will be prepared of the 26 common datums. Here are some examples:
W = WGS84
N = NAD27
O = OSGB36

LOCAL CUSTOM: For special and closed events, the DATUM can be one of ten locally defined options. These are indicated by the digits 0 through 9. One creative use of this is for closed events using fixed maps or venues. These digits could define the maps by number!

BASE-91 ENCODING: Base-91 is used frequently in APRS to improve the resolution of a byte without using Binary and while still using only the printable ASCII character subset. Using only printable ASCII has many benefits and advantages that have been discussed ad nausium elsewhere. Base 91 simply means the value of an ASCII byte after subtracting decimal 33 Thus the character "!" minus 33 is "0" on up to the character "}" minus 33 which has the value "90"

OTHER: There remains 28 other possiblities of DATUM information in the D digit that can be used if needed.

Examples:

!W23! means it is WGS84 and the upper case indicates it is the human readable 3rd digit format. "2" is the third decimal digit of latitude minutes and is human readable. "3" is the added digit of longitude minutes and is also human readable.

!wAb! means it is WGS84 but the lower case indicates this is the added precision to the nearest foot. "A" is the base 91 code for two more digits (65 minus 33 yields "32") and "b" is two more digits of longitude (98 -33 or "...65").

!w:\! would also be WGS84 but with ":" and "\" decoding to two additional digits of "27" to latitude and "59" to longitude to the nearest foot or so.

BASE-91 CONVERSION: In the first !W23! example above, the actual digits are simply added to the existing LAT/LONG for example to add precision to DDMM.mmN/DDDMM.mmW to be come equivalent to DDMM.mm2N and DDDMM.mm3W.

But in the next two examples !wAb! and !w:\! the added two digits cannot simply be added to the ASCII position string, since they can only go from 00 to 90. THus they need to be scaled so that they go from 00 to 99. Do this by multiplying the two digits by 1.10. So for the !wAb! example with added latitude digits of 32, you multiply that by 1.10 to arrive at an actual added digits of 35.2. So the high precision latitude becomes DDMM.mm352N.

APRS Data Units

For historical reasons there is some lack of consistency between units of data in APRS packets — some speeds are in knots, others in miles per hour; some altitudes are in feet, others in meters, and so on. It is emphasized that this specification describes the units of data as they are transmitted on-air. It is the responsibility of APRS applications to convert the on-air units to more suitable units if required.

The default GPS earth datum is World Geodetic System (WGS) 1984 but Continental options such as OSG for the UK are OK.

6 TIME AND POSITION FORMATS

Time Formats

APRS timestamps are expressed in three different ways:

- Day/Hours/Minutes format
- Hours/Minutes/Seconds format
- Month/Day/Hours/Minutes format

In all three formats, the 24-hour clock is used.

Day/Hours/Minutes (DHM) format is a fixed 7-character field, consisting of a 6-digit *day/time* group followed by a single *time indicator* character (**z** or **/**). The day/time group consists of a two-digit day-of-the-month (01–31) and a four-digit time in hours and minutes.

Times can be expressed in *zulu* (UTC/GMT) or *local* time. For example:

`092345`**z** is 2345 hours *zulu* time on the 9th day of the month.
`092345`**/** is 2345 hours *local* time on the 9th day of the month.

It is recommended that future APRS implementations only transmit zulu format on the air.

Note: The time in Status Reports may *only* be in zulu format.

Hours/Minutes/Seconds (HMS) format is a fixed 7-character field, consisting of a 6-digit time in hours, minutes and seconds, followed by the **h** time-indicator character. For example:

`234517`**h** is 23 hours 45 minutes and 17 seconds *zulu*.

Note: This format may *not* be used in Status Reports.

Month/Day/Hours/Minutes (MDHM) format is a fixed 8-character field, consisting of the month (01–12) and day-of-the-month (01–31), followed by the time in hours and minutes zulu. For example:

`10092345` is 23 hours 45 minutes zulu on October 9th.

This format is only used in reports from stand-alone “positionless” weather stations (i.e. reports that do not contain station position information).

Use of Timestamps

When a station transmits a report *without* a timestamp, an APRS receiving station can make an internal record of the time it was received, if required. This record is the *receiving* station's notion of the time the report was created.

On the other hand, when a station transmits a report *with* a timestamp, that timestamp represents the *transmitting* station's notion of the time the report was created.

In other words, reports sent *without* a timestamp can be regarded as real-time, "current" reports (and the *receiving* station has to record the time they were received), whereas reports sent *with* a timestamp may or may not be real-time, and may possibly be (very) "old".

Four APRS Data Type Identifiers specify whether or not a report contains a timestamp, depending on whether the station has APRS messaging capability or not:

		No APRS Messaging	***With APRS Messaging***
(Current/real-time)	***Report without timestamp***	**!**	**=**
(Old/non-real-time)	***Report with timestamp***	**/**	**@**

Stations without APRS messaging capability are typically stand-alone trackers or digipeaters. Stations reporting without a timestamp are generally (but not necessarily) fixed stations.

Latitude Format

Latitude is expressed as a fixed 8-character field, in degrees and decimal minutes (to two decimal places), followed by the upper case letter **N** for north or **S** forsouth.

Latitude degrees are in the range 00 to 90. Latitude minutes are expressed as whole minutes and hundredths of a minute, separated by a decimal point.

For example:

`4903.50`**N** is 49 degrees 3 minutes 30 seconds north.

In generic format examples, the latitude is shown as the 8-character string `ddmm.hhN` (i.e. degrees, minutes and hundredths of a minute north).

Longitude Format

Longitude is expressed as a fixed 9-character field, in degrees and decimal minutes (to two decimal places), followed by the upper case letter **E** for east or **W** forwest.

Longitude degrees are in the range 000 to 180. Longitude minutes are expressed as whole minutes and hundredths of a minute, separated by a decimal point.

For example:

`07201.75W` is 72 degrees 1 minute 45 seconds west.

In generic format examples, the longitude is shown as the 9-character string `dddmm.hhW` (i.e. degrees, minutes and hundredths of a minute west).

PositionCoordinates

Position coordinates are a combination of latitude and longitude, separated by a display Symbol Table Identifier, and followed by a Symbol Code. For example:

`4903.50N/07201.75W-`

The `/` character between latitude and longitude is the Symbol Table Identifier (in this case indicating use of the Primary Symbol Table), and the `-` character at the end is the Symbol Code from that table (in this case, indicating a "house" icon).

A description of display symbols is included in Chapter 20: APRS Symbols. The full Symbol Table listing is in Appendix 2.

Position Ambiguity

In some instances — for example, where the exact position is not known — the sending station may wish to reduce the number of digits of precision in the latitude and longitude. In this case, the `mm` and `hh` digits in the latitude may be progressively replaced by a `␣` (space) character as the amount of imprecision increases. For example:

`4903.5␣N` represents latitude to nearest 1/10th of a minute.

`4903.␣␣N` represents latitude to nearest minute.

`490␣.␣␣N` represents latitude to nearest 10 minutes.

`49␣␣.␣␣N` represents latitude to nearest degree.

The level of ambiguity specified in the latitude will automatically apply to the longitude as well — it is permissible but not necessary to include any `␣` characters in thelongitude.

For example, the coordinates:

`4903.␣␣N/07201.75W-`

represent the position to the nearest minute. That is, the hundredths of minutes of latitude and longitude may take any value in the range `00–99`.

Thus the station may be located anywhere inside a bounding box having the following corner coordinates:

North West corner: 49 deg 3.99 mins N, 72 deg 1.99 mins W
North East corner: 49 deg 3.99 mins N, 72 deg 1.00 mins W
South East corner: 49 deg 3.00 mins N, 72 deg 1.00 mins W
South West corner: 49 deg 3.00 mins N, 72 deg 1.99 mins W

Ambiguity Plots: Since not all positions in APRS are known to the same precision, a significant attribute of all APRS symbols is the provision to show the four ranges of ambiguity of 0.1, 1, 10 and 60 nautical miles. Notice how the object named GUESSED above is not shown with a symbol, but with a circle. This is because this station is transmitting or entered with a 1/10th mile ambiguity with his position. The station's symbol does show up at large area scales, but once you zoom the map below the scale at which the size of the ambiguity is larger than the "size" of the symbol, then the symbol disappears so that there can be no misinterpretation by the person viewing that there is precision where there is not. These ranges of ambiguity are conveyed just like a written position is conveyed. If a position is only known to the nearest degree (60 nm), then only the degrees are sent. If the position is only known to the nearest minute, then only the degrees and integer minutes are transmitted. and So on down to Tenths of a minute. If all digits are known, then the position conveys the full precision inherent in APRS which is to the nearest 100th of a nautical mile or about 60 feet.

Ambiguity in the Spec: There has been lots of confusion over Position Ambiguity caused by the poor wording in the spec that can be incorrectly interpreted as implying a truncation of digits and a lat/long box of ambiguity. . It is not a *truncation* and it is not a box. The position field in APRS is a string field, not a numeric field. . One should place in that field by *inclusion* only the digits the sender wants the receiver to use. . Further the spec implies that this results in a box of ambiguity. This is wrong. because it would imply vastly different sizes of imprecision at the equator and at the poles. . It is clear that the intent of position ambiguity was a range in nautical miles, since Ambiguity was defined in the LATITUDE field only, where the digits do correspond to Nautical Miles. . And they do give the same circular area everywhere on earth.

Plotting Position Ambiguity: The recommended plot of position ambiguity is shown above for GUESSED. . That is, a circle with a radius of the ambiguity centered on the given position. (Not centered in a box of LAT/LONG) . Further, the symbol may be displayed as long as the size of the symbol is larger than the circle of ambiguity. But on higher resolution zooms, when the size of the circle becomes larger than the size of the symbol, then the symbol should NOT be displayed because it implies a location at the center which is incorrect. Only the circle should be displayed at these zooms. Further, the "center" location of this circle should be slightly randomized (say within half the range of ambiguity) so that if there are many stations reporting the same location and ambiguity, that all of their circles will show. . These circles are not intended to be precise sizes or edges of ambiguity, but simply a graphical representation to the viewer that these positions are not well known.

Default NullPosition

Where a station does not have *any* specific position information to transmit (for example, a Mic-E unit without a GPS receiver connected to it), the station must transmit a default null position in the location field.

The null position corresponds to 0° 0' 0" north, 0° 0' 0" west.

The null position should be include the `\.` symbol (unknown/indeterminate position). For example, a Position Report for a station with unknown position will contain the coordinates …`0000.00N\00000.00W.`…

Maidenhead Locator (Grid Square)

An alternative method of expressing a station's location is to provide a Maidenhead locator (grid square). There are four ways of doing this:

- In a Status Report — e.g. `IO91SX/-`
  (`/-` represents the symbol for a "house").
- In Mic-E Status Text — e.g. `IO91SX/G`
  (`/G` indicates a "grid square").
- In the Destination Address — e.g. `IO91SX`. (obsolete).
- In AX.25 beacon text, with the `[` APRS Data Type Identifier — e.g. `[IO91SX]` (obsolete).

Grid squares may be in 6-character form (as above) or in the shortened 4-character form (e.g. `IO91`).

NMEA Data

APRS recognizes raw ASCII data strings conforming to the NMEA 0183 Version 2.0 specification, originating from navigation equipment such as GPS and LORAN receivers. It is recommended that APRS stations interpret at least the following NMEA Received Sentence types:

GGA	Global Positioning System Fix Data
GLL	Geographic Position, Latitude/Longitude Data
RMC	Recommended Minimum Specific GPS/Transit Data
VTG	Velocity and Track Data
WPL	Way Point Location

Altitude

Altitude may be expressed in two ways:

- In the comment text.
- In Mic-E format.

Altitude in Comment Text — The comment may contain an altitude value, in the form **/A=**aaaaaa, where aaaaaa is the altitude in feet. It must contain exactly 6 digits. For example: /A=001234. The altitude may appear anywhere in the comment.

Note: Although not in the official standard, many applications also recognize negative values in the form **/A=**-aaaaa . It must contain exactly 5 digits after the minus sign.

Altitude in Mic-E format — The optional Mic-E status field can contain altitude data. See Chapter 10: Mic-E Data Format.

7 APRS DATA EXTENSIONS

A fixed-length 7-byte field may follow APRS position data. This field is an APRS Data Extension. The extension may be **one** of the following:

- `CSE/SPD` Course and Speed (this may be followed by a further 8 bytes containing DF bearing and Number/Range/Quality parameters)
- `DIR/SPD` Wind Direction and Wind Speed
- `PHGphgd` Station Power and Effective Antenna Height/Gain/ Directivity
- `RNGrrrr` Pre-Calculated Radio Range
- `DFSshgd` DF Signal Strength and Effective Antenna Height/Gain
- `Tyy/Cxx` Area Object Descriptor

Course and Speed

The 7-byte `CSE/SPD` Data Extension can be used to represent the course and speed of a vehicle or APRS Object.

The course is expressed in degrees (001-360), clockwise from due north. The speed is expressed in knots. A slash `/` character separates the two.

For example:

`088/036` represents a course 88 degrees, traveling at 36 knots.

If the course and speed are unknown or not relevant, they can be set to `000/000` or `.../...` or `␣␣␣/␣␣␣` .

Note: In the special case of DF reports, a course of `000` means that the DF station is fixed. If the course is non-zero, the station is mobile.

Wind Direction and Wind Speed

The 7-byte `DIR/SPD` Data Extension can be used to represent the wind direction and sustained one-minute wind speed in a Weather Report.

The wind direction is expressed in degrees (001-360), clockwise from due north. The speed is expressed in knots. A slash `/` character separates the two.

For example:

`220/004` represents a wind direction of 220 degrees and a speed of 4 knots.

If the wind direction and speed are unknown or not relevant, they can be set to `000/000` or `.../...` or `␣␣␣/␣␣␣` .

Power,Effective Antenna Height/Gain/ Directivity

The 7-byte **PHG**phgd Data Extension specifies the transmitter power, effective antenna height-above-average-terrain, antenna gain and antenna directivity. APRS uses this information to plot radio range circles around stations.

The 7 characters of this Data Extension are encoded as follows:

Characters 1–3: **PHG** (fixed)
Character 4: p Power code
Character 5: h Height code
Character 6: g Antenna gain code
Character 7: d Directivity code

PHG Codes

The PHG codes are listed in the table below:

phgd ***Code:***	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	***Units***
Power	0	1	4	9	16	25	36	49	64	81	watts
Height	10 3	20 6.1	40 12.2	80 24.4	160 48.8	320 97.5	640 195.1	1280 390.1	2560 780.3	5120 1560	feet meters
Gain	0	1	2	3	4	5	6	7	8	9	dBi
Directivity	omni	45 NE	90 E	135 SE	180 S	225 SW	270 W	315 NW	360 N		deg

The height code represents the effective height of the antenna *above average local terrain*, not above ground or sea level — this is to provide a rough indication of the antenna's effectiveness in the local area .

The height code may in fact be any ASCII character 0–9 and above. This is so that larger heights for balloons, aircraft or satellites may be specified.

For example:
: is the height code for 10240 feet (approximately 3.1 km).
; is the height code for 20480 feet (approximately 36.2 km), and so on.

The Directivity code offsets the PHG circle by one third in the indicated direction. This means a front-to-back ratio of 2 to 1. Most often this is used to indicate a favored direction or a null, even if an omni antenna is at the site.

An example of the PHG Data Extension:

PHG5132 means a power of 25 watts,
an antenna height of 20 feet above the average local terrain,
an antenna gain of 3 dBi,
and maximum gain due east.

PHGR "probes"

Version 1.2 adds a new variation with an extra character followed by a mandatory / character. This extra character signifies the number of beacons per hour. For numbers above 9, 'A' represents 10, and so on. This allows a listening station to measure reliability, of hearing a given station, by counting the number of beacons heard directly, i.e. no through digipeaters. This new form violates the rule that Data Extensions are always 7 characters.

Example: PHG72604/ means 4 per hour. If 3 were heard in an hour, reception of that station would be 75%.

Range Circle Plot

On receipt, APRS uses the **p**, **h**, **g** and **d** codes to calculate the usable radio range (in miles), for plotting a range circle representing the local radio horizon around the station. The radio range is calculated as follows:

power = $\mathbf{p}^2$

Height-above-average-terrain (haat) = $10 \times 2^{\mathbf{h}}$

gain = $10^{(\mathbf{g}/10)}$

range = $\sqrt{2 \times \text{haat} \times \sqrt{(\text{power}/10) \times (\text{gain}/2)}}$

Thus, for `PHG5132`:

power = $\mathbf{5}^2$ = 25 watts

haat = $10 \times 2^{\mathbf{1}}$ = 20 feet

gain = $10^{(\mathbf{3}/10)}$ = 1.995262

range = $\sqrt{2 \times 20 \times \sqrt{(25/10) \times (1.995262/2)}}$

~ 7.9 miles

As the direction of maximum gain is due east, APRS will draw a range circle of radius 8 miles around the station, offset by 2.7 miles (i.e. one third of 8 miles) in an easterly direction.

Note: In the absence of any PHG data, stations are assumed to be running 10 watts to a 3dBi omni antenna at 20 feet, resulting in a 6-mile radius range circle, centered on the station.

For more details:

The Importance of PHG Range Circles & APRS Mobile Range

Pre-CalculatedRadio Range

The 7-byte **RNG**`rrrr` Data Extension allows users to transmit a pre-calculated omni-directional radio range, where `rrrr` is the range in miles (with leading zeros).

For example, `RNG0050` indicates a radio range of 50 miles.

APRS can use this value to plot a range circle around the station.

Omni-DF Signal Strength

The 7-byte **DFS**`shgd` Data Extension lets APRS localize jammers by plotting the overlapping signal strength contours of all stations hearing the signal. This Omni-DF format replaces the PHG format to indicate DF signal strength, in that the transmitter power field is replaced with the relative signal strength (s) from 0 to 9.

DFS Codes

shgd **Code:**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	***Units***
Strength	0	1	2	3	4	5	6	7	8	9	S-points
Height	10	20	40	80	160	320	640	1280	2560	5120	feet
Gain	0	1	2	3	4	5	6	7	8	9	dB
Directivity	omni	45 NE	90 E	135 SE	180 S	225 SW	270 W	315 NW	360 N		deg

For example, DFS2360 represents a weak signal (around strength S2) heard on an omni antenna with 6 dB gain at 80 feet.

A signal strength of zero (0) is particularly significant, because APRS uses these 0 signal reports to draw (usually black) circles where the jammer is *not* heard. These black circles are extremely valuable since there will be a lot more reports from stations that do not hear the jammer than from those that do. This quickly eliminates a lot of territory.

Bearing and Number/Range/Quality

DF reports contain an 8-byte field /BRG/NRQ that follows the CSE/SPD Data Extension, specifying the course, speed, bearing and NRQ (Number/Range/Quality) value of the report. NRQ indicates the Number of hits, the approximate Range and the Quality of the report.

For example, in:

…088/036/270/729…	course = 88 degrees, speed = 36 knots, bearing = 270 degrees, N = 7, R = 2, Q = 9

If N is 0, then the NRQ value is meaningless. Values of N from 1 to 8 give an indication of the number of hits per period relative to the length of the time period — thus a value of 8 means 100% of all samples possible got a hit. A value of 9 for N indicates to other users that the report is manual.

The N value is not processed, but is just another indicator from the automatic DF units.

The range (R) indicates the approximate area of interest. The range limits the length of the line to the original map's scale of the sending station. The range is 2^R so, for R=4, the range will be 16 miles.

The Q byte is QUALITY and is a relative measure of the degree of unknown in the beam heading. Q is a single digit in the range 0–9, and provides an indication of bearing accuracy. It is best interpreted as a BEAMWIDTH as follows:

Q	Bearing Accuracy
0	Useless
1	< 240 deg
2	< 120 deg
3	< 64 deg
4	< 32 deg

Q	Bearing Accuracy
5	< 16 deg
6	< 8 deg
7	< 4 deg
8	< 2 deg
9	< 1 deg (best)

See original DF.TXT and PROTOCOL.TXT for more information.

If the course and speed parameters are not appropriate, they should have the value `000/000` or `.../...` or `␣␣␣/␣␣␣` .

Area ObjectDescriptor

The 7-byte `Tyy/Cxx` Data Extension is an Area Object Descriptor. The `T` parameter specifies the type of object (square, circle, triangle, etc) and the `/C` parameter specifies its fill color.

Area Objects are described in Chapter 11: Object and Item Reports.

8 POSITION AND DF REPORT DATA FORMATS

Position Reports

Lat/Long Position Reports are contained in the Information field of an APRS AX.25 frame.

The following diagrams show the permissible formats of these reports, together with some examples. The gray areas indicate optional fields, and the shaded (yellow) characters are literal ASCII characters. In all cases there is a maximum of 43 characters after the Symbol Code.

Why? Where did that 43 character limit come from? Chapter 3 stated that the Information part could be up to 256 characters. In practice, we often see much longer comments. The MIC-E format has 'n' for comment length rather than a specific limit.

Lat/Long Position Report Format — without Timestamp

	! *or* =	*Lat*	*Sym Table ID*	*Long*	*Symbol Code*	*Comment (max 43 chars)*
Bytes:	1	8	1	9	1	0-43

Examples

`!4903.50N/07201.75W-Test 001234` no timestamp, no APRS messaging, with comment.

`!4903.50N/07201.75W-Test /A=001234` no timestamp, no APRS messaging, altitude = 1234 ft.

`!49␣␣.␣␣N/072␣␣.␣␣W-` no timestamp, no APRS messaging, location to nearest degree.

`!4903.50N/07201.75Wn` no timestamp, no APRS messaging,

Lat/Long Position Report Format — with Timestamp

	/ *or* @	*Time DHM / HMS*	*Lat*	*Sym Table ID*	*Long*	*Symbol Code*	*Comment (max 43 chars)*
Bytes:	1	7	8	1	9	1	0-43

Examples

`/092345z4903.50N/07201.75W>Test1234` with timestamp, no APRS messaging, zulu time, with comment.

`@092345/4903.50N/07201.75W>Test1234` with timestamp, with APRS messaging, local time, with comment.

Lat/Long Position Report Format — with Data Extension (no Timestamp)						
! or =	Lat	Sym Table ID	Long	Symbol Code	Course/Speed Power/Height/Gain/Dir Radio Range DF Signal Strength	Comment (max 36 chars)
Bytes: 1	8	1	9	1	7	0-36

Example
`=4903.50N/07201.75W#PHG5132` no timestamp, with APRS messaging, with PHG.
`=4903.50N/07201.75W_225/000g000t050r000p001…h00b10138dU2k` weather report.

Lat/Long Position Report Format — with Data Extension and Timestamp							
/ or @	Time DHM / HMS	Lat	Sym Table ID	Long	Symbol Code	Course/Speed Power/Height/Gain/Dir Radio Range DF Signal Strength	Comment (max 36 chars)
Bytes: 1	7	8	1	9	1	7	0-36

Examples
`@092345/4903.50N/07201.75W>088/036` with timestamp, with APRS messaging, local time, course/speed.
`@234517h4903.50N/07201.75W>PHG5132` with timestamp, APRS messaging, hours/mins/secs time, PHG.
`@092345z4903.50N/07201.75W>RNG0050` with timestamp, APRS messaging, zulu time, radio range.
`/234517h4903.50N/07201.75W>DFS2360` with timestamp, hours/mins/secs time, DF, no APRS messaging.
`@092345z4903.50N/07201.75W_090/000g000t066r000p000…dUII` weather report.

Maidenhead Locator Beacon			
[	Grid Locator	]	Comment
Bytes: 1	4 or 6	1	n

Examples
```
[IO91SX] 35 miles NNW of London
[IO91]
```

Sending raw GPS data is discouraged. This was a hack for early trackers with inadequate computing resources to allow conversion to a proper position report. Symbols had to go in the destination field using names like GPSxxx.

Raw NMEA Position Report Format	
NMEA Received Sentence	
$	...,...,...,...,...,...,...
Bytes: 1	25-209

Examples

```
$GPGGA,102705,5157.9762,N,00029.3256,W,1,04,2.0,75.7,M,47.6,M,,*62
$GPGLL,2554.459,N,08020.187,W,154027.281,A
$GPRMC,063909,A,3349.4302,N,11700.3721,W,43.022,89.3,291099,13.6,E*52
$GPVTG,318.7,T,,M,35.1,N,65.0,K*69
```

DF Reports DF Reports are contained in the Information field of an APRS AX.25 frame. The Bearing and Number/Range/Quality (BRG/NRQ) parameters follow the Data Extension field.

Note: The BRG/NRQ parameters are only meaningful when the report contains the DF symbol (i.e. the Symbol Table ID is / and the Symbol Code is \).

Note: If the DF station is fixed, the Course value is zero. If the station is moving, the Course value is non-zero.

DF Report Format — without Timestamp							
! *or* =	*Lat*	*Sym Table ID* /	*Long*	*Symbol Code* \	*Course/Speed* *Power/Height/Gain/Dir* *Radio Range* *DF Signal Strength*	*/BRG/NRQ*	*Comment (max 28 chars)*
Bytes: 1	8	1	9	1	7	8	0-28

Examples

`=4903.50N/07201.75W\088/036/270/729`	no timestamp, course/speed/ bearing/NRQ, with APRS messaging. DF station moving (CSE is non-zero).
`=4903.50N/07201.75W\000/036/270/729`	Same report, DF station fixed (CSE=000).

DF Report Format — with Timestamp								
/ *or* **@**	***Time DHM / HMS***	***Lat***	***Sym Table ID*** **/**	***Long***	***Symbol Code*** ****	***Course/Speed*** / ***Power/Height/Gain/Dir*** / ***Radio Range*** / ***DF Signal Strength***	**/*BRG*/*NRQ***	***Comment (max 28 chars)***
1	7	8	1	9	1	7	8	0-28

Bytes: 1, 7, 8, 1, 9, 1, 7, 8, 0-28

<u>Examples</u>

`@092345z4903.50N/07201.75W\088/036/270/729`	with timestamp, course/speed/ bearing/NRQ, with APRS messaging.
`/092345z4903.50N/07201.75W\000/000/270/729`	with timestamp, bearing/NRQ, no course/speed, no APRS messaging.

9 COMPRESSED POSITION REPORT DATA FORMATS

In compressed data format, the Information field contains the station's latitude and longitude, together with course and speed or pre-calculated radio range or altitude.

This information is compressed to minimize the length of the transmitted packet (and therefore improve its chances of being received correctly under less than ideal conditions).

The Information field also contains a display Symbol Code, and there may optionally be a plain text comment (uncompressed) as well.

The Advantages of Data Compression

Compressed data format may be used in place of the numeric lat/long coordinates already described, such as in the `!`, `/`, `@` and `=` formats.

Data compression has several important benefits:

- Fully backwards compatible with all existing formats.
- Fully supports any comment string.
- Speed is accurate to +/-1 mph up to about 40 mph and within 3% at 600 mph.
- Altitude in feet is accurate to +/- 0.4% from 1 foot to 3000 miles.
- Consistent one-algorithm processing of compressed latitude and longitude.
- Improved position to 1 foot worldwide.
- Pre-calculated radio range, compressed to one byte.
- Potential 50% compression of every position format on the air.
- Potential 40% reduction of raw GPS NMEA data length.
- Additional 7-byte reduction for NEMA GGA altitudes.
- Support for TNC compression at the NMEA source (from the GPS receiver).
- Digipeater compression of old NMEA trackers on the fly.
- Usage is optional in all cases.

