

Update 2017-05-04

APRSIS32

Manual

APRSIS32

wiki dot

on Paper

&

Extended Explanation of APRS

This document is a compilation of the content of APRSIS32 WIKI website

<http://aprsisce.wikidot.com>

and platform

<http://groups.yahoo.com/group/aprsisce>

Furthermore, given content of other websites to deepen the knowledge about APRS

List of hams having done major parts of these reproduced textes and comments:

Lynn W. Deffenbaugh	KJ4ERJ	APRSISCE/32
Steve Daniels	G6UIM	WIKIdot:APRSCE Presentation
Bob Bruninga	WA4APR	APRS Developement & APRS® Holder
Pete Loveall	AE5PL	javAPRSSrvr Copyright© Holder
Roger Bille	SM5NRK	javAPRSFilter Users Guide
Stephen Smith	WA8LMF	APRS Symbols (Rev. H)

The program is known as APRSISCE, APRSISCE/32 and APRSIS32.

This document will use the term APRSIS32

For corrections, amendments, revisions and notes email Helge@SA7SKY.net Helge C. Hartz, SA7SKY

APRSIS32 Tutorials on Video in German language under <http://aprsisce.wikidot.com/de-start>

Daily **Developements** and **Corrections** Online!

<http://sa7sky.net/documents/APRSIS32-Manual.pdf>

Index

Transmit.....	9
Enables	9
APRS-IS Enabled.....	9
AutoSave GPX.....	9
Beaconing Enabled.....	9
CSV Enabled	9
Frequency Monitor	9
GPS Enabled.....	10
Internet Access.....	10
OSM Fetch Enabled.....	10
Sound Enabled.....	10
Telemetry Enabled	10
Ports	10
All.....	10
Log All	10
Reiceive-Only	11
Logging	11
All At Startup.....	11
File Enabled	11
All OFF.....	11
View Logs.....	11
Messages.....	11
Read Message	13
Send Message...	13
Send EMail...	13
Pending Messages.....	13
Announcements.....	15
Active Groups.....	15
Describe Groups	15
My Groups	15
Join Group...	15
Keep Active.....	15
Quietly.....	15
Use CQSRVR.....	15
Only if present	15
IS Only	16
CQ Text.....	16
Accept.....	16
Send.....	16
Delete	16
Auto-Reply	17
Message.....	17
Inactive For.....	17
No More Than	17
Bulletins	18

View	18
Notify.....	18
Create.....	18
Chats	18
Screen.....	19
Dead Reckoning.....	20
Filter Circle (r///nkm)	20
Tracks.....	20
AutoZoom View	20
Brightness	20
Direction Finding.....	21
Follow	21
Find.....	21
Centre	21
ME.....	21
Locked.....	21
Labels	21
Altitude	21
Ambiguity.....	21
Callsign.....	21
Footprint.....	21
Nicknames	22
NWS Calls	22
Speed	22
Weather.....	22
Paths	22
Appearance.....	22
LclRF Only.....	22
Network (Links).....	22
Station (Packet).....	22
ME (Packet)	22
PHG Range Circles	22
Preferred.....	23
TileSet.....	23
View.....	23
ALL	23
NONE.....	23
Chooser.....	23
Altitudes.....	23
Ambiguous	23
Buddies	23
Duplicates	24
FreqSpec.....	24
Freq w/Issues.....	24
Invalids	24
Message-able.....	24
Nicknamed	24
NWS Objects	24

Objects	26
Paths	26
Telemetry.....	26
Tracks	26
Mic-E.....	26
Platforms.....	33
RF.....	33
Shrieks.....	33
Symbols	33
Transport	34
Echolinks.....	34
GeoCaches... ..	34
Configure.....	34
General	34
GeniusBeaconing™ ..	52
Status... ..	54
Aliases	55
AltNet.....	56
Beacon	61
After Transmit	61
Timestamp.....	61
HHMMSS.....	61
Altitude	61
Compressed	61
CourseSpeed	61
Comment	61
Mic-E Notification.....	62
Symbol	62
Why.....	62
Precision	63
Path.....	63
Companions.....	69
DX	69
Map	70
Prefetch	70
Purger Enabled.....	74
Tile Sets.....	74
Messages	80
My Messages.....	80
RF Eavesdrop	80
Eavesdrop	80
But Not NWS.....	80
Hide Queries	80
Notify Query.....	80
Notify New	81
MultiTrack™ New Items	81
Lookup(WHO-IS)	81
Metric.....	81

Mic-E Notifications	81
NWS (http://www.aprs-is.net/wx/).....	83
New Office.....	87
Messages	87
Messages Not *ALL*	87
Messages Notify	87
Notify Line Type	88
Notify Products.....	88
Popup MultiTrack™	90
Show Offices	90
NWS Entry Servers	90
NWS Shapes (http://tinyurl.com/NWS-Shapes)	90
Add File...	92
Enabled	92
Opacity... (10%).....	92
Quality... (100%).....	92
Show Center	92
Objects	92
Create (0 Objs)	92
New JT65.....	94
New Weather	95
Show	95
Max Group Objects (5 Objs)	95
QRU.....	95
Enabled	97
Interval (0 min).....	98
Max Objects.....	98
Range.....	98
Retry Messages.....	98
Overlays.....	107
Add GPX File...	107
Add POS File...	109
Ports	116
New Port.....	118
Simply(KISS).....	118
KISS.....	118
AGW.....	118
FTM350v1.3(RO+GPS)	118
KWD700(KISS).....	120
KWD700(RO+GPS).....	120
KWD710(APRS).....	120
KWD710(Pkt).....	122
CONVerse	123
TEXT	123
CWOP.....	123
IS-Server	124
Local-Server.....	124
APRS-IS	124

NMEA	124
Save Posits	124
Reloaded Paths	124
Save Filter... (0/0)	124
Screen.....	125
Label	125
Altitude	125
Battery.....	125
Circle	125
Crosshairs	125
Date/Time.....	125
GridSquare.....	126
Lat/Lon	126
Orientation.....	126
RedDot	126
Satellites (GPS).....	126
Speed Font Size... (5)	126
Symbol Size... (1)	126
Tracks.....	126
Scroller.....	127
Freeze On Click	128
Show IGate/Digi	128
Filter.....	128
Hide NoParse	128
No Intervals	128
Not Me.....	129
Not Mine	129
RF Only.....	129
Show All.....	129
Status Report	129
Enabled	129
GridSquare.....	129
Timestamp.....	129
DX ()	129
New.....	130
None.....	130
Clear	130
Clear Eavesdrops ()	130
Clear CQ/ALL Messages ()	130
Clear Group Messages ()	130
Clear My Messages ()	130
Clear NWS Messages ()	130
Clear Paths.....	130
Clear Stations ()	130
Clear Tracks	130
Accumulated	130
Configured	131
About	131

Exit	132
Igating	132
Satellite Operation.....	136
Annex.....	139
KISS Mode XML Configuration OpenCmd ⇨ QuietTime	139
TH-D7.....	139
TH-D72.....	139
TM-D700.....	140
TM-D710 (APRS)	140
TM-D710 (PKT)	141
TNC SCS-DSP	142
TNC Tiny2.....	142
TNC TT4	142
SCS PTC-IIIusb	142

Transmit

Causes APRSIS32 to transmit an immediate position packet based on the beaconing configuration. If a GPS is not currently enabled, **Transmit** also allows the current station (ME) to be moved to the center of the screen after a confirmation has been given.

Enables

The **Enables** menu allows you to enable (and disable) various settings within APRSIS32. Currently the TraceLogs are also accessed via this menu.

APRS-IS Enabled

This option enables a live connection to the configured APRS-IS server. Access to the server also requires that Internet Access also be enabled.

AutoSave GPX

If checked, a .GPX file is automatically saved containing your recorded track if your station moves. The file will be named **<callsign-SSID>-YYYYMMDD-HHMM.gpx** and stored in the same directory as the XML configuration file. The "Save Track" on a station popup menu will record the same format file for any station.

Beaconing Enabled

If checked, allows APRSIS32 to generate position updates (beacons) for transmission via APRS. Individual settings on the APRS-IS and RF Ports control where such beacon packets are actually transmitted.

CSV Enabled

If checked, causes APRSIS32 to write a **YYYYMMDD-HHMM.csv** (Comma-Separated Values) file containing each unique location where a beacon was triggered. This value is written whether or not an actual beacon was actually transmitted.

Frequency Monitor

Enable will create a new window for easier spotting of stations transmitting frequency information, especially for mobile operators.

Right click on a Frequency Monitor window to configure it. Selectable options are:

- MultiTrackTM**
- Auto-Close Timer...**
- Font Size...**
- Set Filter...**

Selecting **Enable** ⇒ **Frequency Monitor** will create the window if it does not exist, rather than requiring you to disable and re-enable it to make it visible for re-configuration.

The most recently displayed frequency is remember if you use the above trick to re-display the Frequency Monitor. You should only see the "Sample Frequency" of 146.520 the first time you enable it after a restart.

The Frequency Monitor shows the range and direction to the repeater similar to the "FromME:" in a station information popup.

GPS Enabled

Enables or disables the GPS. If this option is disabled, then a GPS has not yet been configured via **Configure** ⇒ **Ports** ⇒ **NMEA** .

Internet Access

If this option is not checked, all Internet access by APRSIS32 is suspended. This will disable APRS-IS, Map Tile fetching, and Version update checking.

OSM Fetch Enabled

Enables the fetching of new map tiles if a required one is not available in the local cache. Disabling Internet Access also disables this function. The tile purger is also disabled if this function is not checked. (See also **Configure** ⇒ **Map** ⇒ **Purger**)

Sound Enabled

Enables or disables sounds generated by APRSIS32.

Telemetry Enabled

If checked, allows APRSIS32 to generate telemetry packets for transmission via APRS. Individual settings on the APRS-IS and RF Ports control where such telemetry packets are actually transmitted. Even if telemetry is disabled, it will be processed locally to display if requested.

Ports

Enables the data ports, KISS and IP ports, APRS-IS and NEMA ports are enabled elsewhere.

All

Enables all ports listed in the menu.

Log All

Enable logging of all the listed ports.

Reiceive-Only

Disables transmission on the listed ports.

Logging

Enable debug logging of various data.

All At Startup

Activates all of the defined ports upon start of the program. If a port does not connect to its device the program will start anyway but it continues the effort to open the port during the session.

File Enabled

TBD

All OFF

None of the ports are used.

View Logs

View the various debug logs if they are enabled.

Messages

The Messages menu provides access to the APRS message-based features of APRSIS32. If you're not familiar with APRS messaging, you may also want to read [aprs-messaging-explained](#).

Advanced Knowledge Base

APRS Messaging Explained

APRS messaging is a complicated beast that needs a whole bunch of components to be in place and working in order for it to succeed. I'll build up to the full story starting with RF to RF simplex (but your stations have to be in range of each other) and end with RF to IS to RF.

As you might know, APRS uses packet radio's UI packet which is connectionless, non-sequenced, non-guaranteed delivery, and non-acknowledged. Basically, in APRS, a station throws a packet onto the airwaves and has no clue where it went if anywhere. Good for general beaconing, but not so good for messaging. APRS also uses a path similar to packet in that it is in the AX.25 header, but it can contain aliases which to which multiple digipeaters will respond to repeat a packet over larger areas (like WIDE1-1,WIDE2-1, and so forth).

Ok, so an APRS message is actually a standard APRS UI packet, but the destination callsign of the message isn't in the AX.25 header, but is within the data payload, delimited by colons (:KJ4ERJ-15: for instance). APRS messages can be sent either asking for an acknowledgment or not. If an ack is desired, the sender will put a {xxxxx at the end of the message text. The ack sent back by the recipient is just another APRS message directed to the originator with the text of ackxxxxx (and of course NOT including an { ack request!).

So, in an RF simplex situation, the originating APRS station simply sends out an APRS message (typically formatted by an APRS client and sent out through some sort of TNC), the receiver has to hear the packet, decode it successfully in some sort of TNC (AX.25 packets have a checksum), optionally format an ack message, send it back onto the RF where the original sender has to hear it, decode it, and match it up with the original message. Typically APRS client software (even if that software is embedded in a transceiver like a D7, D700, D710 or VX-84) will implement some number of retransmissions until the ack is heard. If a reply is desired, simply reverse the roles of the stations and repeat. And that's the simple case!

Enter the APRS-IS, the Internet side of APRS. There is no "central" APRS server as some people believe. The APRS-IS is simply a series of connections through which APRS packets flow in realtime. Sites like aprs.fi and findu.com are not really part of the APRS-IS, but monitor the packets going through the APRS-IS and store them into a database. The APRS-IS itself stores nothing. It's just a transport.

Note: In order to do direct messaging via the APRS-IS, a verified connection is required. This means you have to have a password that matches your callsign. -1 as a password will not work for messaging as the APRS-IS will not accept any of your packets.

So, sending an APRS message from one -IS connected station to another -IS connected station is just like the RF simplex situation above, but without the need for a TNC and without the possibility of needing to retry in the event of checksums. The -IS only delivers good packets. However, the recipient of a message must be on-line and connected for a successful message interchange. There is no store and forward APRS message delivery (yet).

What about RF to -IS? What's an IGate? How does it figure into messaging? Well, an IGate is simply an APRS RF station that listens for packets on the radio, decodes them, validates their checksums and injects them into the APRS-IS for transport. Once injected, only one copy of any given packet is forwarded to all currently connected -IS nodes (well, there's some filtering in there, but messages go everywhere for the most part). That's the receiving part of an IGate. All IGates at least do that much. IGates can also be configured to gate selected APRS packets from the -IS back out to RF. Typically, this is just APRS message traffic, but other APRS messages can also be gated from the -IS to RF. But wait, you say, the -IS is much faster than the 1200 baud (or 300 on HF) APRS RF channel. You're right, so not all messages get gated. IGate software watches the stations that have been heard on the RF channel and records a list of "recently" (typically 30 minutes) heard "local" stations (limited by used hops). Only messages heard on the -IS targeting one of these recent local stations will be gated to RF.

So, for an -IS station to send a message to an RF station, the APRS client software creates a message for another station and injects it into the APRS-IS. This message goes everywhere and all connected IGates check to see if it trying to go to a local station that it has recently heard. Every IGate that has recently heard the station locally will put the message onto RF. Well, not every IGate, unfortunately, but only those configured to gate APRS messages from the Internet to RF. Once the message is on RF, the intended recipient must hear the packet, decode it, validate the checksum, and issue an ack if requested by the originator. This piece is identical to the RF simplex case above, but this time an IGate (any receiving IGate, not necessarily the one that gated the outbound message) needs to hear, decode, validate, and inject the ack back into the APRS-IS so that the originating client gets the ack and doesn't retransmit.

Ok, so for -IS to RF, we needed at least one bi-directional IGate that had heard the destination station locally recently. What about RF to -IS?

Simply pull the right parts from the above discussions and you'll have it. The RF station formats the message and transmits it. An IGate hears it, decodes it, verifies the checksum and injects it into APRS-IS. Once there, if there is a client connected to the APRS-IS with the destination callsign, it will hear the message and format an ack if requested by the originator. That ack goes back into the -IS as just another message to be gated. Some IGate must have heard the originating station recently locally and will transmit that message back onto the RF where it must be heard, decoded, and verified. If the

ack doesn't make it back out (for instance, because it's only in range of a receive-only IGate), the RF originator will continue retrying the message until the retries are exhausted. The APRS-IS receiving station will receive multiple copies of this message, issuing an ack for each one that never makes it back out onto RF.

So, what about RF to IS to RF (which is what you'll need for someone in Florida to APRS message someone in Virginia unless you're doing it simplex over HF. Here's the sequence:

- 1) Originating station formats an APRS message for the intended recipient
- 2) IGate hears the message, decodes it, verifies checksum and injects into APRS-IS
- 3) The whole world hears the message including a few IGates in Virginia
- 4) Hopefully one of those IGates has recently heard the recipient local enough AND that IGate is configured to gate messages from -IS to RF
- 5) The IGate transmits the message onto RF
- 6) Hopefully the recipient hears the message, decodes it, verifies the checksum, and displays it to the operator.
- 7) Because most of the time acks are requested, the receiver formats an ack message and sends it over RF.
- 8) Repeat step 2 in Virginia
- 9) Repeat step 3 moving to Florida
- 10) Repeat step 4 in Florida (remember, an ack is just another APRS message)
- 11) Repeat step 5
- 12) Repeat step 6 but don't display to the operator, just mark the message received and quit retransmitting.

So you see, there's a WHOLE LOT of stuff that has to a) exist (IGates) b) be configured (gate messages IS to RF), c) work (decoding in remote stations) to even get ONE message through RF-IS-RF. Now, do that round trip for EVERY SINGLE message in a QSO, and you can see the miracle that APRS messaging QSOs represent!

Lynn, KJ4ERJ

Read Message

Brings up the highest priority unread message for a preview or directly into an existing chat window.

Send Message...

Allows entry or selection of a station for which a new Chat window will be opened.

Send Email...

Allows entry or selection of a destination e-mail address to which a short (fitting the subject line of an email program) message may be sent. See also Email.

Pending Messages

If enabled, will initiate a popup menu of sent messages for which an ack has not been received. This will color the Message yellow or orange. See also Message Retries.

Advanced Knowledge Base

Message Retries

So, how does APRSIS32 handle message retries? And how can I tell what's going on? I'm glad you asked!

The first indication that the client has outbound messages on the retry queue is that there's either a Yellow background on "No Msg" or an Orange background on Callsign-SSID in the message pane. Yellow means there's only outbound messages pending and Orange means there's both unread inbounds and outbound pending.

If you double-click on the Yellow pane, or bring up Messages / Pending Messages, one of two things will happen. If there's only one pending message, it will come up for viewing and disposition. If you have more than one, a popup menu will be presented with the pending callsigns. Selecting one of the callsigns from the popup will present the corresponding message.

A Pending Message dialog will show the recipient of the message, the body of the message, the retry status, and offer to Abort, Retry or Ignore. Ignore simply dismisses the dialog and things proceed as if you hadn't looked. Abort will delete the outbound message and kill all pending retries. Retry will cause the message to be immediately retried as if it was Retriggered (see below).

So, what's with all of the Retry, Final Retry, Second Retry, and FINAL retry?

APRSIS32 uses a decaying retry algorithm originally outlined by Bob Bruninga. As of this writing (12/10/2010), it does a total of 7 retries starting at 8 seconds and doubling through a total of 7 retries. This gives a total retry set of $8+16+8+32+64+96+128$ seconds or just under 6 minutes total. That's the main Retry phase. After exhausting that sequence, the dialog will say "Final Retry (N) sent M minutes ago" where M increases with time.

The 8 after the 16 was added on 12/10/2010 after someone noticed that the first retry through a remote IGate was over a minute after the first transmission. This was due to $8+16=24$ being just under the 30 second dupe detector and the next retry after 32 more seconds or 56 seconds after the first transmission. The additional 8 gives APRS-IS retries at 32, 32, 64, 96, and 128 seconds allowing about as quick of a retry as APRS-IS will allow.

And on 10/24/2011 (or so), I discovered that APRS-IS seems to reset the 30 second dupe filter on each duplicate reception totally 'nixing the previous paragraph's assumptions. So, APRSIS32 now suppresses all <30 second message retries to -IS so that the first retry >30 seconds ($T+32$ after $+8+16+8$) will actually pass through the dupe filter instead of being suppressed because of a dupe-filtered retry only 8 seconds prior. Before this correction, the first APRS-IS passed retry was actually 64 seconds ($+8+16+8+32$) seconds after the initial transmission. WAY longer than necessary or desired.

Now, here's the Retrigger feature, known as "Message-On-Heard" on page 10 (20 in Acrobat) in aprs101.pdf (the "bible" of APRS). If a beacon is heard from a station that has exhausted the Final Retry (or you hit the Retry button), a whole new set of retries is initiated. This is indicated by "Second Retry n of m in s seconds" being displayed in the dialog. When the full set of retriggered retries have been exhausted (another 7 transmissions over 6 minutes), the dialog will say "FINAL Retry (N) sent M minutes ago" and stay that way until you either manually re-retrigger (Retry) it or Abort it.

Multiple messages for a single recipient are delivered in the order they were entered. This means that if a particular message to a specific recipient exhausts its retries, all subsequent pending messages will be blocked for that recipient. The Pending Messages list will show all messages for a given recipient, but only the oldest one is actually being transmitted and retried.

Now, which interfaces are messages sent over, you might ask? Here's the deal. The second (retriggered) set of retries are sent over all enabled ports. This is regardless of where or if the station was previously heard. Initial message transmissions and acks are only sent "smart". This means that if the station has not been heard, APRSIS32 doesn't know where it is and sends over all available ports. If the station is known and has not been heard on RF, the message is sent only via APRS-IS. If a station has been heard on RF, the message and/or ack is sent over RF and APRS-IS. Note that

currently (10/24/2011) RF means ALL configured and enabled and message-enabled RF ports, regardless of which one the station was heard over.

One final caveat: If you originally specified that a message be sent RF Only, that will be honored throughout both retry sets

Announcements

This messaging option is designed to easily allow use of the ANSRVR server. The idea being that groups are created for sending announcements to others in the group.

Active Groups

This option requests a message from ANSRVR listing the Active Groups and the number of members of each group.

Describe Groups

This option requests a message from ANSRVR listing the Active Groups, number of members, the description of the group, (if it has one) and the owner of the group.

My Groups

This option requests a message from ANSRVR listing the Groups you are currently a member of, includes number of members of each group.

Join Group...

This option allows you to join a group or create a group if it does not exist. Selecting this option will popup a config box

Keep Active

This tick box enables the **Keep Active** function, a request will be sent every few hours as selected. Recommended to leave this at the default 8 Hours. ANSRVR will drop you from a group if no keep alive or message sent within 12 Hours.

Quietly

This tick box allows you to keep your group connection alive, without sending your CQ Text to every member of the group. If unselected your CQ Text will be sent to every member of the group.

Use CQSRVR

Select this tick box to use CQSRVR instead of ANSRVR. The only difference between the two is that CQSRVR does not allow **Quiet Keep Actives**.

Only if present

Currently not in use.

IS Only

Only send messages and **Keep Actives** via the IS stream if selected, otherwise send via IS and RF.

CQ Text

The text you wish to send to the group and use as a keep alive.

Accept

Update settings and CQ Text quietly. Not sent to every member.

Send

Update settings and CQ Text, send the CQ Text to every member of the group. Can be used in conjunction with CQ Text to send short messages to the group.

Delete

Remove yourself from the group. Note If you are the owner of a group and wish to remove the group, make sure you are the only member of the group and you will need to send U group via a normal message to ANSRVR.

Long messages can be sent via ANSRVR but these need to be done via the normal messaging system.

Advanced Knowledge Base

ANSRVR

Long messages can be sent via ANSRVR but these need to be done via the normal messaging system. Group membership list. Clicking on the list will bring up the same box as the join group box, you can however temporarily suspend your activity within a group

To send a message to an announcement group, you can either change the Announcement's CQ Text and poke Send, or send "CQ <Group> Announcement Text" to ANSRVR via a Send **Message** ⇒ **Chat**.

ANSRVR supports the following APRS message-based commands:

?	Show all groups (# in parens is number of members)
? group	Show member count in group
D	Show groups with descriptions, also sending D <group> description text, will supply a description of the group to the server
L	List groups you are a member of (with member count including self)
J <i>groupname</i>	Join group quietly
CQ <i>groupname</i> text	Join group and send text to all current members, also used to send a message to a group you already belong to.
U <i>groupname</i>	Unjoin group

ANSRVR is like CQSRVR except for the following:

Currently no minimum time between submissions to the group. This is in keeping with the "Announcement" intended use. Ability to monitor a group without sending periodic messages to everyone. Directly supported in APRSIS32! (although CQSRVR is just as supported).

Auto-Reply

Messages ⇒ **Auto-Reply** has **Message** (what to send) and "Inactive For" and "No More Than" sub-menus. "Inactive For" is a delay since the last interaction with APRSIS32 before a configured Auto-Reply will be sent. "No More Than" sets the minimum interval between Auto-Replies to a single station.

Advanced Knowledge Base

From the release notes:

Configure ⇒ **Messages** ⇒ **RF Eavesdrop** supports eavesdropping on just RF-received (not -IS) messages. This lets you display only "locally" heard messages rather than All. Clear Eavesdrops clears these as well.

Messages ⇒ **Auto-Reply** has **Message** (what to send) and **Inactive For** and **No More Than** sub-menus. **InActive For** is a delay since the last interaction with APRSIS32 before a configured Auto-Reply will be sent. **No More Than** sets the minimum interval between Auto-Replies to a single station.

And expanding on them...

The definition of "activity" is any mouse clicks, menu selections, or messaging activity in the client. If you have an "Inactive For" then any configured Auto-Reply Message won't be sent to anyone until after that time has elapsed. Eventually I hope to go back and notify every one that send you messages after the last activity when the "Inactive For" times out, but I haven't figured out where to do that yet. If you set "Inactive For" to Forever, you can see how that would effectively disable the Auto-Reply while still allowing you to have a preferred text selected (rather than None).

Once a station has received an Auto-Reply message, that station won't receive another one until at least "No More Than" has elapsed. However, if you change the Auto-Reply message, all remembered station times are cleared to allow them to receive the new "away" status on their next message (assuming the "Inactive For" has elapsed since you changed the Message).

Clear as Mud? I thought it might be. Play with it and let me know if it's working for you. Oh, and the Auto-Reply messages will show up in your Chat window(s) as well so you will hopefully be able to tell when they fired.

Lynn, KJ4ERJ

Message

Text selection of what you want to be transmitted in Auto-Reply. But can be as well **None** or a **New...** one which makes a new window named Auto-Reply to pop up.

Inactive For

The possible values for **Inactive For** are Immediate, 1, 5, 10, 30 minutes or Forever (effectively disabled).

No More Than

The possible values for **No More Than** are 1, 10, 30, 60 minutes, 1 day, 1 year (doubt you'll get there) or Forever (effectively one shot per message text change). The minimum of 1 in **No More Than** works around the 30 second APRS-IS dupe detector, but still might flood the RF channel if someone sends you a multi-line

message. Each line is a separate APRS message and will receive an Auto-Answer via RF. Not a pretty picture, but that's what Kenwood does when you enable Auto-Answer as far as I can tell, with no delays either! One reply sent for each message received

Bulletins

Creation and viewing of Bulletins

View

This option opens a new window to view the currently received Bulletins. Bulletins will age and disappear from this window over a period of 48 Hours, if they have not been retransmitted. The ageing can be changed, currently only by editing the XML file.

Notify

Selecting this option will cause a pop notification balloon to appear when a new bulletin is received.

Create...

Selecting this will create a popup window to allow creation of a bulletin. The length of a bulletin line is limited to ? Characters, multi line bulletins can be created by assigning a line number to each comment. This is done with the unlabeled drop down menu, next to the enabled tickbox.

Advanced Knowledge Base

Interval selects how often the bulletin is sent out, 0 disables Via-IS/RF tickboxes select to send via IS RF or both.

Clicking send message from the message menu brings up a box to enable entering of the callsign of the station you wish to message. You then proceed to the message window. (clicking a station and selecting send message bypasses the callsign selection box)

Chats

The message window allows entering of up to 300 characters, which are split automatically into 64 character APRS messages. There several options on this window.

Clear - Clears the message window

Copy - Copies the text in the window to the buffer for pasting into another application

Multitrack - Opens a new tracking window centered on the station being messaged

Ack - unselect this if you do not wish to have an acknowledgement of your message. (useful for satellite work)

Best - This option changes it's title. and can be slightly confusing at first

* **Best** - use RF or IS as the program determines the best option.

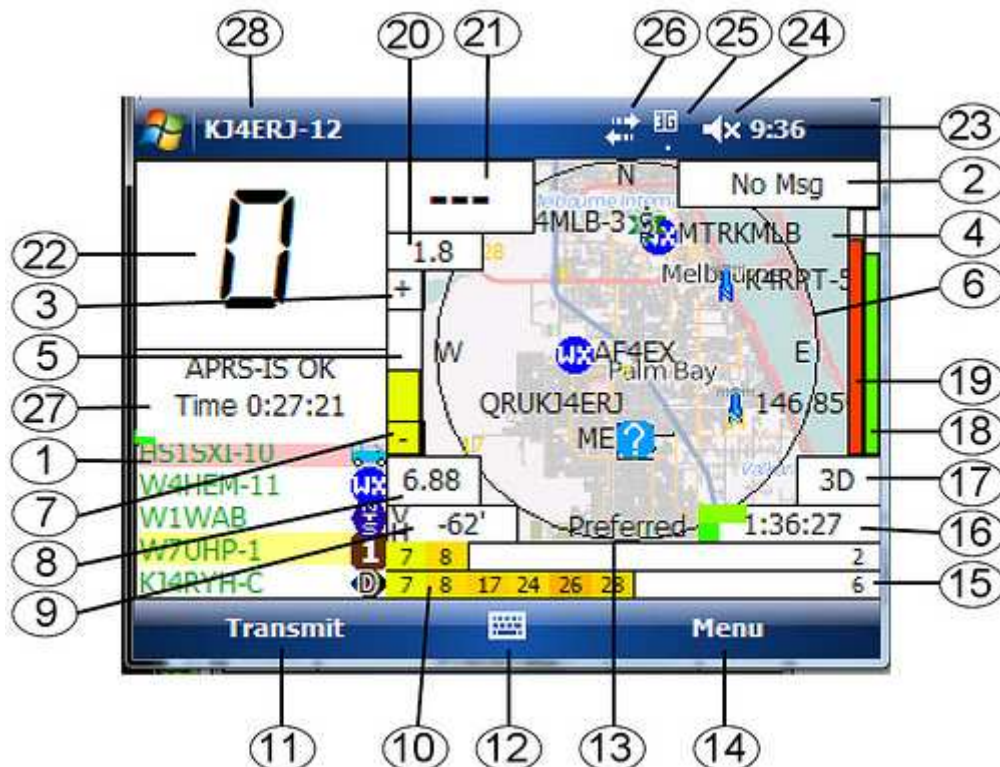
* **RF Only** - Unticked this will send the message via IS only. Ticked the message will go RF only (useful for Satellite work)

Screen

The Screen menu provides access to the more-frequently changed screen and display features of APRSIS32. See also [Configure](#) ⇒ [Screen](#) for additional features.

Advanced Knowledge Base

Screen Dcription (Example shows APRSISCE Version)



- | | |
|---|--|
| 1 Scroller | 15 Number of Satellites |
| 2 Message Notifier | 16 Time/Date |
| 3 Zoom in | 17 GPS Fix Type |
| 4 Map Display | 18 Battery Status |
| 5 Zoom Bar | 19 Station Count |
| 6 Range Circle | 20 Odometer |
| 7 Zoom out | 21 Heading |
| 8 Range Value | 22 Speed |
| 9 GPS Precision and Altitude | 23 Time (Windows Mobile User Interface) |
| 10 GPS Satellite signal level | 24 Volume (Windows Mobile User Interface) |
| 11 Transmit (Windows Mobile User Interface) | 25 Cellular Connection (Windows Mobile User Interface) |
| 12 Keyboard (Windows Mobile User Interface) | 26 Data transfer (Windows Mobile User Interface) |
| 13 Display type | 27 Connection Status (Windows Mobile User Interface) |
| 14 Menu (Windows Mobile User Interface) | 28 Callsign (Windows Mobile User Interface) |

Transparency

The **up/down arrows** will zoom the map in and out just like clicking on the \pm on the left hand scale slider bar. Most of you probably already know that.

Double-clicking in the scale slider bar will set the zoom to that exact point.

Left/right arrows will increase/decrease the transparency (decrease/increase the saturation) of the OSM map backgrounds. This is pretty handy when there's lots of tiny little stations on the screen that you're trying to see.

When you make the OSM maps 100% transparent (the whole way to the left), APRSISCE/32 switches from the Mercator projection that we're all used to and instead places stations according to their range and bearing from the center of the screen. As of a fairly recent version, it also adds lat/lon lines at 10 degree increments. Dragging the world map in this mode is quite interesting, especially if you have stations visible from all over the world. It makes it pretty obvious that things that appear to be East or West of you are in fact, closer if you go "over the pole".

I know there's a name for this projection (Azimuthal equidistant projection thanks to James VE6SRV) because HF operators use something similar from their QTH to figure out the actual compass setting to point there beam to work certain areas of the world. Mercator projections don't make this easy.

Dead Reckoning

Toggles the display of the extrapolated position of a moving station based on their last known speed and heading. Up to three extrapolated positions may be displayed based on the selections in the Screen / Labels / Speed menu. The extrapolated station symbol is connected to the last received position by a line that changes from green to red depending on the time since the last beacon was received. Normal stations will turn fully red and cease being displayed 5 minutes after their last position update. NWS stations received with a speed and heading will continue to dead reckon until their expiration time is reached.

Filter Circle (r///nkm)

TBD

Tracks

Toggles the display of historically captured track lines for stations that have moved. Actual track transmission points will be highlighted for the currently Centered station.

AutoZoom View

Selection

- AutoZoom All
- AutoZoom In
- AutoZoom Out
- Auto Center
- Follow Locked

Brightness

- Bright
- Dim

Direction Finding

Enables the display of direction finding circles

Follow

Menu below

Find

Allows typing a station name into the box and moving the map to their location. The Drop down menu shows all the received stations, if you select the drop down and start typing a station name the drop down, will move through the list narrowing down the selection. Selecting find will move to the station on the map. Multitrack will open a new window centred on the station. If the station has not been received the option to wait for a transmission from the station will appear.

Centre

This option will centre the map on the last selected station. Or your own station if no station has been previously selected.

ME

This option will move the map to centre your own station on screen.

Locked

Will lock the screen to follow the currently selected station.

Labels

Menu below

Altitude

Option to display the altitude of a station if the station broadcasts it. the display unit can be configured under **Configure** ⇒ **Metric** .

Ambiguity

Shows a circle around stations with position ambiguity, the circle shows the area that the station should be within. Ambiguity can be set **Configure** ⇒ **Beacon** Menu.

Callsign

Select the display of callsigns for stations

Footprint

TBD

Nicknames

Nicknames for stations that provide an alternate label (possibly blank) as well as an optional symbol and comment override.

NWS Calls

TBD

Speed

Selects to display the speed of a station. Beaconed displays the speed actually beacons by the station, if the station is moving but not beacons a speed calculate will attempt to display the speed.

Weather

TBD

Paths

TBD

Appearance...