The only minor disadvantages are that the course only resolves to +/- 2 degrees, and this format does not support PHG.

Compressed Data Format

Compressed data may be generated in several ways:

- by APRS software.
- pre-entered manually into a digipeater's beacon text.
- by a digipeater converting raw tracker NMEA packets to compressed.

[In future, there is the possibility that a Kantronics KPC-3 or other tracker TNC will be able to compress data directly from an attached GPS receiver].

In all cases the compressed format is a fixed 13-character field:

`/YYYYXXXX$csT`

where `/` is the Symbol Table Identifier
`YYYY` is the compressed latitude
`XXXX` is the compressed longitude
`$` is the Symbol Code
`cs` is the compressed course/speed or compressed pre-calculated radio range or compressed altitude
`T` is the compression type indicator

Compressed Position Data					
Sym Table ID	***Compressed Lat*** `YYYY`	***Compressed Long*** `XXXX`	***Symbol Code***	***Compressed Course/Speed*** / ***Compressed Radio Range*** / ***Compressed Altitude***	***Comp Type*** `T`
1	4	4	1	2	1

Bytes: (row above)

Compressed format can be used in place of lat/long position format anywhere that …`ddmm.hhN/dddmm.hhW$xxxxxxx`… occurs.

All bytes except for the **/** and **$** are base-91 printable ASCII characters (**!**..**{**). These are converted to numeric values by subtracting 33 from the decimal ASCII character code. For example, **#** has an ASCII code of 35, and represents a numeric value of 2 (i.e. 35-33).

Symbol

The presence of the leading Symbol Table Identifier instead of a digit indicates that this is a compressed Position Report and not a normal lat/long report.

Lat/Long Encoding

The values of `YYYY` and `XXXX` are computed as follows:

`YYYY` is 380926 x (90 – latitude) [base 91]
latitude is positive for north, negative for south, in degrees.

`XXXX` is 190463 x (180 + longitude) [base 91]
longitude is positive for east, negative for west, in degrees.

For example, for a longitude of 72° 45' 00" west (i.e. -72.75 degrees), the math is 190463 x (180 – 72.75) = 20427156. Because this is to base 91, it is then necessary to progressively divide this value by reducing powers of 91, to obtain the numerical values of `X`:

$20427156 / 91^3$ = **27**, remainder 80739
$80739 / 91^2$ = **9**, remainder 6210
$6210 / 91^1$ = **68**, remainder **22**

To obtain the corresponding ASCII characters, 33 is added to each of these values, yielding 60 (i.e. 27+33), 42, 101 and 55. From the ASCII Code Table (in Appendix 3), this corresponds to **`<*e7`** for `XXXX`.

Lat/Long Decoding

To decode a compressed lat/long, the reverse process is needed. That is, if `YYYY` is represented as $y_1y_2y_3y_4$ and `XXXX` as $x_1x_2x_3x_4$, then:

$$Lat = 90 - ((y_1-33) \times 91^3 + (y_2-33) \times 91^2 + (y_3-33) \times 91 + y_4-33) / 380926$$

$$Long = -180 + ((x_1-33) \times 91^3 + (x_2-33) \times 91^2 + (x_3-33) \times 91 + x_4-33) / 190463$$

For example, if the compressed value of the longitude is **`<*e7`** (as computed above), the calculation becomes:

$$\begin{aligned} Long &= -180 + (27 \times 91^3 + 9 \times 91^2 + 68 \times 91 + 22) / 190463 \\ &= -180 + (20346417 + 74529 + 6188 + 22) / 190463 \\ &= -180 + 107.25 \\ &= -72.75 \text{ degrees} \end{aligned}$$

Course/Speed,Pre-Calculated Radio Range and Altitude

The two `cs` bytes following the Symbol Code character can contain either the compressed course and speed or the compressed pre-calculated radio range or the station’s altitude. These two bytes are in base 91 format.

In the special case of `c` = **␣** (space), there is no course, speed or range data, in which case the `csT` bytes are ignored.

Course/Speed — If the ASCII code for `c` is in the range **`!`** to **`z`** inclusive — corresponding to numeric values in the range 0–89 decimal (i.e. after subtracting 33 from the ASCII code) — then `cs` represents a compressed course/speed value:

course = **c** x 4

speed = $1.08^{\mathbf{s}} - 1$

For example, if the `cs` characters are **7P**, the corresponding values of **c** and **s** (after subtracting 33 from the ASCII character code) are 22 and 47 respectively. Substituting these values in the above equations:

course = **22** x 4 = 88 degrees

speed = $1.08^{\mathbf{47}} - 1 = 36.2$ knots

Pre-Calculated Radio Range — If `c` = **{**, then `cs` represents a compressed pre-calculated radio range value:

range = $2 \times 1.08^{\mathbf{s}}$

For example, if the `cs` bytes are **{?**, the ASCII code for **?** is 63, so the value of **s** is 30 (i.e. 63-33). Thus:

range = $2 \times 1.08^{\mathbf{30}}$

~ 20 miles

So APRS will draw a circle of radius 20 miles around the station plot on the screen.

The CompressionType (T) Byte

The `T` byte follows the `cs` bytes. The `T` byte contains several bit fields showing the GPS fix status, the NMEA source of the position data and the origin of the compression.

The `T` byte is not meaningful if the `c` byte is **␣** (space).

Compression Type (T) Byte Format

Bit:	7	6	5	4	3	2	1	0
	Not used	*Not used*	***GPS Fix***	***NMEA Source***		***Compression Origin***		
Value:	0	0	0 = old (last) 1 = current	0 0 = other 0 1 = GLL 1 0 = GGA 1 1 = RMC		0 0 0 = Compressed 0 0 1 = TNC BText 0 1 0 = Software (DOS/Mac/Win/+SA) 0 1 1 = [tbd] 1 0 0 = KPC3 1 0 1 = Pico 1 1 0 = Other tracker [tbd] 1 1 1 = Digipeater conversion		

For example, if the compressed position was derived from an RMC sentence, the fix is current, and the compression was performed by APRSdos software, then the value of `T` in binary is 0 0 1 11 010, which equates to 58 decimal. Adding 33 to this value gives the ASCII code for the `T` byte (i.e. 91), which corresponds to the **[** character.

Thus, using data from all the earlier examples, if the RMC sentence contains (among other parameters) the following data:

Latitude = 49° 30' 00" north
Longitude = 72° 45' 00" west
Speed = 36.2 knots
Course = 88°

and: the fix is current
compression is performed by APRSdos software
the display symbol is a "car"

then the complete 13-character compressed location field is transmitted as:

/	YYYY	XXXX	$	csT
/	**5L!!**	**<*e7**	**>**	**7P[**

Altitude

If the T byte indicates that the raw data originates from a GGA sentence (i.e. bits 4 and 3 of the T byte are 10), then the sentence contains an altitude value, in feet. After compression, the compressed altitude data is placed in the cs bytes, such that:

$$altitude = 1.002^{cs} \text{ feet}$$

For example, if the received cs bytes are **S]**, the computation is as follows:

- Subtract 33 from the ASCII code for each character:
  c = 83 – 33 = 50
  s = 93 – 33 = 60
- Multiply **c** by 91 and add **s** to obtain **cs**:
  cs = 50 x 91 + 60
  = 4610
- Then altitude = 1.002^{4610}
  = 10004 feet

Trackers

Tracker firmware may compress GPS data directly to APRS compressed format. They would use the **!** Data Type Indicator, showing that the position is real-time and that the tracker is not APRS-capable.

If the Position Report is not real-time, then the **/** Data Type Indicator can be used instead, so that the latest fix time may be included.

Compressed Report Formats

Compressed data is contained in the AX.25 Information field, in these formats:

Compressed Lat/Long Position Report Format — no Timestamp

	! *or* ***=***	***Sym Table ID***	***Comp Lat*** YYYY	***Comp Long*** XXXX	***Symbol Code***	***Compressed Course/Speed*** ***Compressed Radio Range*** ***Compressed Altitude***	***Comp Type*** T	***Comment (max 40 chars)***
Bytes:	1	1	4	4	1	2	1	0-40

<u>Examples</u>

`=/5L!!<*e7>␣sTComment`	with APRS messaging. Note the ␣ space character following the > Symbol Code, indicating that there is no course/speed, radio range or altitude. The **sT** characters are fillers and have no significance here.
`=/5L!!<*e7>7P[`	with APRS messaging, RMC sentence, with course/speed.
`=/5L!!<*e7>{?!`	with APRS messaging, with radio range.
`=/5L!!<*e7OS]S`	with APRS messaging, GGA sentence, altitude.

Compressed Lat/Long Position Report Format — with Timestamp

	/ *or* ***@***	***Time DHM / HMS***	***Sym Table ID***	***Comp Lat*** YYYY	***Comp Long*** XXXX	***Symbol Code***	***Compressed Course/Speed*** ***Compressed Radio Range*** ***Compressed Altitude***	***Comp Type*** T	***Comment (max 40 chars)***
Bytes:	1	7	1	4	4	1	2	1	0-40

<u>Example</u>

`@092345z/5L!!<*e7>{?!`	with APRS messaging, timestamp, radio range.

10 MIC-E DATA FORMAT

Mic-E Data Format

In Mic-E data format, the station's position, course, speed and display symbol, together with an APRS digipeater path and Mic-E Position Comment,are packed into the AX.25 Destination Address and Information fields.

The Information field can also optionally contain Mic-E status. The Mic-E Status text can contain the station's Maidenhead locator and altitude.

Mic-E packets can be very short. At the minimum, with no callsigns in the Digipeater Addresses field and no optional telemetry data or Mic-E status text, a complete Mic-E packet is just 25 bytes long (excluding FCS and flags).

Mic-E data format is not only used in the Microphone Encoder unit; it is also used in the PIC Encoder and in the Kenwood and Yaesu radios.

Mic-E Data Payload

The Mic-E data format allows a large amount of data to be carried in a very short packet. The data is split between the Destination Address field and the Information field of a standard AX.25 UI-frame.

Destination Address Field — The 7-byte Destination Address field contains the following encoded information:

- The 6 latitude digits.
- A 3-bit Mic-E position comment, specifying one of 7 Standard Mic-E Position Comment Codes or one of 7 Custom Position Comments or an Emergency Position Comment.
- The North/South and West/East Indicators.
- The Longitude Offset Indicator.
- The generic APRS digipeater path code.

Although the destination address appears to be quite unconventional, it is still a valid AX.25 address, consisting only of printable 7-bit ASCII values (shifted one bit left) — see the *Amateur Packet-Radio Link-Layer Protocol* specification for full details of the format of standard AX.25 addresses.

Information Field — This field contains the following data:

- The encoded longitude.
- The encoded course and speed.
- The display Symbol Code and Symbol Table Identifier.
- An optional field, containing either Mic-E telemetry data (now obsolete) or a Mic-E status text string. The status string can contain plain text, Maidenhead locator or the station's altitude.

Mic-E Destination Address Field

The standard AX.25 Destination Address field consists of 7 bytes, containing 6 callsign characters and the SSID (plus a number of other bits that are not of interest here). When used to carry Mic-E data, however, this field has a quite different format:

Mic-E Data — DESTINATION ADDRESS FIELD Format							
	Lat Digit 1 + Bit A	Lat Digit 2 + Bit B	Lat Digit 3 + Bit C	Lat Digit 4 + N/S Lat Indicator	Lat Digit 5 + Longitude Offset	Lat Digit 6 + W/E Long Indicator	APRS Digi Path Code
Bytes:							

The Destination Address field contains:

- Six encoded latitude digits specifying degrees (digits 1 and 2), minutes (digits 3 and 4) and hundredths of minutes (digits 5 and 6).
- 3-bit Mic-E Position Comment identifier (bits A, B and C).
- North/South latitude indicator.
- Longitude offset (adds 0 degrees or 100 degrees to the longitude computation in the Information field).
- West/East longitude indicator.
- Generic APRS digipeater path (encoded in the SSID).

Destination Address Field Encoding

The table on the next page shows the encoding of the first 6 bytes of the Destination Address field, for all combinations of latitude digit, the 3-bit Mic-E message identifier (A/B/C), the latitude/longitude indicators and the longitude offset.

The encoding supports position ambiguity.

The ASCII characters shown in the table are left-shifted one bit position prior to transmission.

Mic-E Destination Address Field Encoding (Bytes 1–6)

Byte:	1-6	1-3	4	5	6
ASCII Char	***Lat Digit***	***Position Comment A/B/C***	***N/S***	***Long Offset***	***W/E***
0	0	0	South	+0	East
1	1	0	South	+0	East
2	2	0	South	+0	East
3	3	0	South	+0	East
4	4	0	South	+0	East
5	5	0	South	+0	East
6	6	0	South	+0	East
7	7	0	South	+0	East
8	8	0	South	+0	East
9	9	0	South	+0	East
A	0	1 (Custom)			
B	1	1 (Custom)			
C	2	1 (Custom)			
D	3	1 (Custom)			
E	4	1 (Custom)			
F	5	1 (Custom)			
G	6	1 (Custom)			

Byte:	1-6	1-3	4	5	6
ASCII Char	***Lat Digit***	***Position Comment A/B/C***	***N/S***	***Long Offset***	***W/E***
H	7	1 (Custom)			
I	8	1 (Custom)			
J	9	1 (Custom)			
K	space	1 (Custom)			
L	space	0	South	+0	East
P	0	1 (Std)	North	+100	West
Q	1	1 (Std)	North	+100	West
R	2	1 (Std)	North	+100	West
S	3	1 (Std)	North	+100	West
T	4	1 (Std)	North	+100	West
U	5	1 (Std)	North	+100	West
V	6	1 (Std)	North	+100	West
W	7	1 (Std)	North	+100	West
X	8	1 (Std)	North	+100	West
Y	9	1 (Std)	North	+100	West
Z	space	1 (Std)	North	+100	West

Note: the ASCII characters **A**–**K** are not used in address bytes 4–6.

For example, for a station at a latitude of 33 degrees 25.64 minutes north, in the western hemisphere, with longitude offset +0 degrees, and transmitting standard position comment identifier bits 1/0/0, the encoding of the first 6 bytes of theDestination Address field is as follows:

Destination Address Byte:	***1***	***2***	***3***	***4***	***5***	***6***
Latitude Digit	3	3	2	5	6	4
Position Comment	Bit A = 1 (Std)	Bit B = 0	Bit C = 0			
N/S Indicator				North		
Long Offset					+0	
W/E Indicator						West
Dest Address (ASCII Char)	**S**	**3**	**2**	**U**	**6**	**T**

Mic-E Position Comments

The first three bytes of the Destination Address field contain three position comment bits: A, B and C. These bits allow one of 15 position comment types to be specified:

- 7 Standard
- 7 Custom
- 1 Emergency

For the 7 Standard position comments, one or more of the identifier bits is a **1**, shown in the Mic-E Destination Address Field Encoding table as 1 (Std).

For the 7 Custom position comments, one or more of the identifier bits is a **1**, shown in the Mic-E Destination Address Field Encoding table as 1 (Custom).

For the Emergency position comment, all three identifier bits are **0**.

The following table shows the encoding of Mic-E message types, for all combinations of the A/B/C message identifier bits:

Mic-E Position Comment Types

A	*B*	*C*	*Standard Mic-E*	*Custom Mic-E*
1	1	1	M0: Off Duty	C0: Custom-0
1	1	0	M1: En Route	C1: Custom-1
1	0	1	M2: In Service	C2: Custom-2
1	0	0	M3: Returning	C3: Custom-3
0	1	1	M4: Committed	C4: Custom-4
0	1	0	M5: Special	C5: Custom-5
0	0	1	M6: Priority	C6: Custom-6
0	0	0	Emergency	

The Standard position comments and the Emergency position comment have the same meaningfor all APRS stations. The Custom position comments may be assigned any arbitrary meaning.

Note: Support for Custom position comments is optional. Original Mic-E units do notsupport Custom position comments.

Note: If the A/B/C position comment identifier bits contain a mixture of Standard **1**sand Custom **1**s, the position comment type is “unknown”.

Some examples of MIC-E Position Comment type encoding:

First 3 Destination Address Bytes	***IdentifierBits A/B/C***	***Type***	***Meaning***
S32	Standard 1 / 0 / 0	Standard	M3: Returning
F2D	Custom 1 / 0 / Custom 1	Custom	C2: Custom-2
234	0 / 0 / 0	Emergency	Emergency

Destination Address SSID Field

The SSID in the Destination Address field of a Mic-E packet is coded to specify either a conventional digipeater VIA path (contained in the Digipeater Addresses field of the AX.25 frame), or one of 15 generic APRS digipeater paths. See Chapter 4: APRS Data in the AX.25 Destination and Source Address Fields.

Mic-E Information Field

The Information field is used to complete the Position Report that was begun in the Destination Address field. The encoding used is different from the destination address since the content is not constrained to be printable, shifted 7-bit ASCII, as it is in the address. However, full 8-bit binary is not used — all values are offset by 28 and further operations (described below) are performed on some of the values to make almost all of the data printable ASCII.

The format of the Information field is as follows:

Mic-E Data — INFORMATION FIELD Format

	Data Type ID	***Longitude***			***Speed and Course***			***Symbol Code***	***Sym Table ID***	***Mic-E Telemetry Data (obsolete)***
		d+28	m+28	h+28	SP+28	DC+28	SE+28			***Mic-E Status Text***
Bytes:	1	1	1	1	1	1	1	1	1	n

Information Field Data

The first 9 bytes of the Information field contain the APRS Data Type Identifier, longitude, speed, course and symbol data.

The APRS Data Type Identifier is one of:

` Current GNSS data
(but not used in Kenwood TM-D700 radios) .

' Old GNSS data
(or *Current* GNSS data in Kenwood TM-D700 radios).

0x1c Current GNSS data (Obsolete - Rev. 0 beta units only).

0x1d Old GNSS data (Obsolete - Rev. 0 beta units only).

<u>IMPORTANT NOTE</u>: As explained in detail below, some of the bytes in the Information field are non-printing ASCII characters. Use of non-printing characters makes it harder to troubleshoot. In certain circumstances (such as incorrect TNC setup or inappropriate software), someof these non-printing characters may be dropped, making the Information field appear shorter than it really is. This will lead to incorrect decoding. (In particular, if the Information field appears to be less than 9 bytes long, the packet must be ignored).

Longitude Degrees Encoding

The **d+28** byte in the Information field contains the encoded value of the longitude degrees, in the range 0–179 degrees.

(Note that for longitude values in the range 0–9 degrees, the longitude offset is +100 degrees):

Mic-E Longitude Degrees Encoding

Long Deg	*ASCII Char*	*d+28*	*Long Offset*
0	`v`	118	**+100**
1	`w`	119	**+100**
2	`x`	120	**+100**
3	`y`	121	**+100**
4	`z`	122	**+100**
5	`{`	123	**+100**
6	`|`	124	**+100**
7	`}`	125	**+100**
8	`~`	126	**+100**
9	**DEL**	127	**+100**
10	`&`	38	+0
11	`'`	39	+0
12	`(`	40	+0
…			
97	`}`	125	+0
98	`~`	126	+0
99	**DEL**	127	+0

Long Deg	*ASCII Char*	*d+28*	*Long Offset*
100	`l`	108	+100
101	`m`	109	+100
102	`n`	110	+100
103	`o`	111	+100
104	`p`	112	+100
105	`q`	113	+100
106	`r`	114	+100
107	`s`	115	+100
108	`t`	116	+100
109	`u`	117	+100
110	`&`	38	+100
111	`'`	39	+100
112	`(`	40	+100
…			
177	`i`	105	+100
178	`j`	106	+100
179	`k`	107	+100

Note from the table that the encoding is split into four separate pieces:

- <u>0–9 degrees</u>: **d+28** is in the range 118–127 decimal, corresponding to the ASCII characters **v** to **DEL**.

  Important Note: The longitude offset is set to **+100 degrees** when the longitude is in the range 0–9 degrees.

- <u>10–99 degrees</u>: **d+28** is in the range 38–127 decimal (corresponding to the ASCII characters **&** to **DEL**), and the longitude offset is +0 degrees.

- 100–109 degrees: **d+28** is in the range 108–117 decimal, corresponding to the ASCII characters **ℓ** (lower-case letter "L") to **"u"**, and the longitude offset is +100 degrees.
- 110–179 degrees: **d+28** is in the range 38–107 decimal (corresponding to the ASCII characters **&** to **"k"**), and the longitude offset is +100 degrees.

Thus the overall range of valid **d+28** values is 38–127 decimal (corresponding to ASCII characters **&** to **DEL**).

All of these characters (except **DEL**, for 9 and 99 degrees) are printable ASCII characters.

To decode the longitude degrees value:

1. subtract 28 from the **d+28** value to obtain **d**.
2. if the longitude offset is +100 degrees, add 100 to **d**.
3. subtract 80 if 180 ˜ **d** ˜ 189
   (i.e. the longitude is in the range 100–109 degrees).
4. or, subtract 190 if 190 ˜ **d** ˜ 199.
   (i.e. the longitude is in the range 0–9 degrees).

Longitude Minutes Encoding

The **m+28** byte in the Information field contains the encoded value of the longitude minutes, in the range 0–59 minutes:

Mic-E Longitude Minutes Encoding

Long Mins	ASCII Char	m+28	Long Mins	ASCII Char	m+28
0	X	88	**10**	&	38
1	Y	89	**11**	'	39
2	Z	90	**12**	(	40
3	[	91	**13**	)	41
4	\	92	**14**	*	42
5	]	93	**…**		
6	^	94	**56**	T	84
7	_	95	**57**	U	85
8	`	96	**58**	V	86
9	a	97	**59**	W	87

Note from the table that the encoding is split into two separate pieces:

- 0–9 minutes: **m+28** is in the range 88–97 decimal, corresponding to the ASCII characters **X** to **a**.
- 10–59 minutes: **m+28** is in the range 38–87 decimal (corresponding to the ASCII characters **&** to **W**).

Thus the overall range of valid **m+28** values is 38–97 decimal (corresponding to ASCII characters **&** to **a**). All of these characters are printable ASCII characters.

To decode the longitude minutes value:

1. subtract 28 from the **m+28** value to obtain **m**.
2. subtract 60 if **m** ™ 60.
   (i.e. the longitude minutes is in the range 0–9).

Longitude Hundredths of Minutes Encoding

The **h+28** byte in the Information field contains the encoded value of the longitude hundredths of minutes, in the range 0–99 minutes. This byte takes a value in the range 28 decimal (corresponding to 0 hundredths of a minute) through 127 decimal (corresponding to 99 hundredths of a minute).

To decode the longitude hundredths of minutes value, subtract 28 from the **h+28** value.

All of the possible values are printable ASCII characters (except 0–3 and 99 hundredths of a minute).

Speed and Course Encoding

The speed and course of a station are encoded in 3 bytes, designated **SP+28**, **DC+28** and **SE+28**.

The speed is in the range 0–799 knots, and the course is in the range 0–360 degrees (0 degrees represents an unknown or indefinite course, and 360 degrees represents due north).

The encoded speed and course are spread over the three bytes, as follows:

Speed		***Course***
Encoded Speed (hundreds/tens of knots)	Encoded Speed (units) and Encoded Course (hundreds of degrees)	Encoded Course (tens/units)
SP+28	**DC+28**	**SE+28**

SP+28 Encoding

The **SP+28** byte contains the encoded speed, in hundreds/tens of knots, according to this table:

SP+28 Speed Encoding (hundreds/tens of knots)

Speed knots	ASCII Char		SP +28	
0-9	l	0x1c	108	28
10-19	m	0x1d	109	29
20-29	n	0x1e	110	30
30-39	o	0x1f	111	31
40-49	p		112	32
50-59	q	!	113	33
60-69	r	"	114	34
70-79	s	#	115	35
80-89	t	$	116	36
90-99	u	%	117	37
100-109	v	&	118	38
110-119	w	'	119	39
120-129	x	(	120	40
130-139	y	)	121	41
140-149	z	*	122	42
150-159	{	+	123	43
160-169	|	,	124	44
170-179	}	-	125	45
180-189	~	.	126	46
190-199	DEL	/	127	47

Speed knots	ASCII Char	SP +28
200-209	0	48
210-219	1	49
220-229	2	50
230-239	3	51
240-249	4	52
250-259	5	53
260-269	6	54
270-279	7	55
280-289	8	56
290-299	9	57
300-310	:	58
310-320	;	59
…		
730-739	e	101
740-749	f	102
750-759	g	103
760-769	h	104
770-779	i	105
780-789	j	106
790-799	k	107

Note: The ASCII characters shown in white on a black background are non-printing characters.

Note: For speeds in the range 0–199 knots, there are two encoding schemes in existence. Hence there are two columns for the ASCII character, and two columns for the corresponding **SP+28** byte values.

For example, for a speed of 73 knots (i.e. in the range 70–79), the **SP+28** byte may contain either **s** or **#**, depending on the encoding method used. Both are equally valid.

The decoding algorithm described later handles either of these encoding schemes.

DC+28 Encoding

The DC+28 byte contains the encoded units of speed, plus the encoded course in hundreds of degrees:

DC+28 Speed / Course Encoding (units of knots/hundreds of degrees)

Knots (units)	Course (deg)	ASCII Char		DC +28		Knots (units)	Course (deg)	ASCII Char		DC +28	
0	0-99		0x1c	32	28	5	0-99	R	N	82	78
0	100-199	!	0x1d	33	29	5	100-199	S	O	83	79
0	200-299	"	0x1e	34	30	5	200-299	T	P	84	80
0	300-360	#	0x1f	35	31	5	300-360	U	Q	85	81
1	0-99	*	&	42	38	6	0-99	\	X	92	88
1	100-199	+	'	43	39	6	100-199	]	Y	93	89
1	200-299	,	(	44	40	6	200-299	^	Z	94	90
1	300-360	-	)	45	41	6	300-360	_	[	95	91
2	0-99	4	0	52	48	7	0-99	f	b	102	98
2	100-199	5	1	53	49	7	100-199	g	c	103	99
2	200-299	6	2	54	50	7	200-299	h	d	104	100
2	300-360	7	3	55	51	7	300-360	i	e	105	101
3	0-99	>	:	62	58	8	0-99	p	l	112	108
3	100-199	?	;	63	59	8	100-199	q	m	113	109
3	200-299	@	<	64	60	8	200-299	r	n	114	110
3	300-360	A	=	65	61	8	300-360	s	o	115	111
4	0-99	H	D	72	68	9	0-99	z	v	122	118
4	100-199	I	E	73	69	9	100-199	{	w	123	119
4	200-299	J	F	74	70	9	200-299	|	x	124	120
4	300-360	K	G	75	71	9	300-360	}	y	125	121

Note: The ASCII characters shown in white on a black background are non-printing characters.

Note: There are two encoding schemes in existence for the DC+28 byte. Hence there are two columns for the ASCII character, and two columns for the corresponding DC+28 byte values.

For example, for a speed of 73 knots (i.e. units=3) and a bearing of 294 degrees (i.e. in the range 200–299), the DC+28 byte may contain either @ or <, depending on the encoding method used. Both are equally valid.

The decoding algorithm described later handles either of these encoding schemes.

SE+28 Encoding

The **SE+28** byte contains the encoded tens and units of degrees of the course:

SE+28 Course Encoding (tens/units of degrees)

Course (deg)	ASCII Char	m+28
0	0x1c	28
1	0x1d	29
2	0x1e	30
3	0x1f	31
4		32
5	!	33
6	"	34
7	#	35
8	$	36
9	%	37
10	&	38
11	'	39
12	(	40
13	)	41
14	*	42

Long Mins	ASCII Char	m+28
15	+	43
16	,	44
17	-	45
18	.	46
19	/	47
…		
91	w	119
92	x	120
93	y	121
94	z	122
95	{	123
96	|	124
97	}	125
98	~	126
99	DEL	127

Example of Mic-E Speed and Course Encoding

For a speed of 86 knots and a course of 194 degrees, the encoding is:

SP+28: The speed is in the range 80–89 knots. From the **SP+28** encoding table, the **SP+28** byte may be either **t** or **$**.

DC+28: The units of speed are 6, and the course is in the range 100–199 degrees. From the **DC+28** encoding table, the **DC+28** byte may be either **]** or **Y**.

SE+28: The course in tens and units of degrees is 94. From the **SE+28** encoding table, the **SE+28** byte will be **z**.

Decoding theSpeed and Course

To decode the speed and course:

SP+28: To obtain the speed in tens of knots, subtract 28 from the **SP+28** value and multiply by 10.

DC+28: Subtract 28 from the **DC+28** value and divide the result by 10. The quotient is the units of speed. The remainder is the course in hundreds of degrees.

SE+28: To obtain the tens and units of degrees, subtract 28 from the **SE+28** value.

Finally, make these speed and course adjustments:

- If the computed speed is ≥ 800 knots, subtract 800.
- If the computed course is ≥ 400 degrees, subtract 400.

Example ofDecoding the Information Field Data

If the first 9 bytes of the Information field contain **`` `(_fn "Oj/ ``**, and the destination address specifies that the station is in the western hemisphere with a longitude offset of +100 degrees, then the data is decoded as follows:

- **`** is the APRS Data Type Identifier for a Mic-E packet containing current GNSS data.
- **(** is the **d+28** byte. The **(** character has the value 40 decimal. Subtracting 28 gives 12. The longitude offset (in the destination address) is +100 degrees, so the longitude is 100 + 12 = 112 degrees.
- **_** is the **m+28** byte. The **_** character has the value 95 decimal. Subtracting 28 gives 67. This is ™ 60, so subtracting 60 gives a value of 7 minutes longitude.
- **f** is the **h+28** byte. The **f** character has the value 102 decimal. Subtracting 28 gives 74 hundredths of a minute.

Thus the longitude is 112 degrees 7.74 minutes west.

The speed and course are calculated as follows:

- **n** is the **SP+28** byte. The **n** character has the value 110 decimal. After subtracting 28, the result is 82. As this is ™ 80, a further 80 is subtracted, leaving a result of 2 tens of knots.
- **"** is the **DC+28** byte. The **"** character has the value 34 decimal. Subtracting 28 gives 6. Dividing this by 10 gives a quotient of 0 (units of speed). Adding the first part of the speed, multiplied by 10 (i.e. 20) to the quotient (0) gives a final computed speed of 20 knots.
  The remainder from the division is 6. Subtracting 4 gives the course in hundreds of degrees; i.e. 2.
- **O** (upper-case letter "O") is the **SE+28** byte. The **O** character has the value 79 decimal. Subtracting 28 gives 51. Adding this to the remainder calculated above, multiplied by 100 (i.e. 200), gives the final value of 251 degrees for the course.

The last two characters (**j/**) represent the jeep symbol from the Primary Symbol Table.

Mic-E Position Ambiguity

As mentioned in Chapter 6 (Time and Position Formats), a station may reduce the precision of its position by introducing position ambiguity. This is also possible in Mic-E data format.

The position ambiguity is specified for the latitude (in the destination address). The same degree of ambiguity will then also apply to the longitude.

For example, if the destination address is **T4SQZZ**, the last two digits of the latitude are ambiguous (represented by **ZZ**). Then, if the longitude data in the Information field is **(_f** , as in the above example, the last two digits of the computed longitude will be ignored — that is, the longitude will be 112 degrees 7 minutes.

Mic-E Telemetry Data (Obsolete)

This is now obsolete and replaced by the Base 91 Comment Telemetry format described in the Telemetry Data chapter.

If the byte following the Symbol Table Identifier is one of the Telemetry Flag characters (**`**,**'** or 0x1d), then telemetry data follows:

	Optional Mic-E Telemetry Data (obsolete)					
	Telemetry Flag	***Telemetry Data Channels***				
	F	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5
Bytes:	1	1/2	1/2	1/2	1/2	1/2

The Telemetry Flag F is one of:

- **`** 2 printable hex telemetry values follow (channels 1 and 3).
- **'** 5 printable hex telemetry values follow.
- 0x1d 5 binary telemetry values follow (Rev. 0 beta units only).

If F is **`** or **'**, each channel requires 2 bytes, containing a 2-digit printable hexadecimal representation of a value ranging from 0–255. For example, 254 is represented as **FE**.

If F is 0x1d, each channel requires one byte, containing an 8-bit binary value.

For example, if the telemetry data is **'7200007100**, the **'** indicates that 5 bytes of telemetry follow, coded in hexadecimal:

0x72 = 114 decimal
0x00 = 0 decimal
0x00 = 0 decimal
0x71 = 113 decimal
0x00 = 0 decimal

Mic-E Status Text

The packet may include Mic-E status text.The status text may be any length that fits in the rest of the Information field. (Why can't we just call it a comment for consistency with the other position and object report formats?)

The Mic-E status text must not start with **`**,**'** or 0x1d, otherwise it will be confused with [now obsolete] telemetry data.

It is possible to include a standard APRS-formatted position in the Mic-E status text field. (HUH???) A suitable position will cause the APRS display software tooverride any position data the Mic-E has encoded. This is useful if using a Mic-E without a GNSS receiver. (HUH???)

The Mic-E text field can contain any normal Position comment field too, such as PHG.

To distinguish themselves from the original MIC-E device, the early Kenwood radios automatically insert a prefix code at the front of the status text string (i.e. in the 10th character of the Information field):

Kenwood TH-D7: **>**
Kenwood TM-D700: **]**

Later Kenwood models kept the same prefix, to indicate handheld or mobile, then added a single character suffix to the text:

Kenwood TH-D72: **>** … =
Kenwood TH-D74: **>** … ^
Kenwood TH-D75: **>** … &
Kenwood TM-D710: **]** … =

Devices from other manufacturers used a different prefix character of

` which indicates a system is capable of messaging, or

' which indicates a system is not capable of messaging (e.g. a tracker).

A two character suffix indicates the manufacturer and model. Examples:

Yaesu FT2D: **`** … _((messaging capable)
Yaesu FT3D: **`** … _0
Yaesu FT5D: **`** … _3

Byonics TinyTrack 3: **'** … |3 (no messaging)
Byonics TinyTrack 4: **'** … |4

APRS display applications should remove any extra prefix and suffix before displaying the text. If you see something like "_3" at the end of the status text (comment), it is because the application did not remove the device type as it should have.

Rather than having hardcoded device identifiers, it is highly recommended that applications read a file at runtime. When new models are added only this one file needs to be updated rather than modifying the code and making a new release. Machine readable files, with the latest device identifiers, can be found at https://github.com/aprsorg/aprs-deviceid .

MaidenheadLocator in the Mic-EStatus Text Field

The Mic-E status text field can contain a Maidenhead locator.

If the locator is followed by a plain text comment, the first character of the text *must* be a space. For example:

```
IO91SX/G␣Hello  world      (from a Mic-E or PIC-E)
>IO91SX/G␣Hello  world     (from a Kenwood TH-D7)
]IO91SX/G␣Hello  world     (from a Kenwood TM-D700)
```

(**/G** is the grid locator symbol).

Altitude in theMic-E Status Text Field

The Mic-E status text field can contain the station's altitude. The altitude is expressed in the form `xxx}`, where `xxx` is in meters relative to 10km below mean sea level (the deepest ocean), to base 91.

For example, to compute the `xxx` characters for an altitude of 200 feet:

200 feet = 61 meters = 10061 meters relative to the datum
$10061 / 91^2$ = **1**, remainder 1780
1780 / 91 = **19**, remainder **51**

Adding 33 to each of the highlighted values gives 34, 52 and 84 for the ASCII codes of xxx.

Thus the 4-character altitude string is **`"4T}`**

Some examples:

```
"4T}
>"4T}
]"4T}
```

Optional altitude should be *first* after the Mic-E **type** byte.

Note that this comes after any device identifier prefix character.

Mic-E Data in Non-APRS Networks

Some parts of the Mic-E AX.25 Information field may contain binary data (i.e. non-printable ASCII characters). If such a packet is constrained to the APRS network, this should not cause any difficulties.

If, however, the packet is to be forwarded via a network that does not reliably preserve binary data (e.g. the Internet), then it is necessary to convert the data to a format that will preserve it.

Further, if the packet subsequently re-emerges back onto the APRS network, it will then be necessary to re-convert the data back to its original format.

11 OBJECT AND ITEM REPORTS

Objects and Items

Any APRS station can manually report the position of an APRS entity (e.g. another station or a weather phenomenon). This is intended for situations where the entity is not capable of reporting its own position.

APRS provides two types of report to support this:

- Object Reports
- Item Reports

Object Reports specify an Object's position, can have an optional timestamp, and can include course/speed information or other Extended Data. Object Reports are intended primarily for plotting the positions of moving objects (e.g. spacecraft, storms, marathon runners without trackers).

Item Reports specify an Item's position, but cannot have a timestamp. While Item reports may also include course/speed or other Extended Data, they are really intended for inanimate things that are occasionally posted on a map (e.g. marathon checkpoints or first-aid posts). Otherwise they are handled in the same way as Object Reports.

Objects are distinguished from each other by having different Object names. Similarly, Items are distinguished from each other by having different Item names.

Implementation Recommendation: When an APRS Object/Item is displayed on the screen, the callsign of the station sending the report should be associated with the Object/Item.

Replacing anObject / Item

A fundamental precept of APRS is that any station may take over the reporting responsibility for an APRS Object or Item, by simply transmitting a new report with the same Object/Item name.

The replacement report may specify the existing location or a new location.

The original station will cease transmitting an Object/Item Report when it sees an incoming report with the same name from another station.

Killing anObject / Item

To kill an Object/Item, a station transmits a new Object/Item Report, with a "kill" character following the Object/Item name.

Implementation Recommendation: When an Object/Item is killed it should be removed from display on the screen. However, the data associated with the Object/Item should be retained internally in case it is needed later.

Additional Amplifying Comments about Objects

Object Report Format

An Object Report has a *fixed* 9-character Object name field, which may consist ofany printable ASCII characters, including embedded spaces. Trailing spaces are used to make the field 9 characters wide. These are not part of the name. Object names are case-sensitive.

The ; is the APRS Data Type Identifier for an Object Report, and a * or _ separates the Object name from the rest of the report:

- * indicates a live Object.
- _ indicates a killed Object.

- The position may be in lat/long or compressed lat/long format, and the reportmay also contain Extended Data. Compressed Objects are not recommended for use on RF due to incompatibilities. For 1 foot precision, use the !DAO! format.

An Object always has a timestamp.

The Comment field may contain any appropriate APRS data (see the *Comment Field* section in Chapter 5: APRS Data in the AX.25 Information Field).

Why is the Comment field restricted to 36 or 43 characters? Chapter 3 states that the Information part can be up to 256 characters.

APRS 1.1: The compressed version is not recommended.

Object Report Format — with Lat/Long position

	;	Object Name	* or _	Time DHM / HMS	Lat	Sym Table ID	Long	Symbol Code	Course/Speed Power/Height/Gain/Dir Radio Range DF Signal Strength Area Object	Comment (max 36 chars with Data Extension, or 43 without)
Bytes:	1	9	1	7	8	1	9	1	7	0-36/43

Examples

;LEADER˽˽˽*092345z4903.50N/07201.75W>088/036	A live Object. At 2345 hours zulu on the 9th of the month, the "Leader" was in the car at 49°3'30"N/72°1'45"W, heading 88 deg at 36 knots.
;LEADER˽˽˽_092345z4903.50N/07201.75W>088/036	The same Object, now killed.

Object Report Format — with Compressed Lat/Long position - not recommended.

	;	Object Name	* or _	Time DHM / HMS	Compressed Position Data /YYYYXXXX$csT	Comment
Bytes:	1	9	1	7	13	43

Example

;LEADER˽˽˽*092345z/5L!!<*e7>7P[	The "Leader" was in the car at 49°30'00"N/ 72°45'00"W, heading 88 deg at 36.2 knots.

Item Report Format (not recommended)

An Item Report has a *variable-length* Item name, 3–9 characters long. The name may consist of any printable ASCII characters, including embedded space, *except* **!** or **_** because they are the field terminator.

Item names are case-sensitive.

Note that the name field is variable length, rather than fixed, as with objects.

The **)** is the APRS Data Type Identifier for an Item Report, and a **!** or **_** separates the Item name from the rest of the report:

- **!** indicates a live Item.
- **_** is the Item "kill" character.

The position may be in lat/long or compressed lat/long format. There is no provision for a timestamp. The report may also contain Extended Data.

The Comment field may contain any appropriate APRS data (see the *Comment Field* section in Chapter 5: APRS Data in the AX.25 Information Field).

APRS 1.1: The Item Format is not recommended on RF due to incompatibilities.

Item Report Format — with Lat/Long position - Not Recommended

	)	***Item Name***	**!** or **_**	***Lat***	***Sym Table ID***	***Long***	***Symbol Code***	***Course/Speed*** ***Power/Height/Gain/Dir*** ***Radio Range*** ***DF Signal Strength*** ***Area Object***	***Comment (max 36 chars with Data Extension, or 43 without)***
Bytes:	1	3-9	1	8	1	9	1	7	0-36/43

<u>Examples</u>

`)AID  #2!4903.50N/07201.75WA`	First Aid Station #2 is at 49°3'30"N/72°1'45"W. (**/A** is the symbol for Aid Station).
`)G/WB4APR!53␣␣.␣␣N\002␣␣.␣␣Wd`	A rare DX station "somewhere in England". (**\d** is the symbol for DX Spot).
`)AID␣#2_4903.50N/07201.75WA`	The First Aid Station has closed down.

Item Report Format — with Compressed Lat/Long position - Not Recommended

	)	***Item Name***	**!** or **_**	***Compressed Position Data*** `/YYYYXXXX$csT`	***Comment***
Bytes:	1	3-9	1	13	43

<u>Example</u>

`)MOBIL!\5L!!<*e79  sT`	Mobil Gas Station is at 49°30'00"N/72°45'00"W. (**\9** is the symbol for Gas Station).

Area Objects

Using the \ℓ symbol (i.e. the lower-case letter "L" symbol from the Alternate Symbol Table) it is possible to define circle, line, ellipse, triangle and box objects in all colors, either open or filled in, any size from 60 feet to 100 miles.

These Objects are useful for real-time events such as for a search-and-rescue, or adding a special road or route for a special event.

The Object format is specified as a 7-character APRS Data Extension `Tyy/Cxx` immediately following the ℓ Symbol Code. For example:

```
;OBJECT___*ddmm.hhN\dddmm.hhW ℓ Tyy/Cxx
```

where:

`T` is the type of object shape.

`/C` is the color of the object.

`yy` is the square root of (the latitude offset in degrees / 1500).

`xx` is the square root of (the longitude offset in degrees / 1500).

On the receiving end:

offset_in_degrees = (offset_in_packet^2) / 1500

NOTE: This specification originally had a scaling factor of 100. http://www.aprs.org/aprs11/areaobjects.txt contains the correction to make it 1500. All of the modern applications checked use 1500.

The object type and color codes are as follows:

T	*Object Type*
0	Open circle
1	Line (offset: down/right)
2	Open ellipse
3	Open triangle
4	Open box
5	Color-filled circle
6	Line (offset: down/left)
7	Color-filled ellipse
8	Color-filled triangle
9	Color-filled box

/C	*Object Color*	*Intensity*
/0	Black	High
/1	Blue	High
/2	Green	High
/3	Cyan	High
/4	Red	High
/5	Violet	High
/6	Yellow	High
/7	Gray	High
/8	Black	Low
/9	Blue	Low
10	Green	Low
11	Cyan	Low
12	Red	Low
13	Violet	Low
14	Yellow	Low
15	Gray	Low

- Triangles are always isosceles triangles, oriented vertically and the points of reference are the top APEX and the lower right corner.

- Circles are defined by a square box referenced by the upper left corner and the center.

- Lines are referenced to the upper end.

The 'offset reference' position of the object is the upper left corner of the object and the offsets are the distance from the lower right corner (or center of a circle) back to this "offset reference" position. (An exception is the special case of a type-6 line which is drawn down and to the left).

Here are some examples of Object Position Reports. The latitude and longitude offsets are each 4 minutes = 4/60 degree = 100/1500 of a degree), so

yy = xx = √100 = 10.

`;SEARCH␣␣␣*092345z4903.50N\07201.75Wℓ710/310`

A high intensity cyan filled ellipse, yy=10, xx=10

`;SEARCH␣␣␣*092345z4903.50N\07201.75Wℓ8101310`

A low intensity violet filled triangle, yy=10, xx=10

Further, with the line option (Type 1 and Type 6) it is possible to specify a "corridor" either side of the central line. The width of the corridor (in miles) either side of the line is specified in the comment text, enclosed by **{}**.

For example:

`;FLIGHTPTH*4903.50N\07201.75W ℓ 610/310{100}`

A high intensity cyan line, with a 100-mile corridor either side

Note: The color fill option should be used with care, since a color-filled object will obscure information displayed underneath it.

Signpost Objects/Items

Signpost Objects/Items (with the symbol **\m**) display as a yellow box with a 1–3-character overlay on them. The overlay is specified by enclosing the 1–3 characters in braces in the comment field. Thus a signpost with {55} would appear as a sign with **55** on it.

For example:

`)I91␣3N!4903.50N\07201.75Wm{55}`

This was originally designed for posting the speed of traffic past speed measuring devices, but can be used for any purpose.

Implementation Recommendation: Signposts should not display any callsign or name, and to avoid clutter should only be displayed at close range.

.

Obsolete Object Format

Some stations transmit Object reports without the **;** APRS Data Type Identifier. This format is obsolete. Some software may still decode such dataas an Object, but it should now be interpreted as a Status Report.

12 WEATHER REPORTS

Weather Report Types

APRS is an ideal tool for reporting weather conditions via packet. APRS is also ideally suited for the Skywarn weather observer initiative.

APRS supports three types of Weather Report:

- Raw Weather Report (**not recommended -** sending system should reformat data into Complete Weather Report format)
- Positionless Weather Report (**not recommended** - sending system should reformat data into Complete Weather Report format)
- Complete Weather Report

Data Type Identifiers

The following APRS Data Type Identifiers are used in Weather Reports containing raw data:

!	Ultimeter 2000
#	Peet Bros U-II
$	Ultimeter 2000
*	Peet Bros U-II
_	Positionless weather data (not recommended)

In addition, where the raw data has been post-processed (for example, by the insertion of station location information), the four position Data Type Identifiers !, =, / and @ may be used instead. In this case, the Weather Report is identified with the weather symbol /_ or _ in the APRS Data.

Raw Weather Reports (not recommended)

Raw weather data from a stand-alone weather station is contained in theInformation Field of an APRS AX.25 frame.

Raw Weather Formats are not recommended. Sending stations should convert to *complete* format on RF.

	Raw Weather Report Format - Not recommended	
	! *or* # *or* $ *or* *	***Raw Weather Data***
Bytes:	1	n

Examples

`!!006B005803500000----03E9--------002105140000005D`	Ultimeter 2000
`#50B7500820082`	Peet Bros U-II
`$ULTW0031003702CE0069----000086A00001----011901CC00000005`	Ultimeter 2000
`*7007600000000`	Peet Bros U-II

Positionless Weather Reports (not recommended)

Generic raw weather data from a stand-alone weather station is contained in the Information Field of an APRS AX.25 frame.

The WinAPRS "positionless" weather format is not recommended.

Positionless Weather Report Format - Not recommended					
	_	***Time MDHM***	***WeatherData***	***APRS Software*** S	***WX Unit*** uuuu
Bytes:	1	8	n	1	2-4
	Example `_10090556c220s004g005t077r000p000P000h50b09900wRSW` report derived from Radio Shack WX station data.				

APRS Software Type

A Weather Report may contain a single-character code `S` for the type of APRS software that is running at the weather station:

- **d** = APRSdos
- **M** = MacAPRS
- **P** = pocketAPRS
- **S** = APRS+SA
- **W** = WinAPRS
- **X** = X-APRS (Linux)

Weather Unit Type

A Weather Report may contain a 2–4 character code `uuuu` for the type of weather station unit. The following codes have been allocated:

- **Dvs** = Davis
- **HKT** = Heathkit
- **PIC** = PIC device
- **RSW** = Radio Shack
- **U-II** = Original Ultimeter U-II (auto mode)
- **U2R** = Original Ultimeter U-II (remote mode)
- **U2k** = Ultimeter 500/2000
- **U2kr** = Remote Ultimeter logger
- **U5** = Ultimeter 500
- **Upkm** = Remote Ultimeter packet mode
- O = Otracker
- K = Kenwood
- B = Byonics
- Y = Yaesu

Users may specify any other 2–4 character code for devices not in this list.

Weather Data

The format of weather data within a Weather Report differs according to the type of weather station unit, but generically consists of someor all of the following elements:

Weather Data									
	Wind Direction `cccc`	***Wind Speed*** `ssss`	***Gust*** `gggg`	***Temp*** `tttt`	***Rain Last Hr*** `rrrr`	***Rain Last 24 Hrs*** `pppp`	***Rain Since Midnight*** `PPPP`	***Humidity*** `hhh`	***Barometric Pressure*** `bbbbbb`
Bytes:	4	4	4	4	4	4	4	3	6

where: **c** = wind direction (in degrees).
s = sustained one-minute wind speed (in mph).
g = gust (peak wind speed in mph in the last 5 minutes).
t = temperature (in degrees Fahrenheit). Temperatures below zero are expressed as -01 to -99.
r = rainfall (in hundredths of an inch) in the last hour.
p = rainfall (in hundredths of an inch) in the last 24 hours.
P = rainfall (in hundredths of an inch) since midnight.
h = humidity (in %. `00` = 100%).
b = barometric pressure (in tenths of millibars/tenths of hPascal).

Other parameters that are available on some weather station units include:

L = luminosity (in watts per square meter) 000 to 999.
ℓ (lower-case letter "L") = luminosity (in watts per square meter) 1000 and above. (Actual value is 1000 more than 3 digit number.)
(L is inserted in place of one of the rain values).
s = snowfall (in inches) in the last 24 hours. 3 digits.
A decimal point is allowed for non-integer values.
= raw rain counter

Note: The weather report must include at least the MDHM date/timestamp, wind direction, wind speed, gust and temperature, but the remaining parameters may be in a different order (or may not even exist).

Note: Where an item of weather data is unknown or irrelevant, its value may be expressed as a series of dots or spaces. For example, if there is no wind speed/direction/gust sensor, the wind values could be expressed as:

`c...s...g...` or `c␣␣␣s␣␣␣g␣␣␣`

For example, Jim's rain gauge may produce a report like this:

```
_10090556c...s...g...t...P012Jim
```

(The date/timestamp, wind direction/speed/gust and temperature parameters must be included, even though they are not meaningful).

Location of a Raw and PositionlessWeather Stations

APRS cannot display weather data on a map until it knows the location of the sending station. In the case of a station transmitting Raw or Positionless Weather Reports, the station has to occasionally send an additional packet containing its position (using any of the legal lat/long and compressed lat/long position formats described earlier).

Raw Weather Formats not recommended. Microprocessors should convert to *complete* format on RF.

Symbols with Raw and PositionlessWeather Stations

Because Raw and Positionless Weather Reports do not contain a display symbol in the AX.25 Information field, it is possible to specify the symbol in a generic APRS destination address (e.g. `GPSHW` or `GPSE63`) instead. Alternatively, if the weather station is on a balloon, the SSID –11 may be used in the source address (e.g. `N0QBF-11`).

Raw Weather Formats not recommended. Sending stations should convert to *complete* format on RF.

See Chapter 20: APRS Symbols for more detail on the usage of symbols.

Complete Weather Reports with Timestamp and Position

An APRS Complete Weather Report can contain a timestamp and location information, using any of the legal lat/long and compressed lat/long position formats described earlier. An APRS Object may also have weather information associated with it.

Examples of report formats are shown below. Note that the Symbol Code in every case is the _ (underscore). Also, the 7-byte Wind Direction and Wind Speed Data Extension replace the `cccc` and `ssss` fields of a Positionless Weather Report.

Complete Weather Report Format — with Lat/Long position, no Timestamp									
	! or =	Lat	Sym Table ID	Long	Symbol Code _	Wind Directn/ Speed	Weather Data	APRS Software `s`	WX Unit `uuuu`
Bytes:	1	8	1	9	1	7	n	1	2-4

Examples

```
!4903.50N/07201.75W_220/004g005t077r000p000P000h50b09900wRSW
!4903.50N/07201.75W_220/004g005t077r000p000P000h50b. ...wRSW
```

Complete Weather Report Format — with Lat/Long position and Timestamp

	/ or @	Time DHM / HMS	Lat	Sym Table ID	Long	Symbol Code _	Wind Directn/ Speed	Weather Data	APRS Software S	WX Unit uuuu
Bytes:	1	7	8	1	9	1	7	n	1	2-4

Example

```
@092345z4903.50N/07201.75W_220/004g005t-07r000p000P000h50b09900wRSW
```

Complete Weather Report Format — with Compressed Lat/Long position, no Timestamp

	! or =	Sym Table ID	Comp Lat YYYY	Comp Long XXXX	Symbol Code _	Comp Wind Directn/ Speed	Comp Type T	Weather Data	APRS Software S	WX Unit uuuu
Bytes:	1	1	4	4	1	2	1	n	1	2-4

Example

```
=/5L!!<*e7>_7P[g005t077r000p000P000h50b09900wRSW
```

Complete Weather Report Format — with Compressed Lat/Long position, with Timestamp

	/ or @	Time DHM / HMS	Sym Table ID	Comp Lat YYYY	Comp Long XXXX	Symbol Code _	Comp Wind Directn/ Speed	Comp Type T	Weather Data	APRS Software S	WX Unit uuuu
Bytes:	1	7	1	4	4	1	2	1	n	1	2-4

Example

```
@092345z/5L!!<*e7_7P[g005t077r000p000P000h50b09900wRSW
```

Complete Weather Report Format — with Object and Lat/Long position

	;	Object Name	*	Time DHM / HMS	Lat	Sym Table ID	Long	Symbol Code _	Wind Directn/ Speed	Weather Data	APRS Software S	WX Unit uuuu
Bytes:	1	9	1	7	8	1	9	1	7	n	1	2-4

Examples

```
;BRENDA˽˽˽*4903.50N/07201.75W_220/004g005t077r000p000P000h50b09900wRSW
;BRENDA˽˽˽*092345z4903.50N/07201.75W_220/004g005b0990
```

Storm Data

APRS reports can contain data relating to tropical storms, hurricanes and tropical depressions. The format of the data is as follows:

	Storm Data									
	Direction	**/**	***Speed***	***Storm Type*** /ST	***Sustained Wind Speed*** /www	***Peak Wind Gusts*** ^GGG	***Central Pressure*** /pppp	***Radius Hurricane Winds*** >RRR	***Radius Tropical Storm Winds*** &rrr	***Radius Whole Gale*** %ggg
Bytes:	3	1	3	3	4	4	5	4	4	4

where: ST = **TS** (Tropical Storm)
HC (Hurricane)
TD (Tropical Depression).

www = sustained wind speed (in knots).