TBD

LclRF Only

TBD

Network (Links)

TBD

Station (Packet)

TBD

ME (Packet)

TBD

PHG Range Circles

Enables the display of full (fixed station) or half (mobile station) scale range circles as well as setting the desired opacity of said circles. APRSIS32 displays a cardioid shape to indicate any directionality specified by the PHG values. PHG circles are colored green (full) or red (half).

Preferred

Allows a the current view to be Saved or the previously saved view to be restored. The preferred settings saves the map center and zoom level. Direct access to the saved preferred view is also available by double-clicking the Scale number.

TileSet

Provides direct access to select what map will be used as a background. See also [Configure](#) ⇒ [Map](#) ⇒ [Tile Sets](#) for more information.

View

The View Menu allows various filters on what stations will actually be displayed on the map. View settings are retained across restarts but will revert back to ALL to avoid wondering why you don't see any stations when you return to APRSIS32 after having been away for a few days.

ALL

Retains any individually selected options below to allow quick restoration to a filtered view, but displays all stations.

NONE

Unchecks all selections on the view menu. Centered stations and ME will always be displayed.

Chooser...

TBD

Altitudes

TBD

Ambiguous

Displays all stations that are transmitting beacons including position ambiguity. Non-filled circles are drawn around such stations to indicate that they may more may not be precisely at their displayed position.

Buddies

Displays all stations that APRSIS32 considered to be a "Buddy". This includes any station matching any [b/xxx](#) or [f/xxx](#) filter (including wildcards i.e. [b/DF8LS*](#) so DF8LS no matter which SSID will be indicated) and also any station currently centered in any APRSIS32 MultiTrack window.

Duplicates

Displays all stations for which suspected duplicate position packets have been received. Duplicates can occur when a portion of the APRS Infrastructure delays packet delivery for more than the default 30 second duplicate detection filter of the APRS-IS transport system. Duplicate packets can make a mobile station appear to ping-pong between new and old positions. APRSIS32 highlights such points in purple and does not include them in the track lines.

FreqSpec

Shows those stations that specify their frequency in use in the comment / beacon text and so might be tunable directly.

Freq w/Issues

Shows those station that have a mismatch toward standard APRS frequency specification. When clicking them on the map the box opening tells the reason.

Invalids

TBD

Message-able

Displays all stations whose transmissions, either by APRS packet type or application identifier, indicate that they support APRS messaging. If you have ever sent messages to or received messages from a station, it will also be included in the display. These stations will offer "Message" on their popup menu to provide a direct way to open a new chat with that station.

Nicknamed

Displays all stations for which you have defined a Nickname. See also [Screen Labels](#) ⇒ [Nicknames](#) .

NWS Objects

Displays all NWS weather objects. These objects are either directly received via APRS or are internally generated from NWS alerts in conjunction with locally defined shapefiles.

Advanced Knowledge Base

Dale Huguley KG5QD developed a server application which translates the NWS feed into a useable message format for APRS. Dale's application also generated APRS objects for tropical depressions, tropical storms, hurricanes, tornados, severe thunderstorms, and special marine warnings. In 2009, Dale ended the long run of WXSVR. Pete Loveall AE5PL created a new server which is limited to issuing (NWS nomenclature) Immediate and Actual alerts which primarily include warnings that potentially affect public safety.

Extended information can be obtained by sending a finger inquiry to wxsvr.ae5pl.net. The finger capability is supported by Xastir, UI-NWS, and other APRS software. Finger information is only available for active alerts that have been sent to APRS-IS from AE5PL-WX (qAO,AE5PL-WX)

There is now an Australian wxsvr using different area IDs, etc. More information can be found at [Australian WXSvr web site](#).

The databases on this site provide a way to interpret the NWS statements the weather server creates and disseminates to the Internet. APRS+SA, Xastir, UI-NWS, and WinAPRS support use of the NWS shape files to graphically depict the watches and warnings. You can download the files using the "NWS Shape Files" link to the above. The current NWS shapefiles are (AWIPS Counties) [c_07jn12.zip](#), (Zone Forecast Areas) [z_07jn12.zip](#), and (Coastal Marine Zones) [mz07jn12.zip](#). If you are interested in programming with .NET to read shape files, I have updated the .NET MapTools shapelib wrapper found at <http://shapelib.maptools.org> to be fully compatible with .NET 2.0. The MapTools archive found on this web site at <http://www.aprs-is.net/downloads/DotNetshapelibupdate.zip> has been tested with VisualStudio 2008 and is compatible with all .NET 2.0 programming languages. It does not run as unsafe so it should be usable with any .NET application. The archive includes all source and solution files. I have not tested or debugged any of the example and test applications. All applications should only be compiled in x86 mode to maintain compatibility with the shapelib.dll. While this is compiled in x86 (32-bit) mode, it is usable on 64-bit platforms as long as the application is also compiled in x86 mode.

The format for the NWS statements generated by the weather servers is:

Area Specific:

CWAPID>APRS::NWS-TTTTT:DDHHMMz,ADVISETYPE,zcs{seq#

CWAPID>APRS::NWS_TTTTT:DDHHMMz,ADVISETYPE,compressed zcs{seq#

General/Full Text Messages:

CWAPID>APRS::SKYCWA :DDHHMMz,ADVISETYPE,zcs{seq#

Objects:

CWAPID>APRS::CWattttz *DDHHMMzLATLONICONADVISETYPE{seq#

CWAPID>APRS::CWattttz *DDHHMMzLATLONICONADVISETYPE}multiline{seq#

CWA is the NWS office (See databases to the above). The CWA "SEC" is the source for solar reports and is not listed in the databases as it is a nationwide CWA.

PID is the product Code (See database to the above).

TTTTT is ADVIS, WARN, WATCH, etc.

ttttz is the issue time.

DDHHMMz is the expiration time.

LATLONICON is the latitude, longitude, and symbol for the object (standard object format).

ADVISETYPE is things like FLOOD, FLASHFLOOD, SVRTSM, SEVERE_WEATHER, etc.

zcs are comma separated zone codes or county codes (see the databases linked to the left to find the zone or county code for your area), or statement text.

compressed zcs is a compressed listing of zones or counties (see example below) (note underscore in To address instead of hyphen).

multiline is the enclosing line structure for the object (see Multiline Format link above).

{seq# is decoded as:

The first three characters are the "issue time" compressed by assigning 0-9 as themselves, A-Z as 10 thru 35, a-x as 36-59, where it is DHM (Day of the Month-Hour in 24 hour format and Minute). Up to 16 this reads as hexadecimal so {A8B** was issued on the 10th at 08:11 Z.

The next two characters are line numbers which (along with the "From Call") make the packet unique. Any packet with the "from Call" and the first 4 digits of the sequence matching are "associated" packets- in otherwords they are from the same product or portion of product that is defined by UGC codes.

Compressed Zones and Counties Examples:

COZ91-92-KSZ27>29-41-42
Colorado zones 91 and 92, Kansas zones 27, 28, 29, 41, and 42

TXC67-183-203-315-459-LAC17
Texas counties 67, 183, 203, 315, and 459, Louisiana county 17

Pete, AE5PL

Objects

Displays all objects received from APRS. Objects are typically points of interest like repeaters, hospitals, and other non-APRS stations.

Paths

TBD

Telemetry

Displays stations for which Telemetry has been received.

Tracks

Displays stations that have moved and therefore have accumulated tracks to display.

Mic-E

TBD

Advanced Knowledge Base

Mic-E Type Codes		21 Aug 2013
21 Aug 13	added FTM-400DR and FT1D	
26 Oct 11	new Mic-E code T.....*v	www.KissOZ.dk OZ1EKD and OZ7HVO
27 Jan 11	add HinzTec anyfrog	
03 Nov 10	add D72	
24 Mar 10	add Yaesu VX-8G. And updated info on KWD D710	
20 Oct 09	add Yaesu FTM-350	
21 Oct 08	clarify and add notes on D710 present decoding	
18 Aug 08	reverse order of MFR and Version bytes:	
15 Aug 08	set up OTHER category and use ENDING bytes.	
14 Jul 08	original	
Old idea	T.....vM	Prior to 18 August 08
New idea	T.....Mv	allows more flexibiilty in use of "v"

BACKGROUND: Before the formal APRS SPEC was written, there were only two Mic-E products in use. These were the original Mic-E prototype, and the TAPR Mic-E kit. While developing the SPEC for others, the two Kenwood radios, D7 and D700 were in development and so the first byte of the original free-text field was designated as a Mic-E TYPE identifier to identify different devices. This was written into the APRS spec.

It is important that the type of Mic-E device be identified, since some may have very important distinct properties. For example we need to know which devices are message capable or not. In the future, we may want to know which models can auto-tune or not.

ORIGINAL TYPE CODE: The first byte of the original free-field text was set-aside for use in indicating the type of Mic-E device. On page 55 of chapter 10 of the spec, the first two of these TYPE CODES ">" and "]" were defined for the D7 and D700. In addition, the statement was made:

"It is envisioned that other Mic-E compatible devices will be allocated their own type-codes in the future."

The future has arrived and the many thousands of existing APRS radios and APRS clients will recognize the D7 and D700's, but unfortunately, will not recognize anything with newer TYPE CODES without new software.

And since many people use UI-View which cannot be upgraded with new TYPE codes, we had to find a solution when Yaesu and others began considering new APRS radios and Mic-E implementations.

2008: Luckily, the original SPEC also allowed for two other special characters in that location, the (') apostrophe and the (`) grave accent. These were used by the original prototypes for additional BYTE wise analog telemetry fields. Since no one since the mid 1990's was using these, and since all EXISTING radios and clients would pass these (and only these) two other bytes, they were REDEFINED as a new MIC-E TYPE category called "OTHER Mic-E". The (`) character was defined to indicate a message capable Mic-E device and the (') apostrophe would identify Trackers only.

D710 VERSION CODE: For backwards compatibility, the D710 uses the same "]" Type byte as the D700 but adds a special "=" byte on the end to distinguish the new radio. We will use the abbreviated syntax of].....= to represent this Mic-E TYPE code and version code. The seven dots represent free text but it can also contain the special "aaa}" altitude.

VALID TYPE CODE AND VERSION CODES. For the purpose of this Mic-E update to the spec, the written format for the free-field text defined in the original APRS spec will be referred to as

Mic-E TEXT FORMAT "T.....Mv".

Where T is the Mic-E TYPE byte (" ", ">", "]", "'", "`", "")

Where is free field text

Where aaa}.... like Taaa}....Mv is optional altitude

Where "M" represents the Manufacturer Byte

Where "v" represents a version Byte

Using this definition, all of the following examples have been tested to work with existing D7 and D700 radios. The D710 after software version 2.01 also decodes them. The Yaesu VX-8R, VX-8G and FTM-350 as well as TH-D72 are compatible.

LEGACY FORMATS: These formats are to be recognized always in all models now and in the future. Notice that it allows for future VERSION codes for the D7 and D700 series so that even without recognizing the new version code, the radio will be recognized as being in the D700 or D7 family.

.....	Original Mic-E (leading space)	(not message capable)
>.....	TH-D7A walkie Talkie	(message capable)
>.....v	*future TH-D7A upgrade version	
].....	TM-D700 MOBILE Radio	(message capable)
].....=	TM-D710 Mobile Radio	(message capable)
].....v	*future D7xx models version	
>.....=	TH-D72 walkie Talkie (Oct 2010)	(message capable)

NEW FORMATS FOR ALL FUTURE APPLICATIONS: (plus the exceptions above)

```
`.....Mv OTHER Mic-E - display "McE-msg"    or "Mic-Emsg"
'.....Mv  OTHER Mic-E - display "McE-trk"    or "McTrackr"
```

As combinations of "Mv" bytes are added to this table, manufacturers are encouraged to display the correct radio type if known. The following additional bytes have been declared:

```
`....._b Yaesu VX-8 (b = SPACE)                (message capable)
`....." Yaesu FTM-350      [new in Oct 2009]    (message capable)
`.....# Yaesu VX-8G        [new in Mar 2010]    (message capable)
`.....$ Yaesu FT1D         [New in Dec 2012]
`.....% Yaesu FTM-400DR    [new in Aug 2013]
'.....|3 Byonics TinyTrack3 (TT3)
'.....|4 Byonics TinyTrack4 (TT4)
T.....\v Hamhud ?          (awaiting confirmation?)
T...../v Argent ?          (awaiting confirmation?)
T.....^v HinzTec anyfrog
T.....*v APOZxx www.KissOZ.dk Tracker. OZ1EKD and OZ7HVO
T.....~v OTHER (to be used when all other "M" MFR's are used up)
```

Other manufacturer type codes could be as follows. If anyone intends to use any codes PLEASE check with WB4APR first to make sure you are universally compatible.

```
T.....`v
T.....'v
T...../v
T.....\v
T.....-v
T.....:v
T.....;v
T......v
```

It is important to choose ending MFR and Version bytes that will not cause confusion on the end of any random text. Designers of Mic-E type devices are encouraged to advise their intent to use a specific ENDING code (M) so that other designers can include them in their code for display.

DELIMITER BYTE: As one last afterthought, if the MFR wants to really make sure that his "Mv" code on the end of any random text does not add confusion to the text, maybe we should consider preceding it with a forced SPACE byte. This way the Mv code of "-3" would not be too confusing on the end of this "...text-3", but would look cleaner as "...text -3". But it makes the packet one byte longer.

RECEIVING AND DISPLAY OF MIC-E TEXT (STATUS): Since the leading TYPE byte "T" was not as well defined as it could have been in the original Mic-E spec, this byte will be displayed as part of the text on some existing receiving applications that display the received Mic-E text. It is recommended that all future RECEIPT and DISPLAY software consider that this leading TYPE byte will usually be one of the above TYPE characters " ", ">", "]", "'", or "`" and should not display it with the text. This is essential to maintain the common 10x10x8 text formatting of local information for best display on mobile displays.

So, suppress the T-TYPE byte from the text display if it is recognized. But use this information to display the radio type elsewhere as appropriate. Similarly, if the ending "....Mv" bytes are both recognized and matched and decoded and displayed elsewhere, then these two bytes may be suppressed from the display if desired. If these two bytes are not decoded and are not recognized, then they need to be displayed for older radios so the operator can interpret their meaning until his firmware is updated.

When a new type is not recognized, then display "McE-Msg" for a two-way system and "McE-Trk" for a tracker without message capability. If 8 bytes are available for this display, then display "MicE-Msg or McTrackr" etc..

This table is maintained on the APRS web page: www.aprs.org/aprs12/mic-e-types.txt

Adding these two new "" and "" OTHER MIC-E TYPE bytes was only possible beacuse extensive testing showed that those were the only ASCII bytes that the tens of thousands of D7 and D700 radios and UI-View would decode now. Fortunately, the D710 also parses these TYPE bytes, and is fully compatible with version number V2.01 and later.

Bob, WB4APR

Mic-E Test Examples

10 Dec 2008

There have been lots of changes and evolution to the Mic-E formats since the original Mic-E was built by WB4APR and N3MIM back in 1994. Many of these were captured in the APRS spec in 2000, but with the new APRS radios, D710, Yaesu and TinyTrack products, plus the addition of !DAO! precision and FFF.FFFMHz for frequency, there needs to be a consistent test suite of formats to make sure we are all on the same page.

This document interprets the evolution of the Mic-E protocol and shows how to properly place the additions within the Mic-E format.

LATITUDE BYTES: For these examples, assume that the AX.25 TOCALL is loaded with "ABCDEF" which happens to decode this information:

```
LATITUDE: 1234.56N
MESSAGE:  CUSTOM 0
LONGITUDE: West and greater than 100 degrees.
```

LONGITUDE AND CSE/SPD BYTES: I have randomly chosen the "abc123" bytes for the next 9 original bytes of the packet which decodes as:

```
LONGITUDE: 16910.71W
CSE/SPD: 223 degrees and speed of 014m.
```

SYMBOL: I will assume an RV Primary table symbol "R/". But remember that the second byte is the TABLE or OVERLAY BYTE and can be virtually any uppercase printable ASCII character A-Z and 0-9. Also notice that the order of the symbol and table are reversed from the normal way that we typically refer to table and symbol as "/R". This is because the alternate symbol table was added to the Mic-E protocol after the initial TAPR Mic-E kits were in use. Before then, only primary symbols were available.

MIC-E TEXT FIELD: All of the evolution of the Mic-E protocol has been via additions to the original TEXT field so that all future systems are 100% backwards compatible with precious hardware and decoders. There have been no additions to the Mic-E protocol that would have any impact on the above position and symbol data.

The additions are all in the TEXT field:

```
'abc123Rtext... - original Mic-E
'abc123R/text... - original Mic-E with "/" TABLE byte
'abc123R/>text... - added the ">" TYPE byte for the D7
'abc123R/]text... - added the "]" TYPE byte for the D700
'abc123R/]123}text... - added altitude as FIRST 4 bytes...
                        - But altitude is OPTIONAL, so it simply offsets remaining text by 4 bytes.
```

The original APRS spec was published at this point. Then the D710 needed an identifier byte. We added this byte to the end of the text for backwards compatibility to all existing radios and clients.

'abc123R/]123}text...= - D710 added "=" on end for ID

In 2007 we needed many more TYPE bytes for many more radios and hardware; so through exhaustive testing we found that all existing radios could accept the "`" and "" bytes as TYPE bytes without breaking anything. We defined "`" for MSG capable units and the "" one for one-way trackers.

Also we needed to identify the radio manufacturer and the version number since the APxxxx field cannot be used with the Mic-E format. So we added a Manufacturer and Version byte (Mv) on the end of the text in 2008.

'abc123R/`text.....Mv - Message capable (MFR and Version)

'abc123R/"text.....Mv - One-way Tracker (MFR and Version)

DELIMITER: For clarity we recommend that a SPACE might be included before the " Mv" bytes to distinguish them from the text if the MFR byte can be confusing on the end of some text.

ALTITUDE AND FREQUENCY: Next we added FREQUENCY as an optional New "first" 10 byte field in all APRS text formats, not just Mic-E. But to apply this rule in the Mic-E means that it must apply after the earlier optional 123} altitude is considered. In otherwords, the "text..." field is right shifted if 123} altitude is used, and so the definition of "first" bytes is also shifted right 4 bytes before parsing the "text...".

'abc123R/'123}text... Mv - optional altitude must be first

'abc123R/'FFF.FFFMHztext... Mv - FREQ must be first "text".

'abc123R/'123}FFF.FFFMHztext Mv - format if both are used.

NOTE: Thought not shown, it is desirable to transmit a SPACE byte after the Frequency to separate it from other text. Early Yaesu FTM-350's require this space.

PRECISE POSITION DATA !DAO! FORMAT: The optional !DAO! format for 1 foot precision and DATUM is allowed in APRS1.2 to be decoded "anywhere" in the text field, but again, only as long as "anywhere" does not break any other rules. In the Mic-E format, it cannot go before the 123} or the FREQ field. It is recommended that !DAO! always go at the end where it does not detract from other more valuable human readable text, nor any of the other "first" optional fields.

This also applies to any use of the old /A=123456 optional altitude format. It should always be towards the end so that it does not displace valuable bytes at the beginning. Of course there is no reason for this verbose altitude format to ever be in a Mic-E packet, but it is included here for completeness with the spec.

'abc123R/'123}FFF.FFFMHztext.../A=123456 Mv - old-alt later

'abc123R/'123}FFF.FFFMHztext.../A=123456!DAO! Mv - DAO always last

I will summarize all of these formats below so you can use them as a test case...

AX.25 destination address "ABCDEF" AX.25 data field:

'abc123R/text...

'abc123R/>text...

'abc123R/>text...v for future D7's

'abc123R/]text...

'abc123R/]text...=

'abc123R/]text...v for future D7XX

'abc123R/"text..... Mv for all other manufacturers

```
'abc123R/'FFF.FFFMHztext... Mv
'abc123R/'123}text... Mv
'abc123R/'123}FFF.FFFMHztext... Mv
'abc123R/'123}FFF.FFFMHztext.../A=123456 Mv
'abc123R/'123}FFF.FFFMHztext.../A=123456...!DAO! Mv
```

MIC-E FORMAT BYTE: The very first byte shown above as a "" at the very beginning of the packet is the APRS Data-type identifier and has two values for the Mic-E protocol. The apostrophe "" was supposed to indicate an OLD GPS fix, and the grave-accent "" was supposed to mean a valid GPS fix.

But Kenwood got them backwards, and by the time this was discovered, it was too late to change the radio. But this quirk did make it into the original spec. I do not know how most software clients distinguish between these two types of Mic-E format. But the original APRSdos will display "OLD FIX" if the "" type byte is used by any other Mic-E but if it is a Kenwood, then the "oldness" is ignored and the packet is just displayed as kenwood...

NOTES: The decision to add the additional Mic-E-OTHER type byte (' or `) was solidified after the initial D710 was released, so on those D710's the Mic-E position is properly decoded, but the CSE/SPD and text fields are not displayed. But these are being corrected in the next D710 firmware download.

See the original document on these Mic-E TYPE/VERSION bytes:
www.aprs.org/aprs12/mic-e-types.txt

These are linked from the APRS 1.2 Addendum Web page:
www.aprs.org/aprs12.html The APRS1.2 addendum

Bob, WB4APR

APRS Precision And Datum Option

07 May 2008

This proposed 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 can 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/DDMM.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 comment
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 claiming 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 nauseum 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 possibilities 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 become 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.

Bob, WB4APR

Platforms

Allows selection of specific APRS platforms to display. Note that counts of stations determined to be of a specific platform are displayed on this menu. Some of the platforms provide additional breakdowns, but selecting any one of the detailed categories sometimes selects the entire category.

RF

Selection

All

Direct

Local

3rd Party

Shrieks

APRSIS32 parses any !<nonwhitespace>! from comments and groups (up to 32 of) them into a **View** ⇒ **Shrieks** submenu. This allows event coordinators to pick a suitable !EVENT! for stations to put in their beacon comment to provide easy on-screen filtering of only the stations participating in the event. The Spotter Network requires their users to put !SN! in their comment, so you can expect to see !SN! in the **View** ⇒ **Shrieks** menu if any such stations are in your range.

!Shriek!s are also parsed from nickname comments EVEN if the nickname or comment is not enabled! This allows you to nickname your favorite stations with a !WATCHME! comment and then **View** ⇒ **Shrieks** ⇒ **!WATCHME!** to see ONLY those stations! You could even nickname stations as !VIEW1!, !VIEW2!, !VIEW3!, and so forth to build your own local MultiTrack views of specific stations. A single station can even have multiple !VIEWn! tags in the comment to put them on multiple views.

If nicknames are enabled, and a station has a nickname, and that nickname overrides the comment, then ONLY the comment's Shrieks are used. The comment received from the station will be ignored completely for Shriek purposes. After these conditions are no longer met, a new packet from the station must be received to get a proper !Shriek! for the station. (So, why "Shriek"? Search for Shriek at Wikipedia)

Symbols

Allows selection of specific symbols to display. The currently centered station's symbol is sorted to the top of the list to allow easy selection of other stations like the

centered station. Note that counts of stations using a symbol are displayed on this menu.

Remark Beside single specific symbols there are bundled groups of symbols concerning complete themes. These themes are Flight, Marine, Mobile, RFID and Weather. **End of Remark**

Transport

Selection

RF Only

RF + -IS

-IS Only

APRSIS32 attempts to determine the transports available to a given station based on the contents of received packets. This information is used to optimize some communications with stations on various platforms.

Also Digipeater, IGate (these last will display stations that have been observed acting in the specified capacity regardless of what symbol they may be transmitting).

Echolinks...

Will load the actual active Echolink from the server. Due to the fact that this process might take really long a respective box pops up telling so. Clicking YES will continue the procedure.

GeoCaches...

TBD

Configure

The Configure menu provides access to all of the not-so-frequently changed and/or initial setup options of APRSIS32.

General

The General configuration dialog is the first dialog that is presented on a clean installation of APRSIS32 and collects the basic configuration information necessary to operate an APRS-IS client.

MyCall-ssid

This is actually the only field that you **MUST** complete to configure a new APRSIS32 instance. Enter your amateur radio callsign and an optional -SSID. If you are new to APRS and don't know what an -SSID is, just enter your callsign to get started.

Advanced Knowledge Base

Latest updated to clarify the title of this document to be SSID RECOMMENDATIONS not to imply any kind of decoding "standard". That original NMEA decodign standard is covered in the original spec

SSID's have seen two different uses in APRS. Initially as an ICON indicator back in the early 1990's. But that is obsolete for over a decade. Now SSID's are used as an informal way of indicating one of several different typical APRS applications.

Since many small displays for the handheld and mobile operator show nearby APRS station callsigns that flash up on the screen, it is nice to have some idea of what type of station or activity might be involved simply from the callsign SSID without having to push buttons, search lists, or look at maps to find out more about them.

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 SSID's (the limit of 16 comes from the 4 bits available in the AX.25 protocol. Note, The SSID of zero is dropped by most display applications. So a callsign with no SSID has an SSID of 0.

- 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, Blackberry's etc)
- 6 Special activity, Satellite ops, camping or 6 meters, etc
- 7 **walkie talkies**, HT's or other human portable
- 8 **boats**, sailboats, RV's 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

* One-way trackers should best use the -12 one-way SSID indicator because the -9's usually mean a ham in full APRS communication both message and voice. The -9's can be contacted by APRS message or by Voice on his frequency included in his beacon, or on Voice Alert if he is in simplex range. The -12's are just moving Icons on the map and since they have no 2 way communication for ham radio they are not generally of routine interest to other operators.

OBJECTS or INTERNET: In addition, Objects or internet generated stations can have any SSID, not just the original 16, since Objects are not constrained by the AX.25 header and can have a 9 byte name. Here are some common OBJECT/Internet SSID's:

- 63 for PSK63 HF stations
- tt for APRS TouchTone users (DTMF)
- ID for RFID
- A through -Z for Dstar

SSID BACKGROUND: Originally, in 1992, we had to use the SSID as a way of indicating the type of station that transmitted a raw NMEA-0183 GPS sting. But in the mid 1990's we began indicating any of the nearly 200 APRS symbols by the setting of the AX.25 TOCALL of "GPSxyz". The "xyz" characters define the symbol from the standard APRS symbol table www.aprs.org/symbols.html.

The GPSxyz concept worked so well, the original SSID associations are no longer a required part of the spec. But the conventions that evolved from those early SSID's are still encouraged as noted above, for easy recognition of station type or activity by when only the callsign is seen.

The -1, -2, -3, -4 and -15 are kept generic so that anyone with as many as 6 digipeaters, or 6 trackers or 6 weather stations or 6 vehicles can still have unique SSID's for each of his stations. Beyond 6, people will just have to use any SSID that suits their fancy. In some areas there might be 15 digipeaters all under one guy's call!

SSID USAGE: The SSID's also might give a hint as to how someone is getting into APRS whether via satellite, a one-way tracker, a mobile, an HT or even via DTMF or an RFID device or whether he is doing something special.

For example, if you are doing something special, change your SSID to -6 to alert others to your excitement, or to make the track-history begin and end on site, and not be tied to all your other -9 travels. Or use -6 SSID for a packet sent via the ISS or APRS satellite or for a 6 meter test so the successful packet is preserved and not overwritten by the same radio the next time you use it not via the ISS on the 144.800 [US 144.390] national channel. By using separate SSID's the WEB data bases will keep statistics and data separate from when you are working normally on other bands with other SSID's.

So stick to the suggestions above for the obvious applications where you can. Of course these are not rigid. If you have more than 6 digipeaters, use any SSID you want. These are only guidelines to hint at a station's possible application when all you can see easily is the callsign on a screen or in a list....

Bob, WB4APR

Password

The APRS-IS network requires an authentication password before packets will be accepted. This is the same password granted your callsign if you have used other APRS-IS clients in the past. If you are new to APRS or do not remember your APRS-IS password, send an e-mail to Password Request to request one. Make sure you include your name, callsign, and the fact that you need a password for APRSIS32. If your beacons are not showing up on APRS Internet sites like aprs.fi or findu.com, an incorrect password is the most probable cause (second only to not having an active Internet connection! or having Enables / Internet Access unchecked).

Quiet Time

The Quiet Time is the maximum length of time during which a packet should have been heard on that port. If no packet is heard within the specified time, the port is closed and re-opened using all of the configure `<Close/OpenCmd>`s. This is provided as a way to detect a possibly dropped network or bluetooth connection, or maybe even a TNC that has lost its mode and needs a restart to recover. Because there is no heartbeat/keepalive capability on a TNC connection, an RF Port Quiet Time must be set long enough to cover the maximum expected elapsed time between packet receptions. If set too short, the port will unnecessarily close and re-open itself periodically, possibly resulting in dropped packets across the restart. A Quiet Time of zero disables this function.

So, don't set this on a KISS-type port on a Kenwood radio. The default `<Open/CloseCmd>`s will leave the radio's TNC turned off until you manually turn it back on.

Don't set this on any port that doesn't have a steady stream of expected traffic (like an HF APRS radio port). It just doesn't work out very well.

If you have a steady APRS traffic hum on your radio port, set the Quiet Time to 2-3 TIMES the quiet time you'd expect to see on the channel. This gives APRSISCE/32 a chance to reset the channel if it goes quiet too long. If you have sounds enabled, you'll heard bings and bongs while the radio port transitions through the various states of a restart.

Range

This is the range (in 1/10 miles, so multiply by 10) of the default Range filter for APRS-IS. If you want 50 miles, enter 500. If you are operating an IGate and don't want any unnecessary packets from APRS-IS, you can set this to 0 (zero). APRSIS32 uses this value in constructing other components of its automatic-filters.

Advanced Knowledge Base

The client automatically does some filtering for you. It does:

m/dist - My Range filter based on the configured Range (converted from 1/10 mile to km) (Only if Range is non-zero)

If you track or center on someone (right click or popup menu/Center), it adds a:

f/call/dist - Friend Range filter using the same range (changes to b/call if Range is zero)

This filter will be removed when you are no longer tracking a callsign.

If you drag the main window map around to establish a center other than yourself or a specific station, it adds a (if Range is non-zero):

r/lat/lon/dist - Range filter using the map center and the configured range

In the main window, if you check Screen / Filter Circle, an r/lat/lon/radius is added where lat/lon are the center of the circle and radius is the radius of the circle. This filter is only added if the circle radius is greater than the configured Range.

MultiTrack windows automatically add a Buddy (b/call) for the Track (station). If you check "Range" on that station's popup menu in the MultiTrack window, the buddy will change to a Friend (f/call/dist) filter using the configured Range. Due to APRS-IS server filter limits, only a limited number of MultiTrack windows can be Range tracked at any point in time.

If you have a remembered MultiTrack that is not known, a window will open that is "Await(CALL)". Because APRSIS32 doesn't know if CALL is a station or an object, it will add both a b/CALL and o/CALL to your filter string until the station is heard from.

Due to an oversight in the t/n (NWS alert) filter that doesn't pick up alerts issued between 36 and 59 minutes into the hour (a lower-case sequence character), all enabled NWS Entry Servers are added with an e/ entry filter to ensure receipt of all NWS packets from that server. In addition, all enabled NWS Offices are added to a p/ prefix filter.

If you have an Alternate Network selected ([Configure](#) ⇒ [AltNet](#)), a [u/AltNet](#) filter is automatically added for the selected AltNet. When [Configure](#) ⇒ [AltNet](#) ⇒ [None](#) is selected, this automatic u/ filter is removed.

For any other filter, you can do your own. Here's a few of the Add Filter's I've got in a few instances of my APRSIS32 client:

```
r/27.99673/-80.659072/1 b/KJ4DXK* b/KJ4ERJ* b/KJ4OVQ* u/APWW*/APWM* b/W4SGC*
```

r/28.40/-81.50/2000 r/27.99673/-80.659072/1 -b/WINLINK -s/S/S -s//a/DW
 b/KJ4OVQ* b/KJ4ERJ* b/KJ4DXK* b/W4SGC* b/WI8H* -b/WINLINK -s/S/S -s//a/DW

and my APRSISCE phone has the following filter:

b/N4GVA* b/KJ4ERJ* b/KJ4DXK* b/KJ4OVQ* b/W4SGC* s/@/@ u/APWW*/APWM* -b/WINLINK

Can you tell what callsigns I like to keep tabs on?

Oh, and obviously, if you add any additional filter, you only specify the actual filter components as shown above. You do NOT need to prefix it with filter or #filter or any other things that other APRS clients might have required.

For a complete list of APRS-IS-supported filters, please refer to <http://www.aprs-is.net/javAPRSFilter.aspx>

Lynn, KJ4ERJ

Add Filter

If you want to supply any additional filter terms, enter them here. Do NOT include anything but actual filter terms as described at Filters In particular do NOT put a "#filter" prefix as required by some other APRS clients. Unless you have something specific you're interested in, the automatic-filters will probably be sufficient and you can leave this blank.

Advanced Knowledge Base

3.1 javAPRSFilter Users Guide.

New in 3.1 Release

- 1) All call signs are now case sensitive
- 2) New Port of entry filter E
- 3) new Unproto filter U
- 4) D-filter now allow wildcard
- 5) New filter command "filter default"
- 6) T-filter now support distance from a station or object

Introduction:

The APRS-IS full feed today have a lot of traffic and require a high bandwidth, in particular for the APRS-IS server sites, but also for some of you that connect that are only interested in the particular traffic. To accommodate this a number of servers have special regional feeds which filter the traffic. There are also some weather specific feeds available. But all these are setup according to what the server operator "thinks" will be good for the users. Now we can take this one step further.

Now will each of you be able to select what traffic you are interested in and the server will create a unique feed for you. There is great flexibility to construct your personal feed.

How does it work?

Pete Loveall AE5PL have written the APRS-IS server software javAPRSSrvr in java which is used by a number of servers. Pete has been kind to create some hooks into his server software so I have been able to write a filter add-on, javAPRSFilter (also in java). These 2 applications work together to provide this filtering. Status on the APRS-IS servers can be found here:
http://ahubswe.net/aprs_stat.asp

You define the filter by doing the following:

1. Connect and logon to a filter enabled port on a server
2. Send an APRS message to the server requesting the filter you want

Filter commands

There are 12 different kinds of filters that can be used in any combination. Each filter is working independent and is additive to the feed. This mean if the filter finds a match it will be passed to you. The filter commands in the APRS message to the servers call is starting with the word 'filter' (without quotes) and each filter command is delimited by a single space. A message with just 'filter?' (without quotes) will return the current filter definition.

If you send "filter default" it will revert back to the server default filter definition.

#1 Range filter

The range filter will pass all stations and objects within a distance from a set location. It will also pass messages to stations within the filter and positions of the message sender even if they are outside the range. Up to 9 range filters can be used at the same time to extend the areas when you have problem to find a good circle match.