GGG = gust (peak wind speed in knots).

pppp = central pressure (in millibars/hPascal)

RRR = radius of hurricane winds (in nautical miles).

rrr = radius of tropical storm winds (in nautical miles).

ggg = radius of "whole gale" (i.e. 50 knot) winds (in nautical miles). Optional.

Storm data will usually be included in an Object Report, but may also be included in a Position Report or an Item Report.

The display symbol will be either:

\@ Hurricane/Tropical Storm (current position)
/@ Hurricane (predicted future position)

For example, the progress of Hurricane Brenda could be expressed in Object Reports like these:

```
;BRENDA     *092345z4903.50N\07202.75W@088/036/HC/150^200/0980>090&030%040
;BRENDA     *100045z4905.50N/07201.75W@101/047/HC/104^123/0980>065&020%040
```

National Weather Service Bulletins

APRS supports the dissemination of National Weather Service bulletins. See Chapter 14: Messages, Bulletins and Announcements.

13 TELEMETRY DATA

Telemetry Report Format

The AX.25 Information field can contain telemetry data. The APRS Data Type Identifier is **T**.

The report Sequence Number is a 3-character value — typically a 3-digit number, or the three letters **MIC**. In the case of **MIC**, there may or may not be a comma preceding the first analog data value.

There are five 8-bit unsigned analog data values (expressed as 3-digit decimal numbers in the range 000–255), followed by a single 8-bit digital data value (expressed as 8 bits, each containing **1** or **0**).

The Kantronics KPC-3+ TNC and APRS Micro Interface Module (MIM) use this format.

In version 1.2, this was relaxed to allow values in the range of 000-999.

In reality, the restriction of fixed width 3 digits is often ignored. It is common to see much longer variable width values including decimal points, optionally preceded by a minus sign. Most modern applications recognize this more convenient format. Others are encouraged to do the same.

Telemetry Report Format

	T	Sequence No	Analog Value 1	Analog Value 2	Analog Value 3	Analog Value 4	Analog Value 5	Digital Value	Comment
	T	#xxx,	aaa,	aaa,	aaa,	aaa,	aaa,	bbbbbbbb	
Bytes:	1	5	4	4	4	4	4	8	n

Examples
```
T#005,199,000,255,073,123,01101001
T#MIC199,000,255,073,123,01101001
T#151,45.7,2.3,190.0,91.0,-7.3,00001100
```

APRS Base91 Comment Telemetry

The new Base91 Comment Telemetry extension MAY appear in the comment field of any of the three position packet formats ("Normal" uncompressed, Mic-E, and compressed). (Why not in an Object? Intentional exclusion or oversight?)

The Base91 Telemetry extension, if present, MUST appear after the free-form comment text entered by the end-user, but before any DAO or Mic-E type codes. The DAO extension MAY appear after the Base91 Telemetry extension. When the Mic-E Type Code is used, it must appear in the end of the packet.

Base91 telemetry is delimited at both ends by the '|' (pipe / vertical bar) character.

The telemetry sequence counter and telemetry channels are encoded using the Base91 encoding, as it is presently used in Compressed APRS position packets, the altitude extension, and the DAO extension.

Two bytes are transmitted for the sequence counter and each of the channels, giving over 13 bits of resolution (values 0 to 8280). Please note that APRS uses a different definition of Base91 than the internet standard Base91.

While the Base91 encoding provides more resolution and a larger sequence counter range, the transmitting station may use whatever resolution is available from the sensors. Values of 0 to 255 are fine for 8-bit A/D converters - upscaling to 0...8280 is not necessary.

The telemetry sequence counter MAY wrap from 255 to 0 if memory constraints require using a 1-byte variable for storing the counter. Please make sure that it and all of the telemetry values never get values higher than 8280. For example, the sequence number can be safely incremented with:

```
sequence = (sequence + 1) & 0x1FFF;
```

This will make it wrap to 0 after 8191, which will provide plenty of range.

The extension MUST contain at least a sequence counter and one channel of telemetry.

The extension MAY contain up to 5 channels of "analog values" and one 8-bit channel of "binary values", as in the traditional telemetry format.

If binary values are transmitted, they MUST appear last in the extension, after all 5 "analog" channels. They are put into a single Base91 encoded integer, where the LSB (least significant bit) corresponds to B1 of the traditional Telemetry specification, the 8th bit corresponds to B8. Bits 9 to 13 are reserved to future use and will not currently be treated as additional binary values.

Examples of valid Base91 telemetry formats:

```
|ss11|
|ss112233|
|ss1122334455!"|
```

Where ss is the sequence counter, 11 is the first channel, and so on. The '!"' in the end would be the binary values. These examples, while useful for demonstration, would also parse correctly.

```
Sequence:  Base91 'ss' decodes to 7544
Channel 1: Base91 '11' decodes to 1472
Channel 2: Base91 '22' decodes to 1564
Channel 3: Base91 '33' decodes to 1656
Channel 4: Base91 '44' decodes to 1748
Channel 5: Base91 '55' decodes to 1840

Binary values:    '!"' decodes to decimal 1, binary values 10000000,
                  B1 is 1, B2 to B8 are 0.
```

The following minimal telemetry extension has a sequence number of 0, and Channel 1 value of 0:

```
|!!!!|
```

On-Air Definition of TelemetryParameters

In principle, received telemetry data may be interpreted in any appropriate way. In practice, however, an APRS user can define the telemetry parameters (such as quadratic coefficients for the analog values, or the meaning of the binary data) at any time, and then send these definitions as APRS messages. Other stations receiving these messages will then know how to interpret the data.

This is achieved by sending four messages:

- A Parameter Name message.
- A Unit/Label message.
- An Equation Coefficients message.
- A Bit Sense/Project Name message.

The message addressee is the callsign of the station transmitting the telemetry data. For example, if N0QBF launches a balloon with the callsign N0QBF-11, then the four messages are addressed to N0QBF-11.

See Chapter 14: Messages, Bulletins and Announcements for full details of message formats.

Parameter Name Message

The Parameter Name message contains the names (N) associated with the five analog channels and the 8 digital channels. Its format is as follows:

Telemetry Parameter Name Message Data

Note the different byte counts, which include commas where shown. The list may stop at any field.

		A1	A2	A3	A4	A5	B1	B2	B3	B4	B5	B6	B7	B8
	PARM.	N...	,N...	,N...	,N...	,N...	,N...	,N...	,N...	,N...	,N...	,N...	,N...	,N...
Bytes:	5	1-7	1-7	1-6	1-6	1-5	1-6	1-5	1-4	1-4	1-4	1-3	1-3	1-3

Example
`:N0QBF-11 :PARM.Battery,Btemp,ATemp,Pres,Alt,Camra,Chut,Sun,10m,ATV`

Historical Note: The field widths are not all the same (this is a legacy arising from earlier limitations in display screen width). Note also that the byte counts*include* the comma separators where shown.

The list can terminate after any field.

Compatibility note: Many implementations ignore these overly restrictive inconsistent name lengths. It is not uncommon to see names with a dozen characters or more. New/upgraded applications should handle what is in common use.

Unit/Label Message

The Unit/Label message specifies the units (U) for the analog values, and the labels (L) associated with the digital channels:

Telemetry Unit/Label Message Data

Note the different byte counts, which include commas where shown. The list may stop at any field.

		A1	A2	A3	A4	A5	B1	B2	B3	B4	B5	B6	B7	B8
	UNIT.	U...	,U...	,U...	,U...	,U...	,L...	,L...	,L...	,L...	,L...	,L...	,L...	,L...
Bytes:	5	1-7	1-7	1-6	1-6	1-5	1-6	1-5	1-4	1-4	1-4	1-3	1-3	1-3

Example
`:N0QBF-11 :UNIT.v/100,deg.F,deg.F,Mbar,Kft,Click,OPEN,on,on,hi`

Historical Note: Again, the field widths are not all the same, and the byte counts *include* the comma separators where shown.

The list can terminate after any field.

Compatibility note: Many implementations ignore these overly restrictive inconsistent unit lengths. It is not uncommon to see much longer unit descriptions. New/upgraded applications should handle what is in common use.

Equation Coefficients Message

The Equation Coefficients message contains three coefficients (`a`, `b` and `c`) for each of the five analog channels.

Telemetry Equation Coefficients Message Data The list may stop at any field. Value = $a \times v^2 + b \times v + c$																
		A1			***A2***			***A3***			***A4***			***A5***		
	EQNS.	a	,b	,c	,a	,b	,c	,a	,b	,c	,a	,b	,c	,a	,b	,c
Bytes:	5	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n

Example
`:N0QBF-11␣:EQNS.0,5.2,0,0,.53,-32,3,4.39,49,-32,3,18,1,2,3`

To obtain the final value of an analog channel, these coefficients are substituted into the equation:

$$a \times v^2 + b \times v + c$$

where v is the raw received analog value.

For example, analog channel A1 in the above beacon examples relates to the battery voltage, expressed in hundredths of volts, and a = 0, b = 5.2, c = 0. If the raw received value v is 199, then the voltage is calculated as:

$$\text{voltage} = 0 \times 199^2 + 5.2 \times 199 + 0$$
$$= 1034.8 \text{ hundredths of a volt}$$
$$= 10.348 \text{ volts}$$

Bit Sense/Project Name Message

The Bit Sense/Project Name message contains two types of information:

- An 8-bit pattern of ones and zeros, specifying the sense of each digital channel that matches the corresponding label.
- The name of the project associated with the telemetry station.

Telemetry Bit Sense/Project Name Message Data										
	BITS.	***B1*** x	***B2*** x	***B3*** x	***B4*** x	***B5*** x	***B6*** x	***B7*** x	***B8*** x	***Project Title***
Bytes:	5	1	1	1	1	1	1	1	1	0-23

Example
`:N0QBF-11␣:BITS.10110000,N0QBF's Big Balloon`

Thus in the above message examples, if digital channel B1 is 1, this indicates the camera has clicked. If channel B2 is 0, the parachute has opened, and so on.

14 MESSAGES, BULLETINS AND ANNOUNCEMENTS

APRS messages, bulletins and announcements are packets containing free format text strings, and are intended to convey human-readable information. A message is intended for reception by a single specified recipient, and an acknowledgement is usually expected. Bulletins and announcements are intended for reception by multiple recipients, and are not acknowledged.

Messages

An APRS message is a text string with a specified addressee. The addressee is a fixed 9-character field (padded with spaces if necessary) following the **:** Data Type Identifier. The addressee field is followed by another **:**, then the text of the message.

The message text may be up to 67 characters long, and may contain any printable ASCII or UTF-8 characters except **{** which indicates start of Message Identifier.

What is the reason for that length restriction? Chapter 3 stated that the Information part can be up to 256 characters. An article from around the turn of the century mentioned that the Kenwood TH-D7 had a message text limit of 45 characters and the TM-D700 had a limit of 67 characters. Was the spec based on an implementation limit of one product, now long discontinued? In reality, we sometimes see Message Text longer than 67 so receiving applications should be able to handle that.

A message may also have an optional message identifier, which is appended to the message text. The message identifier consists of the character **{** followed by a message number (up to 5 alphanumeric characters, no spaces) to identify the message.

Messages *without* a message identifier are not to be acknowledged.

Messages *with* a message identifier are intended to be acknowledged by the addressee. The sending station will repeatedly send the message until it receives an acknowledgement or it is canceled, or it times out.

Message Format

					Message ID	
	:	***Addressee***	**:**	***Message Text (max 67 chars)***	**{**	***Message No*** xxxxx
Bytes:	1	9	1	0-67	1	1-5

Examples

`:WU2Z␣␣␣␣␣:Testing` A message for WU2Z, containing the text "Testing", no acknowledgement expected. (Note the filler spaces in the 9-character addressee field).

`:WU2Z␣␣␣␣␣:Testing{003` The same message, Message No=003, acknowledgement expected.

`:EMAIL␣␣␣␣␣:msproul@ap.org Test email` An e-mail message (Note: This is an example of how such a message could be constructed. APRS itself does not support e-mail delivery)

Message Acknowledgement

A message acknowledgement is similar to a message, except that the message text field contains just the letters **ack**, and this is followed by the Message Number being acknowledged.

Message Acknowledgement Format				
:	***Addressee***	**:**	**ack**	***Message No*** xxxxx
1	9	1	3	1–5

Bytes: 1, 9, 1, 3, 1–5

Example
`:KB2ICI-14:ack003`

A station should respond to a message only when it is the addressee. (i.e. Do not respond to bulletins or announcements.)

Message Rejection

If a station is unable to accept a message, addressed to itselft, it can send a **rej** message insteadof an **ack** message:

Message Rejection Format				
:	***Addressee***	**:**	**rej**	***Message No*** xxxxx
1	9	1	3	1–5

Bytes: 1, 9, 1, 3, 1–5

Example
`:KB2ICI-14:rej003`

A station should respond to a message only when it is the addressee. (i.e. Do not respond to bulletins or announcements.)

Multiple Acknowledgements

If a station receives a particular message more than once, it will respond with an acknowledgement for each instance of the message.

If a station receives a message over a long path, it may respond with a reasonable number of multiple copies of the acknowledgement, to improve the chances of the originating station receiving at least one of the copies.

In either of these two situations, multiple message acknowledgements should be separated by at least 30 seconds (this is because some network components such as digipeaters will suppress duplicated messages within a 30-second period).

New Message Number Format (December 1999)

The original message number format was 1 to 5 alphanumeric characters. A newer format allows more efficient conversation between two stations by including an ACK within an outgoing message.

This is 100% backwards compatible with all code. The REPLY ACKS are embedded in the 5 digit line number. Old code doesn't care.

The format for the line number for outgoing message numbers is "{MM}AA" where MM is the outgoing message number and AA is the "free ACK" if needed. If no ACK is pending, then the message # is "{MM}".

The non-base91 character "}" was chosen as the separator so that there is no chance that an existing line number could be misinterpreted. Notice, that even if there is no ACK, the presence of the trailing "}" tells the other end that the sender is REPLY-ACK capable and can accept a REPLY-ACK.

This is actually very simple to implement, but it does affect several areas of your code:

1. Send all MSG numbers in the new {MM}AA format. AA is null if none. But AA is only appended at the INSTANT of transmission. For the same MM message, the AA must always represent the "latest" outstanding ACK for that station. It may change at future retries...
2. RECEIVE messages and look for AA. IF AA matches the MM of one of your outgoing messages, then consider that message ACKED.
3. RECEIVE messages and Strip off the AA from the end of the message text before you store it. THis is necessary so that your DUPE checker will still work. Since the AA may be different for multiple copies of the same incoming MM...
4. When you receive a message line XXX.., send the normal existing "ackXXX..". This algoirithm is unchanged. Even if XXX.. is MM}AA then the ack is just the exact copy as before "ackMM}AA".
5. When you receive a message line MM}AA, do #4 above as normal, but also then save the incoming MM number as now it is your "latest ACK" for that station. Now then this becomes the AA REPLY-ACK number for your next outgoing message to that station. (Step #1 above).
6. Note, that in #1, above, that when the user prepares each message line, that it is buffered up with only the {MM} line number on the end. The AA (if pending) is not attached until the instant that packet is transmitted.
7. Thus, in step #2, when you get a new REPLY-ACK, then you simply match up the "AA" with the {MM} in your outgoing message queue. But if you get the old "ackMM}AA" ack, then you must pull out the "MM" here and use IT to match with the outstanding "{MM}" in your outgoing message queue.

Message Groups

An APRS receiving station can specify special Message Groups, containing lists of callsigns that the station will read messages from (in addition to messages addressed to itself). Such Message Groups are defined internally by the user at the receiving station, and are used to filter received message traffic.

The receiving station will read all messages with the Addressee field set to `ALL`, `QST` or `CQ`.

The receiving station will only acknowledge messages addressed to itself, and not any messages received which were addressed to any group callsign.

Note: The receiving station will acknowledge all messages addressed to itself, even if it is operating in an Alternate Net (see Chapter 4: APRS Data in the AX.25 Destination and Source Address Fields).

General Bulletins

General bulletins are messages where the addressee consists of the letters **BLN** followed by a single-*digit* bulletin identifier, followed by 5 filler spaces. General bulletins are generally transmitted a few times an hour for a few hours, and typically contain time sensitive information (such as weather status).

Bulletin text may be up to 67 characters long, and may contain any printable ASCII or UTF-8 characters. In reality, we often see Bulletin Text longer than 67 so receiving stations should be able to deal with that.

General Bulletin Format

	:	BLN	*Bulletin ID* n	␣␣␣␣	:	*Bulletin Text (max 67 characters)*
Bytes:	1	3	1	5	1	0-67

Example
`:BLN3␣␣␣␣:Snow expected in Tampa RSN`

Announcements

Announcements are similar to general bulletins, except that the letters **BLN** are followed by a single upper-case *letter* announcement identifier. Announcements are transmitted much less frequently than bulletins (but perhaps for several days), and although possibly timely in nature they are usually not time critical.

Announcements are typically made for situations leading up to an event, in contrast to bulletins which are typically used within the event.

Users should be alerted on arrival of a new bulletin or announcement.

Announcement Format

	:	BLN	*Announcement Identifier* x	␣␣␣␣	:	*Announcement Text (max 67 characters)*
Bytes:	1	3	1	5	1	0-67

Example
`:BLNQ␣␣␣␣:Mt St Helen digi will be QRT this weekend`

Group Bulletins

Bulletins may be sent to *bulletin groups*. A bulletin group address consists of the letters **BLN**, followed by a single-*digit* group bulletin identifier, followed in turn by the name of the group (up to 5 characters long, with filler spaces to pad the name to 5 characters).

In reality, bulletin text is often more than 67 characters so receiving applications should be ready for that.

Group Bulletin Format

	:	BLN	*Group Bulletin ID* n	*Group Name*	:	*Group Bulletin Text (max 67 characters)*
Bytes:	1	3	1	5	1	0-67

Example
`:BLN4WX␣␣␣␣:Stand by your snowplows` Group bulletin number 4 to the `WX` group. (Note the filler spaces in the group name).

A receiving station can specify a list of bulletin groups of interest. The list is defined internally by the user at the receiving station. If a group is selected from the list, the station will only copy bulletins for that group, plus any general bulletins. If the list is empty, all bulletins are received and generate alerts.

National Weather Service Bulletins

Standard APRS message formats can be used for a variety of other applications. For example, in the United States, special formatted messages addressed to the generic callsign **NWS-**xxxxx are used to highlight map areas involved in weather warnings, using the following format:

National Weather Service Bulletin Format

	:	NWS-xxxxx	:	*NWS Bulletin Text*
Bytes:	1	9	1	n

xxxxx is the severity such as ADVIS, WARN, WATCH, etc.

Example
`:NWS-WARN␣:092010z,THUNDER_STORM,AR_ASHLEY,{S9JbA`
(Note: The "message identifier" `{S9JbA` at the end is for reference only, as **receiving stations do not acknowledge bulletins**).

Note that there is no specific maximum text length limit as with other bulletins.

More Information about APRS NWS Weather Bulletins can be found at https://www.aprs-is.net/wx/ .

NTS Radiograms

APRS can be used to transport NTS radiograms. This uses the existing APRS message format for backwards compatibility, by adding a 3-character NTS format identifier `Nx\` at the start of the APRS Message Text, as follows:

`N#\`number`\`precedence`\`handling`\`originator`\`check`\`place`\`time`\`date
`NA\`address_line1`\`address_line2`\`address_line3`\`address_line4
`NP\`phone number
`N1\`line 1 of NTS message text
`N2\`line 2 of NTS message text
`N3\`line 3 of NTS message text
`N4\`line 4 of NTS message text
`N5\`line 5 of NTS message text
`N6\`line 6 of NTS message text
`NS\`Signature block
`NR\`Received from`\`date_time`\`sent_to`\`date_time

All of these fields comes from the ARRL NTS Radiogram form and are described in the NTS handbook.

Each message line is addressed to the same station.

The `N#\`, `NA\` and `NR\` lines are multiple fields combined for APRS transmission efficiency. The backslash separator is used so that conventional forward slashes may be embedded in messages. (The backslash does not exist in the RTTY or CW alphabets, so it therefore cannot appear in an NTS radiogram).

Each line may be up 67 characters long, including the 3-character NTS format identifier. Lines in excess of 67 characters will be truncated.

There is a maximum of 6 lines of NTS message text.

Note: The `N#\`, `NA\`, `NS\` and `NR\` fields are required. The others are optional.

Serialization of each line is handled by the normal APRS Message ID `{`xxxxx.

An APRS application is not required to understand or generate these messages. The information can be read and understood in the normal message display.

Obsolete Bulletinand Announcement Format

Some stations transmit bulletins and announcements without the `:` APRS Data Type Identifier. This format is obsolete. Some software may still decode such data as a bulletin or announcement, but it should now be interpreted as a Status Report.

Bulletin and Announcement Implementation Recommendations

Bulletins and announcements are seen as a way for all participants in an event/emergency/net to see all common information posted to the group. In this sense they are visualized as a mountain-top billboard or a bulletin board on the wall of an Emergency Operations Control Center.

Information that everyone must see is to be posted there. Information is added and removed. Space is limited. Only so much information can be posted before it becomes too busy for anyone to see everything. Thus things are supposed to be posted, updated, and cleared to keep the big billboard uncluttered and current with what everyone needs to know at the present time. It should not be cluttered with obsolete information.

This can be implemented in an APRS display system as a “Bulletin Screen”. Everyone has this screen, and anyone can post or update lines on this screen. At any instant, everyone in the network sees exactly the same screen. Everything is arranged and displayed in exactly the same way. Thus, everyone, everywhere is looking at the same mountain-top billboard or bulletin board. There is no ambiguity as to who sees what information, in what order at what time.

To make this work, a number of issues should be considered:

- **Sorting**: Bulletins/Announcements are almost always multi-line, and may arrive out of sequence. They must be sorted before presentation on the Bulletin Screen, and re-sorted if necessary when each new line arrives. This includes sorting by originating callsign and Bulletin/Announcement ID.
- **Replacement**: Stations sending bulletins/announcements can send new lines to replace lines sent earlier, re-using the original Bulletin/Announcement IDs. (Note: It is only necessary to re-send replacement lines. It is not necessary to re-send the whole bulletin/announcement). Receipt of a new line with the same Bulletin/Announcement ID as one already received from the same station should result in the existing line being overwritten (replaced).
- **Clearing**: A user should be able to clear any or all of the bulletins/announcements from the Bulletin Screen once he has read them. Any bulletins/announcements that are still valid will re-appear in due course because of the way they are redundantly re-transmitted.
- **Alerts**: On receipt of any new or replacement line for the Bulletin Screen, an alarm should be sounded and re-sounded periodically until the user acknowledges it. Thus, this vital information is “pushed” to the operator. There is no excuse for not being aware of the current bulletin/announcement state — this is important in the hurried and crisis-laden scenario of an APRS event.
- **Logging**: Old bulletins/announcements should be logged in sequential APRS log files in case they are subsequently needed.

15 STATION CAPABILITIES, QUERIES AND RESPONSES

Station Capabilities

A station may define a set of one or more attributes of the station, known as Station Capabilities. The station transmits its capabilities in response to an IGATE query (see below), using the **<** Data Type Identifier.

Each capability is a TOKEN or a TOKEN=VALUE pair. More than one capability may be on a line, with each capability separated by a comma.

Currently defined capabilities include:

```
IGATE,MSG_CNT=n,LOC_CNT=n
```

where `IGATE` defines the station as an IGate, `MSG_CNT` is the number of messages transmitted, and `LOC_CNT` is the number of "local" stations (those to which the IGate will pass messages in the local RF network).

Queries and Responses

There are two types of APRS queries. One is general to all stations and the other is in a message format directed to a single individual station.

Queries always begin with a **?**, are one-time transmissions, do not have a message identifier and should not be acknowledged. Similarly the responses to queries are one-time transmissions that also do not have a message identifier, so that they too are not acknowledged.

Each query contains a Query Type (in upper-case). The following Query Types and expected responses are supported:

Query Type	*Query*	*Response*
APRS	General — All stations query	Station's position and status
APRSD	Directed — Query an individual station for stations heard direct	List of stations heard direct
APRSH	Directed — Query if an individual station has heard a particular station	Position of heard station as an APRS Object, plus heard statistics for the last 8 hours
APRSM	Directed — Query an individual station for outstanding unacknowledged or undelivered messages	All outstanding messages for the querying station
APRSO	Directed — Query an individual station for its Objects	Station's Objects
APRSP	Directed — Query an individual station for its position	Station's position
APRSS	Directed — Query an individual station for its status	Station's status
APRST or **PING?**	Directed — Query an individual station for a trace (i.e. path by which the packet was heard)	Route trace
IGATE	General — Query all Internet Gateways	IGate station capabilities
WX	General — Query all weather stations	Weather report (and the station's position if it is not included in the Weather Report)

If a queried station has no relevant information to include in a response, it need not respond.

A queried station should ignore any query that it does not recognize.

General Queries

The format of a general query is as follows:

General Query Format								
				Target Footprint				
	?	Query Type	?	Lat	,	Long	,	Radius
Bytes:	1	n	1	n	1	n	1	4

Examples

Query	Typical Response
?APRS? General query, with standard posit and status reply.	/092345z4903.50N/07201.75W> >092345zNet Control Center
?APRS?␣34.02,-117.15,0200 General query for stations within a target footprint of radius 200 miles centered on 34.02 degrees north, 117.15 degrees west, with standard posit and status reply. (Note the leading space in the latitude, as its value is positive, see below).	/3402.78N11714.02W- >Digi on low power
?IGATE? General query for IGate stations, with a Station Capabilities reply.	<IGATE,MSG_CNT=43,LOC_CNT=14
?WX? Query for weather stations, with a standard Weather Report reply (without a position), followed by a standard posit.	_10090556c220s004g005t077… /090556z4903.50N/07201.75W>

In the case of an ?APRS? query for stations within a particular target footprint, the latitude and longitude parameters are in *floating point* degrees (*not* in APRS lat/long position format).

- North and east coordinates are positive values, indicated by a leading ␣ (space).
- South and west coordinates are negative values.
- The radius of the footprint is in miles, expressed as a fixed 4-digit number in whole miles.

All stations inside the specified coverage circle should respond with a Position Report and a Status Report.

Directed Station Queries

Queries addressed to individual stations are in APRS message format (except that they never include a message identifier). The addressee is the callsign of the station being queried.

The message text is the Query Type. This is followed optionally by another callsign — this callsign does not need filler spaces as it is at the end of the data.

	Directed Station Query Format					
	:	Addressee	:	?	Query Type	Callsign of Heard Station
Bytes:	1	9	1	1	5	0-9

Examples

Query	Typical Response
`:KH2Z˽˽˽˽˽:?APRSD` A query asking KH2Z what stations he has heard direct.	`:N8UR˽˽˽˽˽:Directs=˽WA1LOU˽WD5IVD…`
`:KH2Z˽˽˽˽˽:?APRSH  N0QBF    ` A query asking for the number of times N0QBF was heard in each of the last 8 hours. (Note the trailing spaces in the callsign following `APRSH`, padding the callsign to 9 characters).	`:N8UR˽˽˽˽˽:N0QBF˽HEARD:˽1˽3˽2˽. ˽. ˽4˽5˽6`
`:KH2Z˽˽˽˽˽:?APRSM` A query asking KH2Z for any unacknowledged or undelivered messages for him. KH2Z responds with all such messages.	`:N8UR˽˽˽˽˽:Testing{003`
`:KH2Z˽˽˽˽˽:?APRSO` A query asking for KH2Z's APRS Objects.	`;LEADER˽˽˽*092345z4903.50N/07201.75W>`
`:KH2Z˽˽˽˽˽:?APRSP` A query asking for KH2Z's position.	`/092345z4903.50N/07201.75W>`
`:KH2Z˽˽˽˽˽:?APRSS` A query asking for KH2Z's status.	`>092345zNet Control Center`
`:KH2Z˽˽˽˽˽:?APRST` A query asking KH2Z for a trace of the route taken to reach him.	`:N8UR˽˽˽˽˽:KH2Z>APxxxx,DIGI1,WIDE*:`
`:KH2Z˽˽˽˽˽:?PING?` The same query, using PING instead of APRST.	`:N8UR˽˽˽˽˽:KH2Z>APxxxx,DIGI1,WIDE*:`

16 STATUS REPORTS

A Status Report announces the station's current mission or any other single line status to everyone. The report is contained in the AX.25 Information field, and starts with the **>** APRS Data Type Identifier.

The report may optionally contain a timestamp.

Note: The timestamp can *only* be in DHM *zulu* format.

The status text occupies the rest of the Information field, and may be up to 62 characters long (if there is no timestamp in the report) or 55 characters (if there is a timestamp). The text may contain any printable ASCII or UTF-8 characters.

Status Report Format

	>	Time DHM z	Status Text (max 62 chars if no timestamp, or 55 chars if there is a timestamp)
Bytes:	1	7	0-62 or 0-55

Examples

`>Net Control Center`	without timestamp.
`>092345zNet Control Center`	with timestamp.

Although the status will usually be plain language text, there are two cases where the report can contain special information which can be decoded:

- Beam Heading and Power
- Maidenhead grid locator

Status Report with Beam Heading and Effective Radiated Power

It is useful to include beam heading and ERP in packets in meteor scatter work. To keep packets as short as possible, these parameters are encoded into two characters, as follows:

`H` = beam heading / 10
(`H`=0–9 for 0–90 degrees, and A–Z for 100–350 degrees).

`P` = ERP code.

ERP	P
10w	1
40	2
90	3
160	4
250	5
360	6
490	7
640	8
810	9

ERP	P
1000w	:
1210	;
1440	<
1690	=
1960	>
2250	?
2560	@
2890	A
3240	B

ERP	P
3610w	C
4000	D
4410	E
4840	F
5290	G
5760	H
6250	I
6760	J
7290	K

The `HP` value appears as the *last* two characters of the status text, preceded by the **^** character — for example, **^B7** means a beam heading of 110 degrees and an ERP of 490 watts.

The `HP` value may be combined with the Maidenhead grid locator (as described below), or with any other plain language status text.

Status Report with Maidenhead Grid Locator

The Maidenhead grid locator may be 4 or 6 characters long, and must immediately follow the **>** Data Type Identifier.

All letters must be transmitted in upper case. Letters may be received in upper case or lower case.

The Symbol Table Identifier and Symbol Code follow the locator.

If the report also contains status text, the first character of the text *must* be a space.

A Status Report with Maidenhead locator can not have a timestamp.

Status Report Format — with Maidenhead Grid Locator							
	Maidenhead Locator			***Sym Table ID***	***Symbol Code***		***Status Text (starting with a space) (max 54 chars)***
>	GG	nn	gg			␣	
Bytes: 1	2	2	2	1	1	1-54	

Examples

`>IO91SX/G`
`>IO91/G`
`>IO91SX/-␣My house` (Note the space ␣ at the start of the status text).
`>IO91SX/-␣^B7` Meteor Scatter beam heading = 110 degrees, ERP = 490 watts.

Transmitting Status Reports

Each station should only transmit a Status Report once every net cycle time (i.e. once every 10, 20 or 30 minutes), or in response to a query.

17 NETWORK TUNNELING AND THIRD-PARTY DIGIPEATING

Third-Party Networks

APRS provides a mechanism for encapsulation of packets which came from a different type of network. This is most often used when an IGate station transmits a packet which has traveled across APRS-IS. However, it is a more general feature that can be used in other situations.

Source Path Header

Prior to forwarding an APRS packet from some other network (e.g. APRS-IS) to RF, the gateway station adds a new header, the } data type indicator, and the original packet.

The prepended address path is known as the Source Path Header. It consists of the source (the gateway name), destination (software type of gateway), and digipeater addresses added by the gateway configuration.

The original sender is in the encapsulated original packet.

Data with Source Path Header

	Source Path Header	Data Type ID	Encapsulated Original Packet
Bytes:	n	1	N
		}	

Source Path Header - in TNC-2 format
An asterisk follows the digipeater callsign heard.

Source Callsign (-SSID)	>	Destination Callsign (-SSID)	0-8 Digipeaters: , Digipeater Callsign (-SSID)(*)	:
1-9	1	1-9	0-81	1

Example

```
WB4APR-14>APxxxx,WIDE1,WIDE2-1:
```

All of the addresses in the source path header must adhere to the AX.25 convention. i.e. 1 to 6 upper case letters and digits, numeric SSID in range 0 (not displayed) to 15. Example:

- WB4APR-14 is the gateway name.
- APxxxx is the system identifier for the gateway. This is useful for troubleshooting. Do not put APRS in this field.
- The gateway adds the desired digipeater via path.

Third-Party Header

The packet which emerged from another type of network contains the Third-Party Header and an Information part.

Original Packet in Third-party Format			
	}	Third-Party Header	Information part of the original data
Bytes:	1	n	n

The Third-Party Header must be in "TNC-2" format.

Third Party Header — "TNC-2" format						
	Original Source and Destination	,	Third-Party Network Identifier ("callsign")	,	Callsign of Receiving Gateway Station (-SSID)	*
Bytes:	3-99	1	1-9	1	1-9	1

Example
`WB4APR-14>APxxxx,TCPIP,G9RXG*:`

The source address in the encapsulated third-party packet is a character string and does not need to adhere to the AX.25 address restrictions. It can contain 1 to 9 printable ASCII characters, other than ">" which is the field terminator. Stations connected directly to APRS-IS, rather than going over RF, often have names with more than 6 characters or two alphanumeric characters for the SSID.

The original source address is retained.

The original destination field (device/system identifier or MIC-E) is retained.

The original digipeater via path field could be kept for special network diagnostic purposes but most of the time it is discarded.

In either case, these are appended to the digipeater via path:

- The Third-Party Network Identifier (e.g. `TCPIP`). This is a dummy "callsign" that identifies the nature of the third-party network.
- The callsign of the receiving gateway station, followed by an asterisk.

The original Information field must not be changed.

An Example of Sending a Message through the Internet

The Scenario:

- WB4APR-14 wants to send an APRS message to G3NRW. This can't be done over the local VHF network because they are in different continents.
-

  In the normal way, WB4APR-14 builds a message packet that contains:

  ```
  :G3NRW␣␣␣␣:Hi  Ian{001
  ```

- WB4APR-14 transmits the packet with his via path `WIDE1-1,WIDE2-1`.

  ```
  WB4APR-14>APxxxx,WIDE1-1,WIDE2-1::G3NRW␣␣␣␣:Hi   Ian{001
  ```

- The Internet gateway K4HG happens to receive this packet over the radio, either directly or after it has been retransmitted by a digipeater or two.
- IGate K4HG sends the packet to an APRS Tier 2 server. Rather than specifying a specific server, the IGate uses a rotate address such as noam.aprs2.net. This provides load balancing across redundant servers and resiliency if one happens to fail.
- The APRS-IS system forwards the message packet to any IGates, including G9RXG , that heard the message addressee (G3NRW) over the radio recently.
- IGate G9RXG encapsulates the message packet into a Third-Party packet:

  ```
  G9RXG>APxxxx,WIDE2-2:}WB4APR-14>APxxxx,TCPIP,G9RXG*::G3NRW␣␣␣␣:Hi    Ian{001
  ```

- IGate G9RXG transmits the packet over the local APRS network.
  - Digipeaters only look at the via path before the `}` data type indicator.
  - IGate stations will not send this to APRS-IS because TCPIP appears in the encapsulated packet's path. This is very important to prevent the same packet from looping around.
  - Other stations, that are interested in the packet contents, remove everything up to and including the `}` third party data type indicator
- Message addressee G3NRW receives the packet, removes the encapsulation, and discoversthat the packet contains a message for him.

  Note: The message sender address might not comply with AX.25 address rules. It could be longer than 6 alphanumeric characters, contain lower case letters, and/or have an SSID which is not numeric. Messaging applications need to handle this case.

  ```
  WB4APR-14>APxxxx,TCPIP,G9RXG*::G3NRW␣␣␣␣:Hi    Ian{001
  ```

- From the remaining header, G3NRW then sees that the message is from WB4APR-14 and sends an acknowledgement.

  ```
  G3NRW>APxxxx,WIDE1-1,WIDE2-1::WB4APR-14:ack001
  ```

- The process repeats in the opposite direction.

More details about the inner workings of an IGate can be found in the APRS-IS Specification:

Limiting Flow from RF to APRS-IS

- Packets with NOGATE or RFONLY, at in the digipeater field, should not be forwarded to APRS-IS by IGate stations. Obviously, you would want to put it at the end, not the beginning. The packet would never get digipeated if you put RFONLY at the beginning like this:

  RFONLY,WIDE1-1,WIDE2-1

  Digipeaters act on the first unused address so they would not see WIDE1-1..

- !x! means no archive. Any packet containing this string should not be archived by any of the APRS-IS data bases. 'x' is literally the lower case X, not a placeholder for any letter.

When receiving this type of packet

The processing of this type of packet depends on the type of station that receives it.

(1) A **digipeater** only looks at the first via path (before the "}") and doesn't need to have any knowledge of third party packets.

(2) An **application** that wants to interpret the information part of the packet must first remove the encapsulation and process what is left over.

(3) An **RF-to-IS IGate** first removes the encapsulation and operates on the rest. If the remaining via path contains TCPIP, the packet must not be forwarded. This prevents loops.

18 APRS FREQENCY SPECIFICATION

FREQUENCY IMPORTANCE IN APRS:

Ham radio's biggest advantage of thousands of frequencies is also its biggest stumbling block at rapidly and efficiently establishing communications under emergent or immediate need or just to chat on a long trip across country. APRS provides a way to determine the operating frequency of the other stations and applications around us or to send to a distant station, a desired contact frequency. This initiative came out of the Hurricane Katrina lessons learned.

APRS, is a single resource for identifying and locating amateur radio operators on both a local and global scale. Over 30,000 stations world wide are currently in the system and you can view them in any area or near any station via any APRS client program or via any number of APRS web pages such as the following:

http://aprs.fi OR
http://map.findu.com/callsign* ... and then select "stations near".
http://www.jfindu.net

FREQUENCY CATEGORIES: There are basically four categories of Frequency specifications in APRS:

- FREQ OBJECTS - originated by local digis. FREQ is the object name
- Fixed FREQ COMMENT - included in any fixed position or object comment
- FREQ STATUS - Frequency included in STATUS packets for NMEA trackers
- Auto FREQ INSERTION - automatically included in position comment

This APRS Frequency initiative is to standardize so that we know the correct frequency, Tone, offset, and maybe bandwidth. In addition, there are many other useful parameters that can be included in these beacons as well. These are the types of local objects that should show up on the mobile APRS display in every local area:

- Every mobile's (or fixed station's) operating frequency
- The locally recommended voice frequency for visitors
- The local IRLP or EchoLink node and current node status
- The local WinLink node and current node status
- The frequency of any Net in progress
- Any other local ham radio asset or net of interest to APRS mobiles

KEEP IT LOCAL! These local fixed FREQ OBJECTS should ONLY be transmitted in the local RF area where they can be immediately used. Transmitting them any farther than DIRECT will not only add QRM and congestion to the local channel they are also just SPAM beyond the local simplex range in areas that cannot use them. The sources of these frequencies are as follows:

- Newer Mobile radios put their operating frequency in their STATUS text (not to be confused with the APRS "STATUS" packet which is a separate packet and is not part of this FREQ spec.

- Local Voice Repeater objects are TX'ed by the local DIGI's TEXTS

- IRLP, EchLink and WinLink objects are generated by their own software and injected into the APRS-IS. From there, each LOCAL IGATE SYSOP decides which IRLP and which Echlink and which WinLINK object is in his immediate area, and he then adds these specific individual objects to his pass-to-RF list via no more than a ONE-HOP-PATH...

RECOMMENDED TRAVELERS VOICE REPEATERS:

Every digi has a coverage area. It is the responsiblity of that digi to transmit an object showing the best recommended travelers voice repeater in that coverage area, DIRECT, once-every-10-minutes. Being direct (no digipeats), they are only received in the vicinity where they are usable. Also, by originating at the Digipeater's high site, there is no impact on the channel load, because the digi will not transmit until the channel is clear. See:

http://aprs.org/localinfo.html

WOTA, DX CLUSTERS, KATRINA: This AFRS Frequency initiative is working in parallel with the post-Katrina ARRL Initiative to provide operating frequency contact information in support of emergency response.

Since the APRS-IS exists worldwide and already supplies information on all APRS stations, it can also accept data from WOTA, DX clusters and Logging programs to serve as a single resource for finding the likely operating frequency of interest for any station.

ADDING FREQUENCY TO APRS: Already, APRS encodes the following information into typical user position packets.

CALL, LAT, LONG, COURSE, SPEED (Grid is calculated)
Station type (one of over 200 symbols)
Antenna Height above average terrain
Antenna Gain
Comment (usually where stations have their frequency)
Software version

FREQUENCY has been added in a backwards compatible manner, either as a fixed formatted field in the first 10 bytes of the existing free-field position comment text or as the noun name of an object packet.

In addition to the frequency information (9 bytes of an OBJECT name) or 10 fixed format bytes in a position comment, there are additional optional format bytes to include additional amplifying information about tone, range, net and meeting times. This information has been carefully formatted to show up well on the 10x10 byte displays of the D7, D700's, the HAMHUD, Yaesu's and other existing devices. A typical entry might be:

"FFF.FFFMHz Tnnn +500 RXXm". The Tone and Offset are fixed field formats followed by additional text such as Range (XX miles or km). If the offset is omitted, then the standard offset in the region will be used by radios on receipt. The tone does not include decimals.

FREQUENCY-in-MESSAGE: Similarly we want to be able to send a frequency in a message to another station, and for his station to either manually or automatically change his voice radio to that contact frequency. With this APRS capability, we can implement almost a ham radio version of cell phones. Just send a message to a distant ham callsign and let the system set up the voice links. See the http://aprs.org/avrs.html

FUTURE RADIO COMPATIBILITY: In response to this frequency initiative both Kenwood and Yaesu have released compatible radios. The Kenwood D72/710 and Yaesu FTM350 will automatically insert their frequency into their position text and on receipt, can parse it and automatically tune to that frequency with a single press of the TUNE/QSY button. See the Automatic Voice Relay System: http://www.aprs.org/avrs.html

RADIO INCOMPATIBILITY: There are some radios (Yaesu) that will not recognise the APRS Frequency Object that uses the FREQUENCY as the OBJECT name. Thus, drivers cannot simply see the Frequency in their STATION LISTS. Nor can they be sorted for easy summary at the top of the list. For these radios to see and react to the frequency object, the frequency must also be listed in the comment field. This greatly limits the amount of data in these objects as the Frequency information has to be included twice.

LOCAL/GLOBAL VOICE CONTACT VIA CALLSIGN ALONE: One of the major initiatives for this frequency capability was to be able to set up end-to-end VOIP voice contact between amateur radio operators knowing only callsigns. By using APRS as the backbone signaling system, Automatic QSY can be implemented and this, combined with APRS messaging and IRLP or EchoLink networks can lead to fully automatic end-to-end voice connectivity between APRS users anywhere on the planet with only the knowledge of a callsign. All it takes is a little software. But that is where we are headed with this.

BACKWARDS COMPATIBILITY: It is also possible that some PC controlled radios could be outfitted with simple external PIC processor that can interrogate the radios via their serial ports and can then insert their frequency information into their position beacon. This could easily be added to the firmware of some of the existing APRS TRACKER devices.

NATIONAL VOICE ALERT FREQUENCY: Presently, APRS operators already have a nationwide voice contact frequency called VOICE ALERT. This system for mobiles simply means that the APRS data radio is set to CTCSS 100 with the volume up. This mutes all packets, but allows the operator to be available for a voice call from anyone in simplex range running the PL 100 tone. Most APRS built-in radios can do this as is. Any APRS mobile packet system can also do this if the TNC is attached to the discriminator prior to the CTCSS squelch circuit. But Voice Alert only works within simplex range. That is why we also need to see the stations other Voice band so we can contact him via the repeater he is currently monitoring.

See: http://aprs.org/VoiceAlert3.html

OTHER VOICE SYSTEMS: All specialized local voice assets should also beacon their operating frequency. The three other global internet amateur radio linked systems, IRLP, EchoLink and Winlink also include provisions for beaconing their POSIITON and FREQUENCY data onto the national APRS channel using the FFF.FFFMHz format. This way, mobiles monitoring the national APRS frequency anywhere in the country can be aware of the position and frequency of all amateur radio assets around them.

APRS FREQUENCY FORMATS:

There are two Frequency formats. The POSITION/OBJ COMMENT format includes the frequency as FFF.FFFMHz in the free field text of a normal position or object report as noted above. The other is called the FREQ OBJECT format because it puts the Frequency in the OBJECT NAME using the format of FFF.FFxyz so that it shows up very clearly on the radio's positions/object list. Of course, an object can also have a frequency in its position comment as well. If both the object name and the comment contain a frequency, then the NAME is considered the transmit frequency for the object and the frequency in its comment text is its crossband or non-standard split receive frequency.

*** NOTE, [08 May] that a dual band APRS radio cannot TUNE or QSY to
a cross-band repeater anyway, without losing the APRS band!

The information is in there for DISPLAY and INFO only.

As noted before, a 10x10x+ format is used for the POSITION COMMENT format for best display on the existing variety of APRS radios. Here are the standard 10-byte formats. Please note that spaces are required where shown. In some cases a "_" may be shown for clarity in this document, but in the actual format, a SPACE should be used:

CLARIFICATION 08 May 2012: Note, if the FREQ is in the object name, (not in the first 10 bytes of the "comment") then the remaining comment text (such as TONE and OFFSET, etc) are left justified in the comment field. Examples for a FREQ OBJECT named FFF.FFxyz:

Example object comments
```
-----------------------
Tnnn oXXX Rxxm ... Tone, Offset +xxx or -xxx, and or range
  Tnnn Rxxm ...      Tone and Range
  Tnnn ...           Tone only (radios will use standard offsets)
  Tnnn -000 ...      Tone and forced simplex
  tnnn oXXX ...      Tone and offset and Narrowband
  1750 oXXX ...      Tone Burst with offset
  l750 oXXX ...      Tone Burst & narrowband (Leading lower case L)
```

EXAMPLE POSITION/OBJECT comments when OBJECT NAME is not FFF.FFxyz:

1st 10-BYTES Frequency Description
```
----------   --------------------------------------------------
  FFF.FF MHz    Freq to nearest 10 KHz
  FFF.FFFMHz    Freq to nearest  1 KHz
```

Examples:
```
   146.52 MHz Enroute Alabama
  147.105MHz AARC Radio Club
  146.82 MHz T107 AARC Repeater (Tone of 107.2)
  146.835MHz C107 R25m AARC     (CTCSS of 107.3 and range of 25 mi)
  146.805MHz D256 R25k Repeater (DCS code and range of 25 km)
  146.40 MHz T067 +100 Repeater (67.8 tone and +1.00 MHz offset)
  442.440MHz T107 -500 Repeater (107.2 tone and 5 MHz offset)
  145.50 MHz t077 Simplex       (Tone of 77.X Hz and NARROW band)
```

2nd 10-BYTES Optional Added fields (with leading space)
```
----------   --------------------------------------------------
  _Txxx RXXm    Optional PL tone and nominal range in miles
  _Cxxx -060    Optional CTCSS tone and -600 KHz offset
  _Dxxx RXXk    Optional DCS code and nominal range in kilometers
  _1750 RXXk    Optional 1750 tone, range in km, wide modulation
  _l750 RXXk    Optional 1750 tone narrow modulation (lower case L)
  _Toff -000    Optional NO-PL, No DCS, no Tone, forced simplex
  _Txxx +060    Optional Offset of +600 KHz (up to 9.90 MHz)
  _Exxm Wxxm    East range and West range if different (N,S,E,W)
  _txxx RXXm    Lower case first letter means NARROW modulation
  _FFF.FFFrx    Alternate receiver Frequency if not standard offset
```

If a frequency is included in the first 10 bytes then "MHz" in mixed case is required to be transmitted. (Case should be case insensitive on receipt to allow for manual typos). Notice that the second 10 byte fields begin with a SPACE shown above as "_" (9 useable bytes) for better reading of the packet when combined with a frequency in the first ten bytes. Do not include the "_" but put a SPACE there in your actual packet. (8 May

2012 clarification: if the first 10 bytes do not contain a frequency, then left justify the TONE and OFFSET without the leading space). Here is the raw packet format for the comments:

```
FFF.FFFMHz comment...           one frequency
FFF.FF MHz FFF.FFFrx comment... for separate TX and RX
FFF.FF MHz T107 R25m comment... for TX, tone and range
FFF.FF MHz T107 oXXX R25m ...   for TX, tone and range
Tnnn oXXX R25m comments....     for a FREQ OBJ <08 May clarity>
Tnnn R25m comments....          for a FREQ OBJ standard offsets
Tnnn comments....               for a FREQ OBJ Tone only
```

The 10x10x8 byte format has been defined so that it shows up well on the TH-D7, TM-D700 and Hamhud displays. Examples are as follows:

```
+------------+    +------------+    +------------+
| >WB4APR-11 |    | >WB4APR-11 |    | >147.105md | <08 May clarity
| FFF.FF MHz |    | FFF.FFFMHz |    | T107 +060  |
|  FFF.FFF M |    |  T107 R17m |    | R25m text. |
+------------+    +------------+    +------------+
```

OTHER COMBINATIONS: It is important to note that the TEXT field used for this frequency already has several possible options. Normally the comment-TEXT field begins after the SYMBOL byte. But there are already several defined 7 byte Data-Extensions that then push the comment-TEXT field further to the right as shown here. In many applications, an optional delimiter is added to make the packet more readable. This optional delimiter (usually a " " or "/") is not explicitly called out in the spec, but should be considered in all following parsing of the text field. All of these can be valid use of Data-Extensions and following FREQUENCY info.

```
!DDMM.mmN/DDDMM.MMW$FFF.FFFMHz ...         begins after symbol ($)
!DDMM.mmN/DDDMM.MMW$CSE/SPD/FFF.FFFMHz ...
!DDMM.mmN/DDDMM.MMW$PHGphgd/FFF.FFFMHz ...
!DDMM.mmN/DDDMM.MMW$DFSshgd/FFF.FFFMHz ...
!DDMM.mmN/DDDMM.MMW$DFSshgd FFF.FFFMHz ... with SPACE insted of /
!DDMM.mmN/DDDMM.MMW$CSE/SPDFFF.FFFMHz ...  w/o delimiter
!DDMM.mmN/DDDMM.MMW$PHGphgdFFF.FFFMHz ...  w/o delimiter
```

LEADING SPACE ISSUE: Althout the optional delimiter shown here is a "/" it can also be a SPACE character too. The original intent of APRS was that any free text field that begins with either of these two delimiters, then they can be ignored and the beginning of the text field begins after them. We believe that Kenwood and maybe others did not allow for this option, and so a leading space will not work. We are trying to clarify this.... now in 2012...

FREQUENCY ON GNSS TRACKERS: APRS was intended to be a two-way comunications system between human operators. For this reason, transmit-only tracking devices should include their monitoring voice frequency in their packet beacon so that they can be contacted. The format is the same. Simply put the frequency in the position text. If the the tracker does not have that flexibility (such as a NMEA tracker, or Mic-E), then put the frequency into the STATUS BEACON text with the leading ">" STATUS formatter (not required in Mic-E):

```
BText >FFF.FFFMHz Tnnn oXXX etc......
```

Where the leading ">" makes this packet an APRS STATUS packet and the "etc..." can be any additional free text (or Tnnn tone) as needed. On recepit, most APRS programs should combine this info with the station data to make the frequency parseable as needed.

D-STAR STATION FORMATS:

To make a voice call with D-STAR station that uses D-STAR repeater, the callsign of the accessing repeater is needed. On the other hand, to make a voice call with D-STAR station that does not use a D-STAR repeater, the frequency of the station's voice channel is needed. Here are the examples for D-STAR QSY information.

Communication without using a repeater station:

```
 1) FFF.FFFMHz D-STAR comment...
```

Communication using a repeater station:

```
 2) FFF.FFFMHz D-STAR>CALLSN A comment...
 3) D-STAR>CALLSN A comment...
```

1st 7-BYTES means D-STAR station format.
2nd 8-BYTES "CALLSN A" means the callsign of access repeater
"CALLSN A" is fixed 8 characters.

(The optional "oXXX" (SHIFT/OFFSET) information is available between "FFF.FFFMHz" and "D-STAR")

RECOMMENDED VOICE REPEATER FREQUENCY OBJECTS:

Please see: http://aprs.org/localinfo.html

OBJECT NAMES: Every New-N Paradigm APRS digipeater is supposed to periodically transmit an OBJECT showing the locally recommended voice repeater frequency for travelers visiting the digi's own coverage area. APRS software should be able to locate these frequencies as well. The format for these frequency object names is as follows:

OBJECTNAME

```
  FFF.FFF-z    5KHz repeaters with up to 62 unique (z) ID's
  FFF.FFFyz    5KHz repeaters with up to 2700 unique (yz) ID's
  FFF.FF-yz   10kHz repeaters with over 3600 unique (yz) ID's
  FFF.FFxyz   10kHz using three xyz unique characters...
```

Choose letters to make your repeater frequency unique in all the world so that it can be easily found by wildcarding the callsign FFF.FF*.

The rest of the OBJECT format contains the TONE and RANGE and any specific information on regular net times and meeting dates. The format for these FREQ OBJECTS is:

```
;FFF.FFFxy*111111zDDMM.hhN/DDDMM.hhWrT079 R25m NETxxxxxx MTGxxxx...
;FFF.FFFxy*111111zDDMM.hhN/DDDMM.hhWrT079 oXXX NETxxxxxx MTGxxxx...
```

(15 Jun 08 change. Eliminated the leading space before the T079)

Where T079 is a tone of 79.7 Hz (always drop the tenths)
Where R25m is a Range of 25 miles (or k for km)

Where oXXX is an offset in 10's of KHz +XXX or -XXX
Where NETxxxxxx is something like "Net Tu9PM" or "Net Tu730"
Where MTGxxxx is something like "Mg3rdTu" and must be 7 bytes
Where ... 9 more bytes are possible but won't show on mobiles

These may at first appear very cryptic, but as a standard, they will become second nature. THe goal is to put as much standard info into the minimum text that will fit nicely on the 10x10x8 displays of the D7, D700 and HAMHUD and on the 20 byte displays of the VX-8R and FTM-350.