Syntax: `r/lat/lon/dist [r/lat1/lon2/dist2 [[r/lat2/lon2/dist2]]`

Where: `r` = range command
`lat` = latitude in degrees. Negative for south
`lon` = longitude in degrees. Negative for west
`dist` = distance in kilometers from lat/lon. I'm sorry we don't use miles here in Sweden ;-)

Samples: `r/55/-4/600` This will pass all traffic for UK
`r/37/-81/1500` This will pass all east cost US traffic

From V3.0 lat and lon can be in decimals. E.g. 58.5

#2 Prefix filter

Note: This filter is kept for backwards compatibility. The Budlist filter now support this functionality. (from V 1.4)

The prefix filter will pass traffic based on if the sender's call starts with a specific pattern.

Syntax: `p/p1/p2/p3...`

Where: `p` = prefix command
`p#` = The prefix (starting) pattern

Samples: `p/K` This will pass all traffic from stations starting with K
`p/SK/F` This will pass stations starting with either SK or F
`p/SM5NRK` This will pass all traffic from SM5NRK and any SSID at the end

#3 Budlist filter

The budlist filter will pass traffic based on exact match of the sender's call or call starts with a specific pattern (from V 1.4). Also the SSID is part of the exact match.

Syntax: `b/call1/p1*/call3/p2*...`

Where: `b` = budlist command
`call#` = The prefix (starting) pattern
`p#` = The prefix (starting) pattern

Samples: `b/SM5NRK` This will pass all traffic from SM5NRK without any SSID
`b/SM5NRK-5/SK5UM` This will pass all traffic from SM5NRK-5
`b/K*` This will pass all traffic from stations starting with K
`b/SM5NRK/F*` This will pass all traffic from SM5NRK and stations starting with F

#4 Type filter

The type filter will pass traffic depending on the packet type. More than one type can be defined in one single command. You can also limit this to a distance around a station or object.

Syntax: t/type
t/type/call/dist

Where: t = type command
type = is one or more of the following letters
p = Position packets
o = Objects
i = Items
m = Message
n = NWS Weather and NWS Area Objects
w = Weather
t = Telemetry
q = Query
s = Status
u = User-defined
call = call of a station or object
dist = distance in km from call to pass this type

Samples: t/p	This will pass all traffic with a position
t/w	This will pass all weather traffic. For positionless weather objects the corresponding position packet will also be sent when it is next heard
t/mos	This will pass all messages, objects and status traffic
t/p/SM5NRK/500	Pass all position packets within 500 km from last known position of SM5NRK

Remember that the APRS message must start with the word filter and the the commands.
The above filters can be combined as explain above. Each filter will however working independent of the others, for example:

filter r/63/16/1000 r/55/-4/600 p/F b/AE5PL t/s

The above filter will pass all traffic within Nordic (range#1) AND UK (range#2) AND stations starting with F (prefix) AND from AE5PL (budlist) AND all status traffic (type).

#5 Symbol filter

The symbol filter will pass traffic based on the symbol in the packet.

Syntax: s/pri/alt/over

Where: s = symbol command
pri = symbols in primary table
alt = symbols in alternate table
over = overlay character (case sensitive)

Samples: s/->	This will pass all House and Car symbols (primary table)
s//#	This will pass all Digi with or without overlay
s//#/T	This will pass all Digi with overlay of capital T

#6 Digipeater filter

The digipeater filter will pass all packets that have been digipeated by a particular station(s). Remember that a packet can many time go different routes to get to APRS-IS and might be digipeated by other stations that is shown. These packets are filtered out by various filters/application

as duplicates. More that one digipeater can be entered and each are OR together. This filter also support wildcard

Syntax: d/digi1/digi2...

Where: s = digipeater command
digi# = digipeater call

Samples: d/SM5NRK-2	Pass all packets digipeated by SM5NRK-2
d/SM5NRK-2/SK5UM	Pass all packets digipeated by SM5NRK-2 or SK5UM
d/SM*/SK*/SL*	Pass all packets that has been digipeated by a digi in Sweden

#7 Area filter

The area filter works the same as range filter but the filter is defined as a box of coordinates. The coordinates can also been seen as upper left coordinate and lower right. South and west are negative. Up to 9 area filters can be defined at the same time.

Syntax: a/latN/lonW/latS/lonE

Where: a = area command
latN = North latitude border (-90 to 90)
lonW = West longitude border (-180 to 180)
latS = South latitude border (-90 to 90)
lonE = East longitude border (-180 to 180)

Sample: a/50/-130/20/-70 This will pass all traffic in US

From V3.0 lat and lon can be in decimals. E.g. 58.5

#8 q Construct filter

The q Construct filter will base all filtering on the q Construct used on the APRS-IS. For more information about q Construct look here: <http://www.aprs-is.net/q.htm>

Syntax: q/con/ana

Where: q = q Construct command
con = list of q Construct to pass (case sensitive)
ana = analysis based on q Construct.
I = Pass positions from IGATES identified by qAr or qAR.

Sample: q/C	Pass all traffic with qAC
q/rR	Pass all traffic with qAr ot qAR
q//I	Pass all position packets from IGATES indentified in other packets by qAr or qAR

#9 Object filter (from V1.4)

Same as BudList but acts on the object names instead of sender's call.

Syntax: o/name1/n2*/name3/n3*...

See Budlist for more information

#10 My Range filter

The my range filter will pass all stations and objects within a distance from your own station. It will use the location sent for the same call as you used when you logged onto the server. This is useful if you have an mobile station with internet connection. It will then always pass the local stations no matter of where you are.

Note: This will not work until a valid position has been sent from the same call-ssid you used when you logon to the server.

Syntax: m/dist

Where: m = my range command
dist = distance in kilometers from lat/lon.

Samples: m/500 This will pass all traffic within 500 km from my location

#11 Friend Range filter

The friend filter works the same as My range filter, except you define which call-ssid should be used (see more above). Up to 9 friend filters can be defined. This is a moving filter so it is following the call-ssid last known position.

Note: This will not work until a valid position has been sent from the call-ssid defined.

Syntax: f/call/dist

Where: f = friend range command
call = call to be used as center of the range
dist = distance in kilometers from lat/lon.

Samples: f/SM5NRK/500 This will pass all traffic within 500 km from SM5NRK's last position.

#12 Port of entry filter

This filter will pass packets which match the CallSSID that follow immediately after the q-construct. This filter supports wildcard.

Syntax: e/call/call/call*...

#13 Unproto filter

This filter will pass packets which match the Unproto (or destination field) in the packet. This filter supports wildcard.

Syntax: u/text/text/te*...

#14 Exclusion filter

All the above filters also support exclusion. By prefixing the above filters with a dash the result will be the opposite. Any packet that matches the exclusion filter will NOT pass. The exclusion filters will be processed first so if there is a match for an exclusion then the packet is not passed no matter any other filter definitions.

Samples: -a/50/-130/20/-70 -b/CW* The area filter says to pass all traffic in US. The -b filter says to exclude any stations that start with CW.
-a/50/-130/20/-70 -s/>j The area filter says to pass all traffic in US. The -s filter says to exclude any stations with Car or Jeep symbols.

Roger, SM5NRK

Advanced Knowledge Base

Server Filter Commands

APRS-IS > APRS-IS Specifications > Connecting to APRS-IS > Server Filter Commands

This is derived from the javAPRSFilter Users Guide. (see above) All commands to javAPRSFilter start with the word "filter" and followed by one or more filter specifications. For instance, to specify all packets near Dallas Texas plus all NWS bulletins, you would use the following line:

filter r/33/-97/200 t/n

The default filter is not to pass anything in addition to what javAPRSSrvr will pass. So a user-defined filter port (14580) will pass messages to the client and any gated station, and nothing else until a filter definition is added. Multiple filter definitions can be setup separated by spaces. If any of the filters find a match the packet is passed.

With javAPRSFilter 3.0, you can prevent the filter from passing certain packets by prefixing the filter parameter with a hyphen (-). This tells the filter to approve any packets that match the include filters **except** those that match the exclude filters. Standard port functionality such as messaging for IGates is not affected.

For instance, to get all stations within 200 km of me except stations with the prefix of CW, I would use:

```
filter m/200 -p/CW
```

javAPRSFilter 3.0 also allows you to use decimal degrees for latitude and 3.1 added the e & u filters, the radius specification for the t filter, and wildcards for the d filter. 3.1 also added the command "filter default" which resets the filter to the predefined filter for that port.

You can check the version of javAPRSFilter that the server is using by accessing its status page, usually on port 14501 (for instance, <http://first.aprs.net:14501>). The status page is also a good way to verify that all of your filter was accepted.

The filter command may be set as part of the login line, as an APRS message to SERVER, or as a separate comment line (#filter r/33/-97/200). The preferred method is to set the command as part of the login which is supported by most current APRS software.

Below are the available filters:

Parameter	Filter Type	Description
r/lat/lon/dist	Range filter	Pass posits and objects within dist km from lat/lon. lat and lon are signed degrees, i.e. negative for West/South and positive for East/North. Up to 9 range filters can be defined at the same time to allow better coverage. Messages addressed to stations within the range are also passed.
p/aa/bb/cc...	Prefix filter	Pass traffic with fromCall that start with aa or bb or cc...
b/call1/call2...	Budlist filter	Pass all traffic from exact call: call1, call2, ... (* wild card allowed)
o/obj1/obj2...	Object filter	Pass all objects with the exact name of obj1, obj2, ... (* wild card allowed)
t/poimqstunw t/poimqstu/call/km	Type filter	Pass all traffic based on packet type. One or more types can be defined at the same time, t/otq is a valid definition. p = Position packets o = Objects i = Items m = Message q = Query s = Status t = Telemetry u = User-defined n = NWS Weather & Weather Objects w = Weather

		<p>Note: The weather type filter also passes positions packets for positionless weather packets.</p> <p>The second format allows putting a radius limit around "call" (station callsign-SSID or object name) for the requested station types.</p>
s/pri/alt/over	Symbol filter	<p>pri = symbols in primary table alt = symbols in alternate table over = overlay character (case sensitive)</p> <p>For example: s/-> This will pass all House and Car symbols (primary table) s//# This will pass all Digi with or without overlay s//#/T This will pass all Digi with overlay of capital T</p>
d/digi1/digi2...	Digipeater filter	The digipeater filter will pass all packets that have been digipeated by a particular station(s) (the station's call is in the path). This filter allows the * wildcard.
a/latN/lonW/latS/lonE	Area filter	The area filter works the same as rang filter but the filter is defined as a box of coordinates. The coordinates can also be seen as upper left coordinate and lower right. Lat/Lon are decimal degrees. South and west are negative. Up to 9 area filters can be defined at the same time.
e/call1/call1/...	Entry station filter	This filter passes all packets with the specified callsign-SSID(s) immediately following the q construct. This allows filtering based on receiving IGate, etc. Supports * wildcard.
u/unproto1/unproto2/...	Unproto filter	This filter passes all packets with the specified destination callsign-SSID(s) (also known as the To call or unproto call). Supports * wildcard.
q/con/ana	q Construct filter	<p>q = q Construct command con = list of q Construct to pass (case sensitive) ana = analysis based on q Construct.</p> <p>I = Pass positions from IGATES identified by qAr or qAR.</p> <p>For example: q/C Pass all traffic with qAC q/rR Pass all traffic with qAr or qAR q//I Pass all position packets from IGATES identified in other packets by qAr or qAR</p>
m/dist	My Range filter	This is the same as the range filter except that the center is defined as the last known position of the logged in client.
f/call/dist	Friend Range filter	This is the same as the range filter except that the center is defined as the last known position of call. Up to 9 friend filters can be defined at the same time.

Peter, AE5PL

Comment

This is the beacon comment sent with APRS position packets. See also Configure Beacon.

Symbol

You can click on the symbol to choose a different symbol for your station. This symbol is used for local display (unless you override it with a Nickname) as well as

being sent with APRS position packets. For a look at all of the available symbols, see symbols

Advanced Knowledge Base

A standard APRS-format position report looks like:

!3612.34N/11518.95W>

The latitude and longitude are expressed in degrees, minutes and decimal fractions of minutes, not degrees-minutes-seconds. (This is the standard NMEA format for lat/long output by GPS receivers, and is also the default format for APRS.) Thus the example above says "36 degrees 12.34 minutes north latitude" and "115 degrees 18.95 minutes west longitude".

The character after the latitude (emphasized in red above) determines if the symbol to be displayed is from the primary set or secondary set below. The character after the longitude (emphasized in green above) determines which symbol out of the selected set will be displayed. The example above says to use the primary set, and to select the red car in the third position from the right in the second row (the character ">" and the car are in the same position in the two tables).

!	"	#	\$	%	&	'	()	*	+	,	-	.	/	0
1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	@
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_	`
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
q	r	s	t	u	v	w	x	y	z	{		}	~		

Choosing a character from the table above causes the symbol in the corresponding position from one of the tables below to be displayed. The cross-hair symbols occupy symbol slots that have not yet been assigned.

Primary Symbol Table - Selected by placing "/" between Lat and Long

Secondary Symbol Table - Selected by placing "\" between Lat and Long

Overlay Symbols

Some of the secondary table symbols can have an alphanumeric overlay. The alternate symbols that can have an overlay are highlighted in yellow in the text grid above. The overlay character can be a single digit 0 through 9 or letter A through Z. It is placed in the position after latitude instead of "\". NOTE: Many APRS applications don't properly implement the overlay technique. In such programs, you will see the base symbol from the secondary set, but not the overlaid character or digit.

This will be most obvious with new generation digipeaters that use the solid green star in the secondary set overlaid with a letter to indicate capability, instead of the traditional star with fixed "D" from the primary set.

Stephen, WA8LMF

Advanced Knowledge Base

APRS SYMBOL OVERLAY and EXTENSION TABLES in APRS 1.2

07 Oct 2013

BACKGROUND: This file addresses new additions proposals (OVERLAYS) to the APRS symbol set after 1 October 2007. The master symbol document remains on the www.aprs.org/symbols/symbolsX.txt page.

NOTE: There was confusion with different copies of this file on different web pages and links. THIS file is now assumed to be the CORRECT one.

UPDATES/REVISIONS/CORRECTIONS:

07 Oct 13 Added new overlays to ships such as Jet Ski, Js
 Added Ham Club symbol as a C overlay on House, C-
 19 Sep 11 Added T and 2 overlays for TX 1 and 2 hop IGates
 Added overlays to (;) portable, to show event types
 23 Mar 11 Added Radiation Detector (RH)
 20 Apr 10 Byonics requested (BY)
 04 Jan 10 Added #A to the table (correcting earlier omission)
 12 Oct 09 Added W0 for Yaesu WIRES nodes
 09 Apr 09 Changed APRStt symbol to overlaid BOX (#A)
 21 Aug 08 Added RFID R=, Stroller B], Radios#Y, & skull&Xbones (XH)
 27 Apr 08 Added some definitions of the numbered circle #0.
 25 Mar 08 Added these new definitions of overlays:

\A	- (BOX symbol) APRStt(DTMF), RFID users, XO (OLPC)
\'	- Was Crash Site. Now expanded to be INCIDENT sites
\%	- is an overlaid Powerplant. See definitions below
\H	- \H is HAZE but other H overlays are HAZARDs. WH is "H.Waste"
\Y	- Overlays for Radios and other APRS devices
\k	- Overlay Special vehicles. A = ATV for example
\u	- Overlay Trucks. "Tu" is a tanker. "Gu" is a gas truck, etc
\<	- Advisories may now have overlays
\8	- Nodes with overlays. "G8" would be 802.11G
\[- \[is wall cloud, but overlays are humans. S[is a skier.
\h	- Buildings. \h is a Ham store, "Hh" is Home Depot, etc.

Previous edition was 4 Oct 2007.

In April 2007, a proposal to expand the use of overlay bytes for the extension of the APRS symbol set was added to the draft APRS1.2 addendum web page. The following document addresses that proposal:

www.aprs.org/symbols/symbols-overlays.txt

For details on Upgrading your symbol set, please see the background information on Symbols prepared by Stephen Smith, WA8LMF:

www.aprs.org/symbols/symbols-background.txt

CONSISTANCY: Since the objective of APRS is consistent, reliable communications at the local level, there has been a hesitance to making significant changes to the APRS symbol set. The Integrity of APRS depends on everyone seeing the same information at the same time. Frequent changes to the symbol sets can actually undermine that integrity and operational utility of APRS and end up with worse outcomes due to miss-communications than the lack of any particular symbol might suggest.

OVERLAY HISTORY: When the overlay symbol set was first defined for the original APRS back in 1995, it had the potential to expand the APRS symbol set from the 94 original primary symbols to a secondary set that could each have as many as 36 different overlays on each of those secondary symbols up to almost 3500 combinations. But some authors then could not easily implement these overlays, except by one-by-one exceptions to their code.

For this reason, a compromise was made with those authors and then eventually written into the APRS spec to limit overlays to only a small subset of alternate symbols. Those original overlayable alternate symbols were labeled with a "#" and called "numbered" symbols. (UIview requires "No." in the symbols.ini file)

STATUS OF OVERLAYS 1 OCTOBER 2007: the APRS symbol set only had a few remaining unused symbol codes that had not yet been defined:

OF THE 94 Primary Symbols. The following were available:

- 10 symbols (/0 - /9) that mostly look like billiard balls now
- 4 symbols /D, /J, /Q, /z were undefined or TBD
- 2 were reserved

OF THE 94 Alternate Symbols. The following were available:

- 3 undefined series \=, \Y, \Z which could do 36 overlays
- 8 series \1 through \8 that can support 36 overlays each
- 3 reserved series.

ADDITIONAL OVERLAY PROPOSAL: But any of the other 79 alternate symbols could all have multiple (36) overlays if they can make sense with the existing underlying basic symbol that we have been using for that basic alternate symbol. That is, any new definition of a previously unused overlay character will have undefined results on all prior APRS systems and should be used with caution. But the symbol set is extensible with these cautions. (See the Proposal that would expand the APRS symbol set to over 3200 at the bottom of this document.)

SYMBOL OVERLAY TABLES: This document will keep track of all definitions of overlays on all ALTERNATE symbols. Although these overlays were originally intended to just overlay a displayable single character on a basic symbol, there is no prohibition against taking the combination of a symbol and specific overlay, and then letting that define a new graphic just for that combination.

The following tables will attempt to keep track of these and any other useful generic applications of overlay characters.

ATM Machine or CURRENCY: # \$

- /\$ = original primary Phone
- \\$ = Bank or ATM (generic)
- U\$ = US dollars
- L\$ = British Pound
- Y\$ = Japanese Yen

POWER PLANT: # %

- /% = DX cluster <= the original primary table definition
- C% = Coal
- G% = Geothermal
- H% = Hydroelectric
- N% = Nuclear

S% = Solar
 T% = Turbine
 W% = Wind

GATEWAYS: #&

/& = HF Gateway <= the original primary table definition
 I& = Igate Generic (please use more specific overlay)
 R& = Receive only IGate (do not send msgs back to RF)
 T& = TX igate with path set to 1 hop only)
 2& = TX igate with path set to 2 hops (not generally good idea)

INCIDENT SITES: #'

\' = Airplane Crash Site <= the original primary deifinition
 A' = Automobile crash site
 H' = Hazardous incident
 M' = Multi-Vehicle crash site
 P' = Pileup
 T' = Truck wreck

HUMAN SYMBOL: #[

/[= Human
 \[= Wall Cloud (the original definition)
 B[= Baby on board (stroller, pram etc)
 S[= Skier * <= Recommend Special Symbol
 R[= Runner
 H[= Hiker

HOUSE: #-

/- = House
 \- = (was HF)
 5- = 50 Hz mains power
 6- = 60 Hz mains power
 B- = Backup Battery Power
 C- = Club, as in Ham club
 E- = Emergency power
 G- = Geothermal
 H- = Hydro powered
 O- = Operator Present
 S- = Solar Powered
 W- = Wind powered

NUMBERED CIRCLES: #0

E0 = Echolink Node (E0)
 I0 = IRLP repeater (I0)
 S0 = Staging Area (S0)
 W0 = WIRES (Yaesu VOIP)

NETWORK NODES: #8

88 = 802.11 network node (88)
 G8 = 802.11G (G8)

PORTABLE SYMBOL: #;

/; = Portable operation (tent)
 \; = Park or Picnic
 F; = Field Day
 I; = Islands on the air
 S; = Summits on the air
 W; = WOTA

ADVISORIES: #< (new expansion possibilities)

/< = motorcycle
 \< = Advisory (single gale flag)

CARS: #>

/> = normal car (side view)
 \> = Top view and symbol POINTS in direction of travel
 E> = Electric
 H> = Hybrid
 S> = Solar powered
 V> = GM Volt

BOX SYMBOL: #A (and other system inputted symbols)

/A = Aid station
 \A = numbered box
 9A = Mobile DTMF user
 7A = HT DTMF user
 HA = House DTMF user
 EA = Echolink DTMF report
 IA = IRLP DTMF report
 RA = RFID report
 AA = AllStar DTMF report
 DA = D-Star report
 XA = OLPC Laptop XO
 etc.

EYEBALL and VISIBILITY #E

/E = Eyeball for special live events
 \E = (existing smoke) the symbol with no overlay
 HE = (H overlay) Haze
 SE = (S overlay) Smoke
 BE = (B overlay) Blowing Snow was \B
 DE = (D overlay) blowing Dust or sand was \b
 FE = (F overlay) Fog was \{

HAZARDS: #H

/H = hotel
 \H = Haze
 RH = Radiation detector (new mar 2011)
 WH = Hazardous Waste
 XH = Skull&Crossbones

RESTAURANTS: #R

\R = Restaurant (generic)
 7R = 7/11
 KR = KFC
 MR = McDonalds
 TR = Taco Bell

RADIOS and APRS DEVICES: #Y

/Y = Yacht <= the original primary symbol
 \Y = <= the original alternate was undefined
 AY = Alinco
 BY = Byonics
 IY = Icom
 KY = Kenwood * <= Recommend special symbol
 YY = Yaesu/Standard* <= Recommend special symbol

GPS devices: #\

/\ = Triangle DF primary symbol
 \\ = was undefined alternate symbol
 A\ = Avmap G5 * <= Recommend special symbol

ARRL or DIAMOND: #a

/a = Ambulance
 Aa = ARES
 Da = DARES Dutch AR Emergency Services
 Ga = RSGB Radio Society of Great Brittan
 Ra = RACES
 Sa = SATERN Salvation Army
 Wa = WinLink

CIVIL DEFENSE or TRIANGLE: #c

/c = Incident Command Post
 \c = Civil Defense
 Rc = RACES
 Sc = SATERN mobile canteen

BUILDINGS: #h

/h = Hospital
 \h = Ham Store ** <= now used for HAMFESTS
 Hh = Home Dept etc..

SPECIAL VEHICLES: #k

/k = truck
 \k = SUV
 4k = 4x4
 Ak = ATV (all terrain vehicle)

SHIPS: #s

/s = Power boat (ship) side view
 \s = Overlay Boat (Top view)
 Cs = receive as Canoe but still transmit canoe as /C
 Js = Jet Ski
 Ks = Kayak
 Hs = Hovercraft

TRUCKS: #u

/u = Truck (18 wheeler)
 \u = truck with overlay
 Gu = Gas
 Tu = Tanker
 Cu = Chlorine Tanker
 Hu = Hazardous

Anyone can use any overlay on any of the overlayable symbols for any special purpose. We are not trying to document all possible such overlays. The purpose of this document is to help keep a list of more common such definitions that have more universal use and for which multiple definitions would lead to confusion.

Future APRS code writers should be aware of where we are going:

www.aprs.org/symbols/symbols-overlays.txt

In the initiative to upgrade APRS symbols, we are wasting some very valuable OVERLAYABLE symbol subgroups with a few nailed down legacy weather symbols. We are proposing to consolidate some of these Weather symbols to open up much more space. Since this is the first time we have considered actually CHANGING some symbol definitions, this can cause problems out there for some users of some legacy systems.

That is why I am posting this plan to the APRS community. If we do this, XASTIR and UIVIEW will be able to download new symbol definitions. But some legacy clients that do not operate from external symbol files will show the wrong symbols for these. If users of those systems foresee some serious problems with the re-arrangement of these symbols, let us know the specific impact and your ideas for a workaround..

The symbols that would be impacted are as follows:

First, consolidate all of the visibility symbols into the old SMOKE symbol and change its meaning to "VISIBILITY", and then differentiate them with the overlay characters.

"\E"	(existing smoke) the symbol with no overlay	
"HE"	(an H overlay) will mean Haze	
"SE"	(an S overlay) will mean Smoke	
"BE"	(a B overlay) will mean Blowing Snow	was \B
"DE"	(a D overlay) will mean blowing Dust or sand	was \b
"FE"	(an F overlay) will mean Fog	was \{

Another category is to expand the RAIN symbol to make it kinda like lots of angled dots coming from the sky, but with an open center so that we can use overlays for a number of common PRECIPITATIONS. The consolidations would be:

"\`"	(existing Rain) would be the symbol with no overlay	
"R`"	(an R overlay) would mean Rain	
"F`"	(an F overlay) would mean Freezing Rain	was \F
"H`"	(an H overlay) would mean Hail	was \:
"D`"	(an D overlay) would mean Drizzle	was \D
"E`"	(an E overlay) would mean sleet	was \e
"S`"	(an S overlay) would mean Snow	was *

etc. and other particulates coming from the sky

Next, I propose expanding the existing RAIN SHOWER "\I" symbol to look like some kind of cloud symbol with specks in it that can be overlaid. (It needs to look different from the next CLOUD symbol). It can then consolidate these symbols:

"RI"	(an R overlay) would mean Rain Shower	
"SI"	(an S overlay) would mean Snow shower	was \G
"LI"	(an L overlay) would mean Lightning	was \J

etc. and other things related to clouds

Next, I propose expanding the existing CLOUD symbol to allow definition of any number of different types of cloud. This needs to also look like a cloud but a different shape and allow for overlays (Maybe this cloud is clear):

"\""	is the existing cloud symbol (would have no overlay)	
"P\""	with P overlay would mean partly cloudy	was \p
"W\""	with W overlay would be a wall cloud	was \[
"F\""	would be Funnel cloud, but the original "\f" will also be retained for now	

All of these initiative will free up a lot of overlayable symbol GROUPS each of which can suport up to 36 different overlays in each group for the future:

#H	for 36 new Hazards (was only Hail)	
#[for 36 new human symbols (was only Wall Cloud)	
#\	for 36 new GPS or navigation equipment	
#B	TBD. \B was only blowing snow	now is BE
#b	TBD. \b was only blowing dust/sand	now is DE
#{	TBD. \{ was only fog	now is FE
#*	TBD. * was snow only	now is S`
#:	TBD. \: was hail only	now is H`
#D	TBD. \D was drizzle only	now is D`
#F	TBD. \F was freezing rain only	now is F`
#e	TBD. \e was sleet only	now is E`
#G	TBD. \G was only Snow shower	now is SI
#J	TBD. \J was only Lightening	now is LI
#p	TBD. \p was only partly cloud	now is P(

Of course future code can fully draw each of these overlays as distinct special symbols in any way they want.

I especially want to hear from Dale Hugley who is a resource for weather, and Stephen Smith who will have to draw them for Uiview. And others with a stake in this...

Bob, WB4APR

Genius

This button provides access to the **Configure** ⇨ **Genius** dialog box.

NMEA

This button provides access to the NMEA GPS configuration. You can run APRSIS32 without a GPS by manually panning and zooming the map and clicking Transmit to confirm your desire to move ME. If you are running APRSISCE on Windows Mobile and have an internal GPS supported by the uSoft GPSapi (most are), you do not need to configure an NMEA GPS.

PHG

The PHG button will prompt for Power, Height Above Average Terrain (HAAT), Gain, and directional characteristics of your station. The resultant PHG string will be automatically inserted into your Comment string. If you are running a non-RF station, you don't need to even think about PHG.

Accept

This will accept all changes and close the dialog.

Cancel

This will close the dialog discarding all changes made. (This may eventually disappear and become an X on the title bar)

GeniusBeaconing™...

APRSIS32 extends the Smartbeaconing concept to a whole new level made possible by running the client on platforms more powerful than a typical PIC.

You should not need to make any changes to the GeniusBeaconing(tm) settings until you have run APRSIS32 for a while. If you don't have a GPS, then you definitely do not need to change anything here except maybe the Max Time which will be the only thing governing your beacon interval.

A good description of the GeniusBeaconing can be found in the description of the red-dot behavior.

Min Time - Under no circumstances will APRSIS32 automatically beacon in less elapsed time than the Min Time seconds.

Max Time - If nothing else triggers a beacon, APRSIS32 will transmit one every Max Time minutes. If there is no GPS input, or if the GPS is disabled, then beacons (if enabled) will be transmitted at this rate.

Time Only - If this is checked, then only the time parameters are used and the remainder of the settings on this dialog are ignored.

Start/Stop - If checked, APRSIS32 will transmit a position beacon whenever you speed drops to zero or leaves zero, governed by the Min Time of course. If you frequently drive in start/stop traffic, you may want to uncheck this to avoid frequent (Min Time) beaconing.

Heading Change - If your heading changes by more than this many degrees since the last beacon, a new beacon will be triggered. If you have problems with frequent beaconing due to a drifting GPS, simply set this value to 180. The Forecast Error will still provide very accurate, predictable beacons. (This parameter may be removed in some future version because Forecast Error works so well).

Forecast Error - This is the "Genius" of GeniusBeaconing(tm). If an outside observer were to calculate your position based on the speed, heading, and elapsed time since your last beacon and your actual current position is more than "Forecast Error" from that position, a beacon will be transmitted. The magnitude and direction of the error between a forecast from your last beacon and your actual current position is shown by the position of the red-dot relative to the range circle. When the red-dot touches the circle, the Forecast Error has been exceeded and a beacon will be triggered (if allowed by Min Time).

Max Distance - If you have "made good" more than this distance since your last beacon, a new beacon will be triggered. Note that this is expressed in 1/10 mile increments, so remember to multiply by 10.

Accept / Cancel - Obvious operation here.

Advanced Knowledge Base

Red Dot Function

The red dot is a visual indicator of the Genius Beaconing's Forecast Error.

Genius Beacons' Forecast Error works by calculating where an outside observer would think you are based on your last beacons speed and heading. It takes that last information and extends it by the time it has been since your last beacon to arrive at an expected lat/lon. This is compared to your actual lat/lon (which, of course, only you know at this point) to derive an error vector (distance and direction). The red dot is plotted inside the circle in that direction and scaled such that the circle is the configured Genius Forecast Error distance.

When the red dot hits the circle, the configured Forecast Error distance has been reached and a new beacon is sent (provided that at least the Min Time has been reached).

So, if you're driving along in a straight line at a steady speed, you won't see the red dot. (It's actually bouncing around a bit underneath your icon in the center of the circle).

If you slow down, that outside observer will still think you're driving fast and will have you further ahead of where you are and the red dot will start moving towards the top of the screen. If you then resume your original speed, the red dot will freeze at that distance ahead. If you go faster than your last beacons speed the red dot will drift back to the center and will then start falling towards the bottom of the screen as your actual location gets ahead of where the elapsed-time forecast would place you.

Slight heading changes are even more fun. If you turn to the left (by less than the configured Heading Change or that will trigger a new beacon), the red dot will begin drifting to the right. Veer right and it will drift to the left. Resume straight line motion in the new heading and the dot will continue to drift as your actual location continues to get further away from the course indicated by the previously beacons speed and heading.

Lynn, KJ4ERJ

Status...

The Status Configuration dialog allows setting both the Beacon Comment (redundant with [Configure](#) ⇒ [General](#) and [the Configure](#) ⇒ [Status](#) ⇒ [Report](#)).

Beacon Comment:

Comment - This is the same comment string configurable in [Configure](#) ⇒ [General](#) and also selectable via [Configure](#) ⇒ [Beacon](#) .

Interval - The default value of 0 causes the comment to be included with every position beacon (if enabled in [Configure](#) ⇒ [Beacon](#). APRSIS32 also supports only including the comment on position beacons at a configurable interval. A position beacon won't be triggered just to carry the comment, but the comment will be sent at most once in each configured interval. Note that some APRS radios think the station has significantly changed if the comment is or is not present, so you might be asked why your station keeps triggering the display on those radios as if the station was newly received.

Status Report - The status report is a completely different specification in APRS than the beacon comment. See chapter 16 in [aprs101.pdf](#) for details.

Enabled - If checked, status report packets will be periodically generated by APRSIS32.

GridSquare - If checked, the current GridSquare will be included in the Status Report. Use this with caution if you are also transmitting position beacons as the position resolution between the GridSquare and Lat/Lon coordinates is significantly different, possibly resulting in a station that ping-pongs between two locations depending on which packet was heard most recently.

Timestamp - If checked, the current timestamp (ddhhmm zulu, unfortunately not hhmmss) will be included in the Status Report. This can be handy in demonstrating packet delays in the APRS RF network possibly in excess of the 30 second APRS-IS duplicate filter causing two instances of a single Status Report to appear on APRS-IS. Without a timestamp, it's hard to prove that the packet was duplicated.

Status - This is the actual text included in the Status Report. Also selectable via [Configure](#) ⇒ [Status](#) ⇒ [Report](#) .

Interval - Status Report packets will be generated every Interval minutes.

PHG - Inserts a Power/Height/Gain string into the Beacon Comment. See [Configure](#) ⇒ [General](#) .

Accept / Cancel

Aliases

Aliases are used within paths of APRS so that users don't need to know the specific digipeaters in order to use the installed RF network. The most popular alias is WIDE in the typical incarnations of [WIDE1-1](#), [WIDE2-1](#) and [WIDE1-1](#), [WIDE2-2](#). Older (pre-WIDEn-N) aliases are RELAY and WIDE (yes, same alias as we're still using). There are also local state or province aliases referred to as [SSn-N](#), but there is no central clearing house for the values of [SS](#). Finally, there are other aliases that allow a packet originator to request specific behaviors from infrastructure components that may process a packet. Some examples of these are [GATE](#), [NOGATE](#), and [RFONLY](#).

APRSISCE/32 needs to know what aliases it might encounter in order to properly count used hops when making decisions on "local" stations for message gating decisions from APRS-IS to RF. To that end, APRSISCE/32 allows the configuration of aliases which will not be counted when encountered within the used portion of a packet's path. This list of known aliases comes pre-populated with well-known aliases which are visible in the Known cascading menu. Selecting a specific alias will toggle it between active (not counted) and inactive (counted).

Note: Until the performance impact of this is determined, you MUST enable at least one of the Alias() Trace Logs before any of the following will work. Enabling Alias(New) is recommended because it doesn't actually increase any logging, but enables the processing necessary for the following functionality. Also, this will probably only be available in Development Mode.