TONES and DCS: We force the tone to 3 bytes always for simplicity of display, since all tones are standardized, they do not need the tenths. So T088, C067, T156 and DXXX are all valid entries. Lower case implies NARROW modulation as in t088, c067, t156 etc.

OBJECT NAME PERMANANCE: Notice that an object name needs to be unique since there are dozens of 146.94 repeaters in the country, and so these objects choose characters for the optional z, yz and xyz characters above to make them unique.

Since these can be upper and lower case and or numbers, this gives 62 unique identifiers for the 5 KHz repeaters and over 3600 for 10 KHz repeaters. If there are still conflicts, then other delimiters such as "_=!^@, etc" can be used in place of the - separator. But alwayscheck to see if they get through on FINDU.COM using the link below. For example, FINDU.COM does not display FREQ OBJECTS that use the (+) character.

To see what freqeuncies are already in use, simply do a wildcard (*) on the end of the FREQ in your favorite APRS-IS web engine. For example, http://map.findu.com/146.94* will provide a list of all 146.94 repeaters currently showing on APRS and their "xyz" modifiers currently in use!

These objects are transmitted with the pseudo permanent date/time stamp of 111111z. This unique time stamp declares this object to be permanent. This means that it should not be replaced by any other similarly named object unless that object is transmitted by the same originating station. This lets the originator of a permanent object update or move his object, but it then prevents his object from being replaced by anyone else's similar object.

DX CLUSTER FORMAT: APRS also decodes the DX CLUSTER format, and this format includes a frequency field too. All APRS software should decode and capture this frequency information also.

THE NEXT SECTION IS ALSO FOUND IN: www.aprs.org/echo-irlp-win.txt)

EchoLink OBJECT FORMATS:

EchoLink object names have the format of EL-123456 so that the node number can be easily seen on the station list of the mobile. This is important, since the mobile user has to have the node number to activate a link. Not the callsign of the node. The rest of the object uses the previous formats above to convey operating frequency and tone, and range. Here is an example OBJECT for EchoLink nodes.

```
;EL-123456*111111zDDMM.--NEDDDMM.--W0FFF.FFFMHz Tnnn STTS CALL...
```
.... <== up to a max of 43 total characters

The "STTS" is only a four byte field for the current stataus of the
node. Maybe "busy", "conf", "off_", "idle" etc. The CALL is the callsign of this node. But we don’t get the

RANGE info. So maybe:

```
;EL-123456*111111zDDMM.--NEDDDMM.--W0FFF.FFFMHz Tnnn oXXX %CALL..
```

.... <== up to a max of 43 total characters

Where % is a single byte to indicate Status. This will show with all of the information on teh front panel of the APRS mobiles as:

```
+------------+
| EL-123456  |
| 438.700MHz |
|  Tnnn Rxxm |
|  %CALL..   |
+------------+
```

The only loss is that the -R or -L does not show for most node callsigns unless they are four character calls. Though this is not needed by the mobile operator

POSITION DETAILS: In the LAT/LONG fields the TWO hundredths digits are shown entered as SPACE bytes so that the object is transmitted as a one-mile ambiguity object. Replace the two "--" bytes above with SPACES for it to work. The E between LAT/LONG makes APRS display the EchoLink symbol. It is hoped that a single central EchoLink server will generate these objects so that the only thing needed to get them on the air in a local area is to have the local IGate sysop to activate the local object by name for pass-to-RF.

ECHOLINK FORMAT USING MIC-E: This format is for the Kenwood D710 radio which reports its position in Mic-E format. This formatwill appear on the air as:

```
MYCALL>LLLLLL,DIGI:'GGGCSD0E]FFF.FFFMHz Tnnn RXXm Ecolink node=
```

The LLLLLL is the latitude bytes, GGGCSD are the longitude and course speed bytes. The "0E" are the ECHOLINK Symbol bytes. And the "]" byte indicates this is a D700 series Mic-E and the "=" on the end indicates it is the D710 version. The "FFF.FFFMHz Tnnn RXXm" is the frequency and range format. And "Ecolink node" is intentionally misspelled to fit nicely on the display of a D700.

IRLP OBJECT FORMATS:

See the Scrip to add this to your IRLP node: http://irlp.kc6hur.net/irlp_scripts.php

IRLP object names are similar and have the format of IRLP-1234 or IRLP12345 if they ever go to 5 digits. This format is so that the node number can be easily seen on the station list of the mobile. The rest of the object uses the previous formats above to convey operating frequency and tone, and range. Here is an example OBJECT for IRLP:

```
;IRLP-1234*111111zDDMM.--NIDDDMM.--W0FFF.FFFMHz Tnnn oXXX STTS CALL...
```

.... <== up to a max of 43 total characters

In the LAT/LONG fields the TWO hundredths digits are shown entered as SPACE bytes so that the object is transmitte as a one-mile ambiguity object. Replace the two "--" bytes above with SPACES for it to work. The I between LAT/LONG makes APRS display the IRLP symbol. It is hoped that a central server will be written for these objects as well so that they can be consistently generated into the APRS-IS and then only gated back to RF locally. The STATUS (STTS) can be something like these: (Idle, off_, busy, etc). For more info, contact Mark Herson n2mh@n2mh.net

WiRES OBJECT FORMATS:

Yaesu WiRES nodes should also appear on the APRS channel. For example for an OBJECT NAME : WIR-1101D the format would be:

Example: WiRES Node Number = "1101D" (Lat:3348.43N/Lon:11802.32W)

```
;WIR-1101D*111111z3348.43NW11802.32W0430.900MHz D023 oXXX STTS...
```
.... <== up to a max of 43 total characters

If an FTM-350 Radio is used as the WiRES node, then it will identify using the Mic-E format and should use this format:

```
MYCALL>LLLLLL,DIGI:'GGGCSD0W`FFF.FFFMHz Tnnn oXXX WiRES node _"
```

WINLINK OBJECT FORMATS:

WinLINK Telpac gateway object names have the format of W?-CALLSIGN (where W? can be WL or W1, W2,... W9 for multiple stations) so that the node callsign can be easily seen on the station list of the mobile. The rest of the object uses the previous formats above to convey operating frequency and range. The baud rate, "bbbb" for packet is inserted in place of the TONE, Txxx for voice. Here is an example OBJECT for injection into the APRS-IS:

```
;WL-AB9XYZ*111111zDDMM.--NWDDDMM.--WaFFF.FFFMHz 1200 oXXX comment
```
.... <== up to a max of 43 total characters

In the LAT/LONG fields the TWO hundredths digits are shown entered as SPACE bytes so that the object is transmitte as a one-mile ambiguity object. Replace the two "--" bytes above with SPACES for it to work. The W between LAT/LONG and "a" symbol makes APRS display the WinLink symbol.

MICROWAVES: Of course FFF.FFFMHz only works to 999.999 MHz, and so we have defined some letter designations above that. There are two methods. One, is to simply use GHz as in XXX.XXXGHz. But this is limited to wideband modes to the nearest MHz. Example is__1.296GHz for 1296 MHz. But to retain one KHz resolution, we use this table of alphabetical extensions:

A96.000MHz would be 1296 MHz
B20.000MHz would be 2320 MHz
C01.000MHz would be 2401 MHz
D01.000MHz would be 3401 MHz
E51.000MHz would be 5651 MHz
F60.000MHz would be 5760 MHz
G30.000MHz would be 5830 MHz
H01.000MHz would be 10,101 MHz
I01.000MHz would be 10,201 MHz
J68.000MHz would be 10,368 MHz
K01.000MHz would be 10,401 MHz
L01.000MHz would be 10,501 MHz

M48.000MHz would be 24,048 MHz
N01.000MHz would be 24,101 MHz
O01.000MHz would be 24,201 MHz

APRS FREQUENCY-IN-MESSAGE FORMATS: A variety of formats have been proposed, but until someone is ready to implement either the auto-QSY on receipt of a message, or to transmit such a message, it is premature to nail down an exact format. But something like:

QSY FFF.FFFMHz! ... Auto QSY now
QSY FFF.FFFMHz? ... Auto QSY unless user hits NO
QSY FFF.FFFMHz. ... QSY if user manually hits TUNE

CONCLUSION: Examples of frequency use in APRS are obvious:

1) Advertising the voice frequency you are monitoring
2) Voice repeater and other local frequency Objects
3) Mobile GNSS map display of surrounding frequency assets
4) IRLP and EchoLink nodes
5) WinLINK Packet nodes
6) EOC operations
7) Long distance travelers
8) ATV repeaters and links, etc...

Since the object of APRS is to facilitate local communications and situational awareness of all surrounding ham radio assets, everyone is encouraged to include their operating frequency in their position packets to make their availability known.

19 USER-DEFINED DATA FORMAT

The APRS protocol defines many different data formats, but it cannot anticipate every possible data type that programmers may wish to send. The User-Defined data format is designed to fill these gaps. Under this system, program authors are free to send data in any format they choose.

The data in the AX.25 Information field consists of a three-character header:

{	APRS Data Type Identifier.
U	A one-character User ID.
X	A one-character user-defined packet type.

The APRS Working Group will issue User IDs to program authors who express a need.

[Keep in mind there is a limited number of available User IDs, so please do not request one unless you have a true need. The Working Group may require an explanation of your need prior to issuing a character. If only one or two data formats are needed, those may be issued from a User ID pool].

For experimentation, or prior to being issued a User ID, anyone may utilize the User ID character of **{** without prior notification or approval (i.e. packets beginning with **{{** are experimental, and may be sent by anyone).

Important Note: Although there is no restriction on the nature of user-defined data, it is highly recommended that it is represented in printable 7-bit ASCII character form.

User-Defined Data Format

	{	***User ID*** U	***User-Defined Packet Type*** X	***User-defined data (printable ASCII recommended)***
Bytes:	1	1	1	n

Examples

`{Q1qwerty`	User ID = Q, User-defined packet type = 1.
`{{zasdfg`	User ID undefined (experimental), User-defined packet type = z.

This is envisioned as a way for authors to experiment and build in features specific to their programs, without the danger of a non-standard packet crashing other authors' programs. In keeping with the spirit of the APRS protocol, authors are encouraged to make these formats public.

The following list contains all of the User IDs and packet types as of April 2020. There is currently no mechanism in place to reserve a code and avoid conflicts.

```
HEADER  AUTHOR   DESCRIPTION
------  -------- --------------------------------------------------
 {BT    KB3UKS   For BRICSAT Telemetry
 {BP    KB3UKS   For BRICSAT Pictures
 {Dx    WB2OSZ   Direwolf software TNC
 {Fnn   WB4APR   nn is a 2 byte descriptor of MITEL GPS data formats
 {HT    WB4APR   For HAM-Trail reporting devices (tbd)
 {ID    WB4APR   RFID tags
 {K1    WB4APR   "K" for Keps and "1" for NASA One line.
 {K2n   WB4APR   For NASA two line elements. "n" is line 1 or 2.
 {KY    KI4YLJ   For KYSAT (University of Kentucky)
 {Q1    WB4APR   For QIKcom-1 data
 {Q2    WB4APR   For QIKcom-2 data
 {Sx    ZL4FOX   for SARTrack a SAR specific set of protocols
 {TP    WB4APR   Telemetry for PCSAT2 Solar Panel Experiments
 {TU    WB4APR   Telemetry for PCSAT2 Solar Panel Experiments
 {x1    J Bennet for WX
 {x2    J Bennet for WX
```

Generally, all formats using this method will be considered optional. No program is required to decode any of these packets, and must ignore any it does not decode. However, it is possible that in the future some of these formats may prove to be of sufficient utility and interest to the entire APRS community that they will be specifically included in future versions of the APRS protocol.

20 OTHER PACKETS

Invalid Data or Test DataPackets

To indicate that a packet contains invalid data, or test data that does not conform to any standard APRS format, the **,** Data Type Identifier is used.

For example, the Mic-E unit will generate such a packet if it detects that a received GNSS sentence is not valid.

Invalid Data / Test Data Format		
	,	***Invalid Data or Test Data***
Bytes:	1	n

Example
`,191146,V,4214.2466,N,07303.5181,W,417.238,114.5,091099,14.7,W/GPS FIX`
Invalid GPS data from a Mic-E unit. The unit has interpreted the **V** character in the received sentence to mean the data is invalid, and has stripped out the `$GPRMC` header.

All Other Packets

Packets that do not meet any of the formats described in this document are assumed to be non-APRS beacons. Programs can decide to handle these, or ignore them, but they must be able to process them without ill effects.

APRS programs may treat such packets as APRS Status Reports. This allows APRS to accept any UI packet addressed to the typical beacon address to be captured as a status message. Typical TNC ID packets fall into this category. Once a proper Status Report (with the APRS Data Type Identifier **>**) has been received from a station it will not be overwritten by other non-APRS packets from that station.

Normally, **HID should be OFF** in all APRS TNCs.

20th century TNCs had a command called HID which enables a special "ID" packet once every 10 minutes if the TNC is used in repeater service. But this is for use on conventional packet channels for Digipeaters, and Nodes to identify their presence. In APRS, this HID function has been replaced by the much more valuable APRS POSITION packet which not only identifies the digi, but also, its type, position, elevation and range. Thus the HID is supposed to be OFF and remain off.

21 APRS SYMBOLS

Three Methods

There are three methods of specifying an APRS symbol (display icon):

- In the AX.25 Information field.
- In the AX.25 Destination Address.
- In the SSID of the AX.25 Source Address.

The preferred method is to include the symbol in the Information field. However, where this is not possible (for example, in stand-alone trackers with no means of introducing the symbol into the Information field), either of the other two methods may be used instead.

The Symbol Tables

There are two APRS Symbol Tables:

- Primary Symbol Table
- Alternate Symbol Table

See Appendix 2 for a full listing of these tables.

The essential difference between the Primary and Alternate Symbol Tables is that some of the symbols in the Alternate Symbol Table can be overlaid with an alphanumeric character. For example, a “car” icon in the Alternate Symbol Table could be overlaid with the digit “3”, to indicate it is car #3.

Symbols capable of taking an overlay are marked as ***[with overlay]***.

None of the symbols in the Primary Symbol Table can be overlaid.

In the tables, each symbol is coded in three ways:

- **/$** or **\$** — for symbols in the Information field.
- **GPSxyz** — for generic Destination addresses containing symbols.
- **GPSCnn** or **GPSEnn** — another form of generic Destination addresses containing systems.

In addition, 15 of the symbols in the Primary Symbol Table have an associated SSID (e.g. a small aircraft has SSID -7). The SSID is intended for use in the AX.25 Source Address of stand-alone trackers which have no othermeans of specifying the symbol.

Symbols in theAX.25 Information Field

A symbol in the AX.25 Information field is a combination of a one-character Symbol Table Identifier and a one-character Symbol Code.

For example, in the Position Report:

```
@092345z4903.50N/07201.75W>088/036…
```

the forward slash **/** is the Symbol Table Identifier and the **>** character is the Symbol Code (in this case representing a "car" icon) from the selected table.

The Symbol Table Identifier character selects one of the two Symbol Tables, or it may be used as single-character (alpha or numeric) overlay, as follows:

Symbol Table Identifier	***Selected Table or Overlay Symbol***
/	Primary Symbol Table (mostly stations)
****	Alternate Symbol Table (mostly Objects)
0-9	Numeric overlay. Symbol from Alternate Symbol Table (*uncompressed* lat/long data format)
a-j	Numeric overlay. Symbol from Alternate Symbol Table (*compressed* lat/long data format only). i.e. a-j maps to 0-9
A-Z	Alpha overlay. Symbol from Alternate Symbol Table

In the generic case, a symbol from the Primary Symbol Table is represented as the character-pair **/$**, and a symbol from the Alternate Symbol Table as **\$**.

Overlays withSymbols in the AX.25 Information Field

Where the Symbol Table Identifier is `0-9` or `A-Z` (or `a-j` with *compressed* position data only), the symbol comes from the *Alternate* Symbol Table, and is overlaid with the identifier (as a single digit or a capital letter).

For example, in the *uncompressed* Position Report:

```
@092345z4903.50N307201.75W>…
```

the digit **3** following the latitude will cause the number "3" to be overlaid on top of the "car" icon (**Note**: Because the symbol is overlaid, the **>** Symbol Code here comes from the *Alternate* Symbol Table).

Similarly, to overlay a "car" icon with the letter "B" in a *compressed* Position Report, the report will look something like:

```
=BL!!<*e7>7P[
```

However, in a *compressed* Position Report, it is not permissible to use a *numeric* Symbol Table Identifier (0-9) — *compressed* positions never start with a digit. If a numeric overlay is required, the report must use a lower-case letter instead (in the range **a-j**) as the Symbol Table Identifier. The lower-case letter is then mapped to the digits **0-9** (i.e. a=0, b=1, c=2, d=3 etc).

Thus, in the *compressed* Position Report:

```
=d5L!!<*e7>7P[
```

the letter **d** maps to overlay character “3”.

As noted above, not all symbols from the Alternate Symbol Table may be overlaid in this way — those that can be overlaid are marked as ***[with overlay]*** in Appendix 2. This means that they are *capable* of taking an overlay, but they do not necessarily need to have one. Thus, for example, the following report uses the car symbol from the Alternate Symbol Table, but does not display an overlay:

```
@092345z4903.50N\07201.75W>…
```

Symbols in the AX.25 Destination Address

Where it is not possible to include a symbol in the Information field, the symbol may be specified in the AX.25 Destination Address instead, using the following generic destination addresses: `GPSxyz`, `GPSCnn`, `GPSEnn`, `SPCxyz` and `SYMxyz`.

The characters `xy` and `nn` refer to entries in the APRS Symbol Tables. For example, from the Primary Symbol Table, a tracker could use the Destination Address `GPSMV␣` or `GPS30` to specify a “car” icon.

The character `z` specifies the overlay character (where permitted), or is a ␣ (space) — the space is a filler character, as all AX.25 addresses must be exactly 6 characters long.

The `GPS/SPC/SYMxy` and `GPSCnn/GPSEnn` addresses can be used interchangeably. Thus, for example, `GPSBM`, `SPCBM`, `SYMBM` and `GPSC12` all specify a “Boy Scouts” icon (from the Primary Symbol Table), and `GPSOM`, `SPCOM`, `SYMOM` and `GPSE12` all specify a “Girl Scouts” icon (from the Alternate Symbol Table).

Overlays with Symbols in the AX.25 Destination Address

If the `z` character in a `GPSxyz`, `SPCxyz` or `SYMxyz` address is not a space, it specifies an alphanumeric overlay character, in the range `0-9` or `A-Z`.

Overlays can only be used with symbols from the Alternate Symbol Table marked with the legend ***[with overlay]***.

For example, if the “car” icon is to be overlaid with a digit “3”, the Destination Address will be `GPSNV3`.

However, even if the address is overlay-capable, it is not actually necessary to specify an overlay; e.g. `GPSNV␣`.

`GPSCnn` and `GPSEnn` symbols can not have overlays.

Symbol in theSource Address SSID

Where it is not possible to include a symbol in the Information field or in the Destination Address, the symbol may be specified in the SSID of the Source Address instead:

SSID-Specified Icons in the AX.25 Source Address Field

SSID	Icon	SSID	Icon
-0	*[no icon]*	-8	Ship (power boat)
-1	Ambulance	-9	Car
-2	Bus	-10	Motorcycle
-3	Fire Truck	-11	Balloon
-4	Bicycle	-12	Jeep
-5	Yacht	-13	Recreational Vehicle
-6	Helicopter	-14	Truck
-7	Small Aircraft	-15	Van

Symbol Precedence

APRS packets should not contain more than one symbol. However, it is conceivably possible to (erroneously) construct a packet containing up to three different symbols.

For example:

	Source Address SSID	*Destination Address*	*Information Field*
	G3NRW-7	GPSMV	!0123.45N/01234.56Wj
Symbol	Small Aircraft	Car	Jeep

In such a situation:

- The symbol in the Information field takes precedence over any other symbol.
- If there is no symbol in the Information field, the symbol in the Destination Address takes precedence over the symbol in the Source Address SSID.

APRS Symbol Attributes

One of the most important aspects of APRS is the display of tactical information on maps using symbols for each station. Each such SYMBOL has several ATTRIBUTES that convey additional meanings. In the original APRS, these additional attributes were displayed with COLORS and circles. Some new software has taken a more simplistic approach and simply usees fixed color ICONS but without any displayed attributes. Thus, the users of these systems are blind to attribute information on their maps without having to click on each one... This operator intensive masking of important information does not do justice in some cases to the original intent of APRS information display to the end users.

See http://www.aprs.org/symbols.html for more details.

SSID Recommendations

It is very convenient to other mobile operators or others, looking at callsigns flashing by, to be able to recognize some common applications at a glance. Here are the recommendations for the 16 possible SSIDs. Note, The SSID of zero is dropped by display applications. So a callsign with no SSID has an SSID of 0.

SSID	***Type of Station***
-0	Your primary station, usually fixed and message capable
-1	Generic additional station, digi, mobile, wx, etc.
-2	Generic additional station, digi, mobile, wx, etc.
-3	Generic additional station, digi, mobile, wx, etc.
-4	Generic additional station, digi, mobile, wx, etc.
-5	Other networks (Dstar, iPhones, Androids, Blackberrys etc.)
-6	Special activity, Satellite ops, camping or 6 meters, etc.
-7	Walkie talkies, HTs or other human portable
-8	Boats, sailboats, RVs or second main mobile
-9	Primary Mobile (usually message capable)
-10	Internet, IGates, echolink, winlink, AVRS, APRN, etc.
-11	Balloons, aircraft, spacecraft, etc
-12	APRStt, DTMF, RFID, devices, one-way trackers*, etc.
-13	Weather stations
-14	Truckers or generally full time drivers
-15	Generic additional station, digi, mobile, wx, etc.

APPENDIX 1: APRS DATA FORMATS

This Appendix contains format diagrams for all APRS data formats. The gray fields are optional. Shaded (yellow) characters are literal ASCII characters.

	AX.25 UI-FRAME FORMAT								
	Flag	Destination Address	Source Address	Digipeater Addresses (0-8)	Control Field (UI)	Protocol ID	INFORMATION FIELD	FCS	Flag
Bytes:	1	7	7	0–56	1	1	1–256	2	2

	Generic APRS Information Field			
	Data Type ID	APRS Data	APRS Data Extension	Comment
Bytes:	1	n	7	n

	Lat/Long Position Report Format — without Timestamp					
	! or =	Lat	Sym Table ID	Long	Symbol Code	Comment (max 43 chars)
Bytes:	1	8	1	9	1	0-43

	Lat/Long Position Report Format — with Timestamp						
	/ or @	Time DHM / HMS	Lat	Sym Table ID	Long	Symbol Code	Comment (max 43 chars)
Bytes:	1	7	8	1	9	1	0-43

	Lat/Long Position Report Format — with Data Extension (no Timestamp)						
	! or =	Lat	Sym Table ID	Long	Symbol Code	Course/Speed Power/Height/Gain/Dir Radio Range DF Signal Strength	Comment (max 36 chars)
Bytes:	1	8	1	9	1	7	0-36

	Lat/Long Position Report Format — with Data Extension and Timestamp							
	/ or @	Time DHM / HMS	Lat	Sym Table ID	Long	Symbol Code	Course/Speed Power/Height/Gain/Dir Radio Range DF Signal Strength	Comment (max 36 chars)
Bytes:	1	7	8	1	9	1	7	0-36

	Maidenhead Locator Beacon			
	[	Grid Locator	]	Comment
Bytes:	1	4 or 6	1	n

	Raw NMEA Position Report Format	
	NMEA Received Sentence	
	$	… , … , … , … , … , … , … , … , … , …
Bytes:	1	25-209

	DF Report Format — without Timestamp							
	! or =	Lat	Sym Table ID /	Long	Symbol Code \	Course/Speed Power/Height/Gain/Dir Radio Range DF Signal Strength	/BRG/NRQ	Comment (max 28 chars)
Bytes:	1	8	1	9	1	7	8	0-28

	DF Report Format — with Timestamp								
	/ or @	Time DHM / HMS	Lat	Sym Table ID /	Long	Symbol Code \	Course/Speed Power/Height/Gain/Dir Radio Range DF Signal Strength	/BRG/NRQ	Comment (max 28 chars)
Bytes:	1	7	8	1	9	1	7	8	0-28

Compressed Lat/Long Position Report Format — no Timestamp								
	! or =	Sym Table ID	Comp Lat YYYY	Comp Long XXXX	Symbol Code	Compressed Course/Speed Compressed Radio Range Compressed Altitude	Comp Type T	Comment (max 40 chars)
Bytes:	1	1	4	4	1	2	1	0-40

Compressed Lat/Long Position Report Format — with Timestamp									
	/ or @	Time DHM / HMS	Sym Table ID	Comp Lat YYYY	Comp Long XXXX	Symbol Code	Compressed Course/Speed Compressed Radio Range Compressed Altitude	Comp Type T	Comment (max 40 chars)
Bytes:	1	7	1	4	4	1	2	1	0-40

Compression Type (T) Byte Format								
Bit:	7	6	5	4	3	2	1	0
	Not used	*Not used*	***GPS Fix***	***NMEA Source***		***Compression Origin***		
Value:	0	0	0 = old (last) 1 = current	0 0 = other 0 1 = GLL 1 0 = GGA 1 1 = RMC		0 0 0 = Compressed 0 0 1 = TNC BText 0 1 0 = Software (DOS/Mac/Win/+SA) 0 1 1 = [tbd] 1 0 0 = KPC3 1 0 1 = Pico 1 1 0 = Other tracker [tbd] 1 1 1 = Digipeater conversion		

Mic-E Data — DESTINATION ADDRESS FIELD Format							
	Lat Digit 1 + Message Bit A	Lat Digit 2 + Message Bit B	Lat Digit 3 + Message Bit C	Lat Digit 4 + N/S Lat Indicator	Lat Digit 5 + Longitude Offset	Lat Digit 6 + W/E Long Indicator	APRS Digi Path Code
Bytes:	1	1	1	1	1	1	1

Mic-E Data — INFORMATION FIELD Format										
	Data Type ID	Longitude			Speed and Course			Symbol Code	Sym Table ID	Mic-E Telemetry Data (obsolete) Mic-E Status Text
		d+28	m+28	h+28	SP+28	DC+28	SE+28			
Bytes:	1	1	1	1	1	1	1	1	1	n

	Object Report Format — with Lat/Long position									
	;	***Object Name***	***** or **_**	***Time DHM / HMS***	***Lat***	***Sym Table ID***	***Long***	***Symbol Code***	***Course/Speed*** ***Power/Height/Gain/Dir*** ***Radio Range*** ***DF Signal Strength*** ***Area Object***	***Comment (max 36 chars with Data Extension, or 43 without)***
Bytes:	1	9	1	7	8	1	9	1	7	0-36/43

	Object Report Format — with Compressed Lat/Long position					
	;	***Object Name***	***** or **_**	***Time DHM / HMS***	***Compressed Position Data*** `/YYYYXXXX$csT`	***Comment***
Bytes:	1	9	1	7	13	43

	Item Report Format — with Lat/Long position								
	)	***Item Name***	**!** or **_**	***Lat***	***Sym Table ID***	***Long***	***Symbol Code***	***Course/Speed*** ***Power/Height/Gain/Dir*** ***Radio Range*** ***DF Signal Strength*** ***Area Object***	***Comment (max 36 chars with Data Extension, or 43 without)***
Bytes:	1	3-9	1	8	1	9	1	7	0-36/43

	Item Report Format — with Compressed Lat/Long position				
	)	***Item Name***	**!** or **_**	***Compressed Position Data*** `/YYYYXXXX$csT`	***Comment***
Bytes:	1	3-9	1	13	43

	Raw Weather Report Format - Not Recommended - Use Complete Weather Report Format	
	! *or* **#** *or* **$** *or* *****	***Raw Weather Data***
Bytes:	1	n

Positionless Weather Report Format - Not Recommended - Should use Complete Weather Report					
	_	Time MDHM	WeatherData	APRS Software s	WX Unit uuuu
Bytes:	1	8	n	1	2-4

Weather Data									
	Wind Direction cccc	Wind Speed ssss	Gust gggg	Temp tttt	Rain Last Hr rrrr	Rain Last 24 Hrs pppp	Rain Since Midnight PPPP	Humidity hhh	Barometric Pressure bbbbbb
Bytes:	4	4	4	4	4	4	4	3	6

Complete Weather Report Format — with Lat/Long position, no Timestamp									
	! or =	Lat	Sym Table ID	Long	Symbol Code _	Wind Directn/ Speed	Weather Data	APRS Software s	WX Unit uuuu
Bytes:	1	8	1	9	1	7	n	1	2-4

Complete Weather Report Format — with Lat/Long position and Timestamp										
	/ or @	Time DHM / HMS	Lat	Sym Table ID	Long	Symbol Code _	Wind Directn/ Speed	Weather Data	APRS Software s	WX Unit uuuu
Bytes:	1	7	8	1	9	1	7	n	1	2-4

Complete Weather Report Format — with Compressed Lat/Long position, no Timestamp										
	! or =	Sym Table ID	Comp Lat YYYY	Comp Long XXXX	Symbol Code _	Comp Wind Directn/ Speed	Comp Type T	Weather Data	APRS Software s	WX Unit uuuu
Bytes:	1	1	4	4	1	2	1	n	1	2-4

Complete Weather Report Format — with Compressed Lat/Long position, with Timestamp											
	/ or @	Time DHM / HMS	Sym Table ID	Comp Lat YYYY	Comp Long XXXX	Symbol Code _	Comp Wind Directn/ Speed	Comp Type T	Weather Data	APRS Software S	WX Unit uuuu
Bytes:	1	7	1	4	4	1	2	1	n	1	2-4

Complete Weather Report Format — with Object and Lat/Long position												
	*	Object Name	*	Time DHM / HMS	Lat	Sym Table ID	Long	Symbol Code _	Wind Directn/ Speed	Weather Data	APRS Software S	WX Unit uuuu
Bytes:	1	9	1	7	8	1	9	1	7	n	1	2-4

Storm Data										
	Direction	/	Speed	Storm Type /ST	Sustained Wind Speed /www	Peak Wind Gusts ^GGG	Central Pressure /pppp	Radius Hurricane Winds >RRR	Radius Tropical Storm Winds &rrr	Radius Whole Gale %ggg
Bytes:	3	1	3	3	4	4	5	4	4	4

Telemetry Report Format									
	T	Sequence No #nnn,	Analog Value 1 aaa,	Analog Value 2 aaa,	Analog Value 3 aaa,	Analog Value 4 aaa,	Analog Value 5 aaa,	Digital Value bbbbbbbb	Comment
Bytes:	1	5	4	4	4	4	4	8	n

Note: Most modern applications recognize variable length fields with decimal points and leading minus sign.