During packet processing, APRSISCE/32 attempts to identify possible new aliases conforming to the **SSn-N** format (a string of alpha characters followed by a single digit either used (*) or followed by a dash and another single digit). These "learned" aliases are visible in the possible cascading menu. Selecting a possible alias will move it to the known list after a confirmation.

Finally, there are some digipeaters that seem to not be marking fully-decremented **n-N** aliases as used. This results in a path containing just **SSn** which implies a **-0**. APRSISCE/32 detects these possible **SSn-0** aliases when they are the first unused path component and makes them visible in the unused-0 cascading menu along with the preceding digipeater and platform of that station. This is done primarily for diagnosing the local network configuration and behavior.

To back up this alias processing, APRSISCE/32 supports the following Trace Logs.

Alias(-0) Shows information on "Unused-0" alias detection and the associated packet. The first appearance of a given alias and digipeater is always logged. Subsequent references are only logged when the trace is enabled.

Alias(First-0) Shows information on packets with no used components where the first path component appears to be an alias without an -SSID, implying -0 which may mean that an **SSn-N** is missing the **-N**.

Alias(Dbl-0) Similar to Alias(First-0), this shows packets whose first unused component appears to be an -SSID-less alias and comes after a path alias that is marked used. This happens with things like **WIDE2*,VA2** and may indicate an **SSn-N** that is missing the **-N**.

Alias(New) Shows all newly discovered "Possible" aliases and the first referencing packet.

Alias(Numeric) When enabled, shows all n-N path components and associated packets. These are most often an incorrectly entered path with extra commas **WIDE,1-1,WIDE,2-1** or missing aliases **WIDE1-1,2-1**.

Alias(Used) When enabled, shows all packets referencing any Possible alias, regardless of New status.

AltNet

ALTNETS are uses of the AX-25 tocall to distinguish specialized traffic that may be flowing on the APRS-IS, but that are not intended to be part of normal APRS distribution to all normal APRS software operating in normal (default) modes. Proper APRS software that honors this design are supposed to IGNORE all ALTNETS unless the particular operator has selected an ALTNET to monitor for.

An example is when testing; an author may want to transmit objects all over his map for on-air testing, but does not want these to clutter everyone's maps or databases.

He could use the ALTNET of "TEST" and client APRS software that respects the ALTNET concept should ignore these packets.

An ALTNET is defined to be ANY AX.25 TOCALL that is NOT one of the normal APRS TOCALL's. The normal TOCALL's that APRS is supposed to process are: ALL, BEACON, CQ, QST, GPSxxx and of course APxxxx.

Advanced Knowledge Base

APRS TO-CALL VERSION NUMBERS

06 FEB 2017

WB4APR

06 Feb 17 added APIExx for W7KMV's PiAPRS system
 25 Jan 17 added APSFxx F5OPV embedded devices - was APZ40
 16 Dec 16 added APYSxx for W2GMD's Python APRS
 14 Nov 16 Added APINxx for PinPoint by AB0WV
 09 Nov 16 added APNICx for SQ5Eku <http://sq5eku.blogspot.com/>
 24 Oct 16 added APTKPT TrackPoint N0LP, removed APZTKP
 24 Aug 16 added APK004 for Kenwood THD-74
 29 Apr 16 added APFPRS for FreeDV by Jeroen PE1RXQ
 25 Feb 16 Added APCDS0 for Leon Lessing ZS6LMG's cell tracker
 21 Jan 16 added APDNOx for APRSduino by DO3SWW
 18 Nov 15 Added APSTPO for N0AGI
 03 Nov 15 Updated APAND1 and APDRxx for androids
 26 Oct 15 added APZ247 for UPRS
 09 Sep 15 added APHTxx for HMTracker by IU0AAC
 06 Aug 15 added APMTxx for LZ1PPL for tracker
 27 Apr 15 added APZMAJ for Martyn M1MAJ DeLorme inReach Tracker
 21 Apr 15 added APB2MF & APR2MF DL2MF - MF2APRS Radiosonde
 06 Apr 15 added APAVT5 SainSonic AP510 - a 1watt tracker
 ...
 11 Jan 12 added APYTxx for YagTracker and updated Yaesu APY008/350

In APRS, the AX.25 Destination address is not used for packet routing as is normally done in AX.25. So APRS uses it for two things. The initial APxxxx is used as a group identifier to make APRS packets instantly recognizable on shared channels. Most applications ignore all non APRS packets. The remaining 4 xxxx bytes of the field are available to indicate the software version number or application. The following applications have requested a TOCALL number series:

APn 3rd digit is a number
 AP1WWX TAPR T-238+ WX station
 AP1MAJ Martyn M1MAJ DeLorme inReach Tracker
 AP4Rxy APRS4R software interface
 APnnnD Painter Engineering uSmartDigi D-Gate DSTAR Gateway
 APnnnU Painter Engineering uSmartDigi Digipeater
 APA APAFxx AFilter.
 APAGxx AGATE
 APAGWx SV2AGW's AGWtracker
 APALxx Alinco DR-620/635 internal TNC digis. "Hachi" ,JF1AJE
 APAXxx AFilterX.
 APAHxx AHub
 APAND1 APRSdroid (pre-release) <http://aprsdroid.org/>
 APAMxx Altus Metrum GPS trackers
 APAVT5 SainSonic AP510 which is a 1watt tracker
 APAWxx AGWPE
 APB APBxxx Beacons or Rabbit TCPIP micros?
 APB2MF DL2MF - MF2APRS Radiosonde for balloons

APBLxx BigRedBee Beeline
 APBLO M0del Rocketry K7RKT
 APBPQx John G8BPQ Digipeater/IGate
 APC APCxxx Cellular applications
 APCBBx VE7UDP Blackberry Applications
 APCDS0 Leon Lessing ZS6LMG's cell tracker
 APCLEY EYTraker GPRS/GSM tracker by ZS6EY
 APCLWX EYWeather GPRS/GSM WX station by ZS6EY
 APCLEZ Telit EZ10 GSM application ZS6CEY
 APCWP8 John GM7HHB, WinphoneAPRS
 APCYxx Cybiko applications
 APD APD4xx UP4DAR platform
 APDDxx DV-RPTR Modem and Control Center Software
 APDFxx Automatic DF units
 APDGxx D-Star Gateways by G4KLX ircDDB
 APDHxx WinDV (DUTCH*Star DV Node for Windows)
 APDInn DIXPRS - Bela, HA5DI
 APDKxx KI4LKF g2_ircddb Dstar gateway software
 APDNOx APRSduino by DO3SWW
 APDOxx ON8JL Standalone DStar Node
 APDPRS D-Star originated posits
 APDRxx APRSdroid Android App <http://aprsdroid.org/>
 APDSXX SP9UOB for dsDigi and ds-tracker
 APDTxx APRStouch Tone (DTMF)
 APDUxx U2APRS by JA7UDE
 APDWxx DireWolf, WB2OSZ
 APE APExxx Telemetry devices
 APECAN Pecan Pico APRS Balloon Tracker
 APERXQ Experimental tracker by PE1RXQ
 APF APFxxx Firenet
 APFGxx Flood Gage (KP4DJT)
 APFIxx for APRS.FI OH7LZB, Hessu
 APFPRS for FreeDV by Jeroen PE1RXQ
 APG APGxxx Gates, etc
 APGOxx for AA3NJ PDA application
 APH APHKxx for LA1BR tracker/digipeater
 APHAXn SM2APRS by PY2UEP
 APHTxx HMTracker by IU0AAC
 API APICQx for ICQ
 APICxx HA9MCQ's Pic IGate
 APIExx W7KMV's PiAPRS system
 APINxx PinPoint by AB0WV
 APJ APJAx JavAPRS
 APJExx JeAPRS
 APJIXx jAPRSIGate
 APJSxx javAPRSSrvr
 APJYnn KA2DDO Yet another APRS system
 APK APK0xx Kenwood TH-D7's
 APK003 Kenwood TH-D72
 APK004 Kenwood TH-D74
 APK1xx Kenwood D700's
 APK102 Kenwood D710
 APKRAM KRAMstuff.com - Mark. G7LEU
 APL APLQRU Charlie - QRU Server
 APLMxx WA0TQG transceiver controller
 APM APMxxx MacAPRS,
 APMGxx MiniGate - Alex, AB0TJ
 APMIxx SQ3PLX <http://microsat.com.pl/>

- APMTxx LZ1PPL for tracker
- APN APNxxx Network nodes, digis, etc
 - APN3xx Kantronics KPC-3 rom versions
 - APN9xx Kantronics KPC-9612 Roms
 - APNAxx WB6ZSU's APRServe
 - APNDxx DIGI_NED
 - APNICx SQ5EKU <http://sq5eku.blogspot.com/>
 - APNK01 Kenwood D700 (APK101) type
 - APNK80 KAM version 8.0
 - APNKMP KAM+
 - APNMxx MJF TNC roms
 - APNPxx Paccom TNC roms
 - APNTxx SV2AGW's TNT tnc as a digi
 - APNUxx UIdigi
 - APNXxx TNC-X (K6DBG)
 - APNWxx SQ3FYK.com WX/Digi and SQ3PLX <http://microsat.com.pl/>
- APO APRSpoint
 - APOLUx for OSCAR satellites for AMSAT-LU by LU9DO
 - APOAxx OpenAPRS - Greg Carter
 - APOTxx Open Track
 - APOD1w Open Track with 1 wire WX
 - APOU2k Open Track for Ultimeter
 - APOZxx www.KissOZ.dk Tracker. OZ1EKD and OZ7HVO
- APP APPTxx KetaiTracker by JF6LZE, Takeki (msg capable)
- APQ APQxxx Earthquake data
- APR APR8xx APRSdos versions 800+
 - APR2MF DL2MF - MF2APRS Radiosonde WX reporting
 - APRDxx APRSdata, APRSdr
 - APRGxx aprsg igate software, OH2GVE
 - APRHH2 HamHud 2
 - APRKxx APRStk
 - APRNOW W5GGW ipad application
 - APRRTx RPC electronics
 - APRS Generic, (obsolete. Digis should use APNxxx instead)
 - APRxxx >40 APRSmax
 - APRxxx <39 for OH2MQK's igate
 - APRTLM used in MIM's and Mic-lites, etc
 - APRtfc APRSTraffic
 - APRSTx APRStt (Touch tone)
- APS APSxxx APRS+SA, etc
 - APSARx ZL4FOX's SARTRACK
 - APSCxx aprsc APRS-IS core server (OH7LZB, OH2MQK)
 - APSFxx F5OPV embedded devices - was APZ40
 - APSK63 APRS Messenger -over-PSK63
 - APSK25 APRS Messenger GMSK-250
 - APSMSx Paul Dufresne's SMSGTE - SMS Gateway
 - APSTMx for W7QO's Balloon trackers
 - APSTPO for N0AGI Satellite Tracking and Operations
- APT APT2xx Tiny Track II
 - APT3xx Tiny Track III
 - APTAXx K4ATM's tiny track
 - APTIGR TigerTrack
 - APTKPT TrackPoint N0LP
 - APTTxx Tiny Track
 - APTWxx Byons WXTrac
 - APTVxx for ATV/APRN and SSTV applications
- APU APU1xx UIview 16 bit applications
 - APU2xx UIview 32 bit apps

APU3xx UIview terminal program
 APUDRx NW Digital Radio's UDR (APRS/Dstar)
 APV APVxxx Voice over Internet applications
 APVRxx for IRLP
 APVLxx for I-LINK
 APVExx for ECHO link
 APW APWxxx WinAPRS, etc
 APWAxx APRSISCE Android version
 APWSxx DF4IAN's WS2300 WX station
 APWMxx APRSISCE KJ4ERJ
 APWWxx APRSISCE win32 version
 APX APXnnn Xastir
 APXRnn Xrouter
 APY APYxxx Yeasu
 APY008 Yaesu VX-8 series
 APY350 Yaesu FTM-350 series
 APYTxx for YagTracker
 APYSxx for W2GMD's Python APRS
 APZ APZxxx Experimental
 APZ247 for UPRS NR0Q
 APZ0xx Xastir (old versions. See APX)
 APZMAJ Martyn M1MAJ DeLorme inReach Tracker
 APZMDR for HaMDR trackers - hessu * hes.iki.fi]
 APZPAD Smart Palm
 APZTKP TrackPoint, Nick N0LP (Balloon tracking)(depricated)
 APZWIT MAP27 radio (Mountain Rescue) EI7IG
 APZWKR GM1WKR NetSked application

Authors with similar alphabetic requirements are encouraged to share their address space with other software. Work out agreements amongst yourselves and keep me informed.

REGISTERED ALTNETS

ALTNETS are uses of the AX-25 tocall to distinguish specialized traffic that may be flowing on the APRS-IS, but that are not intended to be part of normal APRS distribution to all normal APRS software operating in normal (default) modes. Proper APRS software that honors this design are supposed to IGNORE all ALTNETS unless the particular operator has selected an ALTNET to monitor for.

An example is when testing; an author may want to transmit objects all over his map for on-air testing, but does not want these to clutter everyone's maps or databases. He could use the ALTNET of "TEST" and client APRS software that respects the ALTNET concept should ignore these packets.

An ALTNET is defined to be ANY AX.25 TOCALL that is NOT one of the normal APRS TOCALL's. The normal TOCALL's that APRS is supposed to process are: ALL, BEACON, CQ, QST, GPSxxx and of course APxxxx.

The following is a list of ALTNETS that may be of interest to other users. This list is by no means complete, since ANY combination of characters other than APxxxx are considered an ALTNET. But this list can give consintecy to ALTNETS that may be using the global APRS-IS and need some special recognition:

TEST - A generic ALTNET for use during testing
 PSKAPR - PSKmail . But it is not AX.25 anyway

Bob, WB4APR

Beacon

The **Configure** ⇒ **Beacon** menu provides direct access to the various allowed components and contents of a position beacon.

After Transmit

In a semi-mobile (not fixed) GPS-less installation, it may be desirable to not transmit a position beacon on initial startup until the last known position can be confirmed and possibly corrected. If this option is checked, APRSIS32 will not transmit a position beacon until the Transmit menu option has been manually activated. Given that the Transmit option allows the user to move "ME" to the center of the screen, this allows the actual current position to be verified and/or changed before any position beacons are sent.

Timestamp

If checked, a timestamp is included in each position packet. The default timestamp is ddhhmm zulu unless HHMMSS is also checked.

HHMMSS

Modifies the timestamp to use the hhmmss format. Including this timestamp on RF beacons can allow easy detection and identification of delayed duplicate packets being injected into APRS-IS beyond the default 30 second duplicate detection interval. When delayed packets are introduced, a mobile station can appear to ping-pong along a path as obsolete positions arrive after newer positions.

Altitude

If checked, the current altitude is included in the position packet. Note that altitude is only supported with an active GPS and then only if the GPS is providing an altitude in the fix. Fixed (non-GPS) stations cannot specify an altitude for transmission.

Compressed

If checked, position packets are transmitted in a compressed format per chapter 9 of aprs101.pdf. This Compressed format provides more resolution than the humanly readable format. This compression is NOT Mic-E which is a format parsed and displayed by APRSIS32, but not supported for transmitting positions.

CourseSpeed

If checked and a GPS is providing speed and heading data, a CSE/SPD (Chapter 7 of aprs101.pdf) is included in the position packet. If compressed is checked, then either Altitude or Course/Speed can be selected, but not both as the APRS protocol doesn't support it.

Comment

The Comment cascading menu provides direct access to a set of "remembered" comment texts. You can also specify a New comment text (optionally remembered) or None to eliminate the comment from being transmitted with the position beacons.

Mic-E Notification

Selection

!EMERGENCY!
 !OFF-DUTY!
 !ENROUTE!
 !INSERVICE!
 !RETURNING!
 !COMMITTED!
 !SPECIAL!
 !PRIORITY!
 !ALARM!
 !ALERT!
 !WARNING!
 WXALARM!
 !TESTALARM!

Symbol

The Symbol cascading menu provides direct access to a set of "remembered" symbols. You can also select a New symbol using the same dialog as used for [Configure](#) ⇒ [General](#) .

Why

If checked, causes APRSIS32 to include the Transmit Pressure that triggered the position beacon to be included in the position packet. This can be quite handy when determining the cause of too-frequent beaconing or to see which trigger is actually causing various beacons along a route. Because this text is included in the station's comment, some APRS radios will consider each beacon to have significantly changed causing a new display on each reception.

Advanced Knowledge Base

APRSISCE puts a line of text below the "APRS OK" line that shows the pressure to transmit an updated position. Can you tell me what your's is saying? It will be something like:

Dist nn%	- Traveled Percentage of Max Distance
Fore nn%	- Forecasted Percentage of Max Distance
Proj nn%	- Percentage of Forecast Error
Head nn%	- Percentage of Heading Change
Time hh:mm:ss	- Count down from Max Time
STOP	- Stopped having been moving
START	- Started having been stopped
FORCE	- Clicked the Transmit button
LostFix	- GPS Lost a Fix
GotFix	- GPS Acquired a Fix

It DOES say "APRS OK", right? If not, then you're having trouble getting connected to an APRS-IS server.

There are also three colored bars below that text that indicate (top to bottom): **Transmit Pressure** increasing left to right (Transmits when it gets to the right, see also last line)

APRS-IS Quiet Time increasing left to right, reconnects if it ever gets to the right
Transmit Throttle decreasing right to left (Counts down Min Time after beacon)

These bars aren't very wide, but the normal behavior is that the top one will crawl to the right and a beacon will be sent when it hits the right edge. At that time, the bottom bar will jump immediately to the right and start shrinking to the left as the Min Time elapses.

The center bar will grow from the left until a new packet is received from APRS-IS. If it ever gets to the right edge (set by Quiet Time which should be 60 seconds), the APRS-IS connection is dropped and a new connection is established. APRS-IS servers send out a heartbeat every 20-30 seconds if no packet has been sent, so this bar should just hover in the left half of the display, growing longer and then disappearing only to start over. If your filter has lots of stations coming in, you might never actually see this bar (although you station list will be scrolling fast!).

Precision

The default precision for a non-compressed position beacon is 1/100 minute of latitude and longitude. APRSIS32 supports increased precision via the !DAO! extension Datum as well as ambiguous position as described in Chapter 6 (page 24) of aprs101.pdf). Both 1 digit (1/1000 minute - humanly readable) and 2 digit (91/10000 minute - don't ask) additional precision is supported as well as all levels of ambiguity. Compressed packets are, by definition, accurate to within 1 foot.

Path...

Specifies the path to use when transmitting position beacons. (see New Paradigm or just set it to WIDE1-1,WIDE2-1 as a "safe" value) This is likely to be relocated in some future APRSIS32 revision to allow more discrete setting of different paths on different ports.

Advanced Knowledge Base 1

The New-EU WIDEn-N Paradigm

19 March 2006

WB4APR

Updated 25 Nov 08 because the New-N paradigm was adopted 20 Nov 08 at the final plenary of the Cavtat-IARU-Region 1 conference based on the proposal of the norwegian NRLL (paper C5-33, one worldwide aprs-standard). The paper C5_09 of the Danish Radio Society (european-aprs-standard) has been withdrawn by the EDR.

Updated 3 July 2006

Updated 21 Sept 2006

The APRS network in Europe in many areas is unreliable due to the same lack of consistent user settings and network digipeater settings that had evolved in the USA forcing us to finally implement the New-N Paradigm and get consistency and traceability throughout the network. See the map on www.aprs.org/newN/NewEU-map.GIF

The following history of APRS paths is important to understand:

RELAY	was simple digipeating but NO dupe checking
WIDE	was simple digipeating but NO dupe checking*
TRACE	simple digipeating but with call substitution & no dupe ck*
WIDEn-N	was N flooding with DUPE-CHECKING, but no traceability
TRACEn-N	was N TRACEing with DUPE-CHECKING and with TRACEABILITY.

For Optimum operation, APRS networks need n-N Hopping, Dupe Checking and path Traceing.

To eliminate huge numbers of dupes, the first step is to eliminate RELAY, and WIDE and TRACE paths. Next we needed to get n-N hopping and Traceing at the same time. Since 90% of all users were already using WIDEn-N, and since it is extremely difficult to get tens of thousands of users to change their habits, it was clear that there was only one answer. That is, to move WIDEn-N support over to the Traceable parameter and eliminate TRACEn-N.

Doing it this way, had no impact on the users at all. And only the digipeaters had to change. And the change could be done one-at-a-time with no impact on users. Slowly the system would become one universal standard "WIDEn-N" everywhere and fully Traceable at the same time.

Since we had to change the parameters at ALL the digis to get rid of RELAY, WIDE and TRACE anyway, it was just as easy to implement number 2 above. Once we put WIDEn-N into the fully-traceable

UITRACE function, then there was no need for the "TRACEn-N" any longer. So this then left UIFLOOD unused. So we put the STATE or LOCAL or REGIONAL alias of SSn-N in its place so that the "untraceable" UIFLOOD parameter would still be useful without adding QRM out of the area. In Europe these regions are based on a combination of two iso-standards, ISO 3166-1 country code and ISO 3166-2 subdivision code.

The New-N paradigm in the USA achieved a two to four-fold reduction in dupes and QRM in some regions that had been heavily using the old RELAY and WIDE parameters. This improvement was striking!

NOTE: Admittedly, most European digipeaters evolved after the USA, and so many of them did not have the bad DUPE problem of the old RELAY, WIDE and TRACE because they used more software based TNC's and digipeaters that used the better dupe-checking algorithms that had evolved at that point. But still, there simply is no reason any longer to support all of these original legacy paths. Lets SIMPLIFY APRS! Everything is obsolete except:

WIDEn-N with full dupe-checking and Traceability.

On 20 November 2008, Europe finally decided it was easier to change the digis to Traceable WIDEn-N than to try to convince all the users to change their PATHS to TRACEn-N. At this point, there is no longer any New-EU-Paradigm, as it is now a global New-N Paradigm.

BACKGROUND DATA: To assess the situation a 24 hour snapshot of 94,000 packets was taken in Europe on 17 March 2006 (did not include data from the UK). The data showed the following percentages of those using paths of WIDEn-N or TRACEn-N. It turns out, this was only about half of all paths! The rest were using the obsolete paths of RELAY and WIDE and TRACE! Here were the numbers using WIDEn-N.

WIDE7-7	18%	TRACE7-7	9%
WIDE6-6	2%	TRACE6-6	3%
WIDE5-5	10%	TRACE5-5	4%
WIDE4-4	5%	TRACE4-4	1%
WIDE3-3	36%	TRACE3-3	5%
WIDE2-2	12%	TRACE2-2	1%
WIDE1-1	2%	TRACE1-1	0%
Using both WIDEn-N & TRACEn-N!			5%

The numbers for the UK were similar but they showed a 4:1 preference for using the TRACEn-N system so that they have packet traceability.

The GOOD news: Clearly WIDEn-N was favored *and* about half of the users were being considerate and keeping their QRM down with 3-3 and 2-2 packets. Also this showed that simplifying the network to only use WIDEn-N could be done with little impact to existing users. And simply

swapping the WIDEn-N support over to the UITRACE parameter would allow all WIDEn-N paths to be traceable just like we did in the USA.

The BAD news: The biggest LOAD, however, was caused by the huge numbers of 7-7 and 5-5 packets and packets with BOTH WIDEn-N and TRACEn-N in the same packet! For each of these additional hops, the number of QRM copies can triple! And for only a few 7-7 users, they can cause QRM over much of the continent.

Our final suggestion for the New-EU Paradigm was:

- 1) The best path for fixed stations is WIDEn-N (small N, 2 or less)
- 2) The best path for Mobiles is WIDE1-1,WIDEn-N (Small N, 2 or less)
- 3) Educate users to use routine N no larger than their ALOHA area
- 4) Change the digis (swap TRACEn-N and WIDEn-N) so Wn-N is traceable.

IGATE RULES: IGates are also big contributors to the health of the APRS network. They should set their path to their local area only (usually one hop, WIDE1-1) and should not flood their area with unwanted QRM.

New-N Paradigm DIGIPEATER RULES:

- 1) In saturated areas, TRAP large values of N-N by setting call- substitution aliases of WIDE7-7, 6-6, 5-5, etc. These will digi once and then stop further routing.
- 2) Home stations should **not** be digipeaters UNLESS:
 - a) Station is high and is *-not-* covered by a BIG digi. They should support WIDEn-N, and regional SSn-N.
 - b) *-or-* Station is needed for a fill-in digi for nearby mobiles who cannot be heard by the big digi.
Support ****only**** the WIDE1-1 alias.
- 3) For NETWORK reliability, All digipeaters should:
 - a) Put their PHGxxxx in their position comment
 - b) Put "Wn, SSSSn" in the beacon to show users proper settings
 - c) The SSn shows what regional SSSSn-N they support
 - d) Use the DIGI symbol with "S" overlay to show it is new-N Paradigm.
 - d) *-or-* use the "1" overlay for WIDE1-1 ONLY home digis.
- 4) Move WIDEN-N support to the UITRACE parameter so that all WIDEn-N traffic is traceable.
- 5) Support for WIDE, TRACE or TRACEn-N should be eliminated immediately to force users to reset their settings. Providing an ongoing crutch for a period will only delay the problem.
- 6) With time, WIDEn-N will become fully traceable as DIGI owners move WIDEn-N support over to the UITRACE parameter.
- 7) Digipeaters should not beacon beyond 2 hops. Use the proportional pathing algorithm for DIGI beacons. That is, set for a local (direct, no hops) beacon rate of 10 minutes. If possible, then also a one-hop beacon once every 30 minutes. Then a 2-hop beacon once an hour.

TRACKERS: All future mobile GPS tracker designs should consider the Proportional-Pathing algorithm that sends 1 minute updates local-direct, every 2nd minute via one hop, every 4th minute via 2 hops and optionally every 8th minute via 3 hops. See www.aprs.org/newN/ProportionalPathing.txt

TIPS:

- Be sure all digis transmit their PHGxxxx and "Wn, SSSn" data
- Use the UI-PHG add-on to see the range and coverage of all digipeaters and stations in your area

- Use UI-ALOHA add-on to see the ALOHA range for your UIVIEW station. It analyses traffic heard by your station to compute that area that is a 100% saturated 1200 baud APRS channel. Keep your N hops within that area.

By simplifying the network to only WIDEn-N and WIDE1-1,WIDEn-N guidance and telling users to limit their N's to the minimum needed in their area, then a vast improvement in reliability and throughput can be achieved in the European System.

For more in-depth understanding, please review the details in the web page:
www.aprs.org/fix14439.html

Bob, WB4APR

continue

Advanced Knowledge Base 2

Proportional Pathing and Decayed Beacons

10 NOV 08

WB4APR

10 Nov 08 updated to include trackers
 23 Mar 06 Original draft

Proportional Pathing was introduced in 2006 as an additional improvement on the original APRS decaying algorithm. It was immediately implemented in all digipeaters under the New-N Paradigm and also in the new Kenwood D710 radio. It is recommended for ALL future APRS applications.

With PROPORTIONAL PATHING, packets go frequently locally and direct, and less frequently at more and more hops. For digipeaters, this significantly cut the load on the network. Now digis ID once every 10 minutes local direct, once every 30 minutes via 1 hop, and once an hour via 2 hops. The local 10 minute updates do not QRM with any users, since the digi will not TX if the channel is busy.

For Mobiles, Proportional Pathing provides good local tracking continuity and presence, but overall, much less load on the APRS network. Also, it works for both routine operations and special events without any setting changes by the operator! Operator setting errors is one of the biggest problems with paths on the APRS network. Here is what Proportional Pathing does for a mobile:

- Every minute the packet goes DIRECT path (no digi)
- Every 2nd minute it goes via WIDE1-1 path
- Every 4th minute it goes via WIDE1-1,WIDE2-1
- Every 8th minute it goes via WIDE1-1,WIDE2-2 (only if 3 hops is OK)

The good news is that any existing KPC-3+ Digipeater OR tracker can do this already. See:
www.aprs.org/txt/P-Pathing-kpc3.txt

Proportional Pathing of digis and mobiles drastically reduces channel load and should be included in all new designs and firmware upgrades. The beauty of Proportional pathing is that it does not need to be tweaked by the user as he changes from routine to special event or goes from area to area. Always his packets are higher rate close in and lower QRM at a distance. Trackers can pre-program these settings and never worry again. Of course some simple on/off switch options should be considered. One is the HOPS 2/3 switch, and the other is the RATE 2/3 switch.

The HOP switch determines how far out the longest path goes. In the above example it is set to 2. If it was "3" then every 8th minute a packet would go via WIDE1-1,WIDE2-2 for a total of 3 hops.

The RATE switch determines how rapidly the QRM falls off with distance. In the above examples, the rate is "/2" meaning the rate goes down by two at each additional hop. But if the RATE switch is set

to "/3" then the QRM is further reduced since at each additional HOP tier the rate is 1/3rd the rate. For example:

- Every minute his packet goes DIRECT path (no digi)
- Every 3rd minute it also goes via WIDE1-1 path
- Every 9th minute it also goes via WIDE1-1,WIDE2-1
- Every 27th minute it also goes via WIDE1-1,WIDE2-2

The beauty of Proportional Pathing is simply that other than the on/off switch options above, it does not need precise tweaking for most situations. Yet works at either extreme, local and multi-hop.

OPEN ROAD: The local 1 minute update rate is also ideal on the open road out of range of all digis. With joint-crossing speeds of 140 MPH on the open road, you need a 1 minute rate if you want any chance of hearing another local APRS mobile running Voice Alert.

DECAYED BEACONING: It should be noted that APRS was never designed for fixed rates. Because APRS recognizes the priority that new fresh data should have over old stale data! Too many APRS clones ignored this fundamental principal and operate at fixed rates that are either too HIGH and cause QRM, or are too low and provide little real-time continuity.

The Decay algorithm was fundamental because only new-changed data was transmitted 'now', and that if the data did not change, then the time to the next transmission of that data would double. So, for positions for example, set to a 1 minute rate, if the mobile stopped for any reason, then the next packet would go out 2, then 4, then 8 then 16 and then 30 minutes later. As soon as the position changed (new data) then the rate was reset back to the selected rate (1 minute). Too bad so many APRS clones and follow-on devices simply used fixed rates for simplicity even when the data is not changing. This really added QRM to the network!

The decay algorithm is trivial to implement. Mobiles should simply compare new position data with the last transmitted data, and if unchanged, double the period to the next packet (to a max of 30 mins).

RESULTS:

The result is that for close in work or meeting someone or local events or whatever, the tracker gets good smooth high rate tracking (1 minute for example) and minimum QRM to anyone else. For the local area (each digi he can hit), he gets a comfortable 2 minute rate. For a region and those watching out 2 hops, he gets a 4 minute rate. And for observers a long distance away, he gets an 8 minute rate. And with decayed algorithm, if he stops, he also generates less QRM too. Yet if he operates his tracker at a special event, even without making any changes, he is also seen once a minute direct which is a good rate for an event.

This resolves the tradeoff between too high a rate (QRM)and too low a rate (LATENCY, and LOW TRACK QUALITY)...

PROPORTIONAL PATHING / Decayed Beaconing would cut the QRM from each mobile by a factor of 8 or more while moving, and 30 to 1 when stopped. And since mobiles are the largest load on the network, this would make a HUGE difference to the network while still providing good local tracking.

HUMAN INTERFACE:

The options for setting up a mobile APRS unit (tracker) in my opinion should simply be the selection of the HOP count and RATE divisor as introduced above. The basic rate is always 1 minute (if changing):

HOP SWITCH: 2 or 3

RATE SWITCH: /2 or /3

This is just a continuation of the original APRS decay-algorithm which causes older packets to be transmitted less and less often and new data to be updated quickly. In the case with Proportional Pathing, it also makes NEAR data more important than FAR data as well as NEW data versus old duplicate data.

Also, since the WIDE1-1,WIDEn-N should work everywhere, this one-size-fits-all approach would make setting up a mobile tracker trivial, and help make the network immune to user set-up errors.

This is part of the recommended APRS1.2 Addendum: <http://www.aprs.org/aprs12.html>

Bob, WB4APR

Configure Aliases

31 AUG 12

KJ4ERJ

Aliases are used within paths of APRS so that users don't need to know the specific digipeaters in order to use the installed RF network. The most popular alias is WIDE in the typical incarnations of WIDE1-1 and WIDE2-1 or WIDE2-2. Older (pre-WIDEn-N) aliases are RELAY and WIDE (yep, same alias as we're still using). There are also local state or province aliases referred to as SSn-N, but there is no central clearing house for the values of SS. Finally, there are other aliases that allow a packet originator to request specific behaviors from infrastructure components that may process a packet. Some examples of these are GATE, NOGATE, and RFONLY.

APRSISCE/32 needs to know what aliases it might encounter in order to properly count used hops when making decisions on "local" stations for message gating decisions from APRS-IS to RF. To that end, APRSISCE/32 allows the configuration of aliases which will not be counted when encountered within the used portion of a packet's path. This list of known aliases comes pre-populated with well-known aliases which are visible in the Known cascading menu. Selecting a specific alias will toggle it between active (not counted) and inactive (counted).

Note: Until I determine the performance impact of this, you MUST enable at least one of the Alias() Trace Logs before any of the following will work. Enabling Alias(New) is recommended because it doesn't actually increase any logging, but enables the processing necessary for the following functionality. Also, this will probably only be available in Development mode.

During packet processing, APRSISCE/32 attempts to identify possible new aliases conforming to the SSn-N format (a string of alpha characters followed by a single digit either used (*) or followed by a dash and another single digit). These "learned" aliases are visible in the Possible cascading menu. Selecting a possible alias will move it to the known list after a confirmation.

Finally, there are some digipeaters that seem to not be marking fully-decremented n-N aliases as used. This results in a path containing just SSn which implies a -0. APRSISCE/32 detects these possible SSn-0 aliases when they are the first unused path component and makes them visible in the Unused-0 cascading menu along with the preceding digipeater and platform of that station. This is done primarily for diagnosing the local network configuration and behavior.

To back up this alias processing, APRSISCE/32 supports the following Trace Logs.

Alias(-0) - Shows information on "Unused-0" alias detection and the associated packet. The first appearance of a given alias and digipeater is always logged. Subsequent references are only logged when the trace is enabled.

Alias(New) - Shows all newly discovered "Possible" aliases and the first referencing packet.

Alias(Numeric) - When enabled, shows all n-N path components and associated packets. These are most often an incorrectly entered path with extra commas (WIDE,1-1,WIDE,2-1) or missing aliases (WIDE1-1,2-1).

Alias(Used) - When enabled, shows all packets referencing any Possible alias, regardless of New status.

Lynn, KJ4ERJ

Companions

TBD

DX

DX in APRSISCE/32 is defined as receiving a position packet with no hops used from a station located further away than some minimum distance. The current furthest DX may be included in the periodic Status Report packet and also returned in response to a ?DX directed query. Additionally, newly set DX stations beyond a configurable distance may be immediately reported via a DX cluster formatted packet and may also cause an immediate Status Report to be issued.

The following menu options govern the DX behaviors of APRSISCE/32.

Min Dist specifies the minimum distance required for a packet to be considered "DX". The parameter is specified in miles or kilometers depending on the **Configure** ⇒ **Metric** ⇒ **Distance** setting.

Min Trigger specifies the minimum distance that may trigger an immediate transmission of either a DX cluster or Status Report packet provided that no such packet has been triggered within the past Min Interval minutes, or the new DX is further than the previous DX and from a different station.

Min Interval specifies the minimum time in minutes that must elapse before an additional DX cluster packet can be triggered for the current DX station even if said station is getting further away.

Window specifies the time in minutes before the furthest DX station and distance is forgotten opening the way for another station to become DX at a closer distance. Expiration of a DX station at the end of the Window time does not trigger any transmissions, but the next Status Report will report a different DX status.

Max Ever shows and optionally clears the furthest DX station ever recorded by APRSISCE/32.

Ignore provides visibility to an optional list of stations that will not be considered for DX, regardless of distance. Stations are added to this list via their station popup menu provided that they have been recently received directly over an RF port. Individual or all exception stations can be cleared by selecting from this menu.

To be considered for DX, a received packet must meet the following conditions:

- *Position-containing Packet*
- *Received via RF*
- *Not a Third-Party Packet*
- *No Hops Used*
- *Not an Object*
- *Not At lat/lon 0,0*
- *Not on the Ignore List*
- *Further than Min Dist*

Once these conditions have been met, the packet is inserted into a distance-ordered list by time. Any older/closer packets are dropped from the list and the new distance is this distance. In the event of a tie on distance, the new station ID replaces the older one.

If a new furthest station is further than Min Trigger, then an immediate notification may occur in the following conditions:

- *New station further than original*
- *Longer than Min Interval since last trigger*

In the event of a distance tie, or a station that is now further than before, a trigger will only happen if Min Interval minutes have elapsed since the last one. This serves as a safety to prevent a notification per packet for a steadily retreating station.

Notifications, when triggered, may be a DX cluster format packet via any RF port configured to transmit said packets or a Status Report if it is configured to include DX information. DX transitions suppressed by the Min Interval, Window expirations, or less than Min Trigger distance will be indicated in the next interval-based Status Report or may be issued in response to a directed ?DX query.

Map

Selection

Prefetch

(Note: this page may be a bit redundant as it is composed of three different e-mail interactions. It'll get cleaned up, merged, and maybe split out eventually.)

What's with this new feature of "Prefetching" OSM Map Tiles? Why would I want to clutter up my hard drive with such stuff? Why not just let them download when they're needed?

Well, with the new KISS and AGW interfaces, APRSIS32 has moved beyond the assumption of Internet connectivity. In an RF-only APRS environment, it would still be nice to see the OSM map tiles; hence the prefetch option.

To use it, simply zoom in or out and pan around until your map view shows the area whose maps you want to prefetch. Then simply select Screen/Map/Prefetch.

APRSIS32 will calculate how many tiles will be necessary to have the current zoom and two additional zoom levels at the current size of your window. When you accept this confirmation, ALL of those tiles are queued for fetching and you get to watch an expanding circle moving slower than you've probably seen it move to date. Don't be alarmed if the number of tiles queued doesn't match the original estimate. The second number takes into account tiles that were already resident in your tile cache.

After prefetching the tiles for a fairly large area, I zoom down a bit and continue prefetching more details and panning around fetching more and more details. It is, however, important to wait for each prefetch to complete before dragging the map around. APRSIS32 is smart enough to think that you've moved the map, so you must not need the stuff that was queued, so it will flush the queue and start over at the new location. Probably not what you want.

When you DO go mobile without an Internet connection, just toggle off Enables/OSM Fetch Enabled and APRSIS32 will not attempt to load any additional tiles. If you zoom in further than you did while prefetching, APRSIS32 will simply stretch the closest available tile resulting in a grainier looking map than you would otherwise have, but at least you have a map!

Again, let me know if any of you are using this feature and how I can make it better (other than making the prefetch-queued tiles not flush when you pan around to queue more). BTW, double-clicking an empty space on the map will tell you how many tiles covering how much space you currently have on your machine as well as how long it has been since they've been referenced (since creation currently on Windows Mobile). The XML configuration file has a parameter called OSM.RetainDays, but you really don't want to mess with this. Instead, see Configure / Map / Tile Sets / Original (or your own custom tile set) and uncheck the Purge Enabled box. You can also change the purger days from this same configuration dialog.

Alternatively:

There's a menu option under Screen / Map / Prefetch that will handle the prefetching of the maps. Then you need to configure the map tile set to make sure the tile purger doesn't do away with them. Here's the procedure:

Only do the following when you have an Internet connection and time to wait for the prefetcher to pull in the map tiles! We're talking thousands of them!

- 1) Zoom and pan the map so that an area of interest is all that is visible. On Win32 you can even size the window to help focus on the interested area.
- 2) Select Screen / Map / Prefetch. The client will tell you how many tiles cover the area including several zoom levels closer. Don't panic, this will be a large number!
- 3) Once you click Yes, the client will queue all of the missing map tiles. This can take a while (even just to queue them) so be patient. Once all have been queued, the

actual count will be displayed. This may be less than the original count because some of the tiles have probably already been cached.

4) The yellow circle will begin expanding as the tiles are fetched, but you don't need to wait for it. Double-clicking an empty spot on the map will provide you some visibility into the queue length and the states of the threads doing the retrieval.

5) You can now repeat steps 1 through 4 at various zoom levels to make sure you have all of the tiles you want. If you miss some, the client will simply stretch the next higher zoom level to cover the hole if/when you get into one.

You can still pre-fetch additional tiles whenever you have a connection. However, when you're going to be connectionless, I would recommend unchecking the **Enables ⇨ OSM Fetch** Enabled. This will prevent the client from even attempting a connection to the OSM server for any missing tiles in the event it needs one while you know you'll be out of coverage. It also turns off the tile purger, but only until you enable the fetcher again, so if you're really going Internet-less, you probably want to manually disable the purger by unchecking the Purge Enabled box in your current tile set. See **Configure ⇨ Map ⇨ Tile Sets ⇨ Original** (or your own custom tile set) and uncheck the Purge Enabled box. You can also change the purger days from this same configuration dialog.

Question: Is there a reason to limit the prefetch depth? Why not allow the user the ability to download to any depth?

I limit depth because of the 4x factor that increasing zoom levels represents. When I say that I pre-fetch the next lower N zoom levels, it means the ENTIRE area expanded by 4x for each zoom level closer. Many times, this is not what the user really wants to see in detail. For instance, why pre-fetch lots of ocean tiles just because the lower left corner of the United States happens to be a peninsula?

I would entertain going deeper if the prefetch constrained itself to the same area that would be viewed by zooming in at the current screen size, but again, that's probably not what most users are looking for. Once you get in to a certain area, they really want the 4x expansion as it goes deeper.

My own manual method is to get a reasonable area at a reasonable zoom and trigger the prefetch. I then zoom in 2 or 3 times and pan around triggering more prefetches until I think I've got enough queued. Then I zoom in to a reasonable street level and pan around slowly while watching for the fuzziness to disappear. At my origination and destination points, I zoom in really deep to have the final detail and/or back off 1 or 2 levels and trigger a final few prefetches.

The client does do some automatic prefetching of one zoom level further out and the 4 tiles one level closer (along with one additional tile at the current zoom level out each edge of the screen) each time a new tile is fetched from the server. This is in anticipation of traveling into those areas or doing a small zoom for less or more detail.

Note that the automatic prefetchs of map tiles due to panning and zooming are canceled if your attention has moved elsewhere (by more panning and zooming) before the tiles were fetched. Manual Prefetch requests are not canceled nor cancellable short of closing the client.

Question KC9GQR: When prefetching tiles, is there a way to change how many zoom levels is prefetched? I think its 5 but can you change it to like 15 in the xml file?

See my response to N7FMH to see why this is not necessarily a good idea. It has to do with what area coverage is desired for the increasing zoom levels, all of the area originally covered or just the center as if you had manually zoomed the whole way in at a constant screen center. The client currently does the former on the prefetch which results in a 4x tile count for each zoom closer. The latter does a constant tile count per level, but doesn't end up fetching detail for ALL of the area originally viewed.

See the table below (source Wiki) to get an idea of the total tile count for various zooms.

The zoom parameter is an integer between 0 (zoomed out) and 18 (zoomed in). 18 is normally the maximum, but some tile servers might go beyond that.

0	1 tile covers whole world	1 tile
1	2 × 2 tiles	4 tiles
2	4 × 4 tiles	16 tiles
n	2n × 2n tiles	22n tiles
12	4096 x 4096 tiles	16.777.216
16	Maximum zoom for OpenCycleMap (mostly)	232 = 4.294.967.296 tiles
17		17.179.869.184 tiles
18	Maximum zoom for Mapnik layer	68.719.476.736 tiles

For your suggestion of 15, if you did this while zoomed out to the planet view, you'd end up fetching (left column is zoom tile count, right column is cumulative total).

1	1	
2	4	5
3	16	21
4	64	85
5	256	341
6	1024	1365
7	4096	5461
8	16384	21845
9	65536	87381
10	262144	349525
11	1048576	1398101
12	4194304	5592405
13	16777216	22369621

14	67108864	89478485
15	268435456	357913941
16	1073741824	1431655765
17	4294967296	5726623061
18	17179869184	22906492245

Oh, and you'd have to multiply all of those numbers by the number of tiles it takes to fill the screen from which you did the Prefetch.

Purger Enabled

Purger-Enables or disables the tile purger. Eventually the maximum age of cached tiles will be configurable as well.

Tile Sets

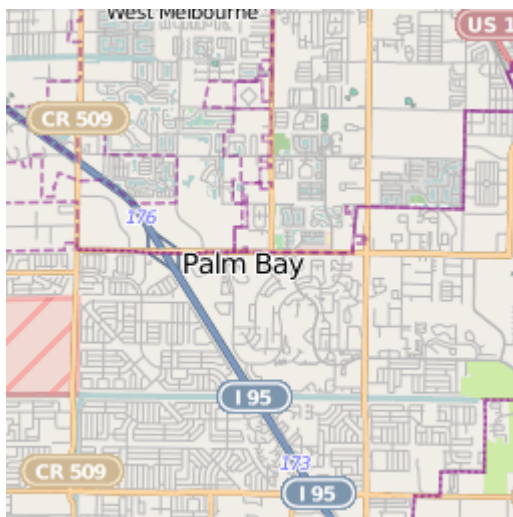
APRSIS32 uses the OpenStreetMap tile set as a default background map. These map tiles are part of an Open Source mapping project which has the intention of creating a freely distributable non-encumbered map of the world for use in all kinds of projects, such as APRSIS32. Most of the map sources normally encountered have copyright or other use restrictions.

Map Tiles

The OSM maps are delivered as 256 X 256 pixel tiles that APRSIS32 downloads from the server, and keeps in a cache on the local storage medium. These tiles are drawn on screen to produce the maps that are used to plot stations on. The tiles are purged from the cache after a preset time to keep the local storage from being filled. It is possible to change the purge time to have the tiles available for as long as desired, including disabling the purge time to allow the tiles to remain forever.

Default Server

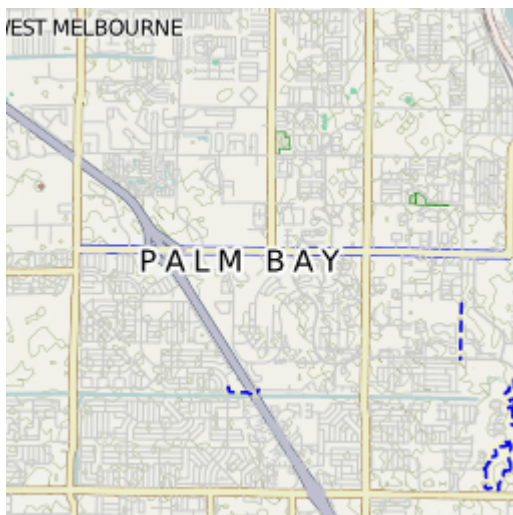
The default tile set "Original" is the Mapnik style used as the default at the OpenStreetMap website. There are a number of different tilesets available for use with APRSIS32. They all use the same map database, but use different rendering styles which may highlight different features, or just use different colour sets that may be more desirable to the end user.



Original (Mapnik)



AeroMap.us



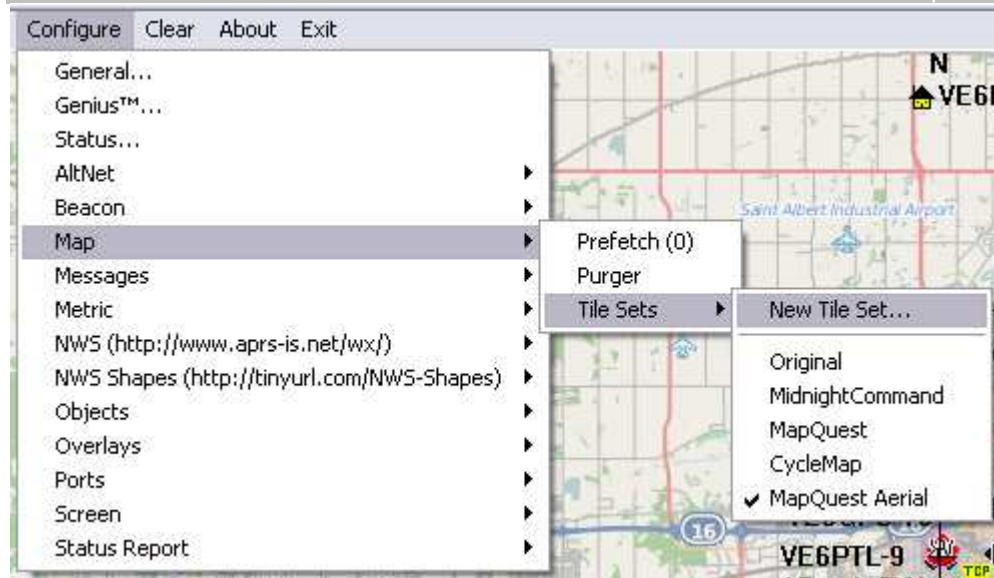
CycleMap



MapQuest

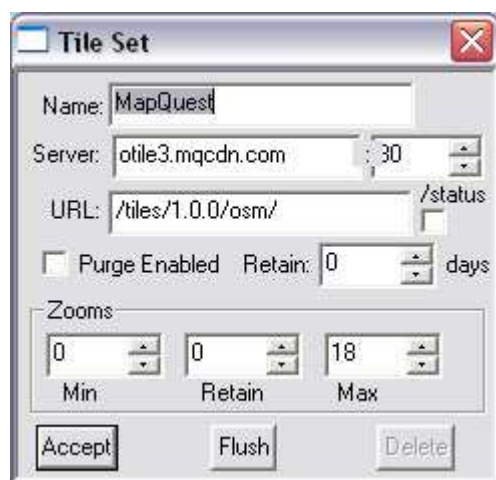
Configure Tile Sets

To configure APRSIS32 to use alternate tile sets, you will need to select **Configure**
⇒ **Map** ⇒ **Tile Sets** ⇒ **New Tile Set**



A configuration dialog box will open up where you can enter the information required to set up your new tile set.

Note: Whatever Tile Set configuration you last Accepted becomes the default tile set for all new windows and across APRSIS32 restarts. The default tile set is indicated by the checkmark in the **Configure** ⇒ **Map** ⇒ **Tile Sets** menu (MapQuest Aerial in the screen-shot above).



Name is the name you want to use for the tile set.

Server is the base URL for the server that will be providing the map tiles.

80 is the port number used to access the tiles.

URL is the directory qualifier to point to the location of the tiles on the server

status check here if the server supports tile change status updates

Purge Enabled check here to have tiles purged from your file storage medium

Retain sets the number of days to keep tiles before being deleted

Zooms Min Retain Max configures the minimum and maximum zoom levels to be retained even if the global tile purger is configured to delete tiles.

Available tile servers

Name	Server	URL
CycleMap	b.tile.opencyclemap.org	/cycle/
MapQuest	otile3.mqcdn.com	/tiles/1.0.0/osm/
FreeMap	tilesrv.sucs.org	/ofm/
MassTopo (Only MA)	tile2.toposm.com	/ma/final/
MapQuest Aerial	otile3.mqcdn.com	/naip/
MidnightCommand	c.tile.cloudmade.com	/BC9A493B41014CAABB98F0471D759707/999/256/
Red Alert	a.tile.cloudmade.com	/BC9A493B41014CAABB98F0471D759707/8/256/

Be aware that some of the above servers only support limited geographic areas, and may not cover your area, or may only support limited zoom levels.

Above are some tile sets that have been predetermined. Rather than include every tile set that exist, some education is reasonable to expect from users that wish to use an alternate set. By closely examining the examples below, many more tile sets may be found and utilized.

Now, there are more tile servers available that can be used. Using the original tile paradigm, they just need to fit the pattern that is used by the tile fetching algorithm where the URL used to fetch the tiles is as such:

b.tile.opencyclemap.org/cycle/\$z/\$x/\$y.png

Where b.tile.opencyclemap.org is the server, /cycle/ is the "URL", and the desired tile is designated by elements; z, x and y.

zoom (\$z)

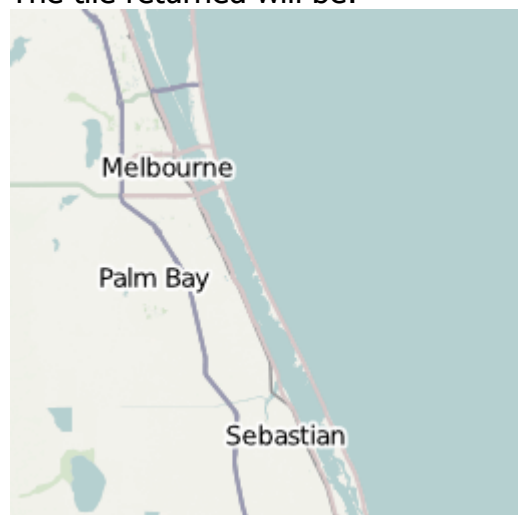
horizontal (\$x)

vertical (\$y)

Entering elements, z = 9, x = 141 and y = 214, will produce a Server/URL:

<http://b.tile.opencyclemap.org/cycle/9/141/214.png>

The tile returned will be:



New Tile Paradigm

The new tile paradigm has added more options. The original format still works. However, if the server has the requirement of re-ordered tile elements, z, y and x, or y, x and x, or an inverted y element is necessary, then more information needs to be added to the URL.

Specifically, the tile elements z, x and y need to be added with the correct formatting. The proper format is %z, %x and %y (or %!y for inverted y). Note: The server may need to be in a specific order. The tile extension for anything other than PNG will be needed as well. Current extension options are, JPG and PNG.

An example of a server that requires re-ordered tile elements is: Server: aeromaps.us

The URL will demonstrate the new tile paradigm: URL: /Z%z/%y/%x.png

An example link: <http://aeromaps.us/Z9/214/141.png>

If the server returned tiles in the same manner as the OSM servers, everything after the /Z in the URL would not be needed. Since x and y are re-ordered (y=214 and x=141), all elements will need to be added.

An example link: <http://aeromaps.us/Z9/214/141.png>

Concatenating the server and URL will produce: aeromaps.us/Z%z/%y/%x.png

The final result for z=9, x=141 and y=214:



For more information on the aeromaps.us server, please check:

<http://aeromaps.us/Home.html>

An example of a server that requires an inverted y is:

Server: earthncseamless.s3.amazonaws.com

The URL will demonstrate the use of an inverted-y element:

URL: /%z/%x/%!y.png

Inverted-Y = (2z-1)-y

If we want a tile from z=9, x=141 and y=214, then y must be inverted. Y becomes 297. The link to the tile becomes:

<http://earthncseamless.s3.amazonaws.com/9/141/297.png>

The tile result is:



For more information on the earthnc.com server, please check:

<http://earthnc.com/online-nautical-charts>

The aeronautical and nautical tiles above were found by experimenting with URLs. During general surfing, new maps may be found. By examining the "behind the scenes" data, sometimes a useful link to a tile can be found. One such method for exploring behind the scenes data: Firefox: Toolbar > tools > Page Info > Media.

Suppose while examining some "behind the scenes" data, a link like this is found:

server.arcgisonline.com/ArcGIS/rest/services/World_Topo_Map/MapServer/tile/15/12104/9922.jpg

How would this be used? Some experimentation is in order. First, a simple copy and paste into a browser address bar will produce this:



Determining which element is z, x and y has to be done. Z is pretty easy. The X and Y are not always easy. Figuring this out tends to be bit trial and error. The simplest method is to navigate to the same location with APRSIS32 and check the current tile in the center by double clicking. By comparing the elements in the link above and the return in APRSIS32, it is simple to determine what each element is. In the example above, x and y are simply re-ordered.

The Server and URL still must be determined. The server starts at the beginning and continues up to the first stroke [/]. In this

example: Server: server.arcgisonline.com

The URL is the rest of the line, starting with the stroke. Each element will be replaced by its placeholder and in the same order as in the link. The result becomes: URL: /ArcGIS/rest/services/World_Topo_Map/MapServer/tile/%z/%y/%x.jpg

Notice the .jpg at the end of the URL. The tiles from the ArcGIS server are not png as required in the past. This must be included at the end of the URL.

In the event two elements match but the third is "wrong". It is probably using the Invert-Y. As in the nautical tile example above, use the %!y and try it.

Looking around at the various servers may be enlightening. X and y may be depicted as i and j or maybe as "x=" and "y=". A little experimenting will often result in a new tile. At minimum, a bit of an education may be acquired.

One user has used 25-30 different sets, and the count continues, each with varying degrees of usefulness. Just how useful is a set of 1812 World Map Tiles to APRS anyway? Yet, they are available for an explorer.

GoogleMaps / YahooMaps

While GoogleMaps, YahooMaps and a number of other online mapping services serve up map tiles in a manner similar to OpenStreetMap, they have restrictive licenses that do not allow unencumbered access to the map tiles, as well as offline tile caching and usage restrictions. Due to these limitations, APRSIS32 does not attempt to make use of the tiles from these sources.

Messages

This menu drop down is used to configure what messages are received.

My Messages

Selecting this option will enable reception and display of any message addressed to your callsign regardless of the SSID the message is sent to.

RF Eavesdrop

This option enables the display of any message received over RF regardless of who the message is addressed to.

Eavesdrop

Enables the reception and display of any message that the software sees, depends very much on your APRS server filter settings. NOT recommended if you are receiving a full feed

But Not NWS

Eavesdrop but not National Weather Service.

Hide Queries

This option hides any queries someone makes to your station such as ?ARPSH etc. Otherwise queries will show in the message box.

Notify Query

When a query arrives at your station a popup balloon will appear to inform you of the query, otherwise the query will just show under messages. In any case, the query will be automatically answered.

Notify New

When a new message arrives a popup balloon will appear to inform you, handy if you have the software minimized, but somewhat bothersome when Windows queues notifications on an unattended machine.

MultiTrack™ New Items

When an object arrives as an item in a message, each object will show in a new multitrack window, if not selected the objects will only show on the main map.

Lookup(WHO-IS)

This option allow selection of the amount of data received when a WHO-IS search is done, if checked WHOIS will hopefully respond with the address related to the callsign looked up. Positive result only if station has posted its details in QRZ.com. Selection is:

Brief (1Line)
Full Address

Metric

This configure option selects which measurements you would like to display as a metric rather than imperial value. Defaults are imperial.

Altitude - Selects between displaying altitude in feet ft (unchecked) or metres m (checked)

Distance- Selects between displaying distance in statute miles sm (unchecked) or kilometres km (checked)

Pressure - Selects between displaying atmospheric pressure in inches inch (unchecked) or milliBars mb (checked)

Rainfall

Selects between displaying inches in feet (unchecked) or millimetres mm (checked)

Temperature - Selects between displaying temperature in Fahrenheit F (unchecked) or Celsius C (checked)

Wind (Knots) - Selects between displaying wind speed in mph / kmh (unchecked) or knots (checked) [-Unchecked km/h when Distance checked-].

Mic-E Notifications

Selection

Suppress Notifications

Suppress EMERGENCY Too

EMERGENCY! Msg New Active Flash Red

Special Msg Yellow
 Priority Msg New Flash Orange
 Alarm Msg New Flash Orange
 Alert Msg Yellow
 Warning Msg New Orange
 Wx Alarm Msg New Flash Orange
 TEST Alarm Msg New Active Flash Crimson
 unknown DarkKhaki

Advanced Knowledge Base

EMERGENCY CODES for APRS

13 Feb 07

The Mic-E format used by many devices including the Kenwood TH-D7 and TM-D700 radios have the provision for sending 3 status bits with these meanings in each position report:

EMERGENCY
 PRIORITY
 SPECIAL
 COMMITTED
 RETURNING
 INSERVICE
 ENROUTE
 OFF-DUTY

In the original APRS, these bits were to trigger these on-receipt-responses:

SPECIAL - would CENTER the map on the symbol
 PRIORITY - would CENTER and ZOOM to 8 mile local range scale
 EMERGENCY - CENTER and ZOOM and ALARM beeps until cleared by operator.

And also the color attribute for the SYMBOL for these stations would be shown in these colors:

SPECIAL - Yellow
 PRIORITY - Orange
 EMERGENCY - Red

There has long been a proposal that other software should be able to send these same status bits. But, since the bits are specially encoded inside the unique Mic-E format, they are not available to other applications that do not use the Mic-E format.

Hence, it has been proposed that the following special strings can be interpreted like the original Mic-E bits. They are bracketed by exclamation points (!) to set them out from normal text and they are expected to be the first bytes of the free-field comment field of any station. The first bytes of the comment field of any POSITION packet or OBJECT packet are located in the same place and if PHG data is in the same packet, then these special designators follow the PHG data.

!TESTALARM!
 !EMERGENCY!
 !PRIORITY!
 !SPECIAL!
 !COMMITTED!
 !RETURNING!
 !INSERVICE!
 !ENROUTE!
 !OFF-DUTY!

Notice that an additional !TESTALARM! function has been added that is supposed to trigger the same circuits and algorithms on receipt as a real !EMERGENCY! field, except that it is CLEARLY supposed to convey it is a TEST.

An Additional Proposal (has already been implemented in APRS+SA and XASTIR), is the recognition of these additional CENTER and ZOOM triggers (but without the surrounding "!" bytes:

```
!ALARM!
!ALERT!
!WARNING!
!WXALARM!
!EM!
```

And I would suggest that we add an additional level of on-receipt-responses is called FLASH for these new not-quite emergencies. This function is one step under the EMERGENCY BEEPING, and adds an unmistakable SILENT FLASH of "something" that an attendant operator cannot miss.

EXAMPLES:

```
!DDMM.hhN/DDDMM.hhW$!PRIORITY! Special Event in progress
!DDMM.hhN/DDDMM.hhW$PHG3530!PRIORITY! Special Event in progress

;OBJECTNAME*DDHHMMzDDMM.hhN/DDDMM.hhW$!PRIORITY! Special Event...
;OBJECTNAME*DDHHMMzDDMM.hhN/DDDMM.hhW$PHG3530!PRIORITY! Special...
```

This proposal is fully backwards compatible with all existing systems in that the free-field bytes are used. Hence they will at least display on all known existent systems.

However, since this is an APRS1.2 Addendum Proposal, there is no expectation of any functional backward compatibility or guaranteed response to these special coded fields in any prior software.

Hence use of this proposed feature should only be expected on pre-tested end-to-end future systems.

Bob, WB4APR

NWS (<http://www.aprs-is.net/wx/>)

The US National Weather Service (NWS) puts out information that can be displayed on your APRSIS32 display. This information includes Severe weather warnings. First thing you need to know, is which NWS office you want to display information from.

[Forecast Offices](#)

Is one resource website for finding the office abbreviation. The other thing that you will need to do is Download the NWS shape files from [US NWS](#) or for Australian offices [Here](#) There are a lot of them, and it depends on what you want to see for which ones you download. (Do Not use files dated in the future)

Configure ⇒ NWS Shapefiles

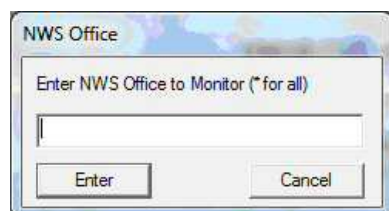
Once you have the files downloaded, you have to point the program to them.



Add each individual file, and enable them. Set your display settings.

Configure ⇒ **NWS** ⇒

New Office ⇒ **Enter NWS Office to Monitor**



In this box, you enter the 3 letter abbreviation for the office you want to receive alerts from [Forecast Offices](#)

If you enter "*" into this box, you will get alerts from ALL NWS offices. This is a lot of information, especially during storm seasons. Make sure that your PC and internet connection will support this much information.

Messages ⇒ This is a check box for receiving messages containing NWS information. If you want to see what creates the shapefiles then check this, if you can't read programming language, then you might want to leave this unchecked.

Message NOT all ⇒ This is for messages that are not directed to ALL stations

Messages Notify ⇒ Do you want to be notified of these messages

Notify Line Type ⇒ What line types do you want to see on your screen.

Notify LineType	▶	✓	? Undefined LineType (0/0 Prods)
Notify Products	▶	✓	a Red solid Tornado Warning (12/12 Prods)
Popup MultiTrack™	▶	✓	b Red dashed Tornado Watch (4/4 Prods)
Show Offices	▶	✓	c Red double-dashed (7/7 Prods)
NWS Entry Servers	▶	✓	d Yellow solid Severe Thunderstorm Warning (15/15 Prods)
✓ *ALL*		✓	e Yellow dashed Severe Thunderstorm Watch (8/8 Prods)
✓ BGM		✓	f Yellow double-dashed (8/8 Prods)
BTV		✓	g Blue solid Test Warning (1/1 Prods)
BUF		✓	h Blue dashed Test Watch (3/3 Prods)
CTP		✓	i Blue double-dashed (9/9 Prods)
✓ ILM		✓	j green solid (0/0 Prods)
LNK		✓	k green dashed Mesoscale Discussion Areas (0/0 Prods)
OAX		✓	l green double-dashed (8/8 Prods)

Default is all checked.

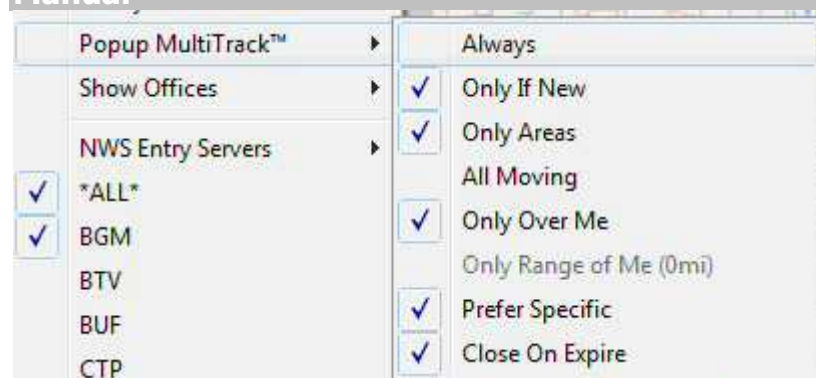
Notify Products ⇒ This determines which weather Icons you want to be notified of

Notify Products	▶	None (75)	
Popup MultiTrack™	▶	a Tornado Warning	▶
Show Offices	▶	b Tornado Watch	▶
NWS Entry Servers	▶	c red double-dashed	▶
✓ *ALL*		d Severe Thunderstorm Warning	▶
✓ BGM		e Severe Thunderstorm Watch	▶
BTV		f yellow double-dashed	▶
BUF		g Test Warning	▶
CTP		h Test Watch	▶
✓ ILM		i blue double-dashed	▶
LNK		l green double-dashed	▶
OAX			
OKX			

CFW Coastal Flood Warning (d)
DSW Dust Storm Warning (d)
GLE Gale Warning (d)
HMW Hazardous Materials Warning (d)
HWW High Wind Warning (d)
ICE Ice Warning (d)
LEW Law Enforcement Warning (d)
SMW Special Marine Warning (d)
SNW Snow Warning (d)
SQL Squall Warning (d)
STS Severe Thunderstorm (d)
SVR Severe Thunderstorm Warning (d)
TRW Tropical Storm Warning (d)
TSW Tsunami Warning (d)
WSW Winter Storm Warning (d)

Default is all enabled.

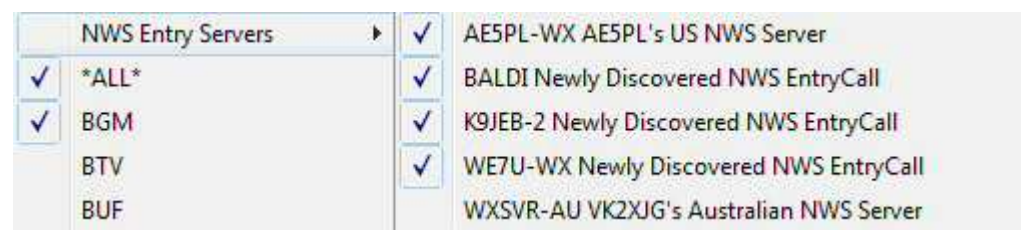
Notify Multitrack ⇒ Under what circumstances do you want a multitrack window to pop up



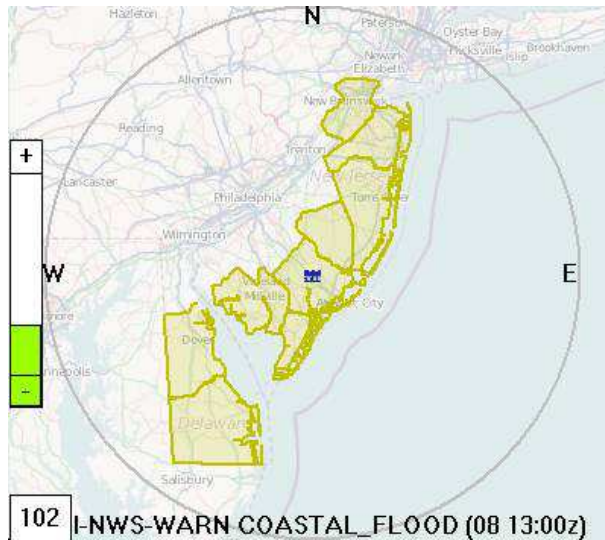
Multitracks are memory intensive, be sure that you want to see what you check. Strongly suggested that the windows close when the warning they display expires. **Show Offices** ⇒ This will highlight the office coverage area on your main display. If you have more than one office selected, it will overlap.