Telemetry Parameter Name Message Data Note the different byte counts, which include commas where shown. The list may stop at any field.														
	PARM.	A1 N…	A2 ,N…	A3 ,N…	A4 ,N…	A5 ,N…	B1 ,N…	B2 ,N…	B3 ,N…	B4 ,N…	B5 ,N…	B6 ,N…	B7 ,N…	B8 ,N…
Bytes:	5	1-7	1-7	1-6	1-6	1-5	1-6	1-5	1-4	1-4	1-4	1-3	1-3	1-3

Telemetry Unit/Label Message Data
Note the different byte counts, which include commas where shown. The list may stop at any field.

		A1	A2	A3	A4	A5	B1	B2	B3	B4	B5	B6	B7	B8
	UNIT.	U...	,U...	,U...	,U...	,U...	,L...	,L...	,L...	,L...	,L...	,L...	,L...	,L...
Bytes:	5	1-7	1-7	1-6	1-6	1-5	1-6	1-5	1-4	1-4	1-4	1-3	1-3	1-3

Telemetry Equation Coefficients Message Data
The list may stop at any field. Value = $a \times v^2 + b \times v + c$

		A1			A2			A3			A4			A5		
	EQNS.	a	,b	,c	,a	,b	,c	,a	,b	,c	,a	,b	,c	,a	,b	,c
Bytes:	5	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n

Telemetry Bit Sense/Project Name Message Data

		B1	B2	B3	B4	B5	B6	B7	B8	Project Title
	BITS.	x	x	x	x	x	x	x	x	
Bytes:	5	1	1	1	1	1	1	1	1	0-23

Message Format

					Message ID	
	:	Addressee	:	Message Text (max 67 chars)	{	Message No xxxxx
Bytes:	1	9	1	0-67	1	1-5

Message Acknowledgement Format

	:	Addressee	:	ack	Message No xxxxx
Bytes:	1	9	1	3	1–5

Message Rejection Format

	:	Addressee	:	rej	Message No xxxxx
Bytes:	1	9	1	3	1–5

	General Bulletin Format					
	:	**BLN**	***Bulletin ID*** n	␣␣␣␣␣	**:**	***Bulletin Text (max 67 characters)***
Bytes:	1	3	1	5	1	0-67

	Announcement Format					
	:	**BLN**	***Announcement Identifier*** x	␣␣␣␣␣	**:**	***Announcement Text (max 67 characters)***
Bytes:	1	3	1	5	1	0-67

	Group Bulletin Format					
	:	**BLN**	***Group Bulletin ID*** n	***Group Name***	**:**	***Group Bulletin Text (max 67 characters)***
Bytes:	1	3	1	5	1	0-67

	National Weather Service Bulletin Format			
	:	**NWS-**xxxxx	**:**	***NWS Bulletin Text***
Bytes:	1	9	1	n

	General Query Format							
				Target Footprint				
	?	***Query Type***	**?**	***Lat***	**,**	***Long***	**,**	***Radius***
Bytes:	1	n	1	n	1	n	1	4

	Directed Station Query Format					
	:	***Addressee***	**:**	**?**	***Query Type***	***Callsign of Heard Station***
Bytes:	1	9	1	1	5	0-9

Status Report Format		
>	Time DHM z	Status Text (max 62 chars if no timestamp, or 55 chars if there is a timestamp)
1	7	0-62 or 0-55

Bytes: row gives the byte counts above.

Status Report Format — with Maidenhead Grid Locator							
	Maidenhead Locator			Sym Table ID	Symbol Code		Status Text (starting with a space) (max 54 chars)
>	GG	nn	gg			␣	
1	2	2	2	1	1	1-54	

Data with Source Path Header		
Source Path Header	Data Type ID	Rest of the original data
n	1	n

Source Path Header — "TNC-2" Format An asterisk follows the digipeater callsign heard.					
Source Callsign (-SSID)	>	Destination Callsign (-SSID)	0-8 Digipeaters		:
			,	Digipeater Callsign (-SSID)(*)	
1-9	1	1-9	0-80		1

Source Path Header — "AEA" Format Obsolete - shown for historical reference only. An asterisk follows the source or digipeater callsign heard.					
Source Callsign (-SSID)(*)	0-8 Digipeaters		>	Destination Callsign (-SSID)	:
	>	Digipeater Callsign (-SSID)(*)			
1-10	0-80		1	1-9	1

Third-party Format			
	}	***Third-Party Header***	***Rest of the original data***
Bytes:	1	n	n

Third Party Header — "TNC-2" format								
	Source Path Header (without "unused" digipeaters, * or :)	**,**	***Third-Party Network Identifier ("callsign")***	**,**	***Callsign of Receiving Gateway Station (-SSID)***	*****	**:**
Bytes:	n	1	1-9	1	1-9	1	1

Third Party Header — "AEA" format Obsolete - Shown for historical reference only.								
	Source Path Header (without "unused" digipeaters, destination, * or :)	***Third-Party Network Identifier ("callsign")***	**>**	***Callsign of Receiving Gateway Station (-SSID)***	*****	**>**	***Destination Callsign from Source Path Header (-SSID)***	**:**
Bytes:	2-90	1-9	1	1-9	1	1	1-9	1

User-Defined Data Format				
	{	***User ID*** U	***User-Defined Packet Type*** X	***User-defined data (printable ASCII recommended)***
Bytes:	1	1	1	n

Invalid Data / Test Data Format		
	,	***Invalid Data or Test Data***
Bytes:	1	n

Agrelo Format				
	%	***Bearing*** nnn	**/**	***Quality*** n
Bytes:	1	3	1	1

APPENDIX 2: THE APRS SYMBOL TABLES

(Each highlighted character in the Alternate Symbol Table may be replaced with an overlay character).

PRIMARY SYMBOL TABLE

/$	GPS xyz	GPS Cnn	Icon	
/!	BB	01	Police, Sheriff	
/"	BC	02	*[reserved]*	
/#	BD	03	Digi (green star with white center)	
/$	BE	04	Phone	
/%	BF	05	DX Cluster	
/&	BG	06	HF Gateway	
/'	BH	07	Small Aircraft (SSID –7)	
/(	BI	08	Mobile Satellite Groundstation	
/)	BJ	09	Wheelchair (handicapped)	
/*	BK	10	Snowmobile	
/+	BL	11	Red Cross	
/,	BM	12	Boy Scouts	
/-	BN	13	House QTH (VHF)	
/.	BO	14	X	
//	BP	15	Dot	
/0	P0	16	Numerical Circle	**Obsolete. Use the "Circle with overlay" symbol instead (code \0).**
/1	P1	17	Numerical Circle	
/2	P2	18	Numerical Circle	
/3	P3	19	Numerical Circle	
/4	P4	20	Numerical Circle	
/5	P5	21	Numerical Circle	
/6	P6	22	Numerical Circle	
/7	P7	23	Numerical Circle	
/8	P8	24	Numerical Circle	
/9	P9	25	Numerical Circle	
/:	MR	26	Fire	
/;	MS	27	Campground (Portable ops)	
/<	MT	28	Motorcycle (SSID –10)	
/=	MU	29	Railroad Engine	
/>	MV	30	Car (SSID –9)	
/?	MW	31	File Server	
/@	MX	32	Hurricane Future Prediction (dot)	
/A	PA	33	Aid Station	
/B	PB	34	BBS or PBBS	
/C	PC	35	Canoe	

ALTERNATE SYMBOL TABLE

\$	GPS xyz	GPS Enn	Icon
\!	OB#	01	Emergency (and overlays)
\"	OC	02	*[reserved]*
\#	OD#	03	Digi (green star) *[with overlay]*
\$	OE#	04	Bank or ATM (green box)
\%	OF#	05	Power Plant with overlay
\&	OG#	06	I=Igate R=RX T=1hopTX 2=2hopTX
\'	OH#	07	Crash (&now incident sites)
\(	OI#	08	Cloudy (other clouds w overlay)
\)	OJ	09	Firenet MEO, MODIS Earth Obs.
*	OK	10	AVAIL (SNOW moved to ` ovly S)
\+	OL	11	Church
\,	OM	12	Girl Scouts
\-	ON#	13	House (H=HF) (O = Op Present)
\.	OO	14	Ambiguous (Big Question Mark)
\/	OP	15	Waypoint Destination (Note 1)
\0	A0#	16	Circle (IRLP/EchoLink/WIRES)
\1	A1	17	
\2	A2	18	
\3	A3	19	
\4	A4	20	
\5	A5	21	
\6	A6	22	
\7	A7	23	
\8	A8#	24	802.11 or other network node
\9	A9	25	Gas Station (blue pump)
\:	NR	26	AVAIL (Hail ==> ` ovly H)
\;	NS#	27	Park/Picnic Area + Overlay Events
\<	NT#	28	ADVISORY (one WX flag)
\=	NU#	29	APRStt Touchtone (DTMF Users)
\>	NV#	30	Cars & Vehicles *[with overlay]*
\?	NW	31	Information Kiosk (blue box with ?)
\@	NX	32	Hurricane/Tropical Storm
\A	AA#	33	Box: DTMF,RFID,XO *[with overlay]*
\B	AB	34	AVAIL (BlwngSnow ==> E ovly B
\C	AC	35	Coast Guard

Note 1: Red dot (with overlay) marks a mobile's destination. Drawn with a line between the mobile and its waypoint destination.

As a rough approximation, they can be categorized:

- Mobile Primary Symbols: !'<=>()*0CFOPRSUXY[\/^abefgjkpsuv
- Mobile Alternate Symbols: >KOS^ksuv
- Weather Primary Symols: _W
- Weather Alternate Symbols: ([*:<@BDEFGHIJTUW_efgptwy{

APRS SYMBOL TABLES (continued)

(Each highlighted character in the Alternate Symbol Table may be replaced with an overlay character).

PRIMARY SYMBOL TABLE			
/$	GPS xyz	GPS Cnn	Icon
/D	PD	36	
/E	PE	37	Eyeball (events, etc.)
/F	PF	38	Farm Vehicle (Tractor)
/G	PG	39	Grid Square (6-character)
/H	PH	40	Hotel (blue bed icon)
/I	PI	41	TCP/IP on air network station
/J	PJ	42	
/K	PK	43	School
/L	PL	44	PC user
/M	PM	45	MacAPRS
/N	PN	46	NTS Station
/O	PO	47	Balloon (SSID –11)
/P	PP	48	Police
/Q	PQ	49	
/R	PR	50	Recreational Vehicle (SSID –13)
/S	PS	51	Space Shuttle
/T	PT	52	SSTV
/U	PU	53	Bus (SSID –2)
/V	PV	54	ATV
/W	PW	55	National Weather Service Site
/X	PX	56	Helicopter (SSID –6)
/Y	PY	57	Yacht (sail boat) (SSID –5)
/Z	PZ	58	WinAPRS
/[	HS	59	Jogger, Human/person
/\	HT	60	Triangle (DF)
/]	HU	61	Mail/Postoffice (was PBBS)
/^	HV	62	Large Aircraft
/_	HW	63	Weather Station (blue)
/`	HX	64	Dish Antenna
/a	LA	65	Ambulance (SSID –1)
/b	LB	66	Bicycle (SSID –4)
/c	LC	67	Incident Command Post
/d	LD	68	Fire Department
/e	LE	69	Horse (equestrian)
/f	LF	70	Fire Truck (SSID –3)

ALTERNATE SYMBOL TABLE			
\$	GPS xyz	GPS Enn	Icon
\D	AD#	36	DEPOTS (Drizzle ==> ' ovly D)
\E	AE	37	Smoke (& other vis codes)
\F	AF	38	AVAIL (FrzngRain ==> `F)
\G	AG	39	AVAIL (Snow Shwr ==> I ovly S)
\H	AH#	40	Haze (& Overlay Hazards)
\I	AI	41	Rain Shower
\J	AJ	42	AVAIL (Lightening ==> I ovly L)
\K	AK	43	Kenwood ht (w)
\L	AL	44	Lighthouse
\M	AM#	45	MARS (A=Army,N=Navy,F=AF)
\N	AN	46	Navigation Buoy
\O	AO	47	Overlay Balloon (Rocket = \O)
\P	AP	48	Parking
\Q	AQ	49	Earthquake
\R	AR#	50	Restaurant
\S	AS	51	Satellite/PACsat
\T	AT	52	Thunderstorm
\U	AU	53	Sunny
\V	AV	54	VORTAC Nav Aid
\W	AW#	55	NWS Site *[with overlay]*
\X	AX	56	Pharmacy Rx
\Y	AY#	57	Radios and devices
\Z	AZ	58	AVAIL
\[	DS#	59	Wall Cloud (& humans w Ovrly)
\\	DT#	60	New overlayable GPS symbol
\]	DU	61	AVAIL
\^	DV#	62	other Aircraft ovrlys (2014)
_	DW#	63	WX Stn with digi (green) *[w/ ov'lay]*
\`	DX	64	Rain (all types w Ovrly)
\a	SA#	65	ARRL,ARES,WinLINK,Dstar, etc
\b	SB	66	AVAIL (Blwng Dst/Snd => E ovly)
\c	SC#	67	CD triangle RACES/SATERN/etc
\d	SD	68	DX Spot (from callsign prefix)
\e	SE	69	Sleet (& future ovrly codes)
\f	SF	70	Funnel Cloud

APRS SYMBOL TABLES (continued)

(Each highlighted character in the Alternate Symbol Table may be replaced with an overlay character).

PRIMARY SYMBOL TABLE			
/$	***GPS xyz***	***GPS Cnn***	***Icon***
/g	LG	71	Glider
/h	LH	72	Hospital
/i	LI	73	IOTA (Islands on the Air)
/j	LJ	74	Jeep (SSID –12)
/k	LK	75	Truck (SSID –14)
/l	LL	76	Laptop
/m	LM	77	Mic-repeater
/n	LN	78	Node (black bulls-eye)
/o	LO	79	Emergency Operations Center
/p	LP	80	Rover (puppy dog)
/q	LQ	81	Grid Square shown above 128m
/r	LR	82	Repeater
/s	LS	83	Ship (power boat) (SSID –8)
/t	LT	84	Truck Stop
/u	LU	85	Truck (18-wheeler)
/v	LV	86	Van (SSID –15)
/w	LW	87	Water Station
/x	LX	88	X-APRS (Unix)
/y	LY	89	Yagi at QTH
/z	LZ	90	
/{	J1	91	
/|	J2	92	*[Reserved]*
/}	J3	93	
/~	J4	94	*[Reserved]*

ALTERNATE SYMBOL TABLE			
\$	***GPS xyz***	***GPS Enn***	***Icon***
\g	SG	71	Gale Flags
\h	SH#	72	Store. or HAMFST Hh=HAM store
\i	SI#	73	BOX or points of Interest
\j	SJ	74	Work Zone (steam shovel)
\k	SK#	75	Special Vehicle SUV,ATV,4x4
\l	SL	76	Area Symbols (box, circle, etc)
\m	SM	77	Value Sign (3 digit display)
\n	SN#	78	Triangle *[with overlay]*
\o	SO	79	Small Circle
\p	SP	80	AVAIL (PrtlyCldy => (ovly P
\q	SQ	81	AVAIL
\r	SR	82	Restrooms
\s	SS#	83	Ship/Boat (top view) *[with overlay]*
\t	ST	84	Tornado
\u	SU#	85	Truck *[with overlay]*
\v	SV#	86	Van *[with overlay]*
\w	SW#	87	Flooding (Avalanches/Slides)
\x	SX	88	Wreck or Obstruction ->X<-
\y	SY	89	Skywarn
\z	SZ#	90	OVERLAYED Shelter
\{	Q1	91	AVAIL? (Fog ==> E ovly F)
\\|	Q2	92	*[Reserved]*
\}	Q3	93	
\~	Q4	94	*[Reserved]*

Originally, APRS had 192 symbols. It was expanded to thousands, in 2007, by allowing every alternate symbol to have 36 alphanumeric overlays.

The new symbol sets are extensible and evolving so they should not be compiled into applications. Instead, applications should read these files, or their equivalents, at runtime to make upgrading easier.

- Current SYMBOLS SPEC (version 1.1 plus non conflicting parts of 1.2).
- New version 1.2 symbols with new expandable Overlay blocks.

For more details see APRS Symbol Upgrade 2014 (Not part of 1.1, but a peek into where this is going.)

APPENDIX 3: 7-BIT ASCII CODE TABLE

In addition to listing the ASCII character codes in their usual form, this table also expresses the hexadecimal codes for the ASCII digits 0–9 and the upper-case letters A–Z in *shifted* form; i.e. shifted one bit left. This is particularly useful for decoding callsigns and Mic-E position information contained in the address fields of AX.25 frames.

Part 1: Codes 0–31 decimal (00–1f hexadecimal)

Dec	Hex	Char		
0	00	NUL	CTRL-@	
1	01	SOH	CTRL-A	Start of Header
2	02	STX	CTRL-B	Start of Text
3	03	ETX	CTRL-C	End of Text
4	04	EOT	CTRL-D	End of Transmission
5	05	ENQ	CTRL-E	Enquiry (Poll)
6	06	ACK	CTRL-F	Acknowledge
7	07	BEL	CTRL-G	Bell
8	08	BS	CTRL-H	Backspace
9	09	HT	CTRL-I	Horizontal Tab
10	0a	LF	CTRL-J	Line Feed
11	0b	VT	CTRL-K	Vertical Tab
12	0c	FF	CTRL-L	Form Feed
13	0d	CR	CTRL-M	Carriage Return
14	0e	SO	CTRL-N	Shift Out
15	0f	SI	CTRL-O	Shift In
16	10	DLE	CRTL-P	Data Link Escape
17	11	DC1/XON	CTRL-Q	Device Control 1
18	12	DC2	CTRL-R	Device Control 2
19	13	DC3/XOFF	CTRL-S	Device Control 3
20	14	DC4	CTRL-T	Device Control 4
21	15	NAK	CTRL-U	Negative Acknowledge
22	16	SYN	CTRL-V	Synchronous Idle
23	17	ETB	CTRL-W	End of Transmission Block
24	18	CAN	CTRL-X	Cancel
25	19	EM	CTRL-Y	End of Medium
26	1a	SUB	CTRL-Z	Substitute
27	1b	ESC	CTRL-[	Escape
28	1c	FS	CTRL-\	File Separator
29	1d	GS	CTRL-]	Group Separator
30	1e	RS	CTRL-^	Record Separator
31	1f	US	CTRL-_	Unit Separator

Part 2: Codes 32–127 decimal (20–7f hexadecimal), including hex codes for shifted 0–9/A–Z

Dec	Hex	Char	Shifted
32	20	˽	40/41 (space)
33	21	!	
34	22	"	(inv commas)
35	23	#	
36	24	$	
37	25	%	
38	26	&	
39	27	'	(apostrophe)
40	28	(	
41	29	)	
42	2a	*	
43	2b	+	
44	2c	,	(comma)
45	2d	-	(minus)
46	2e	.	(dot)
47	2f	/	
48	30	0	60/61
49	31	1	62/63
50	32	2	64/65
51	33	3	66/67
52	34	4	68/69
53	35	5	6a/6b
54	36	6	6c/6d
55	37	7	6e/6f
56	38	8	70/71
57	39	9	72/73
58	3a	:	
59	3b	;	
60	3c	<	
61	3d	=	
62	3e	>	
63	3f	?	
64	40	@	
65	41	A	82/83
66	42	B	84/85
67	43	C	86/87
68	44	D	88/89
69	45	E	8a/8b
70	46	F	8c/8d
71	47	G	8e/8f
72	48	H	90/91
73	49	I	92/93
74	4a	J	94/95
75	4b	K	96/97
76	4c	L	98/99
77	4d	M	9a/9b
78	4e	N	9c/9d
79	4f	O	9e/9f

Dec	Hex	Char	Shifted
80	50	P	a0/a1
81	51	Q	a2/a3
82	52	R	a4/a5
83	53	S	a6/a7
84	54	T	a8/a9
85	55	U	aa/ab
86	56	V	ac/ad
87	57	W	ae/af
88	58	X	b0/b1
89	59	Y	b2/b3
90	5a	Z	b4/b5
91	5b	[	
92	5c	\	
93	5d	]	
94	5e	^	
95	5f	_	(underscore)
96	60	`	(grave accent)
97	61	a	
98	62	b	
99	63	c	
100	64	d	
101	65	e	
102	66	f	
103	67	g	
104	68	h	
105	69	i	
106	6a	j	
107	6b	k	
108	6c	l	
109	6d	m	
110	6e	n	
111	6f	o	
112	70	p	
113	71	q	
114	72	r	
115	73	s	
116	74	t	
117	75	u	
118	76	v	
119	77	w	
120	78	x	
121	79	y	
122	7a	z	
123	7b	{	
124	7c	|	
125	7d	}	
126	7e	~	
127	7f	DEL	

APPENDIX 4: DECIMAL-TO-HEX CONVERSION TABLE

Dec	Hex
128	80
129	81
130	82
131	83
132	84
133	85
134	86
135	87
136	88
137	89
138	8a
139	8b
140	8c
141	8d
142	8e
143	8f
144	90
145	91
146	92
147	93
148	94
149	95
150	96
151	97
152	98
153	99
154	9a
155	9b
156	9c
157	9d
158	9e
159	9f

Dec	Hex
160	a0
161	a1
162	a2
163	a3
164	a4
165	a5
166	a6
167	a7
168	a8
169	a9
170	aa
171	ab
172	ac
173	ad
174	ae
175	af
176	b0
177	b1
178	b2
179	b3
180	b4
181	b5
182	b6
183	b7
184	b8
185	b9
186	ba
187	bb
188	bc
189	bd
190	be
191	bf

Dec	Hex
192	c0
193	c1
194	c2
195	c3
196	c4
197	c5
198	c6
199	c7
200	c8
201	c9
202	ca
203	cb
204	cc
205	cd
206	ce
207	cf
208	d0
209	d1
210	d2
211	d3
212	d4
213	d5
214	d6
215	d7
216	d8
217	d9
218	da
219	db
220	dc
221	dd
222	de
223	df

Dec	Hex
224	e0
225	e1
226	e2
227	e3
228	e4
229	e5
230	e6
231	e7
232	e8
233	e9
234	ea
235	eb
236	ec
237	ed
238	ee
239	ef
240	f0
241	f1
242	f2
243	f3
244	f4
245	f5
246	f6
247	f7
248	f8
249	f9
250	fa
251	fb
252	fc
253	fd
254	fe
255	ff