NWS Entry Servers ⇒ what servers do you want to receive data from



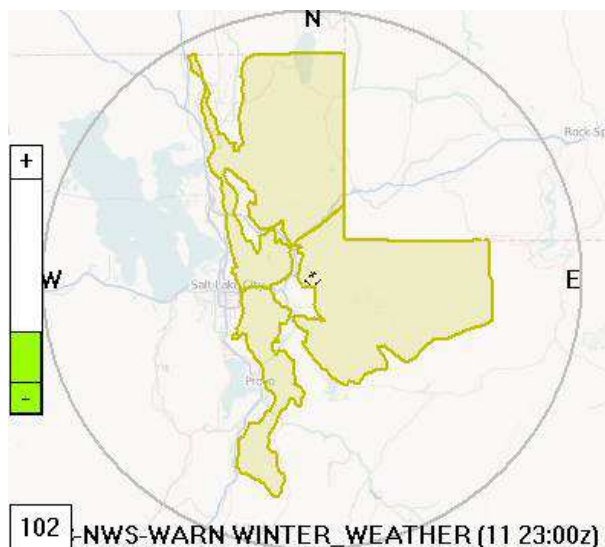
This will auto check servers. Notice the Australian server. If you don't care about the weather, make sure you uncheck it. Under that will be a list of configured NWS offices, the checked ones are the ones you will display information for. Again, make sure that your PC and Internet connection can support the data that you're requesting. If the program locks up shortly after configuring NWS offices, you should think about reducing the number of offices you display. Example NWS alerts:



PHICFWaq3 via WE7U-WX

NWS-WARN COASTAL_FLOOD
Coastal Flood Warning
Multi: 936/62048 Points 442 Parts
Source: DEZ2>4-NJZ12>14-20>27
Issued: 06 19:52z
Expire: 08 13:00z (1d 15h44m)
Duration: 1d 17h08m

Sym: Flooding (\w)
App: Generic APRS
FromME: 866mi NNE (21°)
GridSquare: FM29pm75ee
3931.2998N 07441.4129W
2012-11-06 19:52:33 (1h23m)
PHICFW->



SLCWSWaqj via WE7U-WX

NWS-WARN WINTER_WEATHER
Winter Storm Warning
Multi: 410/1246 Points 3 Parts
Source: UTZ7>9
Issued: 06 20:49z
Expire: 11 23:00z (5d 1h43m)
Duration: 5d 2h11m

Sym: Snow Shwr (\G)
App: Generic APRS
FromME: 1943mi NW (305°)
GridSquare: DN40is68rs
4047.1884N 11116.6343W
2012-11-06 20:49:52 (26m47s)
SLCWSW->

Adam, KC2ANT

New Office...

see above

Messages

see above

Messages Not *ALL*

see above

Messages Notify

see above

Notify Line Type

Selection

- ? Undefined LineType
- a Red solid Tornado Warning
- b Red dashed Tornado Watch
- c Red double-dashed
- d Yellow solid Severe Thunderstorm Warning
- e Yellow dashed Severe Thunderstorm Watch
- f Yellow double-dashed
- g Blue solid Test Warning
- h Blue dashed Test Watch
- i Blue double-dashed
- j Green solid
- k Green dashed Mesoscale Discussion Areas
- l Green double-dashed

Notify Products

Selection & Sub-Selection

None

- a Tornado Warning
 - AVW Avalanche Warning (a)
 - BZW Blizzard Warning (a)
 - EQW Earthquake Warning (a)
 - FFW Flash Flood Warning (a)
 - FLW Flood Warning (a)
 - FRW Fire Warning (a)
 - HUW Hurricane Warning (a)
 - NUW Nuclear Power Plant Warning (a)
 - RHW Radiological Hazard Warning (a)
 - TCY Cyclone Warning (a)
 - TOR Tornado Warning (a)
 - VOW Volcano Warning (a)
- b Tornado Watch
 - AVA Avalanche Watch (b)
 - FFA Flash Flood Watch (b)
 - HUA Hurricane Watch (b)
 - TOA Tornado Watch (b)
- c Red double-dashed
 - BFA Bushfire Alert (c)
 - CDW Civil Danger Warning (c)
 - EAN Emergency Action Notification (c)
 - EVI Evacuation Immediate (c)
 - FFS Flash Flood Statement (c)
 - HLS Hurricane Statement (c)
 - LAE Local Area Emergency (c)
- d Severe Thunderstorm Warning

- CFW Coastal Flood Warning (d)
- DSW Dust Storm Warning (d)
- GLE Gale Warning (d)
- HMW Hazardous Materials Warning (d)
- HWW High Wind Warning (d)
- ICE Ice Warning (d)
- LEW Law Enforcement Warning (d)
- SMW Special Marine Warning (d)
- SNW Snow Warning (d)
- SQL Squall Warning (d)
- STS Severe Thunderstorm (d)
- SVR Severe Thunderstorm Warning (d)
- TRW Tropical Storm Warning (d)
- TSW Tsunami Warning (d)
- WSW Winter Storm (d)
- e Severe Thunderstorm Watch
 - CFA Coastal Flood Watch (e)
 - FLA Flood Watch (e)
 - HWA High Wind Watch (e)
 - SVA Severe Thunderstorm Watch (e)
 - SWW Strong Wind (e)
 - TRA Tropical Storm Watch (e)
 - TSA Tsunami Watch (e)
 - WSA Winter Storm Watch (e)
- f Yellow double-dashed
 - FLS Flood Statement (f)
 - FOG Fog Advisory (f)
 - FST Frost Advisory (f)
 - FWW Fire Weather (f)
 - RWA Road Weather Alert (f)
 - SPS Special Weather Statement (f)
 - SPW Shelter in Place Warning (f)
 - SVS Severe Weather Statement (f)
- g Test Warning
 - DMO Practice/Demo Warning (g)
- h Test Watch
 - NPT National Periodic Test (h)
 - RMT Required Monthly Test (h)
 - RWT Required Weekly Test (h)
- i Blue double-dashed
 - BWA Bushwalking Advice (i)
 - CAE Child Abduction Emergency (i)
 - CEM Civil Emergency Message (i)
 - EAT Emergency Action Termination (i)
 - FBN Fire Ban (i)
 - NIC National Information Center (i)
 - NMN Network Message Notification (i)
 - SGW Sheep Weather (i)

TOE	911 Telephone Outage Emergency (i)
I	Green double-dashed
???	Undefined NWS Product ID (I)
ADR	Administrative Message (I)
ZZZ	Internal APRSIS32 Test (I)

Popup MultiTrack™

Selection

Always

Only If New

Only Areas

All Moving

Only Over Me

Only Range of Me (1609km) (if metric)

Prefer Specific

Close on Expire

Show Offices

Selection

Show Coverage

Show Individuals

NWS Entry Servers

Selection

AE5PL-WX AE5PL's US NWS Server

WXSVR-AU VK2XJG's Australian NWS Server

NWS Shapes (<http://tinyurl.com/NWS-Shapes>)

What can I actually do with ShapeFiles and what should I see?

First, you need to follow the instructions at WA8LMF's excellent page at http://wa8lmf.net/aprs/get_nws_shapefiles.htm . He has links to the pages with the .ZIP files for the US NWS ShapeFiles, but you may also want the Australian ones from Australian Shapefiles <http://wxsvr.aprs.net.au/shapefiles.html> Download the .ZIP files and unzip them all into a single new directory where ever you want to put it. Note: it is *MUCH* easier if you just dump them all into a *single* directory! (Links to the currently valid NWS shape files are also maintained at <http://www.aprs-is.net/wx>)

Once that is done, you need to tell APRSIS32 which file(s) you want it to use. Be careful with the US NWS ones as they've got pending revisions on two of them that aren't actually valid yet. Pick the ones whose date has past, not the ones whose date is in the future.

Use **Configure** ⇒ **Shapefiles** ⇒ **Add File** and pick each of the .SHP files that came out of the .ZIPs. The corresponding, same-named .SHX and .DBF files need to be in the same directory for this all to work. As each file is added, the Open File dialog re-appears to add additional file(s) as needed. When all desired files have been added, hit Cancel.

Once you have the shapefiles enabled, you'll probably want to enable receiving alerts from your local NWS office (or CWA - County Warning Area). You can get your county's code at NWS County Codes <http://www.aprs-is.net/WX/NWSCodes.aspx> just pick your state from the drop-down and scan for your county. Or you can center the map at the point of interest and hit **Configure / NWS Shapes / ShowMe**. APRSIS32 will pop up a MultiTrack window for each zone that the point is within. Some of these zones will start with a 3 character code that is the NWS office (CWA) responsible for that area. Once you have your CWA, use **Configure / NWS / New Office...** to enable it in APRSIS32. You can add as many offices as you need to cover your area of interest.

After that, you wait....and wait...and wait...for bad weather to happen. Or just **Configure** ⇒ **NWS** ⇒ **New Office...** and add * to get weather for the whole world (ok, at least as much of the world as is pushing weather alerts into APRS-IS). Within 15-30 minutes you should have stuff popping up in Australia. They've consistently had alerts running for me.

If you want to see useless "callsigns", check **Screen** ⇒ **Labels** ⇒ **NWS Calls**. These labels are a concatenation of the 3 character Seq and the 6 character county/zone of any given alert. Quite useless if you ask me, which is why they're suppressed by default.

Ok, with that all done, you should be able to pan and zoom around and see just how closely the shapefile boundaries match the underlying OSM map tiles. Freaky if you ask me!

WE7U is now feeding ALL NWS alerts into FireNet servers for display via APRS clients. To access this stream, you need to reconfigure your APRS-IS feed to come from firenet.us instead of an APRS-IS server. All APRS-IS data is available through FireNet and MORE!

Configure ⇒ **Ports** ⇒ **APRS-IS**

Click Device

Change the DNS or IP to "firenet.us" (without the quotes)

Keep your port the same (14580, most likely)

Ok and Accept and APRSIS32 will switch to the new server.

If you don't like all the water stations that you'll start getting, add "-s/w" to your **Configure** ⇒ **General** ⇒ **Add Filter** string (again, without the quotes).

Note: if you already have a -s/ element in your filter, just add the w to the first part.

Add File...

see above

Enabled

see above

Opacity... (10%)

TBD

Quality... (100%)

TBD

Show Center

TBD

Objects

Continue below

Create (0 Objs)

Create a new object at the centre of the screen.

Enabled

A tick box to enable transmission of the object.

Group

The object Group you wish the object to belong to, entering a group will add the group if it does not exist. Groups to be used for QRU must begin with a ?

Kill Flag

Selecting this tick box, will send out a kill request and remove the object from all receiving stations maps.

Symbol

Click the symbol to bring up a box to change the object symbol.

Name

Enter a name for the object, the more characters the better to help ensure uniqueness.

PHG

If the object is capable of transmission or reception, this button will enable you to create a PHG range circle.

Cmnt

Comment that is displayed on other stations displays. This should display any important information pertaining to the object. Repeater objects should contain Tone information, Range, scheduled nets, or any other important information. A format that sticks to breaking up information by groups of 10 characters to include spaces will conform to Mobile Radio Displays.

Item**DHMz**

7-character timestamp: day-of-the-month, hour, minute, zulu only

HMSH

7-character timestamp: , hour, minute, second. zulu only

111111z

Permanent time stamp

Interval

The interval between broadcasts of the object in minutes, "net cycle time" suggests this should be between 10 and 30 minutes. (For QRU query only objects set to 0)

Via IS/RF

Selects sending via RF, IS or both

DF

Pie shaped Direction Finding object that originates from your specified location and expands in the specified direction. This direction is the bearing to the transmitter as determined by your DF equipment. The shape is defined by

Bearing, Compass direction of the strongest signal heard.

Range, which is a judgment call based on s-meter readings and experience.

Quality, How accurate is your DF equipment.

DFS

Places a circle around the location of your reading based on

Strength, S Meter Units.

HAAT, Height above AVERAGE terrain. (PHG)

Gain, The gain of your antenna.

Direct, Directionality of your antenna if any.

RF Path

sets the RF path to send the object over, defaults to the beacon path. Paths should use WIDE rather than the older trace

Object creation is done via the configure/objects menu. and can also be done by right clicking a location and selecting the lat/lon option. To create an object move the map until the location you wish to place the object is centered on the map, it very useful to have crosshairs enable for this (configure/screen/crosshairs), and zoom in as appropriate.

When you are centered create the object via the configure/objects/create menu, you will get a popup menu. Enter a unique name, and a comment. It's useful to use all 9 characters for the name, to help ensure uniqueness. Click on the symbol, this will bring up a popup to enable selection of a symbol.

Remember to enable and set an interval this should be roughly the same as your beacon interval.

Select a time stamp option the default is DHMz (Day Hour Minute zulu) The PHG button enables the setup of Power Height Gain of the object should the object be a transmitter so that an RF range circle can be displayed. (more soon)

The DF and DFS buttons allow the setup of direction finding objects (more soon). See <http://ki6psp.blogspot.com/2011/05/may-fox-hunt-in-indian-wells-valley.html>

Remember that objects should be useful for LOCAL information. Investigate your local environment to see which path is most useful to serve YOUR local area. Object creation can also be completed by right clicking on the screen where you wish to place an object and selecting the GPS coordinates displayed and choosing create object here.

If you've misplaced your LAST CREATED OBJECT you can right click on the correct location, select the GPS coordinates and select Move (last object created name) here.

DF Objects can also be created with the right click option and then following the normal directions for DF/DFS object. The theory is that were DF object shaded area's overlap is the area of your hidden transmitter(s).

New JT65

TBD

Advanced Knowledge Base

JT65, developed and released in late 2003, is intended for extremely weak but slowly-varying signals, such as those found on troposcatter or Earth-Moon-Earth (EME, or "moonbounce") paths. It can decode signals many decibels below the noise floor, and can often allow amateurs to successfully exchange contact information without signals being audible to the human ear. Like the other modes, multiple-frequency shift keying is employed; unlike the other modes, messages are transmitted as atomic units after being compressed and then encoded with a process known as forward error correction (or "FEC"). The FEC adds redundancy to the data, such that all of a message may be successfully recovered even if some bits are not received by the receiver. (The particular code used for JT65 is Reed-Solomon.) Because of this FEC process, messages are either decoded correctly or not decoded at all, with very high probability. After messages are encoded, they are transmitted using MFSK with 65 tones.

Operators have also begun using the JT65 mode for contacts on the HF bands, often using QRP (very low transmit power); while the mode was not originally intended for such use, its popularity has resulted in several new features being added to WSJT in order to facilitate HF operation.

[Wiki](#)

New Weather

Prompts for a file name and location and otherwise works like a new object. If the weather file cannot be accessed during a poll, the object automatically disables itself with a message in the Trace Log which is also brought to the front. File must be in WxNow.txt-format data file input.

Show

Show all of your configured objects on screen. (They are not broadcast)

Max Group Objects (5 Objs)

The maximum number of objects in a group that will be sent, in response to a direct query from a station. Groups with more objects will respond with an error message.

QRU

Overview

The intention of QRU is to allow remote stations to send a query to the call QRU and retrieve objects of interest local to themselves, not necessarily local to the QRU server.

For example sending RPT2M to QRU will get the nearest 2M repeaters to your location, with the frequency and tone, in a format compatible with the tune function of some radios. So send a query and you have the nearby repeater objects on your map or described in text.

Client use

QRU servers display on your map with QRUcallsign and an Info Kiosk symbol.

To query a server to see which if any have objects within range of your station and what groups they hold, send INFO to QRU. Servers with objects in range will respond with a list of supported groups: de CALL-SSID:group(#) group(#)... Max O@Mmi/km where group is the name of the group, # is the number of objects in the group within range of the querying station, O is the maximum number of objects sent in response to a query, and M is the default (soon) range configured for that QRU server.

Appending a distance to the query (in miles or kilometers based on individual QRU server settings) will override each server's default range setting. INFO 20000 for example will get a response from QRU servers worldwide.

Once you have determined which groups are available sending GROUP to QRU where GROUP is the name of the group, should get a server to respond with the objects it has local to you, subject to the maximum amount of objects limit set on the server and also the range from your station. Again appending a distance to the query will override the server's default distance, but will not override the max objects that the server will send.

QRU servers can also be queried directly. For servers with a low amount of objects per group they will respond with a list of objects, servers with a large amount of objects will respond with an error message listing the number of objects per group the server holds.

Servers will attempt to limit the number of objects sent in a combination of max objects and range.

Servers will send item as message objects where ever possible, in an attempt to avoid large amounts of objects showing and also to better pass through remote igates.

Recommended Common Groups

In order for QRU servers to most effectively provide local information to visitors, a set of common group names is necessary. The list below serves as a starting point for groups you may want to configure.

Group	Description
AMBU	Ambulance
FIRE	Fire Departments
FOOD	Local Restaurants
FUEL	Gas/Diesel stations
HOSP	Hospitals
POLI	Police Departments
POST	Post Offices
VETE	Veterinarian
CLUB	Amateur Radio Clubs
STOR	Amateur Radio Shops
ECHO	EchoLink
RP23	23cm repeaters
RP70	70cm repeaters
RP2M	2m repeaters
RP6M	6m repeaters
RP10	10m repeaters
RP23	23cm repeaters
RP13	13cm repeaters
RP3C	3cm repeaters
RD3C	3cm D-Star repeaters
RD13	13cm D-Star repeaters
RD70	D-Star 70cm repeaters

RD2M	D-Star 2m repeaters
RD23	D-Star 23cm Repeaters
RT23	23cm Television repeaters
RT13	13cm Television repeaters
RT3C	3cm Television repeaters
LIFEBOAT	Lifeboats
SRAIL	Steam Railways
T2SRV	Approximate locations of Tier2 APRS Servers
WOTA	Wainwrights On The Air peaks

Server Setup

To enable access to the QRU server configuration, there must first be at least one object created, within a query group folder (queryable groups start with a ? in the local configuration).

Object creation is done via the **Configure ⇒ Objects** menu.

To configure an object move the map until the location is centred on the map, it very useful to have crosshairs enable for this (**Configure ⇒ Screen ⇒ Crosshairs**) and zoom in as appropriate. Remember to enable and set an interval and the vias. If you only want the objects to go out as a result of a query make sure the interval is set to 0, otherwise they will be sent on the scheduled interval also.

Remark: QRU<callsign> object is transmitted every **Configure ⇒ Objects ⇒ QRU ⇒ Interval**, but only if QRU is enabled and some suitably enabled ?group objects are configured. This object is an InfoKiosk Symbol (non-configurable) and includes a Range spec. See <http://aprs.fi/?call=QRUKJ4ERJ> for an example. If you want to see all QRU objects, add o/QRU* to your **Configure ⇒ General ⇒ Add Filter**. This object is transmitted on both RF and -IS ports, provided that Object packets are enabled on said ports. **End of Remark**

When you are centred, create the object via the configure/objects/create menu, you will get a popup menu. Enter a unique name, comment and select a symbol.

QRU Settings

The difference with QRU is that created objects must be within a group beginning with ? such as ?HOSP. Once the group is created with the first object, the group will be available from the drop down box. QRU settings will be available under **Configure ⇒ Objects** after the first queryable group is created, it will be greyed out until then.

Enabled

Enable the QRU server function of the software.

Interval (0 min)

The interval for normal transmission of objects. Best set to 0 (Disable) if you have a large number of objects.

Remark: But it does not transmit QRU objects (provided they are individually set to 0 (zero), instead makes the **QRU<callsign>** object visible to the public with the kiosk icon and the comment showing QRU server objects available (both cannot be change manually) **End of Remark**

Max Objects

Sets the limit to the amount of objects that will go out to a direct query to your callsign. If you have more objects in a group than this, a message will be sent with the total number of objects, but not the objects themselves.

Range

Sets the maximum range for each object, stations querying QRU must be within this distance of an object, the server holds for the server to respond with the object. (this can be overridden by the client's request)

Retry Messages

Enables the retry system system for objects sent in item as a message format

Advanced Knowledge Base 1

Radio Tuning Format (QSY)

Some APRS-aware radios, notably the Kenwood D710 and D72 (?) and the Yaesu FTM-350, are capable of tuning the voice "side" of the radio to a frequency received in an APRS packet from another station. In order for this to work, the frequency, tone, and offset must be formatted in a very specific fashion. The following describes the format that is known to be the most compatible with the various QSY-capable radios.

A brief description of how to sort the correct format to enable auto tune of objects, for permanent objects and QRU ones, thanks to Mike

442.725MHz T114 +500 comment

or

147.345MHz T107 +060 comment

works universally on older and newer Kenwood and Yaesu mobile and portable radios for the position comment. Note that the MHz is CaSe SeNsItIvE (the z is the only lower case)!

The corresponding digipeater object syntax would be (in KPC format):

;442.725+ *111111z4158.60N/08739.29Wr442.725MHz T114 +500 comment

;147.345+R*111111z4208.15N/08748.38Wr147.345MHz T107 +060 comment

The comment for a repeater frequency is a 3.3 decimal frequency followed by a 3 digit tone (or CTCSS or DCS) with no decimal and the the offset in 10KHz units i.e. 145.650MHz T077 -060

where the R in the 2m listing serves to make the repeater object unique... not everyone can list 147.345+ since it is a common repeater pair. Some can list 147.345+R, 147.345+MI, etc. The total object name is 9 characters, so you have 2 or 3 characters after the frequency. Note that the object name must also be space-filled to 9 characters. Many people think the + or - in a frequency object

name sets the offset direction, but that is only a visual clue for a human operator. It's the +060 or -060 or whatever that sets the direction AND offset.

The other digi format mentioned on WB4APR's site:

```
;FFF.FFxyz*11111zDDMM.hhN/DDMM.hhWrTnnn RXXm NETxxxxxx MTGxxxxx
```

will work with newer firmware versions. So, safe to use the one I suggested above to make it work for nearly everyone regardless of whether they updated their firmware or not.

You may be tempted to list your repeater offset as decimal MHz (+6.0M), but that's not the standard. Offsets are specified in 10KHz increments so a 600KHz offset would be +060 or -060. For 440, you might use +500 to get a 5MHz offset.

Steven, G6UIM

[continue](#)

[Advanced Knowledge Base 2](#)

APRS Freq Spec - AFRS (Automatic Frequency Reporting System)

09 May 12

WB4APR

Ham radio's biggest advantage of thousands of frequencies is also its biggest stumbling block at rapidly and efficiently establishing com-munications 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.APRSworld.net>

<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

The purpose of the AFRS 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 responsibility 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://www.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 (37 bytes max) (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>

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 signalling 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.

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 POSITION 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, 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: 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
1750	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
_1750 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. (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
FFF.FF MHz
FFF.FFF M
```

```
>WB4APR-11
FFF.FFFMHz
T107 R17m
```

```
>147.105md
T107 +060
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/DDDDMM.MMW\$FFF.FFFMHz ...	begins after symbol (\$)
!DDMM.mmN/DDDDMM.MMW\$CSE/SPD/FFF.FFFMHz ...	
!DDMM.mmN/DDDDMM.MMW\$PHGphgd/FFF.FFFMHz ...	
!DDMM.mmN/DDDDMM.MMW\$DFSshgd/FFF.FFFMHz ...	
!DDMM.mmN/DDDDMM.MMW\$DFSshgd FFF.FFFMHz ...	with SPACE insted of /
!DDMM.mmN/DDDDMM.MMW\$CSE/SPDFFF.FFFMHz ...	w/o delimiter
!DDMM.mmN/DDDDMM.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.

FREQUENCY ON GPS TRACKERS: APRS was intended to be a two-way communications 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 receipt, most APRS programs should combine this info with the station data to make the frequency parseable as needed.

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*11111zDDMM.hhN/DDMM.hhWRT079 R25m NETxxxxxx MTGxxxx...
;FFF.FFFxy*11111zDDMM.hhN/DDMM.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 always check 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 frequencies 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 11111z. 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 status of the node. Maybe "busy", "conf", "off_", "idle" etc. The CALL is the callsign of this node. But we dont 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 format will 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 transmitted 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.--NWDDDDMM.--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 transmitted 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 GPS 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.

Bob, WB4APR

Overlays

continue below

Add GPX File...

This option allows you to load a GPX file. Simply, click on Add GPX file, find the file of interest it will load.

GPX files can be a bit varied. This can cause interesting results since not every feature of the GPX may be supported. Try it out anyway, you will probably like it!

The scope here is not about GPX, but if you want to learn more about it, check out: GPX: the GPS Exchange Format (<http://www.topografix.com/gpx.asp>).

There are many tools that can create a GPX file. It may be a GPS application that can read the tracks, routes and waypoints inside a GPS. Since this is the exchange format, it is almost always a supported input/output. Another tool could be a web application.

No matter how the GPX is created, it can be a very useful tool. There are many waypoint files of potentially interesting items. They could be waterfalls, covered bridges, WX transmitters or maybe even the Bat Cave. The file might contain tracks

of places others have been. Content describing a track or route might be the directions to get from here to there. Or, maybe it defines a route/track of an event (e.g., bicycle race to raise awareness/money for Multiple Sclerosis).

Monitoring resources in the course of an event often utilizes APRS. And having the course, waypoints, etc, defined is quite useful.

Mapping software comes in many varieties and at varying cost. If your route stays on a drivable surface found on Google Maps, then there is a great online tool. That tool is GMAPtoGPX. The best thing to do is just to try it.

Please visit the GMAPtoGPX information page to get running with this tool (<http://www.elsewhere.org/journal/gmaptogpx/>).

Follow the Instructions. Do the first step! This is what makes the tool operational. Steps 2-5 are included below with a little more information added.

Once the first step is done, the next step is to acquire "Driving Directions" from Google Maps. Choose a starting point and a finish point. The "directions" Google returns may not be what you need or want. Hover over the route and you will see "Drag to change route". Modify the route as desired.

Note: there are only so many points along a route that can be added (~50?). If you have a route that needs more than 50 points to complete, break the route into sections.

Once you are satisfied with your route, click on the bookmark for GMAPtoGPX. This will open a window with text inside it. The data inside is basically what is needed for a GPS to follow. Read again, the third bullet under Notes on the GMAPtoGPX information page.

For cases other than stuffing the data back into a GPS, it is usually useful to click on "Full" just above the text. This will expand the data to add many more points.

The next step is to create a GPX file. A quick select all data (left-click in the data field and press CTRL-A). Copy the data (CTRL-C) and paste the data into NotePad (CTRL-V). Save the file with the file extension GPX (e.g, Here2There.gpx).

I like the fifth step on the information page. Try it and see what happens! So, load the file into APRSIS32. This is accomplished by: **Configure** ⇒ **Overlays** ⇒ **Add GPX file...**

Not only can you see the route, it can be "driven". This may cause APRSIS32 to fetch tiles that are local. Repeat at the required zoom levels. Always remember the tile usage requirements!

Add POS File...

This option allows you to load and enable/disable APRSdata/UI-View-style .POS overlay files.

Overlay files allow you to show information about objects on your map which are not transmitted to other stations, but only displayed locally.

Enabled files are remembered and re-loaded across client restarts as well. Comments on the position lines are used, but the status lines are not yet supported.

For more information and sample .POS files, read VE3KBR's excellent guide (http://www.ve3kbr.com/aprs/aprs_overlays.htm).

An example .POS file is (note: the first line is a description, not a position):

```
* MB7Ux Unattended 24/7 Digis - Created by M0CYP
MB7UA!5147.85N\00101.84En
MB7UA>144.800 Point Clear, Essex
MB7UB!5125.70N\00224.43Wn
MB7UB>144.800 G4KVI
MB7UC!5155.31N\00200.63Wn
MB7UC>144.800 G4GVZ www.g0lgs.myby.co.uk/grg
MB7UH!5154.53N\00053.93En
MB7UH>144.800 G6NHU www.mb7uiv.co.uk
```

And another one:

```
* APRS (MIC-E) Over Voice Repeater
MB7IP!5246.08N\00209.56Wn
MB7IP>144.800 < GB3IP 145.7625 CTCSS 103.5
```

History:

Overlays were first known to exist in APRSdata where it says:

"APRSdata uses all of the POS*.POS data files that you already have. To see these in any version of APRSdos, just hit the MAPS-OVERLAYS commands. These POS*.POS files are distributed with each of the major APRSdos Map distributions."

And here's a link to the original documentation with this priceless client-centric statement:

"Do not worry about exact lat/longs. Just use your eyeball on the most detailed APRS map that you have. The exact location will never matter, since anyone who is viewing the overlay data will be using the SAME map to VIEW it as you did to estimate the position in the first place."

Plagiarizing the UI-View Manual (UI-VIEW32V203MANUAL.pdf):

The format of an overlay file is very simple. The first line is a description, subsequent lines are location lines or status text lines. Location lines consist of up to nine characters for a callsign/identifier, followed by a '!', followed by a position beacon. Status text lines consist of a callsign/identifier, a '>' , and some status text.

NOTE - UI-View32 overlay files are not fully compatible with APRS overlay files, because APRS doesn't support the status text lines. However, if you create overlay files for UI-View32 and include status text, anyone wanting to use them with APRS can easily remove the status text lines.

And from Chip, VA3KGB:
How to create your own Overlay File

Create the overlay file using an ASCII text editor such as "Notepad". Notepad is great for creating short overlay files but for large files such as creating overlay files for the IRLP Nodes, I have a spreadsheet that reads in the raw data , lets me check for errors in the data manually and then when I'm happy with the raw data, the spreadsheet then creates the files for WinAPRS, UI-View, and xastir for me.

Overlay files are one line per object except for UI-View and APRSIS32. With UI-View and APRSIS32, you can have a two lines to convey more information.

Included with the overlay files, I also add a text readme file listing the overlay files, explaining where the files need to be located, and any other information that the APRS operator should be aware of before using the file. xastir users must turn OFF Object/Item transmit before using an overlay otherwise all the objects in the overlay get transmitted out over the APRS network!

Format for a one line overlay file for WinAPRS, UI-View, APRSIS32

```
* 144 MHZ IRLP Nodes Callsign overlay for APRS
* irlp.net node data: 08/12/2008
* This file created using irlpnodelist-aprs.xls by CJ Chapman, VA3KGB
VE7RHS!4916.19N/12315.00WnNode 1000,145.270,T100.0
VE7ISC!4913.64N/12358.45WnNode 1003,146.640(-600)
VE7VIC!4825.80N/12335.99WnNode 1030,146.840(-600),T100.00
VE7BYN!5049.52N/11859.48WnNode 1041,147.570
VE7KJA!5035.03N/12705.61WnNode 1064,146.445
```

The Asterisk (*) denotes that this is a comment and is ignored by the APRS program. I usually add comment lines to identify what the overlay file is used for, the date the file was created, and the spreadsheet or source data that was used to generate the file.

Each line is a separate Object.

The format using one line as an example is:

```
VE7RHS!4916.19N/12315.00WnNode 1000,145.270,T100.0
```

Callsign No APRS Message Lat Primary or Alt Icon Table Lon Icon comment

```
VE7RHS ! 4916.19N / 12315.00W n Node 1000,145.270,T100.0
```

The callsign (or identifying label) should be 9 characters or less or it will get truncated.

The (!) tells the system not to send the object out as an APRS message. or is it saying primary or alternate table for the Icon.

The Lat is in degrees and decimal minutes format (60°10.92' N 24°31.86' E) stripping out the degree and minute characters and is to 2 decimal places.

In between the Lat and Long is the symbol which identifies which symbol table to use for the icon to be displayed.

The Long is in degrees and decimal minutes format (60°10.92' N 24°31.86' E) stripping out the degree and minute characters and is to 2 decimal places.

The Icon is from APRS Icon tables and uses the alpha/numeric character in conjunction with the Primary/Alt table symbol to identify the graphical icon to display.

The comment field should be kept short, and in this example indicates the IRLP NODE ID, the Frequency and Tone required. Commas (,) are used to separate items within the comment. In the comment field using a spreadsheet to build the overlay files there will be comma with a space between them indicating that the source file had no information for that portion of the comment. If building the file by hand you can delete the space(s) and extra comma(s).

Format for a two line overlay file for UI-View and APRSIS32

* 144 MHZ IRLP Nodes Callsigns with Cities overlay for UI-View or APRSIS32

* irlp.net node data: 08/12/2008

* This file created using irlpnodelist-aprs.xls by CJ Chapman, VA3KGB

```
VE7RHS!4916.19N/12315.00WnNode 1000,145.270,T100.0
```

```
VE7RHS> IRLP Node: 1000, Vancouver, BC, Canada
```

```
VE7ISC!4913.64N/12358.45WnNode 1003,146.640(-600)
```

```
VE7ISC> IRLP Node: 1003, Nanaimo, BC, Canada
```

```
VE7VIC!4825.80N/12335.99WnNode 1030,146.840(-600),T100.00
```

```
VE7VIC> IRLP Node: 1030, Victoria, BC, Canada
```

The format for the first line is as per the one line format above.

The format for the second line as an example is:

```
VE7RHS!4916.19N/12315.00WnNode 1000,145.270,T100.0
```

```
VE7RHS> IRLP Node: 1000, Vancouver, BC, Canada
```


Callsign Comment-identifier Comment

VE7RHS > IRLP Node: 1000, Vancouver, BC, Canada

Callsign will be exactly the same as the first line.

Comment field again is short and in this example indicated that it is IRLP Node 1000 located in Vancouver BC Canada. (Commas (,)) separate the items in the comment. In the comment field using a spreadsheet to build the overlay files there will be comma with a space between them indicating that the source file had no information for that portion of the comment. If building the file by hand you can delete the space(s) and extra comma(s).

Converting GPX Waypoint and POS data to Objects

In some cases, the new data is useful as an object. Once a waypoint or position is on the map, it is simple to convert to an object. Find the point on your display and left-click on it. In the dialog window, the last option is MAKE OBJECT. Select it and the Create Object dialog box opens which will have data from the GPX or POS already loaded for the select point. Perform any editing necessary and ACCEPT. Keep in mind that unique names make your object unique.