APPENDIX 5: GLOSSARY

Altitude	1. In Mic-E format, the altitude in meters relative to 10km below mean sea level.
	2. In Comment text, the altitude in feet above mean sea level.
Announcement	An APRS message that is repeated a few times an hour, perhaps for several days.
Announcement Identifier	A single letter A-Z that identifies a particular announcement.
Antenna Height	In NMEA sentences, the height of the antenna in meters relative to mean sea level. (The antenna height in GNSS NMEA sentences fluctuates wildly and should only be used if DGPS correction is applied).
APRS	Automatic Packet Reporting System.
APRS Data	The data that follows the APRS Data Type Identifier in the AX.25 Information field and precedes the APRS Data Extension.
APRS Data Extension	A 7-byte extension to APRS Data. The Data Extension includes one of Course/Speed, Wind Direction/Wind Speed, Station Power/Antenna Effective Height/Gain/Directivity, Pre-Calculated Radio Range, DF Signal Strength/Effective Antenna Height/Gain, Area Object Descriptor.
APRS Digipeater Path	A digipeater path via repeaters with station names or more often aliases.
APRS Data Type Identifier	The single-byte identifier that specifies what kind of APRS information is contained in the AX.25 Information field.
Area Object	A user-defined graphic object (circle, ellipse, triangle, box and line).
ASCII	American Standard Code for Information Interchange. A 7-bit character code conforming to ANSI X3.4 (1968) — see Appendix 3 for character definitions.
AX.25	Amateur Packet-Radio Link-Layer Protocol.
Base 91	Number base used to ensure that numeric values are transmitted as printable ASCII characters. To obtain the character string corresponding to a numeric value, divide the value progressively by decreasing powers of 91, and add 33 decimal to the result at each step. Printable characters are in the range **!**..**{**. Used in compressed lat/long and altitude computation.
Bulletin	An APRS message that is repeated several times an hour, for a small number of hours. A General Bulletin is addressed to no-one in particular. A Group Bulletin is addressed to a named group (e.g. WX).
Bulletin Identifier	A single digit 0-9 that identifies a particular bulletin.
Destination Address field	The AX.25 Destination Address field, which can contain an APRS destination callsign or Mic-E encoded data.
DF Report	A report containing DF bearing and range.
DGPS	Differential GPS. Used to overcome the errors.
DHM	7-character timestamp: day-of-the-month, hour, minute, zulu or local time.
DHMz	7-character timestamp: day-of-the-month, hour, minute, zulu only.
Digipeater	A station that relays AX.25 packets. A chain of up to 8 digipeaters may be specified.
Digipeater Addresses field	The AX.25 field containing 0–8 digipeater callsigns (or aliases).
Directivity	The favored direction of an antenna. Used in the PHG Data Extension.
DX Cluster	A network host that collects and disseminates user reports of DX activity.
ECHO	A generic APRS digipeater callsign alias, for an HF digipeater.
Effective Antenna Height	The height of an antenna above the local terrain (not above sea level). A first-order indicator of the antenna's effectiveness in the local area. Used in the PHG Data

	Extension.
ERP	Effective Radiated Power. Used in Status Reports containing Beam Heading and Power data (typically for meteor scatter use).
FCS	Frame Check Sequence. A sequence of 16 bits that follows the AX.25 Information field, used to verify the integrity of the packet.
GATE	A gateway between HF and VHF APRS networks. Used primarily to relay long-distance HF APRS traffic onto local VHF networks.
GGA Sentence	A standard NMEA sentence, containing the receiving station's lat/long position and antenna height relative to mean sea level, and other data.
GLL Sentence	A standard NMEA sentence, containing the receiving station's lat/long position and other data.
GMT	Greenwich Mean Time (=UTC=zulu).
GNSS	Global Navigation Satellite System. A GNSS is a constellation of satellites providing signals from space that transmit positioning and timing data to devices with the appropriate receivers. The receivers then use this data to determine a person's location. There are currently 4 GNSS constellations: GPS, Galileo, GLONASS, and BeiDou.
GPS	Global Positioning System. A global network of 24 satellites that provide lat/long and antenna height of a receiving station.
GPSxyz	An APRS destination callsign that specifies a display symbol from either the Primary Symbol Table or the Alternate Symbol Table. Some symbols from the Alternate Symbol Table can be overlaid with a digit or a letter. Used by trackers that cannot specify the symbol in the AX.25 Information field.
GPSCnn	An APRS destination callsign that specifies a display symbol from the Primary Symbol Table. The symbol can not be overlaid. Used by trackers that cannot specify the symbol in the AX.25 Information field.
GPSEnn	An APRS destination callsign that specifies a display symbol from the Alternate Symbol Table. The symbol can not be overlaid. Used by trackers that cannot specify the symbol in the AX.25 Information field.
HMS	1. In NMEA sentences, a 6-character timestamp: hour, minute, second UTC.
	2. In APRS Data, a 7-character timestamp: hour, minute, second, zulu or local.
ICQ	International CQ chat.
IGate	A gateway between a VHF and/or HF APRS network and the Internet.
Information field	The AX.25 Information field containing APRS information.
Item	A type of display object.
Item Report	A report containing the location of an APRS Item.
Killed Object	An Object that an APRS user has assumed control of.
knots	International nautical miles per hour.
KPC-3	A Terminal Node Controller from Kantronics Co Inc.
Longitude Offset	An offset of +100 degrees longitude (used in Mic-E longitude computation).
LORAN	Long Range Navigation System (a terrestrial precursor to GPS).
Maidenhead Locator	A 4- or 6-character grid locator specifying a station's position.
MDHM	8-byte timestamp: month, day, hour, minute (used in positionless weather station reports).
Message	A one-line text message addressed to a particular station.
Message Acknowledgement	An optional acknowledgement of receipt of a message.
Message Group	A user-defined group to receive messages.
Message Identifier	A 1–5 character message identifier (typically a line number).
Mic-E	Originally Microphone Encoder, a unit that encodes location, course and speed information into a very short packet, for transmission when releasing the microphone PTT button. The Mic-E encoding algorithm is now used in other devices (e.g. in the

PIC-E and the Kenwood TH-D7/TM-D700 radios).

Mic-E Message Identifier A 3-bit identifier (A/B/C) specifying a standard Mic-E message or custom message code.

Mic-E Message Code A 3-bit code specifying a Standard or Custom Mic-E message.

MIM Micro Interface Module. A complete telemetry TNC transmitter on a chip.

mph miles per hour.

Net Cycle Time The time within which it should be possible to gain the complete picture of APRS activity (typically 10, 20 or 30 minutes, depending on the number of digipeaters traversed and local conditions). Stations should not transmit status or position information more frequently unless mobile, or in response to a Query.

NMEA National Marine Electronic Association (United States). Producer of the *NMEA 0183 Version 2.0* specification that governs the format of Received Sentences from navigation equipment (such as GNSS and LORAN receivers). See Appendix 6 for a reference to NMEA sentence formats.

NMEA (Received) Sentence The ASCII data stream received from navigation equipment (such as GNSS receivers)conforming to the NMEA 0182 Version 2.0 specification. APRS supports five NMEA Sentences: GGA, GLL, RMC, VTG and WPL.

NRQ Number/Rate/Quality. A measure of confidence in DF Bearing reports.

Null Position Default position to be reported if the actual position is unknown or indeterminate. The null position is 0° 0' 0" north, 0° 0' 0" west.

NWS National Weather Service (United States).

Object A display object that is (usually) not a station. For example, a weather front or a marathon runner.

Object Report A report containing the position of an object, with optional timestamp and APRS Data Extension.

PHG APRS Data Extension specifying Power, Effective Antenna Height/Gain/Directivity.

PIC Programmable Interface Controller.

PIC-E A PIC implementation of the Mic-E microphone encoder.

Position Ambiguity A reduction in the accuracy of APRS position information (implemented by replacing low-order lat/long digits with spaces). Used when the exact position is not known.

Position Report A report containing lat/long position, optionally with timestamp and Data Extension.

Pre-Calculated Radio Range A station’s estimate of omni-directional radio range (in miles). Used in compressed lat/long format.

Query A request for information. Queries may be addressed to stations in general or to specific stations.

Range Circle Usable radio range (in miles), computed from PHG data.

Response A reply to a query.

RMC Sentence A standard NMEA sentence, containing the receiving station’s lat/long position, course and speed, and other data.

RTCM Radio Technical Commission for Maritime Services. The RTCM SC104 data format specification describes the requirements for differential GPS data correction.

Sentence See NMEA (Received) Sentence.

Signpost A special signpost icon that displays user-defined variable information (such as a

	speed limit or mileage) as an overlay.
Skywarn	A weather spotter initiative coordinated by the United States National Weather Service.
Source Address Field	The AX.25 Source Address field, containing the callsign of the originating station. A non-zero SSID specifies a display symbol.
Source Path Header	The digipeater path followed prior to a packet entering a Third-Party Network.
SPCL	A generic APRS destination callsign used for special stations.
SSID	Secondary Station Identifier. A number in the range 0-15, as an adjunct to an AX.25 address. If the SSID in a source address is non-zero, it specifies a display symbol. (This is used when the station is unable to specify the symbol in the AX.25 Destination Address field or Information field).
Station Capabilities	A list of station characteristics that is sent in reply to a query.
Status Report	A report containing station status information (and optionally a Maidenhead locator).
Switch Stream Character	A character normally used for switching TNC channels.
Symbol	A display icon. Consists of a Symbol Table Identifier/Symbol Code pair. Generically, **/$** represents a symbol from the Primary Symbol Table, and **\$** represents a symbol from the Alternate Symbol Table.
Symbol Code	A code for a symbol within a Symbol Table.
Symbol Table Identifier	An ASCII code specifying the Primary Symbol Table (**/**) or Alternate Symbol Table (****). The Symbol Table Identifier is also implicit in GPSCnn and GPSEnn destination callsigns.
Target Footprint	A target area for queries. The querying station asks for responses from stations within a specified number of miles of a lat/long position.
TH-D7	A combined VHF/UHF handheld radio and APRS-compatible TNC from Kenwood.
TM-D700	A combined VHF/UHF mobile radio and APRS-compatible TNC from Kenwood.
Third Party Network	A non-APRS network that does not understand AX.25 addresses (e.g. the Internet).
Third-Party Header	A Path Header with the Third-Party Network Identifier and the callsign of the receiving gateway inserted.
TNC	Terminal Node Controller. A combined AX.25 packet assembler/disassembler and modem.
Trace	An APRS query that asks for the route taken by a packet to a specified station.
Tracker	A unit comprising a GNSS receiver (to obtain the current geographical position) and aradio transmitter (to transmit the position to other stations).
Tunneling	Passing APRS AX.25 traffic through a third-party network that does not understand AX.25 addressing. The AX.25 addresses are carried as data (in the Source Path Header) through the tunneled network.
UI-Frame	AX.25 Unnumbered Information frame. APRS uses only UI-frames — that is, it operates entirely in connectionless (UNPROTO) mode.
UNPROTO Path	The digipeater path to the destination callsign.
UTC	Coordinated Universal Time (=zulu=GMT).
VTG Received Sentence	A standard NMEA sentence, containing the receiving station's course and speed.
WIDEn-N	A generic APRS digipeater callsign alias, for a digipeater with wide area coverage (N=0-7). As a packet passes through a digipeater, the value of N is decremented by 1 until it reaches zero. The digipeater keeps a record of each packet (excluding the digipeater via path) as it

passes through, and will not digipeat the packet again if it has digipeated it already within the last 28 seconds. A digipeater should insert its own callsign in the digipeater path so a receiving station will know the path taken by the packet.

WPL Sentence A standard NMEA sentence, containing waypoints.

WX Weather.

Ziplan A cheap twisted-pair LAN connecting PCs via their serial I/O ports. Designed for use in emergency situations.

Zulu UTC/GMT.

Units Conversion Table

To convert from	*to*	*multiply by*	*divide by*
feet	meters	0.3048	
meters	feet		0.3048
miles	km	1.609344	
km	miles		1.609344
miles	nautical miles	0.8689762	
nautical miles	miles		0.8689762
miles per hour (mph)	knots	0.8689762	
knots	miles per hour (mph)		0.8689762
knots	meters / second	0.51444'	
meters / second	knots		0.51444'
miles per hour (mph)	meters / second	0.44704	
meters / second	miles per hour (mph)		0.44704

Fahrenheit / Celsius Temperature Conversion Equations

$$F = (C \times 1.8) + 32$$

$$C = \frac{(F - 32) \times 5}{9}$$

APPENDIX 6: REFERENCES

AX.25 Link Access Protocol for Amateur Packet Radio, version 2.2, July 1998
http://www.tapr.org/pdf/AX25.2.2.pdf

NMEA 0183 Interface Standard
https://www.nmea.org/content/STANDARDS/NMEA_0183_Standard

NMEA Sentence Formats, in the *Garmin GPS25 Technical Reference Manual*
http://www.garmin.com/manuals/spec25.pdf

Maidenhead Locator, in the *IARU Region 1 VHF Manager's Manual*
http://www.scit.wlv.ac.uk/vhfc/iaru.r1.vhfm.4e/index.html

APRS NWS Weather bulletin format https://www.aprs-is.net/wx/

APRS-IS Specifications https://aprs-is.net/Specification.aspx

APPENDIX 7: DOCUMENT RELEASE HISTORY

Date	*Doc Version*	*Status / Major Changes*
10 Oct 1999	1.0 (Draft)	Protocol Version 1.0. First public draft release.
3 Dec 1999	1.0.1g	Protocol Version 1.0. Second public draft release. Much extended, incorporating packet format layouts, APRS symbol tables, compressed data format, Mic-E format, telemetry format.
30 Apr 2000	1.0.1m	Protocol Version 1.0. Third public draft release. Major additions/changes to the draft 1.0.1g specification: • Added a section on Map Views and Range Scale. • Changed Destination Address SSID description (specifying generic APRS digipeater paths) to apply to *all* packets, not just Mic-E packets. • Changed APRS destination "callsigns" to "destination addresses". • Added `TEL*` to the list of generic destination addresses. • Added brief explanations of how several generic destination addresses are used. • Added "Grid-in-To-Address" (but marked as obsolete). • Extended the description of the Comment field, with pointers to what can appear in the field. • Added explanation of base 91. • Added paragraph on lack of consistency in on-air units, and default GNSS datum = WGS84. • APRS Data Type Identifiers Table: marked Shelter Data and Space Weather as reserved DTIs. marked the `-` DTI as unused (previously erroneously allocated to Killed Objects). marked the `'` DTI to mean *Current* Mic-E data in Kenwood TM-D700 radios. marked the `` ` `` DTI as *not used* in Kenwood TM-D700 radios. • Position Ambiguity: need only be specified in the latitude — the longitude will have the same level of ambiguity. • Added the options of `.../...` and `␣␣␣/␣␣␣` to express unknown course/speed. • Added DFS parameter table. • Added Quality table for BRG/NRQ data. • Position, DF and Compressed Report formats: split the format diagrams into two parts (with and without timestamps). • DF Reports: added notes: BRG/NRQ data is only valid when the symbol is `/\`. CSE=000 means the DF station is fixed, CSE non-zero means the station is moving. • Compressed position reports: corrected the multiplication/division constants for encoding/decoding. • Mic-E chapter rewritten and expanded. Emphasized the need to ensure that non-printing ASCII characters are not dropped. Corrected the Mic-E telemetry data format. • Expanded the introductory description of Objects/Items. All Objects must have a timestamp. • Added Area Object Extended Data field to Object and Item format diagrams. • Added Object/Item format diagrams with compressed location data. • Killed Objects/Items: now indicated by underscore after the name. (continued on the next page)

Date	*Doc Version*	*Status / Major Changes*
	1.0.1m (continued)	• Re-categorized weather reports: Raw, Positionless and Complete. • Added a statement that temperatures below zero are expressed as -01 to -99. • Added the options of `...` and `␣␣␣` to express unknown weather parameter values. • Corrected the storm data format. Also, central pressure is now `/ppppp` (tenths of millibar). • Corrected the telemetry parameter data (now APRS *messages* instead of AX.25 UI *beacons*). • Added optional comment field to the Telemetry (`T`) format. • Added a section describing the handling of multiple message acknowledgements. • Added a section on NTS radiograms. • Added Bulletin/Announcement implementation recommendations. • Queries and Responses: Query Names (e.g. APRSD): all upper-case. A queried station need not respond if it has no relevant information to send. A queried station should ignore any query type that it does not recognize. APRSH: callsigns must be padded to 9 characters. • Added `PING` as a synonym of `APRST`. • Extended meteor scatter ERP beyond 810 watts, and added a lookup table. • Maidenhead Locator: all letters must be transmitted in upper case, but may be received in either upper or lower case. • Changed the definition of non-APRS packets — these are not APRS Status Messages, but may optionally be treated as such. • APRS Symbols chapter substantially rewritten.. • Added section on Symbol Precedence (where more than one symbol appears in an APRS packet). • Clarified some of the descriptions in the APRS Symbol Tables. • Added overlay capability to the `\a` symbol (ARES/RACES etc). • Separated the 7-bit ASCII table from the Dec/Hex (0x80-0xff) conversion table. • Added several new entries and a units conversion table to the Glossary. • Added new references to NMEA sentence formats and Maidenhead Locator formats.

Date	Doc Version	Status / Major Changes
29 Aug 2000	1.0.1	Protocol Version 1.0. Approved public release. Minor additions/changes to the draft 1.0.1m specification: • Added Foreword. • Replaced section on Map Views and Range Scale. • APRS Software Version No: added **APD**xxx (Linux aprsd server). • APRS Data Type Identifier: Designated **[** as Maidenhead grid locator (but noted as obsolete). • Position Ambiguity: added a bounding box example. • Compressed Position Formats: for course/speed, corrected the range of possible values of the "c" byte to 0–89. • Mic-E: replaced the latitude example table, to show more explicitly how the N/S/E/W/Long offset bits are encoded. • Mic-E: removed the paragraph stating that there must be a space between the altitude and comment text — no space is required. • Mic-E: removed the note on inaccurate altitude data, as GPS Selective Availability has been switched off. • Object Reports: added timestamps to some of the examples (an Object Report must always have a timestamp). • Signposts: can be Objects or Items. • Storm Data: changed central pressure format to **/**pppp (i.e. to the nearest millibar/hPascal). • Storm Data: Hurricane Brenda examples: inserted a leading zero in the central pressure field (central pressure is 4 digits). • Telemetry Data: Added **MIC** as an alternative form of Sequence Number. **MIC** may or may not be followed by a comma. • Messages: added the reject message format. • Appendix 1: Agrelo format: changed the separator between Bearing and Quality to **/**. • Symbol Table: changed **/(** symbol from "Cloudy" to "Mobile Satellite Ground station". • Reformatted the Units Conversion Table.

Date	***Doc Version***	***Status / Major Changes***
18 Feb 2024	1.1	Merged in updates from http://www.aprs.org/aprs11.html This copy of the APRS SPEC Addendum 1.1 was approved by the APRS-**WG** . (Vote was called 30 July 2004). This APRS1.1 page is always maintained current such as the SYMBOLS and TOCALLS links below. All new updates and additions to APRS since 2004 are found on the **APRS 1.2** proposed draft addendum. **APRS SPEC FINALIZED as of July 2004:** This APRS Specification version 1.1 represents additions, corrections, and deletions since the original APRS1.0 spec of 21 June 2000. This edition represents the state of the APRS protocol and its usage through July 2004. As of this date, the **state of the APRS users** showed almost 27,000 stations worldwide of which: • 33% are fixed or home stations (includes WX stations) • 36% are infrastructure (30% digis and 6% IGates) • 31% are mobiles **CORRECTIONS:** • Page 18 X1J exception. Abandoned/deprecated in 2012. Now the ! DTI is only at the beginning. ✓ • Page 55 Mic-E Altitude. Optional altitude should be *first* after the Mic-E **type** byte. (*type* byte was extended in addendum 1.2) ✓ • Page 25 refers to a 'WPT' NMEA sentence type. Curt, WE7U assumes that should be 'WPL'? ✓ • The Antenna Gain in the PHG format on page 28 is in dBi. ✓ • Page 34 DF NRQ is not defined in spec. See original DF.TXT and PROTOCOL.TXT ✓ • Page 45, We now avoid the term "Mic-E Messages" and use the term "Mic-E Position Comment" for these bits. ✓ • Page 48, 100-109 goes to "u" instead of DEL and 110-179 is 38-107 going to "k" instead of DEL. ✓ • **AX-25 Table:** The AX-25 UI-Frame TABLE lists Flags at the end of a packet being 2 bytes. . . The correct number is 1, and it may be shared with the next packet. (From VK2TDS). ✓ • **Default Paths of WIDE2-2, not RELAY or WIDE.** (page 11) (updated June 2006) ✓ • Normally, **HID should be OFF** in all APRS TNC's (page 91). ✓ • **SSID Conventions** in user calls (page 95) ✓ • **DATUM:** Standard is WGS-84, but Continental options such as OSG for the UK are OK.(page 21) See !DAO! concept in APRS1.2 ✓ • **Range Scale:** The standard *view size* descriptor for APRS displays. (**map example**). (page 11) ✓ • **Time Stamps on receipt** (page 23). ✓ **SYMBOL UPDATES:** • **See all about Symbols!** ✓ • **SSID Conventions** for quick visual identification ✓ • **Updated SYMBOL tables**. . But see also all the APRS1.2 proposals **for new Symbol Extensions** ✓ • **Overlayable Symbols** subset. . These were always defined as overlays in the original APRS. ✓ • **Upgrading your symbol set** gives background on fixing up your symbol set. ✓ • **JUST MOBILE PRIMARY SYMBOLS:** *!'<=>()*0CFOPRSUXY[\^abefgjkpsuv* <== [added !F\ ] ✓ • **JUST MOBILE ALTERNATE SYMBOLS:** *>KOS^ksuv* <==[removed /0An] ✓ • **JUST WEATHER PRIMARY SYMBOLS:** *_ and W* ✓ • **JUST WEATHER ALTERNATE SYMBOLS:** *([*:<@BDEFGHIJTUW_efgptwy{* ✓

- **Timeout Old Stations (fade-to-gray)**. Now 80 mins instead of 2 hrs to account for stations via satellites (page 10). ✓
- **Symbol Attributes** for map displays. (page 92) And Position Ambiguity (p-24) ✓
- **DIGI Overlay Characters** (new for page 11) ✓
- **Have PHG Range Circles** to account for real-world PHG range due to multipath & fading. ✓ (page 29) **See all about PHG** ✓
- **New WAYPOINT symbol:** Red dot (with overlay) marks a mobile's destination. Drawn with a line between the mobile and its waypoint destination. ✓

OBJECT (and NAME) CLARIFICATIONS:

- **Name/Call equality and SPACES-in-names Manifesto** ✓
- **OBJECTS:** Amplifying comments on ownership, killing and equality with stations.(page58) ✓
- **OBJECT names should not have any punctuation** that cannot be converted to valid Wapoints on most GNSSs. That would depend on GNSS Vendor. Where is this defined? ×
- **AREA Objects** are poorly described in the spec. (page 60) ✓
- **Polygon and Line OBJECTS** (new for page 31) Link is now longer valid. Same information is now at More Information about APRS NSW Weather can be found at https://www.aprs-is.net/wx/ ✓
- **Compressed Objects:** not recommended for use on RF due to incompatibilities (page 58) For 1 foot precision, use $GPGGA, or 3rd party compressed posits, or the proposed !DAO! format. ✓
- **ITEM Format** is not recommended on RF due to incompatibilities (page 37) ✓

UPDATED TABLES and INFO:

- **WB2OSZ description of SSID - C bits** ✓
- **TO-CALLS and Version ID's** updated (page 14) ✓
- **Experimental Formats** (page 89) ✓

WEATHER RELATED ISSUES:

- **Raw Weather Formats** not recommended. Microprocessors should convert to *complete* format on RF (pg 62) ✓
- **Weather Details**. Amplifying comments on the original spec 1.0. (page 64) ✓
- **The WXSVR Protocols** managed by Dale Hugley. **See example**. See description: **MOBILE.TXT**. ✓
- APRS-IS (Internet System): ✓
- **NOGATE and RFONLY** in the RF DIGI field should not be forwarded into the APRS-IS by IGates. ✓
- **!x!** means **no archive**. Any packet containing this string should not be archived by any of the APRS-IS data bases. Positions, status or messages. ✓
- **APRS-IS Core** and **Tier-2** servers web pages and **how they work**. ✓
- **The q-construct:** Marks the source of entry of all packets into the APRS-IS. ✓ (Referred reader to the APRS-IS documentation.)
- **IGATE Status Report Format:** Left_bracket then *"IGate, MSG_CNT=N, LOC_CNT=N"* ✓

APRS 1.1 SPEC Operating Conventions for the Good of the APRS Network:

- **The New-N Paradigm** obsoleted old RELAY and WIDE and TRACE for 300-500% network improvement ✓
- **APRS Voice-Alert** for instant voice contact with any APRS mobile
- **Reply-ACKS Algorithm:** Really makes message QSO's FLY! (page 73) ✓
- **Recommended DIGI paths** and striving for *network protection* ahead of too much user flexibility. ✓
- **Digipeater ID rates** for optimum APRS networks ✓
- **Auto-Answer messages** are SPAM to the network in most cases. To limit their impact, all original APRS clients adhered to these rules for auto-answer messages:

		1) Any such AA message text should begin with AA: 2) Any such AA message should not have a Line number (so it does not cause acks) 3) Any such AA message is only sent once on each incoming message packet (no forced retries) 4) Any such AA message should default to OFF on power up. 5) Any such AA message should be canceled when the client detects the return presence of the operator 6) Any such AA message can be ended with a }yy REPLY-ACK if the software supports it. Auto-Answer seems to be a Kenwood invention. It does not belong here but we should start compiling a list of non-standard extensions. ✕ **The above is the complete APRS 1.1 Spec Addendum.** It was a compilation of all the accumulated feedback from users and authors about errata, errors, typos and any omissions in the spec that had been accumulated from the time of the original spec to June 2004. This APRS errata page had been running continuously for years and was updated whenever these items were discovered and each of these items were widely published and discussed on the APRSSIG and the APRSSPEC working group for public discussion. But it was decided best to finally Freeze the accumulation in June 2004 as APRS1.1 and then after public posting approve it. Then to begin working on any new issues as APRS1.2.

Date	Doc Version	Status / Major Changes
18 Feb 2024	1.2	Merged in updates from http://www.aprs.org/aprs12.html **APRS 1.2 SPEC ADDENDUM PROPOSALS:** The APRS protocol was well defined by the 1996 time frame in the original APRSdos docs. But it was then formalized in the **APRS1.01** publication in 2000. The first **APRS 1.1 ADDENDUM** was completed and approved in July 2004. . This APRS1.2 addendum contains all the spec updates since then. . For a comparison of what has been implemented in what clients, please see the Capabilities Chart. But first we will list as a reminder all of those items that have already been included in **1.1:** *Editor's Note: Addendum 1.2 is much different than 1.1. Much of it is redundant, recapping what was in 1.1, or "...ideas that have been proposed for further discussion and eventual adoption ..." I've tried to select only the actual specification additions which have been implemented.* **SPEC ADDITIONS:** • **Proposal (28 Apr 2020):** Allowing 000-999 in addition to the original 000-255 telemetry fields in the five channel T#sss,111,222,333,444,555... format. ✓ • **PHG in Mic-E format**. The Mic-E text field can contain any normal Position comment field too. Such as PHG. (important for Mic-E hardware digipeaters). ✓ • **Original Mic-E Telemetry Format is deprecated.** Overtaken by **MFR TYPE codes.** A new Telemetry format has been developed **See spec** . ✓ • **High precision !DAO! & datum option** with precision to 1 ft with included datum. ✓ • **Adding Frequency to packets** Lets you find someone in the Freq domain! ✓ • **New UTF-8 text encoding and decoding** where possible. ✓ • **Voice Frequency Format** for including operating frequency in APRS. **see spec**. ✓ • **Mic-E TYPE Codes** for different capabilities. See a set of **Mic-E Examples!**. ✓ • **Probes (PHGR)** addition to **PHG** for real-time measuring of Network Reliability. ✓
	1.2b	• Explain p-persistent CSMA for channel access. This was probably not in the original because everyone was using legacy TNCs at the time and the TNC took care of it.
	1.2c	• Changed GPS to GNSS because there are now other similar systems. • Barometric pressure field should be a total of 6 characters, not 5. • Overhauled Third-Party Packet section. • Add metric units to PHG table.

END OF DOCUMENT