Advanced Knowledge Base

APRSdata.TXT

APRSdata Sumary

12 March 2002

WB4APR

APRSdata is a copy of APRSdos with additional features for serving up "Tiny-Web-Pages" to Kenwood Mobiles and HT's in your area. The initial idea was to serve up Satellite data, but then it was trivial to add any other position related databases as well. Here are the default Databases:

- * Puts any Satellites in view up as Objects automatically. Each sat shows its up and downlink frequencies and present doppler. Below 4 degrees, the uplink frequency is replaced with the elevation.
- * Can give VOICE alerts to D700 users of satellites in view
- * Maintains an 80 min sat schedule in users DX lists every 10 mins
- * Responds with a packet of info for any of these special Query Commands:

HELP	- Responds with a list of commands to mobile's DX LIST
DIGIS,IGATE,WX	- Nearest DIGI,WX or Igate heard on the air as an OBJECT
SATS	- Updates the SAT SKED in the mobile user's DX LIST

- * Responds to these Queries with the closest object in the XXX.POS:

ATV,CLUB,NET,FD	- ATV repeaters, CLUBS, on-air nets or FD sites
Crashes	- Known crash sites to avoid errors during SAR
NWR, NWS	- NOAA WX Radio & Freq, or Natl Wx Service Site
Voice	- Nearest Voice repeater of significance
DX	- Nearest DX cluster
HOSP, FD	- Nearest Hospital or Field Day Site
RS	- Nearest Radio Shack or Radio Store
PARK,CAMP	- Nearest PARK (includes Camps) or just CAMPgrounds
XXXX	- Nearest Object in any other POS\XXXX.POS file

OTHER

- Will list these others files (NOT YET IMPLEMENTED)

SATELLITE SETUP: Please limit this capability to only those satellites that are workable with the mobile or HT so that you do not clutter up 144.39 with data of little use to anyone. At this date, I recommend ISS, UO-14, UO-22, AO-27, AO-40, PCsat and Sapphire. To enable these satellite alerts and skeds, select the POSITIONS list and move the cursor down to the satellites to be announced when they come in view. Hit ENTER to select the satellite and then "U" for uplink in the usual fashion to mark an object for uplink. If the satellite is above the horizon, then it will become yellow and uplinked immediately. If, as is more likely, it is not in view, it will be marked as a killed object as dark blue for subsequent uplink later when it does come into view.

HF SATELLITES: Since the Mode-A RS birds can be heard on a mobile HF rig, you can also just mark the satellite on the P-LIST for "L" to be sure it is included in the periodic DX LIST satellite schedule, but it will not be announced as an object to minimize QRM. The RS birds are in higher orbits and in view for up to 20 minutes or more at a time and it is felt that one-minute objects would be too much for the local net. Yet HF users still will see its AOS in their DX lists.

LEO SATELLITE TIMING: Satellites to be uplinked on 144.39 are transmitted when they pass above 1 deg and then about 80 seconds later using the normal decay rate. But this jump to 160 seconds trips the selection process so that the next time through the satellite list, in 10 to 20 seconds, it will again be rescheduled for uplink and the process repeats.

AO40 and AO10 TIMING: For the elliptical satellites, the same rule applies, but the delay between the transmissions is about 10 minutes. Due to the large number of modes, the UPLINK frequency is not shown but is replaced with an elevation angle when it is above 1 deg. Otherwise the center of the analog UHF passband is shown. Also, in the DXlist schedules, these satellites are only shown as " is UP" instead of an AOS time.

SATELLITE SCHEDULES: Once every 10 minutes, a single Satellite schedule will be transmitted as a DX spot. This will be captured on everyone's DX LIST and can show as many as 4 satellites. To force the transmission of a new schedule, hit NEXT-SCHED. Similarly, any user can request a satellite schedule with a QUERY:?SATS command. New in version DA0, satellites can be marked with a "L" for inclusion in the DX LIST separately without also forcing them as objects.

VOICE ANNOUNCEMENTS on the D700: To enable a voice announcement to alert anyone with a D700 and voice chip, simply select that station in the POSITION list and select the VOICE option with the "V" command. Once a minute (I will reduce this later when I get around to it) it will send out a message that will speak "U-O-1-4 LOW" if the satellite is between 5 and 19 degrees. "Medium" if it is 20 to 29 deg and "High" if it is above 30 deg. To allow users to request this data without prior action by the SYSOP, APRSdata will send voice announcements to any D700 that has his CUSTOM-1 comment set. This does add substantial message traffic to the channel load, so users should not do this except when really needed.

LOCATION: APRSdata uses the position of the mobile for its search. If the mobile wants to request the nearest Object closest to some other location, then he should temporarily change his posit to that location. If a Mobile has not transmitted a posit or it is 0000/0000, then APRSdata responds with a message that no data is available without knowing the stations location.

SYSOP FUNCTIONS:

While serving as this community resource for others, APRSdata still is usable by the local operator as a normal APRSdos program and assumes the operator is fully familiar with APRSdos. For more indepth discussion of my overall APRS/SATELLITE concept, see ASTARS.TXT or APRStk.TXT or better go to my WEB page to see the latest. <http://www.ew.usna.edu/~bruninga/astars.html> [broken link]

Just let APRSdata run on a PC in the corner. See the SETUP in the final section below for initial set up. In addition to the normal APRSdos displays for the local operator, APRSdata also:

- * Predicts satellite passes over the next 80 mins and displays the times and peak elevations graphically. (Use NEXT and/or FILE-FAST commands) It will do this EVERY time you redraw the map if you set ALT-S-OTHER to REDRAW. Or otherwise, on a slow PC, leave REDRAW OFF, and you can do it manually. In any case, to serve users, it will do it automatically after your keyboard has been inactive for 10 minutes.
- * Displays the names, Freqs and Modes of all amateur satellites. Hit the LIST-FREQS command.
- * Displays the basic Keplerian Elements for easy comparison with new or old elements. Hit the LIST-KEPS command.
- * Displays locations of satellites on the map as objects.
- * Shows the azimuth and elevation to any satellite selected.
- * Shows a set of elevation rings showing the range to satellites above 0, 5, 20 and 40 degrees from your location. Note that this is dependent on satellite altitude and defaults to 800 km. To change this use INPUT-EL-RINGS.
- * Won't let you miss a pass. If you are zoomed in below a range to see current satellites in view, it will paint the map the color of the current maximum elevation of any pass in progress.
- * Uplinks these satellite objects to your local APRS LAN to inform others. To do this select them with UPLINK command on the P-List. APRSdata will transmit them once per minute when in view, and then kills them once below -2 deg. Every 10 minutes when the SCHEDULE is drawn automatically due to no-operator present, then APRSdata will also send out a DX SPOT containing all passes in the next 80 minutes. If there is an operator present, you can do this at any time with the NEXT-SKED command. Or users can request it with the Q-SATS message.
- * See how many people query your data base on the QUERY line on the HEARD log.
- * Use CONTROLS-FILTERS-IGNORE to ignore all off screen stations. This helps you control how rapidly your LISTS fill up with local traffic.
- * WX-RawCnvt command now causes APRSdata to retransmit Raw UII wx packets as APRS formatted packets so that D7/D700 users can display them.
- * NEXT "-" command displays PREVIOUS 2 hours of passes so you can see what satellites you just missed.

HOW THIS WAS IMPLEMENTED: Here is what I had to do to make this work:

1) Queries are sent as Messages

- * You send the message to "QDOS, QUERY or ?" ...
- * In the message you send ?XXXXX, Where XXXXX is the data you want.

2) The kenwoods do not recognize the concise "ITEM" format which was designed for this application. Thus, I have to use the more verbose OBJECT format

3) APRSdos does not accept the ACK from APRSdata. THus, APRSdos users that request data must remember to ERASE the message request once the data comes in. (Until I get a new modified APRSdos out there)

4) The OLD D7's will not recognize a DX-SPot (the Sat Schedule) unless it is to a NON-APRS TO-CALL. Thus, these packets are sent to RESORC. And this takes a lot of overhead for APRSdos to switch back and forth with the UNPROTO command. Some TNC's dont do this well. SO each SYSOP should test with a D7 on the air to see if his TNC is putting in the UNPROTO RESORC VIA correctly.

5) The new COMPRESSED format for objects will not work with old D7's. So APRSdata forces NORMAL format.

DATABASE SETUP:

APRSDATA should default your station to use the INFOSERVER ICON (/?). APRSdata uses all of the POS*.POS data files that you already have. To see these in any version of APRSdos, just hit the MAPS-OVERLAYS commands. These POS*.POS files are distributed with each of the major APRSdos Map distributions. To differentiate them and match them to the area the users is in, each file uses a .XXX extension that matches the MAPLIST extension for that area. Thus if you are in the .EAS east coast distribution, then your Radio Shack file should be named POS\RSHACKS.EAS. You MUST rename all of these to the appropriate new file name conventions shown below. Be sure you have loaded the Emaps,Wmaps,Mmaps, SEmaps,NWmaps or Cmaps.ZIP file for your area.

But since APRSdata will only be serving your local area, and we need to make the file names short and recognizable, and different from the national .XXX distribution files, you need to COPY your old local files to these .POS files. APRSdata will only use .POS files for its QUERY responses:

RS.POS	(was RSHACKS.XXX)
NWS.POS	(Was NWSPONS.POS)
NWR.POS	(Was NWR.XXX)
CRASH.POS	(Was CRASHES.XXX)
ZIP.POS	(Was ZIP.XXX but distributed separately in ZIPCODES.ZIP)
ATV.POS	(Was ATV.POS)
VOICE.POS	(Was Voice.xxx)
DX.POS	A new file we are working on
CLUB.POS	My local area file is included as a sample
NET.POS	My local area file is included as a sample
HOSP.POS	My local area file is included as a sample
PARK.POS	My local area file is included as a sample

*** IMPORTANT*** Be sure to keep a copy of ALL of your *.POS files under a DIFFERENT EXTENSION, say *.MY, so that if you download a NEW copy of APRSDATA then my SAMPLE files will overwrite your .POS files and you will need to RE-COPY yours back..

You can build any local file you want. Such as FOOD or GAS, etc. Just use the format of these files and the NAME of the file will be its Query Name. For each such file you use, send me the NAME of it, so we can work up a common list of expected data bases...

The DIGI,IGATE and WX data does not come from POS*.POS files, but from what you have heard on the air. The advantage of this is for stand-alone WX stations that don't normally show on the Kenwoods. But here, APRSdata converts their raw-PEET-format to standard APRS format for display on the radios. Now, since you do not want DX stations to overwrite local stations, APRSdata comes configured with CONTROLS-FILTERS-LIMIT set to 20 to try to limit DX stations from entering your P List and overwriting locals. (CFL=20 means it will ignore packets with DIGI Fields longer than 20 characters).

COORDINATION: It is important to coordinate the location, paths and coverage of each APRSdata station. Here is what you can do:

- 1) On D-PAGE, do a JUST-APRDA to find others using APRSdata
- 2) Set your OUTGOING UNPROTO path to the minimum for your intended area
- 3) Set your C-F-L filter to a small enough value to not accept Queries from beyond this area. This is not easy to do with WIDEn-N, but will work in other areas.

APRSDATA FILE STRUCTURE:

APRSDOS has many built-in files.

The .POS files are single files that contain nationwide data. All other files are broken down to smaller "regional" files that have the same extension as the regional APRS map distribution files. For

example, all EAST coast files are .EAS. TO make these same categories work for users using the USA distribution, the default .APR extension is used. These .APR files are only "representative" selectoins and are NOT complete. Once the user zooms into the regional MAPLIST, then it will use the specific file for that region...

DATA FILES	M-OVRLY MENU	OLD FILE EXTENSION	ADATA USES	For the default USA distribution these .APR files are used
ATV	A	.POS		
CRASHES	C		.XXX	.APR 10% of them
DIGIS	D	.POS		
FREQS	F	.POS		
GATES	G	.POS		
NWSPOSNS	N	.POS		
RSHACKS	R		.XXX	.APR 10% of them
STORES	S	.POS		
VOICE	V		.XXX	.APR
NWR	N		.XXX	.APR 25% of them
ZIPCODES	Z		.XXX	.APR incomplete
OTHER.....	O	Prompts for file name		

The .APR files are selected samples so that when using the USA (.APR) distribution, something shows up. Once the user selects one of the regional MAPLIST.XXX, then the correct .XXX file for his area will be used.

Bob, WB4APR

Ports

In order to get APRSIS32 to talk to a GPS, a radio or indeed anything that provides or sends information, you need to configure a port. So how do you configure a port and what in the world does TCP/IP have to do with KISS and NMEA?

This nine minute video shows you the basics:

http://www.youtube.com/watch?feature=player_embedded&v=o0A_ZyMkY9M

APRSIS32 supports two different ways of connecting to serial devices. The "normal" expected one would be via a "normal" Windows COM port with baud rates, parity, stop bits and such. However, I also support connected to a TCP/IP source of this same data. This allows the client to connect to remotely served TCP/IP serial ports which may or may not be virtualized by Windows. I also happen to have some TCP/IP servers that remotely serve out COM ports from one Windows machine to another. So...

When you click the KISS or NMEA buttons on the new configuration dialog, you will be asked what sort of connection it will be, TCP/IP, BlueTooth, or COMn Serial. Most of you will probably want COMn Serial for a normal RS-232 port. Clicking that will reveal a Port Configuration dialog with drop-downs of the known COM ports and support baud rates along with radio buttons for Parity and Data/Stop bits. (Ignore the Xmit checkbox). When you have things set up for your GPS or KISS TNC, click OK.

After configuring a COM port, the KISS or NMEA button will take you directly back to the Port Configuration dialog. If you clear the COM port and click OK, the next time you configure that port, the TCP/IP question will be asked.

The AGW interface only support TCP/IP connections to AGWPE. The TCP Configuration dialog prompts for an IP address or DNS name of the host where AGEPE is executing. Port 8000 is the default for AGW and should not be changed unless you know what you're doing.

After configuring the appropriate port(s), you can then enable whichever feature you want to use. If any communications errors are encountered, you may need to disable and re-enable the feature to get things moving again. I need to do more work on failure recovery.

If you are using a Bluetooth serial adapter for either KISS or NMEA GPS connections (I use a Delorme BlueLogger as my GPS), you can either go through the pairing process and assign a COM port outside of the APRSIS32 client or you can click Bluetooth back at the beginning. If you pair to an Outgoing COM port, follow the instructions above for configuring a COMn Serial port. If you opt for the direct Bluetooth, APRSIS32 will poll for available devices and present a drop-down of the device names. Select one of those for use and proceed into the main port configuration. Whenever a direct Bluetooth port is opened, the devices are again enumerated and a direct connection to the device is established. If a PIN is required, your Bluetooth stack will prompt for it.

Drop a note to the list if you are using any of these new features. I'm anxious to see how they work out and won't know if they're working or not if you don't tell me!

The Quiet Time is the maximum length of time during which a packet should have been heard on that port. If no packet is heard within the specified time, the port is closed and re-opened using all of the configure `<Close/OpenCmd>`s. This is provided as a way to detect a possibly dropped network or bluetooth connection, or maybe even a TNC that has lost its mode and needs a restart to recover. Because there is no heartbeat/keepalive capability on a TNC connection, an RF Port Quiet Time must be set long enough to cover the maximum expected elapsed time between packet receptions. If set too short, the port will unnecessarily close and re-open itself periodically, possibly resulting in dropped packets across the restart. A Quiet Time of zero disables this function.

So, don't set this on a KISS-type port on a Kenwood radio. The default `<Open/CloseCmd>`s will leave the radio's TNC turned off until you manually turn it back on.

Don't set this on any port that doesn't have a steady stream of expected traffic (like an HF APRS radio port). It just doesn't work out very well.

If you have a steady APRS traffic hum on your radio port, set the Quiet Time to 2-3 TIMES the quiet time you'd expect to see on the channel. This gives APRSIS32 a

chance to reset the channel if it goes quiet too long. If you have sounds enabled, you'll hear bings and bongs while the radio port transitions through the various states of a restart.

The Port configuration callsign box is disabled and will probably be removed in an upcoming release. It was a misguided original thought to have the ability to specify different callsign-SSIDs for different ports within a single APRSIS32 instance. I've since decided, after discussions on this list, that that approach would be overly complex when the same or better functionality can be achieved simply by running a second APRSIS32 instance.

New Port...

Selection of **New Ports...** will pop up a box with following option.

Simply(KISS)

TBD

KISS

TBD

AGW

TBD

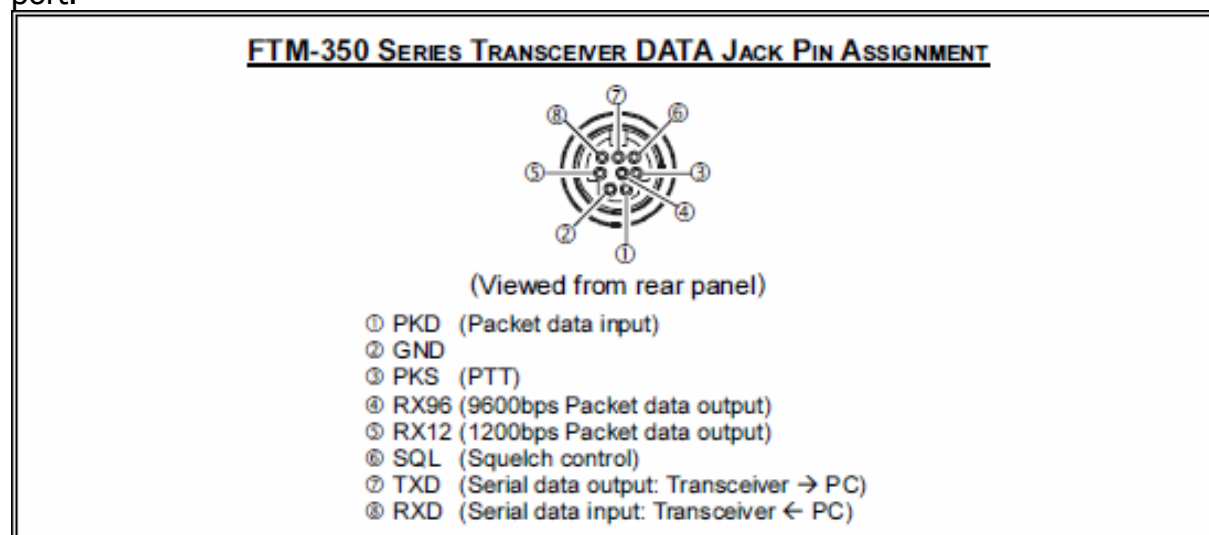
FTM350v1.3(RO+GPS)

The FTM-350 is supported by APRSIS32 with a Port Type of FTM350v1.3(RO) provided you have firmware version 1.3 loaded. This creates a receive only port as Yaesu doesn't (yet as of 1.3 firmware) support transmitting from computer control. Yaesu may at some point support a bi-directional packet radio port, but at this time, receive only is supported. You will be able to see the locations of stations being received by the radio, but will not be able to send packets from APRSIS32 to RF.

The E16 menu (v1.3 firmware) needs to be configured to enable packet output on the FTM-350AR. Set menu option E16's OUTPUT to PACKET for the raw APRS packets to be delivered out the serial port of the FTM-350AR.

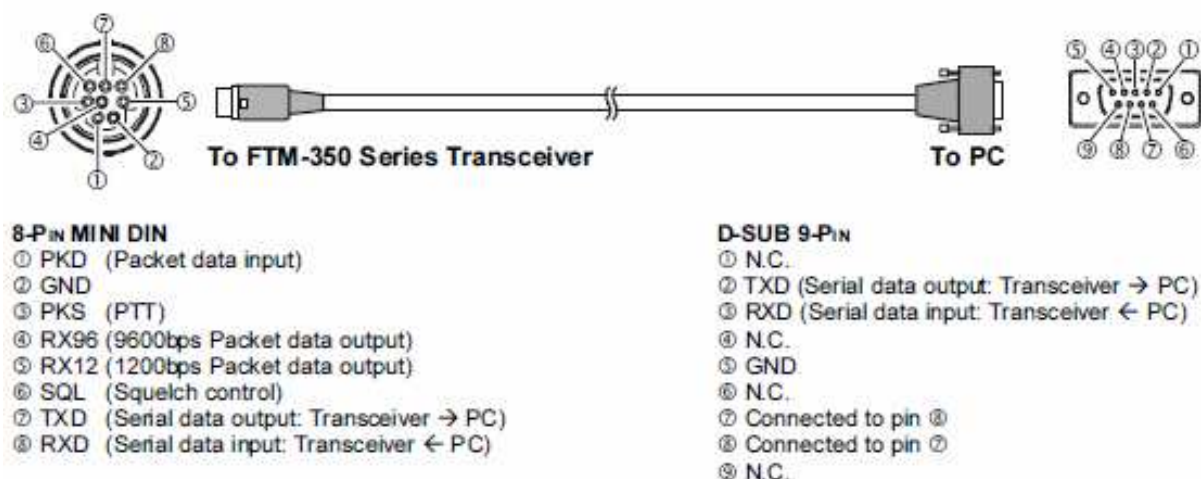
SET MODE ITEM	FUNCTION	AVAILABLE VALUES (DEFAULT: BOLD)
E16 COM PORT SETTING	Sets the COM port setting.	1 SPEED: 4800bps / 9600bps / 19200bps 2 OUTPUT: OFF / GPS OUT / PACKET / WAYPOINT 3 INPUT: OFF / GPS IN 4 WP FORMAT: NMEA9 / NMEA8 / NMEA7 / NMEA6 5 WP FILTER: ALL / MOBILE / FREQ / OBJ/ITEM / DIGI / VOIP / WEATHER / YAESU / C RINGER / R RINGER

The Yaesu FTM-350AR manual contains the pinout information for the radio's data port.



Yaesu also sells a ready made cable that can be used to connect the radio to your computer's serial port, called the CT-142.

CT-142 CABLE CONNECTIONS



KWD700(KISS)

TBD

KWD700(RO+GPS)

TBD

KWD710(APRS)

The Kenwood TM-D710 has the ability to push all of the APRS packets heard out the COM port (Menu 604 set to ON) that is found on the back of the control head when it is running in full APRS mode (Use TNC button to set to APRS).

APRSIS32 supports this mode by being able to parse the data being presented out the COM port during normal APRS operations, and displaying the information on the computer screen. By default, APRSIS32 still attempts to assume full control of the TM-D710 radio. It is possible to configure APRSIS32 in APRS mode to remain a passive component of the complete package. This allows the Kenwood TM-D710 to still be the component in control of the APRS operations, while APRSIS32 becomes the supplementary display, augmenting the APRS information available to the user.

The Kenwood TM-D710 is designed as a nearly fully self-contained APRS station, with the ability to perform most of the common APRS functions built right in. With a mapping GPS unit attached, one is able to not only send live position updates, but also have incoming positions from other stations displayed on the GPS as well. Weather station information as well as messages can be displayed on the TM-D710 screen.

Messaging can be initiated from the TM-D710 using the DTMF keyboard in a manner similar to texting on a cellular phone, but there are many who find this to be a less than desirable input method.

By having APRSIS32 monitoring the COM port, it is possible to decode and display all of the information received by the TM-D710. Everything that you can see on the TM-D710 display, as well as positions plotted on the attached GPS display can be seen on the APRSIS32 instance. In addition, you will be able to decode and display packets that the TM-D710 does not have the ability to decode/display. Packets such as DF reports, multiline objects, as well as recording tracks left by stations as they travel about and others can be visualized on the APRSIS32 display.

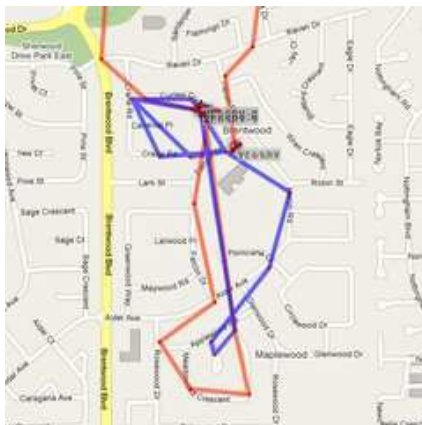
So why not just run in KISS mode, and let APRSIS32 take full control of the TM-D710? In KISS mode, APRSIS32 has no access to the GPS unit attached to the TM-D710. You'll either need to supply another GPS unit feeding the APRSIS32 instance, or disconnect the GPS from the TM-D710, and connect it into the device running APRSIS32. APRSIS32 does not have the facility for providing APRS waypoints to the GPS either, so the ability to have APRS icons displayed on the GPS, and subsequent functions such as navigating to the APRS POIs on the GPS are no longer available. The TM-D710 can also be configured to display the information received in packet on screen automatically. This is a function also not supported by APRSIS32.

By having the Kenwood TM-D710 running in native mode, you get all of the functionality available in the TM-D710, augmented by the abilities of APRSIS32. You lose nothing, and gain a great deal of added functionality through the use of the combination of these devices.

When in APRS mode, all of the packets received by the TM-D710 are passed along to the APRSIS32 instance for processing. Also included are the GPS positioning strings from the attached GPS unit (Menu ??? set to ??? and KWD710(APRS) port in APRSIS32 has "GPS/NMEA" checked). This simplifies the interconnection of the TM-D710 to the APRSIS32 instance, with a simple single serial connection between the devices. This can be accomplished with a simple serial cable connection, or a wireless solution as well.

The trick comes in when APRSIS32 wants to have access to the TNC in order to send packets outbound through the radio. The COM port was designed to be a monitoring port only, allowing a program such as APRSIS32 to listen to what the radio was hearing. However, there are a number of commands that can be accessed via the COM port which allow access to the internals of the control head and TNC.

APRSIS32 makes use of these internal commands (TC 0, TC 1) in order to assume control of the radio so it can send packets. In doing so, APRSIS32 sends a number of commands to the radio, including a command to set the source callsign (MYCALL). If the TM-D710 is left in control, and APRSIS32 is attempting to assert control, you can end up with some interesting effects. Note that APRSIS32 cannot provide digipeater support in this mode because the UNPROTO ... VIA ... command does not support used (*) path components.



The basic tenet espoused is that every APRS instance running should have it's own unique callsign-ssid. When running in APRS mode, it is possible to have position packets being sent by both the TM-D710, as well as APRSIS32. This can create what appears to be many more position packets than desired. You can also end up with a situation where the TM-D710 may be sending position packets, only to have the APRSIS32 instance take over, and vice-versa. When attempting to follow such a station, it can become difficult to keep track of where they have been... The image to the right shows a track that started off being created by the Kenwood TM-D710 as VE6SRV, but partway around Meadowood Crescent, APRSIS32 took over command, and changed the callsign to VE6SRV-9.

An interesting side effect of this phenomenon is that the APRS software running in the TM-D710 control head is not aware of the callsign change in the TNC, and upon hearing it's own packets being digipeated, the TM-D710 will display the packets on screen as if they were originated from a different station, and will also pass the packet waypoints to the attached GPS. This mode of operation incidentally is a mode

desired by a number of APRS operators, which is usually expressed as the "Why can't I see my own packets?", or "How can I tell where I am?" queries on various message boards. It does work on as a handy way of knowing when your packets are being digipeated, and received back by your station.

How can you configure APRSIS32 to run in APRS mode without usurping control from the Kenwood TM-D710 radio? Well, you really can't if you expect APRSIS32 to ever transmit because on the first transmission, MYCALL will be set which changes the D710's identity until it's next TNC reset. However, if you avoid messaging from APRSIS32 (or even receiving a message to which an ack must be sent), the following might work for you.



Start by creating a new port, selecting KWD710(APRS), and the appropriate COM port for your hardware. Once you have the hardware configuration selected, you'll end up with the Port Configuration screen as seen to the left. There are a number of check boxes on the screen that allow you to customize the packet types being sent on that port by APRSIS32.

RF to IS allows packets heard on this RF port to be passed to the APRS-IS stream. (i-gating to the internet)

IS to RF Probably unchecked to avoid transmissions - allows packets heard on the APRS-IS to be passed to the RF network for stations heard locally. (i-gating to RF)

Me not 3rd Doesn't work due to UNPROTO/VIA restrictions - keeps packets from your own station using other SSIDs from being passed as third party traffic when gated from IS to RF

Enabled Checked - Enables communications on this port

Bulletins/Obj Unchecked - Allows APRSIS32 to transmit bulletins and/or objects on this port

Messages Checked - If you want messaging to work on APRSIS32 (See warning above)

Xmit Enable Unchecked if you want to NEVER assume control of the radio, Checked if you expect even message acking to work from APRSIS32

Beacon Unchecked - You're relying on the D710 to be beaconing

Telemetry - Unchecked

GPS/NMEA - Checked - To allow APRSIS32 to pick up passthrough NMEA data from the D710-connected GPS

For XML raw datas see Annex

KWD710(Pkt)

TBD

CONVerse

TBD

TEXT

TBD

CWOP

Selection will create a receive-only CWOP port by the following procedure:

- 1) **Configure** ⇒ **Ports** ⇒ **New Port**
- 2) Select a Type: **CWOP** from the drop-down
- 3) Name the port appropriately (i.e. Weather or **WX-CWOP**)
- 4) Yes, you want a **TCP/IP Port**
- 5) Set the IP or DNS to **cwop.aprs.net**
- 6) Set the port to **10152** for a full, world-wide feed
- 7) **Check Enable** ⇒ **RFPorts** ⇒ **<Your Name>** to start receiving data

I don't do filters on this port yet, so I don't think 14580 will work.

See also: [APRS Servers](#)

If you want to quit getting this data, just uncheck **Enable** ⇒ **RFPorts** ⇒ **<Your Name>**.

Advanced Knowledge Base

I was asked to weigh in on this regarding what those stations are and why they don't show up on APRS-IS. Considering it is because of my software they don't show up, I guess that is a reasonable qualification.

The CWOP (Citizens Weather Observation Program) is a volunteer weather reporting system implemented a little over 10 years ago using APRS-IS as its reporting network. This was done because it was born out of ham stations already reporting on APRS (and therefore APRS-IS) and promoted by Steve Dimse (findU owner/operator) and a few people in NOAA (the US national weather service).

About 3 years ago (I think), we moved the CWOP stations (CW or DW followed by 4 numbers) to their own network because of some poor programming techniques that caused massive spikes on the core servers and connected downstream servers. They have operated on their own network since then and have grown from about 3700 stations to over 5000 stations.

Now the beauty of this is that while their reports don't show up on APRS-IS (separate network), our weather reports show up on their servers. This is accomplished by a read-only server that takes feeds from APRS-IS and the CWOP servers. It is this feed that aprs.fi is using to populate its database. There is not bidirectional operation here; this is unidirectional to the read-only server and then database applications like findU and aprs.fi can pull their information from there or from APRS-IS (or both).

Hope this helps explain what you are seeing and why it won't affect your everyday operation. Basically, APRS-IS and CWOP go to a read-only server which can only be accessed by a database server or other server software that doesn't feed back into either network. Even if the CWOP traffic was fed back in, it would be blocked at the core servers as they do not use validated logins.

Pete, AE5PL

IS-Server

TBD

Local-Server

TBD

APRS-IS

TBD

NMEA

TBD

Save Posits

Finally added **Configure** ⇒ **Save Posits** ⇒ **Filter...** and **Reload Paths**. The former shows how many stations will be saved with the current filter (0 if you have an invalid filter or no filter) and allows entry of an APRS-IS-style filter. The latter does a double-pass over the saved posits during reload in an attempt to re-establish the paths between them. A double-pass is necessary as a path is not recorded unless the station is already known, so the first loading pass generates an incomplete path picture. If Reload Paths is not checked, then NO paths will be reloaded (speeding the reload somewhat), but allowing new paths to be learned quickly as the digi/igate stations will (hopefully) have been reloaded (if you restart soon enough that your saved stations don't age out 30 seconds after loading).

For the adventurous among you, there's a new XML element called `<SavedPositFilter>` that you'll find at the bottom of the file. Set it to an APRS-IS-style filter (b/* for all) and it will save and reload the matching most-recent posit packets when the client is closed and restarted. I do NOT recommend this on a full-feed instance unless you really clamp down on the filtering! (Read: VERY slow to restart!) Also, an amazing number of posits contain non-printable characters. It turns out that the XML format doesn't support saving characters with ASCII values less than 32, so some stations may not be restored. Also, the saved posits are just that, posit packets. There's no recalling of the RF/IS status of said station, but they all appear to have been heard from APRS-IS. That said, the AGE of the posit is recorded and restored and used for expiring old stations, so if your client is down for a long time, you can expect to see the loaded stations for up to 30 seconds before they disappear due to age.

Reloaded Paths

TBD

Save Filter... (0/0)

TBD

Screen

The Configure menu provides access to all of the not-so-frequently changed and/or initial setup options of APRSIS32.

Label

Selection

Allow Overlap

Callsign ⇒ Not Me

 ⇒ Not Mine

Footprint ⇒ Min Altitude (10000ft)

 ⇒ Max Altitude (300000ft)

Geocache ⇒ GCnnnnn

 ⇒ Type

 ⇒ Container

 ⇒ Diff/Terr

 ⇒ Max Labls...

Max Visible...

Altitude

TBD

Battery

TBD

Circle

TBD

Crosshairs

Selection

Always

Never

Timed

Date/Time

Selection

None

Local

GMT/UTC

GPS only

Performance

Date Only

Default

Time Only

Custom ⇒ Date On 0

...

⇒ Date On 9

GridSquare

TBD

Lat/Lon

TBD

Orientation

Selection

Wide

Narrow

Automatic

RedDot

TBD

Satellites (GPS)

TBD

Speed Font Size... (5)

TBD

Symbol Size... (1)

TBD

Tracks

Selection

Follow ⇒ Color... (black)

⇒ Blocks... (32)

⇒ Width... (1pxl)

Other ⇒ Fixed Color... (black)

⇒ Rotating...

⇒ Add Color

⇒ aqua

...

⇒ yellow

- ⇒ Points... (32)
- ⇒ Width... (1pxl)

Scroller

The scroller is the list of packets that shows to the left of the map. Sometimes it's short (Wide Orientation) and sometimes it fills the left side (Narrow Orientation), but in either case, it displays packet information as it is received, governed by the options below.

The scrolling packet window shows packets for stations (or objects) that were recently received. This window communicates many things about the packet as follows:

Callsign-SSID - The assigned station callsign, or tactical station identifier

- * packet was received from RF (AGW, KISS or TEXT ports)
- # packet was received from RF inside a 3rd party packet
- @ packet was digipeated by your station
- *n packet was total received n times (IS, direct RF & digipeated RF)
- (n) multiple copies of the packet were received via APRS-IS



Symbol - current station symbol including all spec-specified assumptions based on:

- 1) Explicit statement in packet
- 2) Weather data is a weather symbol
- 3) GPSxxx specification in ToCall (Page 92/103 of aprs101.pdf)
- 4) Assumption based on -SSID value (Page 93/103 of aprs101.pdf)
- 5) APRSIS32 also uses ~ for Telemetry data if no other symbol was previously known

Clicking on an entry in the scrolling list will bring up that station's popup menu as well as present the packet for that entry across the bottom of the map.

Note If the scrolling region gets a red border when you click it, it is frozen for review. Left click anywhere BUT on the scroller (or right click on the scroller) to unfreeze the list and allow new entries to be displayed.

Note Also: The contents and behavior of the Packet scroller are controlled by Configure / Scroller Menu options so YMMV.

Text Color

Black : No position known for the station

Red- Yellow- Green- Scaled color based on proximity (Red = close, Green = far)

The scale of this conversion is the circle on the map. If the station is within the circle, the color will vary from Green (outer edge) to Red (dead center) based on how close the station is to the coordinates at the center of the screen.

Reversed Red- Yellow- Green- Scaled as above, but reverse video for callsigns that seem to be in your extra filter as a "buddy" (b/) or "friend" (f/). This may get some false positives until I add explicit Buddy and Friend filter configuration.

Background Color

"Invalid"	See details in the upcoming "Invalids" trace log
Purple	Duplicate packet
Pink	Too Quick packet (sub second)
Red	Too Fast packet (>20,000mph or double rolling average)
Aqua	"Restart" packet (long story, under development)
Gray	Unknown Invalid (should never be seen)
Yellow	Other APRSIS32 users have a yellow background unless something else overrides the color (like buddy).

Note that none of the following settings are retro-active. They are only checked when new packets arrive to determine if they will be added to the scroller. Once a packet is or is not added, it will be there or not until it scrolls off the top of the list.

Freeze On Click

If checked, left clicking on the scroller will freeze it, preventing any new packets from being added. Subsequently clicking on any station ID will present the station popup menu for that station as well as provide a preview of the packet at that entry across the bottom of the map. Clicking outside the scroller (or right clicking on the scroller) will un-freeze it and immediately add any new packets that were received while frozen. Right-clicking on a non-frozen scroller will bring up the station popup menu immediately and allow the scroller to continue moving in the background.

Show IGate/Digi

Adds the IGate (APRS-IS) or Digi (RF) that is determined to have last handled the packet. Requires configuration of <PathAlias> elements in the XML configuration to eliminate locally-defined aliases. Under Windows Mobile, the station ID is replaced by the IGate/Digi ID for space considerations.

Filter...

Allows entry of an APRS-IS-style filter to control which packets are added to the scroller.

Hide NoParse

Hides packets that failed the parser (see also the NoParse Trace Log).

No Intervals

Hides APRSIS32 Internally generated packets (see also the Packets(Internal) Trace Log)

Not Me

Hides packets from the configured callsign, even digipeats of your own packets.

Not Mine

Hides packets from the same base callsign, regardless of -SSID.

RF Only

Shows ONLY packets received from RF Ports (handy for IGates, boring on an APRS-IS-only instance)

Show All

Forces ALL packets to be added to the scroller. Handy to set your preferences and toggle to ALL sometimes.

Status Report

The Status Configuration dialog allows setting both the Beacon Comment (redundant with Configure General and the Configure Status Report).

Selection of conditions to transmit the beacon comment. The comment itself can be changed either via [Configure](#) ⇒ [General](#) , [Configure](#) ⇒ [Beacon](#) or [Configure](#) ⇒ [New...](#) ⇒ [Status Text](#) (redundancy to each other).

Enabled

If checked, status report packets will be periodically generated by APRSIS32.

GridSquare

If checked, the current GridSquare will be included in the Status Report. Use this with caution if you are also transmitting position beacons as the position resolution between the GridSquare and Lat/Lon coordinates is significantly different, possibly resulting in a station that ping-pongs between two locations depending on which packet was heard most recently.

Timestamp

If checked, the current timestamp (ddhhmm zulu, unfortunately not hhmmss) will be included in the Status Report. This can be handy in demonstrating packet delays in the APRS RF network possibly in excess of the 30 second APRS-IS duplicate filter causing two instances of a single Status Report to appear on APRS-IS. Without a timestamp, it's hard to prove that the packet was duplicated.

DX ()

TBD

New...

Opens a Status Text box to immediate change the content.

None

No status report is transmitted when activated.

Clear

Continue below

Clear Eavesdrops ()

If you have eavesdrop enabled in the messages configure menu. This option will remove all messages received that are NOT addressed to your callsign.

Clear CQ/ALL Messages ()

TB Removes all messages received addressed to ALL, CQ and QST.

Clear Group Messages ()

TBD

Clear My Messages ()

Removes all messages addressed to your callsign, including those addressed to different SSID's if My Messages is configured.

Clear NWS Messages ()

If you have setup to use the National Weather Service, this option will clear all received messages.

Clear Paths

TBD

Clear Stations ()

This option will clear the map of all displayed stations.

Clear Tracks

TBD

Accumulated

Clear AltNets

Clear the list of received Alternate Networks.

Clear Bulletins

Clear the list of received Bulletins.

Reset MaxWidth Station	Resets the width of the scrolling station list, to a minimum size, until a station with a longer name is received.
Clear Tactical	Clears the list of received tacticals.
Clear Telemetry	Clears the list of received telemetry.
Configured	
Clear Auto Replies	Clears all of the configured Auto Reply responses. A new auto reply comment will need to be added afterwards. If auto replies are in use.
Beacon Comments	Clears all of the configured beacon comments. The current Beacon comment will not be removed.
Clear Status Reports	Clears all of the configured status reports. A new Status report will need to be entered if in use.
Clear Symbols	Clears the list of symbols except the symbol currently in use for the station.

About

Displays the Version of the software that you are using, and copyright information.

The important part is that the about window also checks to make sure you have the latest version of the software. And gives the option to download a newer version if one is available.

If you like to follow the the developement version then this has to be done in the XML-file: `<Update.Development>1</Update.Development>`

This can be reversed to 0 (Zero) to stansard version. Anyhow bugs having been caused by the development version previously will not be illiminated. So always back up a XML-file of your standard version before manually changing anything.

Advanced Knowledge Base 1

Development Verion

To those that were asking where the Development version is, you need to edit the `<Update.Development>` tag in the APRSIS32.XML file following the instructions at Editing XML.

Once the element is set to 1, the normal version check (**About** box) and update mechanism automatically tracks the latest development build instead of the released build. Changing back to 0 will go back to the release build for the upgrade checker, but...

Be aware that some development versions make changes to the XML configuration file, so save a copy somewhere or set up the development execution from a different directory. APRSIS32 does make automatic version-dated copies of the About-upgraded .EXEs and .XML configuration files, but I have not yet documented how to use them (easy enough though, rename a matching pair to their original names and double-click).

I make no promise that any given Development release will function properly in all areas, although that is certainly my goal. As the product increases in features and function, it also increases in complexity and every increase in complexity introduces more and more places for errors in the code. My track record is pretty good (I think there's only been 3 or 4 times that the Dev release crashed in different installations), but please don't perform this change unless you're willing to put up with buggy software, even for a little while (wait, you're running Windows, you're used to putting up with buggy software, right?).

Lynn, KJ4ERJ

continue

Advanced Knowledge Base 2

Editing XML Configuration

WARNING! There are parameters in the XML configuration file that can completely hose the proper operation of the APRSIS32 client. Please save a copy of your current configuration before changing ANYTHING!

WARNING! A busted XML file will cause APRSIS32 to prompt for a completely new configuration. Make sure you save a current copy !

- 1) Close the client. If you don't do this, any changes you make to the XML file will be lost when you DO close the client as it is completely rewritten by the client on close.
- 2) Locate and edit APRSIS32.xml (or APRSISCE) . On Windows Mobile, it will be in your My Documents folder and probably synced to your desktop. Make sure you wait for the sync to finish after closing the client. On Win32, it will be in the default directory where you run the client.
- 3) Edit this file with a plain text editor. Find your tag and make sure you edit ONLY the value between the `<TagName>` and `</TagName>` elements. Disrupting these matching pairs will result in a failure to load your XML file. When this happens, the client will prompt for a new configuration. You DID make a copy before editing it, right?
- 4) Save the file and wait for it to re-sync back over to your Windows Mobile device before restarting the client. Otherwise, the client will update the file and your changes will probably be overwritten by ActiveSync.
- 5) If the client prompts for a configuration like it did the first time you ran it, you hosed something. It's time to go back to your saved copy and start over. You DO have one, right?
Oh, and did I caution you to save a copy of the XML file before you edit it?

Lynn, KJ4ERJ

Exit

End of session. If Exit Confirmations are enabled, a Restart option is also provided.

Igating

With the Dev 2012/09/04 21:26 release, you can now gate arbitrarily filtered packets from APRS-IS to RF. I hope you've been playing with the Control-G Filter Test, because that's the foundation of the function.

1. Please do NOT just go set up a filter for the first time and turn it loose on the local

RF! If you haven't run your proposed filter for a few hours or preferably days, and monitored the FilterTest trace log, then you're probably not ready to gate such packets from APRS-IS to RF.

2. I don't recommend just throwing in a buddy (b/) or prefix (p/) filter in and using it to gate packets. APRSISCE/32 is a particularly bad client to gate from -IS to RF in this fashion as it transmits telemetry that doesn't really lend any value on the RF side. I recommend a +t/p (note the leading plus) to your IGate filter to constrain the gating to ONLY posit packets. Any filter term can be prefixed with a plus (+) to make it a REQUIRED (read: AND, not OR) match.

3. The IGate filter can only select packets that are already coming from the APRS-IS feed to your IGate via your existing filter. This means that if you, like me, run your IGate with a zero range filter, it's unlikely that you'll be receiving any packets that you might be interested in filtering for transmission. Make sure your APRS-IS feed is delivering your desired packets via **Configure / General / Add Filter** or **Range** before saying that the IGate filter isn't working. If it doesn't come in, it won't go out.

4. To disable filtered -IS to RF IGating, simply disable the IGate(Filter) Trace Log.

To get it all going, do the following:

1. Ensure that your desired packets are coming in from the APRS-IS
2. Hit **Control-G** and enter in your proposed filter. This will bring up the FilterTest trace log.
3. Monitor the FilterTest trace log long enough to ensure that your filter is giving you what you want. You may need to iterate 1 and 2 to fine-tune the filter. Give it LOTS of time before going on.
4. Hit **Control-I** to convert the Test Filter to the IGate Filter. This will bring up the IGate(Filter) trace log that will show you the actual hits and 3rd party packets being transmitted. The outbound RF path is taken from **Configure / Beacon / Path** for now.
5. Disable the IGate(Filter) trace log or restart APRSISCE/32 to stop filtered -IS to RF IGating.

Note: This is just the first incarnation of filtered -IS to RF IGating. Multiple filters will eventually be supported as well throughput constraints of some kind as well as filter-specific outbound paths.

Supplement information:

If you set a TestFilter (**Control-G**), then hit **Control-I** (Tab) for IGate, AND meet a bunch of other conditions, you should get asked if you want to "Set IGate Filter To...

". If you say "Yes", the displayed current TestFilter will be used to arbitrarily gate packets from -IS to RF, but ONLY as long as all of the conditions are still met. The only way to change the filter is to change the Test Filter and then do another Control-I. The only way to clear the IGate filter is to restart APRSISCE/32, however if you disable the IGate(Filter) trace log, then gating will cease. Conditions: Development mode, APRS-IS has IStoRF checked, At least one Xmit-enabled RF Port, FilterTest Trace Log is Enabled.

The IGate Filter enabled is not remembered, but the value is. After a restart, the Control-G filter will offer it as a default which can then be re-adopted into the IGate Filter via Control-I. This is to make selecting the filter a very conscious decision until additional safeties are put in place.

Double-clicking the APRS-IS OK pane will show IStoRF: n if filtered packets have been gated.

Don't gate messages from -IS to RF if the station was heard inside a 3rd party packet. IGate capabilities and the ?IGATE query will include a PKT_CNT similar to MSG_CNT to show how many filtered packets have been gated since the program was started.

Don't count a 3rd-party wrapped station as HeardOnRF, only HeardAs3rd.

Don't gate messages from -IS to RF if source station was heard locally via RF per #2 at <http://www.aprs-is.net/IGateDetails.aspx>. I use the same timer and hops as "Recently Local", namely 30 minutes and 2 used hops (double-click vertical bar just right of the map). These will eventually be configurable.

Disable MeNot3rd (wasn't using port configuration checkbox anyway) as it borks other things. Good idea, but I need to think more on it.

Detect and log, but don't learn, unused -0 alias following a used alias. Still bogus, but don't know what digi might be busted or if it is an original SSn sans -N. W4POX-5

```
W4POX-5>SVUP0Q,KB4ZIN-5*,WIDE1*,K4MQF-1*,NV4FM-5*,WIDE2*,VA2,qAR,KG4LAA:'h.#1#/]Portsmouth EOC
```

Include known aliases in the unused -0 detection. Yep, that highlights a BUNCH more potentially bogus digipeaters! You actually might not want to run Alias(-0) enabled on a full feed now. (W1WAB-9 via KG4YZY-10)

```
W1WAB-9>R7TR8R,NI4CE-12*,KG4YZY-10*,WIDE2:`n1.1 j/"46}
```

Correct New check when forcing output to the Alias(-0) trace log. Funny what a big difference a ! makes.

Break out Alias(-0) to Alias(-0) Alias(Dbl-0) and Alias(First-0). See

<http://aprsisce.wikidot.com/menu:configure-aliases> (at the bottom) for details on what they mean.

Corrected View / Platforms / Unrecognized to only look at non-objects. It was mistakenly only counting objects when it's the object originator that has the platform!

Replace View / Platforms / None count with Non-Object count instead of total station count. This is the denominator for the percentage calculations so they'll make more sense now.

Expand View / Platforms / Experimental like Unrecognized to see the values after the >APZ.

Promote "aprx de OH2MQK" to a dedicated View / Platform category with 388 concurrent users over the past 2 hours.

Update platform determinator to trump weather-packet implied platform with ToCall-defined platform. SQ3FYK WX/Digi was incorrectly appearing as WinAPRS.

You can expect the View / Platform numbers for the "APRS Software Types" listed on page 63 of aprs101.pdf to go down as these were being improperly categorized relative to the packet's ToCall. These are APRSdos, MacAPRS, pocketAPRS, APRS+SA, WinAPRS, X-APRS.

Promote "SQ3FYK WX/Digi" to a dedicated View / Platform category now that it's no longer mis-categorized due to Weather.

Divorce Weather's "APRS Software Type" platform from the Station's platform type. The weather's ostensible platform type is displayed along with the Unit in the station's popup window.

Move APTTxx (Tiny Trak) from Others to Byonics cascade in View / Platforms.

Promote "OpenAPRS" from Others to a dedicated View / Platform category with 72 concurrent users over the past 2 hours.

Identify UISS platform as {UISSnn} in the comment. They really need to get an assigned ToCall as they're using >APRS generically.

Break out UI-View versions based on {UIVxxx} in comment and/or APU1/2/3xx ToCall.

Lynn, KJ4ERJ [from email Rev 2012/09/04 21:26]

Satellite Operation

Control-O (Oscar) opens the world of Satellites. This is only accessible if you use the development version of APRSIS32.

Satellite objects include AOS: (invisible) and LOS: (visible) information in their comment in delta time format.

Revamp of the Satellite code to show AOS LOS and expected elevation, TLE downloads rather than hardcoded internal TLEs. You can even paste temporary TLEs directly into the Satellite Details window (still initiated by Control-O for Orbits).

Support TLEs coming from a web URL - Auto-loads at startup or on demand via Satellite Details window.

Support TLEs coming from clipboard via the "Paste TLE" on the Satellite Details window.

Satellite tracks and circles, in fact not just satellite but ANY lines.

If the old TLE set was downloaded, it will offer to view ALL of the satellites.

Advanced Knowledge Base

You will then be given the option to view all satellites or just the ISS. In either case, once satellite tracking is activated, a "Satellite Details" window will be available from the Left-Click popup menu. From there you can select what satellite objects are actually generated and under what conditions. The checkbox in Satellite Details means:

Checked - Always generate satellite objects

Unchecked - Never generate satellite objects

Tri-State (faded check) - Generate killed satellite objects 10 minutes before AOS and after LOS, generate Visible (V overlay) satellite objects while in view.

TLEs for the satellites come from Celestrak Amateur TLE and are filtered to the currently active satellites listed below. To completely turn off satellite tracking, simply close the Satellite Details window and answer the confirmation prompt. If the client is closed with satellite tracking active, it will automatically re-activate on restart with the most recent checkbox settings intact.

The common reasons for errors in tracking satellites are the following:

Very old TLE data, especially with the ISS please use less than week old data, TLE and tracking in general is an estimate and the newer the data the better the estimate. (Not so much an issue with APRSISCE/32 until we support user-supplied TLEs). Note that not all TLE sources are created equal. See below for a specific example.

Also make sure the time on your PC is accurate, windows has a time sync function thats good enough for satellite tracking. But I recommend Meinberg for accurate timing and automatic correction.

And the time zone setting is also important. Don't say you're in the Eastern Time Zone and set the time for UTC or vice versa. Your time zone is used to offset the displayed time to UTC for satellite orbit calculations.

Also of course make sure you have not got your location wrong! This is not the center of the screen, but the location of "ME"!

Here's the list of currently known active satellites:

/* From Steve via AMSAT Status */

Name	Cat No.	Name	Cat No.
SRM	99999	CO-58	28895
Duchifat-1	40021	VO-52	28650
ARRISSat-1	37772	AO-51	28375
FASTRAC 2	37380	RS-22	27939
FASTRAC 1	37227	CO-57	27848
OREOS	37224	CO-55	27844
HO-68	36122	SAUDISAT 1C	27607
ITUpSAT1	35935	NO-44	26931
BEESAT	35933	SAUDISAT 1B	26549
SwissCube	35932	SAUDISAT 1A	26545
SO-67	35870	ARISS	25544
KKS-1	33499	SO-33	25509
STARS	33498	GO-32	25397
PRISM	33493	FO-29	24278
Rs-30	32953	RS-15	23439
CO-66	32791	IO-26	22826
DO-64	32789	AO-27	22825
Compass-1	32787	AO-16	20439
Co-65	32785	UO-11	14781
GENESAT-1	29655	AO-7	7530

On 10/14/2011 I researched a report that APRSISCE/32 was 2+ minutes different in an AOS forecast than Ham Radio Delux. After tracking down the sources of their respective TLEs, I found that APRSISCE/32's source of Celestrak Amateur TLE had the following from day 286 of year 11.

ISS (ZARYA)

1 25544U 98067A 11286.81211653 .00009560 00000-0 11861-3 0 2381

2 25544 51.6369 283.5817 0016212 325.3429 93.8255 15.60650478739436

HRD was using TLEs from AMSAT NASAbare TLE which included the following from day 285 of year 11.

ISS

1 25544U 98067A 11285.91900602 .00007956 00000-0 10000-3 0 02347

2 25544 051.6409 288.0749 0016408 320.9217 117.1587 15.60589610739299

A source TLE difference of Just under 9/10 of a day resulted in a 2 minute difference in the predictions. Yes, TLE freshness DOES make a difference. (It didn't help that the ISS had just undergone two orbital boosts in the past week, but still...)

As there are a lot of changes needed to send messages through satellites it is best to have a dedicated instance. To do this, simply make a new empty directory and copy APRSIS32.EXE into that directory. When you double-click the .EXE, it will prompt for new configuration as it originally did. If you're used to running the Development version, you'll have to set that again in the APRSIS32.XML file. It is perfectly fine to point multiple instances to a single copy of the OSM Map tiles.

You'll want to configure an APRS-IS filter to receive satellite-related traffic.

Set **Configure** ⇒ **General** ⇒ **Add Filter** to b/RS0ISS*/W3ADO-1/ARISS/FAST1 d/RS0ISS*/W3ADO-1/ARISS/FAST1

RS0ISS* will pick up the ARISSAT, the ISS and maybe others. W3ADO-1 is NO-44 (aka PCSat). FAST1 is one of the FASTRAC sister satellites. The b/ (buddy) will give you their beacons. The d/ (digi) will give you all packets digipeated by one of them.

You'll want to disable unnecessary things on the APRS-IS port.

In **Configure** ⇒ **Ports** ⇒ **APRS-IS**, uncheck Telemetry, Beacon, and Bulletin/Obj. You'll definitely want to uncheck IS to RF Igating since you don't want to transmit other people's -IS messages to the satellites. Leave that for them. Unchecking Beacon is a good idea so that your Beacons only go out RF and they'll only show up in APRS-IS if you were successfully copied by someone else.

In **Configure** ⇒ **Ports** ⇒ **<YourRFPort>**, uncheck Telemetry, Bulletin/Obj, IS to RF (redundant, but safer) and Me not 3rd (not sure why you'd uncheck this one, but it's a modifier on IS to RF anyway). Keeping RF to IS checked (both here and in the APRS-IS port) will send any RF-received packets to APRS-IS providing SGATE (Satellite Gateway) operation.

If you send APRS Messages via the satellites, make sure you uncheck Ack since you're not really likely to get any. This also eliminates the APRSIS32 automatic retries and leaves your transmitted text in the Chat box for manual retransmissions. You can monitor the APRS-IS stream on another instance to see if you've actually been digipeated by the satellite and gated by another SGATE (APRS-IS will never send you back your own packets).

You should also change the "Best" checkbox until it says "RF Only" and is checked. The "Send" button will also say "Send RF". In this configuration, any message you send will only go via RF and not via APRS-IS, but yet acks (if any) and incoming messages will still be accepted from both RF and APRS-IS.

it is Handy to have **Enables** ⇒ **View Logs** ⇒ **<YourRFPort>** window open and enabled see if you got through or not, and also your normal instance (with the same Add Filter shown above) to monitor the Packet Scroller (I often get through and am gated but don't see my own RF digi).

You will need to change your **Configure** ⇒ **Beacon** ⇒ **Path** depending on the Satellite you are attempting to use:

Satellite	Baud	Up Freq.	Down Freq.	Path
ISS	1200	145.825MHz	145.825MHz	ARISS or RS0ISS-4
PCSAT (NO-44)	1200	145.825MHz	145.825MHz	ARISS or W3ADO-1

FAST1	1200	145.825MHz	437.345MHz	FAST1
FAST1	9600	145.980MHz	437.345MHz	FAST1
FAST2	1200	435.025MHz	145.825MHz	N/A
FAST2	9600	437.345MHz	145.825MHz	N/A

NO-44 works in sunlight only but is capable of transatlantic work. It is also recommended to add SGATE to your path, if you are gating to the internet, so ARISS,SGATE for ISS

Annex

KISS Mode XML Configuration OpenCmd ⇒ QuietTime

TH-D7

```
<OpenCmd>^M~/</OpenCmd>
<OpenCmd>^M~/</OpenCmd>
<OpenCmd>XFLOW OFF</OpenCmd>
<OpenCmd>FULLDUP OFF</OpenCmd>
<OpenCmd>KISS ON</OpenCmd>
<OpenCmd>RESTART!!0</OpenCmd>
<CloseCmd>^192^255^192~!!0</CloseCmd>
<CloseCmd>^C^C^C~!!0</CloseCmd>
<CloseCmd>TC 1!TS 1</CloseCmd>
<CloseCmd>TN 2,0!TN 2,0</CloseCmd>
```

TH-D72

```
<OpenCmd>^C^C^C~!!0</OpenCmd>
<OpenCmd>TC 1!?!1.0</OpenCmd>
<OpenCmd>TC 1!TS 1!1.0</OpenCmd>
<OpenCmd>TN 1,1!TN 1,1</OpenCmd>
<OpenCmd>TC 0!0.050</OpenCmd>
<OpenCmd>^M!cmd:!1</OpenCmd>
<OpenCmd>DIG OFF</OpenCmd>
<OpenCmd>HB 1200</OpenCmd>
<OpenCmd>GPST $GPRMC</OpenCmd>
<OpenCmd>LTM 0</OpenCmd>
<OpenCmd>LTMH OFF</OpenCmd>
<OpenCmd>LOC E 0</OpenCmd>
<OpenCmd>M ON</OpenCmd>
<OpenCmd>MCOM ON</OpenCmd>
<OpenCmd>MS OFF</OpenCmd>
<OpenCmd>CONO OFF</OpenCmd>
<OpenCmd>P 250</OpenCmd>
<OpenCmd>UI OFF</OpenCmd>
<OpenCmd>UIF NOID</OpenCmd>
<OpenCmd>UIT %</OpenCmd>
<OpenCmd>X OFF</OpenCmd>
<OpenCmd>HEADER OFF</OpenCmd>
<OpenCmd>NE ON</OpenCmd>
<OpenCmd>NO OFF</OpenCmd>
<OpenCmd>ECHO OFF!cmd:!1.0</OpenCmd>
<OpenCmd>BEACON EVERY 60!cmd:!1.0</OpenCmd>
<OpenCmd>TC 1!?!EH!1</OpenCmd>
```



```

<OpenCmd>TC 1!TS 1!1.00</OpenCmd>
<CloseCmd>TC 0!!0.050</CloseCmd>
<CloseCmd>^M!cmd:!1</CloseCmd>
<CloseCmd>LTM 0</CloseCmd>
<CloseCmd>TC 1!?EH!1</CloseCmd>
<CloseCmd>TC 1!TS 1</CloseCmd>
<CloseCmd>TN 0,0!TN 0,0</CloseCmd>

```

TM-D700

```

<OpenCmd>^C</OpenCmd>
<OpenCmd>AI 1!AI 1</OpenCmd>
<OpenCmd>TC 1</OpenCmd>
<OpenCmd>TNC 2!TNC 2</OpenCmd>
<OpenCmd>BC 0,0!BC 0,0</OpenCmd>
<OpenCmd>FQ 0014480000,0!FQ 0014480000,0</OpenCmd> (00144390000,0 for USA)
<OpenCmd>SFT 0!SFT 0</OpenCmd>
<OpenCmd>DTB 0!DTB 0</OpenCmd>
<OpenCmd>BC 1,1!BC 1,1</OpenCmd>
<OpenCmd>TC 0!cmd:!5</OpenCmd>
<OpenCmd>ECHO OFF</OpenCmd>
<OpenCmd>BEACON EVERY 0</OpenCmd>
<OpenCmd>ECHO OFF</OpenCmd>
<OpenCmd>BEACON EVERY 0</OpenCmd>
<OpenCmd>DIGI OFF</OpenCmd>
<OpenCmd>GBAUD 4800</OpenCmd>
<OpenCmd>HBAUD 1200</OpenCmd>
<OpenCmd>LTMON 0</OpenCmd>
<OpenCmd>LTMHEAD ON</OpenCmd>
<OpenCmd>LOC E 0</OpenCmd>
<OpenCmd>MON ON</OpenCmd>
<OpenCmd>MCOM ON</OpenCmd>
<OpenCmd>MSTAMP OFF</OpenCmd>
<OpenCmd>CONOK OFF</OpenCmd>
<OpenCmd>PACLEN 250</OpenCmd>
<OpenCmd>UIDIGI OFF</OpenCmd>
<OpenCmd>UIFLOOD %</OpenCmd>
<OpenCmd>UITRACE %</OpenCmd>
<OpenCmd>XFLOW OFF</OpenCmd>
<OpenCmd>HEADER OFF</OpenCmd>
<OpenCmd>NEWMODE ON</OpenCmd>
<OpenCmd>NOMODE OFF</OpenCmd>
<CloseCmd>MON OFF</CloseCmd>
<CloseCmd>TC 1</CloseCmd>
<CloseCmd>TNC 1</CloseCmd>
<CloseCmd>BC 1,1!BC 1,1</CloseCmd>

```

TM-D710 (APRS)

```

<OpenCmd>^C^C^C~!!0</OpenCmd>
<OpenCmd>TC 1!?!1.0</OpenCmd>
<OpenCmd>TC 1!TS 1!1.0</OpenCmd>
<OpenCmd>TN 1,0!TN 1,0</OpenCmd>
<OpenCmd>TC 0!!0.050</OpenCmd>
<OpenCmd>^M!cmd:!1</OpenCmd>
<OpenCmd>DIG OFF</OpenCmd>
<OpenCmd>GB 4800</OpenCmd>
<OpenCmd>HB 1200</OpenCmd>
<OpenCmd>GPST $GPRMC</OpenCmd>

```

```

<OpenCmd>LTM 0</OpenCmd>
<OpenCmd>LTMH OFF</OpenCmd>
<OpenCmd>LOC E 0</OpenCmd>
<OpenCmd>M ON</OpenCmd>
<OpenCmd>MCOM ON</OpenCmd>
<OpenCmd>MS OFF</OpenCmd>
<OpenCmd>CONO OFF</OpenCmd>
<OpenCmd>P 250</OpenCmd>
<OpenCmd>UI OFF</OpenCmd>
<OpenCmd>UIF NOID</OpenCmd>
<OpenCmd>UIT %</OpenCmd>
<OpenCmd>X OFF</OpenCmd>
<OpenCmd>HEADER OFF</OpenCmd>
<OpenCmd>NE ON</OpenCmd>
<OpenCmd>NO OFF</OpenCmd>
<OpenCmd>ECHO OFF!cmd:!1.0</OpenCmd>
<OpenCmd>BEACON EVERY 0!cmd:!1.0</OpenCmd>
<OpenCmd>ECHO OFF!cmd:!1.00</OpenCmd>
<OpenCmd>BEACON EVERY 0!cmd:!1.00</OpenCmd>
<OpenCmd>TC 1!?EH!1</OpenCmd>
<OpenCmd>TC 1!TS 1!1.00</OpenCmd>
<CloseCmd>TC 0!!0.050</CloseCmd>
<CloseCmd>^M!cmd:!1</CloseCmd>
<CloseCmd>LTM 0</CloseCmd>
<CloseCmd>TC 1!?EH!1</CloseCmd>
<CloseCmd>TC 1!TS 1</CloseCmd>
<CloseCmd>TN 0,0!TN 0,0</CloseCmd>

```

TM-D710 (PKT)

```

<OpenCmd>^C^C^C~!!0</OpenCmd>
<OpenCmd>TC 1!?</OpenCmd>
<OpenCmd>TC 1!TS 1</OpenCmd>
<OpenCmd>TN 2,0!TN 2,0</OpenCmd>
<OpenCmd>^M!cmd:!5</OpenCmd>
<OpenCmd>DIG OFF</OpenCmd>
<OpenCmd>GB 4800</OpenCmd>
<OpenCmd>HB 1200</OpenCmd>
<OpenCmd>GPST $GPRMC</OpenCmd>
<OpenCmd>LTM 10</OpenCmd>
<OpenCmd>LTMH OFF</OpenCmd>
<OpenCmd>LOC E 0</OpenCmd>
<OpenCmd>M ON</OpenCmd>
<OpenCmd>MCOM ON</OpenCmd>
<OpenCmd>MS OFF</OpenCmd>
<OpenCmd>CONO OFF</OpenCmd>
<OpenCmd>P 250</OpenCmd>
<OpenCmd>UI OFF</OpenCmd>
<OpenCmd>UIF NOID</OpenCmd>
<OpenCmd>UIT %</OpenCmd>
<OpenCmd>X OFF</OpenCmd>
<OpenCmd>HEADER OFF</OpenCmd>
<OpenCmd>NE ON</OpenCmd>
<OpenCmd>NO OFF</OpenCmd>
<OpenCmd>ECHO OFF!cmd:!1.0</OpenCmd>
<OpenCmd>BEACON EVERY 0!cmd:!1.0</OpenCmd>
<OpenCmd>ECHO OFF!cmd:!1.00</OpenCmd>
<OpenCmd>BEACON EVERY 0!cmd:!1.00</OpenCmd>
<OpenCmd>^C^C^C~!!0</OpenCmd>
<CloseCmd>^M!cmd:!1</CloseCmd>

```

```
<CloseCmd>LTM 0</CloseCmd>
<CloseCmd>TC 1!TS 1</CloseCmd>
<CloseCmd>TN 0,0!TN 0,0</CloseCmd>
```

TNC SCS-DSP

```
<OpenCmd>^027~!!1</OpenCmd>
<OpenCmd>^064^075!!0</OpenCmd>
<CloseCmd>^192^255^192~!!1</CloseCmd>
<CloseCmd>^027~!!1</CloseCmd>
```

TNC Tiny2

```
<OpenCmd>^M~</OpenCmd>
<OpenCmd>^M~</OpenCmd>
<OpenCmd>XFLOW OFF</OpenCmd>
<OpenCmd>FULLDUP OFF</OpenCmd>
<OpenCmd>KISS ON</OpenCmd>
<OpenCmd>RESTART!!0</OpenCmd>
<CloseCmd>^192^255^192~!!0</CloseCmd>
<CloseCmd>^C^C^C~!!0</CloseCmd>
```

TNC TT4

```
<OpenCmd>~!!0</OpenCmd>
<CloseCmd>~!!0</CloseCmd>
```

SCS PTC-IIIusb

a. FSK (FSK300)

```
<OpenCmd>^027~!!1</OpenCmd>
<OpenCmd>^M~!!1</OpenCmd>
<OpenCmd>QUIT!cmd:</OpenCmd>
<OpenCmd>FSKA 600</OpenCmd> check your value (it is up to ALC limit)
<OpenCmd>TONES 2</OpenCmd>
<OpenCmd>MARK 1800</OpenCmd>
<OpenCmd>SPACE 1600</OpenCmd>
<OpenCmd>TRX Frequency 10147.6</OpenCmd>
<OpenCmd>PAC!pac:</OpenCmd>
<OpenCmd>BAUD 300!pac:</OpenCmd>
<OpenCmd>^064^075!!0</OpenCmd>
<CloseCmd>^192^255^192~!!2</CloseCmd>
<CloseCmd>^M~!pac:!1</CloseCmd>
<CloseCmd>QUIT!cmd:</CloseCmd>
```

b. RPR (RPR300)

```
<OpenCmd>^027~!!1</OpenCmd>
<OpenCmd>^M~!!1</OpenCmd>
<OpenCmd>QUIT!cmd:</OpenCmd>
<OpenCmd>PSKA 250</OpenCmd> check your value (it is up to ALC limit)
<OpenCmd>TRX Frequency 10147.3</OpenCmd>
<OpenCmd>PAC!pac:</OpenCmd>
<OpenCmd>BAUD R300!pac:</OpenCmd>
<OpenCmd>^064^075!!0</OpenCmd>
<CloseCmd>^192^255^192~!!2</CloseCmd>
<CloseCmd>^M~!pac:!1</CloseCmd>
<CloseCmd>QUIT!cmd:</CloseCmd>
```